Maintenance Guide

INTERACTIVE™ UNIX®

OPERATING SYSTEM

SunSoft

A Sun Microsystems, Inc. Business

INTERACTIVE™ UNIX® System V/386 Release 3.2 Maintenance Guide

2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

Part No: 801-7872-10
Revision A, June 1994

SunSoft
A Sun Microsystems, Inc. Business

Contents

About This Book

This guide provides detailed information on most aspects of configuring and maintaining INTERACTIVE™ UNIX® System V/386 Release 3.2 from SunSoft and its extensions.

The maintenance procedures are designed to be as self-explanatory as possible, with easily accessible on-line help. This guide provides more information than is available in the on-line help facility and gives additional information necessary for users with nonstandard hardware configurations or special system needs. (The screens shown in this document are for purposes of illustration. Your screens may vary slightly.)

Who Should Use This Book

This guide is intended for users of the INTERACTIVE UNIX Operating System who need to modify, maintain, or fix their system. Novice system administrators or system administrators unfamiliar with the INTERACTIVE UNIX System may refer to Part 2 — "System Administration for New Users" in the *INTERACTIVE UNIX System V/386 Release 3.2 User's Guide*. It provides step-by-step instructions for basic system administration tasks.

Before You Read This Book

If you are new to the INTERACTIVE UNIX Operating System and other UNIX-based systems, read Part 1 — "INTERACTIVE UNIX System Primer" in the *INTERACTIVE UNIX System User's Guide*. Refer to Part 2 — "System Administration for New Users" in that same book to learn how to shut down and reboot the system, install user accounts, back up and maintain files on the system, and tailor the system to match your requirements.

Also, refer to the *INTERACTIVE UNIX System V/386 Release 3.2 Release Notes* for any last minute caveats, problems, or workarounds.

How This Book Is Organized

Each chapter in this guide is briefly described below.

Chapter 1, "The Character User Interface," gives a general description of the INTERACTIVE UNIX System Character User Interface (CUI), which is the interface used by the `sysadm` system administration program and `kconfig`, the kernel configuration program. It explains how to get help at any time while using the system and how to use the menus and forms to perform tasks.

Chapter 2, "The System Administration Program," gives a general description of how the system administration program is used to perform system administration tasks and describes how to access its menus using the `sysadm` command.

Chapter 3, "Creating and Using INTERACTIVE UNIX System File Systems," describes the procedures for creating and maintaining INTERACTIVE UNIX System file systems that are run from the command line.

Chapter 4, "File System Maintenance," explains how to use `fsck` from the command line to check and repair file systems, how bad blocks are detected and handled, the fixed disk layout, how to recover from a system crash, and how to mount diskette-based file systems when using DOS-FSS. It also discusses the various file system types available in the INTERACTIVE UNIX Operating System.

Chapter 5, "Using kconfig to Tailor Your System Kernel," describes the `kconfig` program interface, and the options available through it for configuring your kernel. Using `kconfig`, you can add, modify, and remove drivers, add and remove facilities, add and modify default parameters for memory size, add and modify tunable system parameters, display the High Performance Device Driver configuration, configure the High Performance Device Driver, and build and install a new kernel. This chapter also explains what to do if the system does not boot.

Chapter 6, "Hardware Compatibility and Configuration," contains descriptions of the hardware requirements, software setup, configuration information, and possible conflicts for the drivers included with the INTERACTIVE UNIX Operating System.

Chapter 7, "Adding Modems, Printers, and Other Serial Devices," explains the basic networking procedures required to add serial devices to your system and the software configuration that is required.

Chapter 8, "Basic Networking Administration," describes the administration of the Basic Networking utilities (primarily `uucp`) that allow your system to communicate with other UNIX System computers using either dial-up or hard-wired communication lines.

Chapter 9, "Electronic Mail," explains the electronic mail options available with the INTERACTIVE UNIX System and provides pointers to the information for setting up the mail.

Chapter 10, "The Berkeley Line Printer Spooler (lpr) System," describes the structure and installation procedure for the line printer spooling system.

Chapter 11, "Using a Serial Terminal as the Console," describes how to use the `chgcon` utility to select a console other than the default. The various options available through `chgcon` are described and explained.

Chapter 12, "Setting Up an International Environment," provides information you need if you have installed the International Supplement optional subset. This chapter explains how to set up a properly functioning international environment on an INTERACTIVE UNIX System. It also summarizes the internationalization features and provides tips for C programmers who want to develop internationalized applications.

Chapter 13, "Configuring Network Drivers," describes how to install and configure network drivers on the INTERACTIVE UNIX System.

Chapter 14, "Configuring INTERACTIVE TCP/IP," describes how to add a network interface, configure a network, and activate a TCP/IP interface.

Chapter 15, "INTERACTIVE NFS Management," describes how to set up and maintain INTERACTIVE NFS®, SunSoft's distributed computing file system. It provides explanations and step-by-step instructions for options available with the INTERACTIVE NFS extension.

Chapter 16, "Configuring INTERACTIVE X11," describes the functionality of the INTERACTIVE X11 optional extension. It discusses how to build new servers, remove servers, make default servers, modify configuration information, install new X drivers, add X users, and delete X users.

Appendix A, "Setting Up Your Printer," describes how to configure the System V `lp` print system.

Appendix B, "Troubleshooting," answers some of the most frequently asked questions that arise during the installation and configuration of the INTERACTIVE UNIX Operating System to help you to troubleshoot problems with your system.

Appendix C, "Kernel Error Messages," lists the kernel panic, warning, and notice messages along with possible actions you can take.

Appendix D, "Advanced Internationalization Features," provides advanced tips for C programmers who want to develop internationalized applications.

Related Books

Part 2 — "System Administration for New Users" in the *INTERACTIVE UNIX System User's Guide* contains instructions for common system administration tasks not discussed in this guide.

Consult the *INTERACTIVE UNIX System V/386 Release 3.2 Installation Guide* for information on how to install the INTERACTIVE UNIX System or optional software.

Conventions

Various font types are used in this guide to distinguish between information you type, information displayed by the system, and items that either you or the system replace with a variable. These are the type changes and symbols used in this book.

Typeface or Symbol	Meaning	Example
AaBbCc123	An initial capitalized key name refers to that key on your keyboard	Press Enter to store the file. Press Control-d to quit.
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.profile` file. Use `ls -a` to list all files. `$ You have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`$` **`rlogin host2`** `password:`
AaBbCc123	Placeholder: replace with a real name or value; book titles; words to be emphasized; new terms	To delete a file, type `rm` *filename*. See the *INTERACTIVE UNIX System User's Guide*. You *must* be `root` to do this.

Code samples are included in plain boxes and may display the following:

UNIX C shell prompt	`%`
UNIX Bourne and Korn shell prompt	`$`
UNIX superuser prompt	`#`

Note – Keys on your keyboard may be labeled differently than those shown in this guide. For example, the Enter key is labeled Return on some keyboards. If your hardware or software vendor supplies additional documentation with your system, read that documentation for information on key names before you continue.

Plain boxes represent screen displays, system responses, file contents, path names, or program code. They may contain text that you type, which is indicated by bold lettering. For example:

```
login: tony
*** Welcome to the UNIX Operating System. ***
$
```

Highlighting on screens is represented by reverse type. For example:

```
 Disk      File      Machine    Software   User      Help      Quit
Statistics on fixed disk(s) usage
┌──────────────────────────┐SYSTEM ADMINISTRATION
│ Current Disk Usage       │
│ Diskette Management   -> │
│ Fixed Disk Management -> │
│ Quit                     │
└──────────────────────────┘
```

References of the form *name*(n) refer to an entry called name in section "n" of the reference manual or manual entries associated with that product or as stated in the documentation.

Throughout this guide, the following full documentation titles are referenced in shortened versions as follows:

Full Title	Shortened Version
INTERACTIVE UNIX System V/386 Release 3.2 User's Guide	*INTERACTIVE UNIX System User's Guide*
INTERACTIVE UNIX System V/386 Release 3.2 Installation Guide	*INTERACTIVE UNIX System Installation Guide*
INTERACTIVE UNIX System V/386 Release 3.2 Hardware Compatibility List	*INTERACTIVE UNIX System Hardware Compatibility List*

The Character User Interface 1

The `sysadm` and `kconfig` programs both make use of the CUI System. This system is designed to be as self-explanatory as possible. If you've never used it before, you should read, either in this guide or on-line, the short summary of how to get help at any time and how to use the menus and forms to perform tasks.

One way to learn about the system is to look at it on-line while reading about it here. You can look at the forms on-line without altering your system; no actions are taken until you explicitly "instruct" `sysadm` to do so. The system displays a warning screen and requires you to confirm your action before any potentially destructive changes are initiated. You can also run `sysadm` in a demonstration mode by typing:

```
$ sysdemo
```

In this mode, all `sysadm` actions are disabled, and you can safely explore the system.

Using the CUI Help Facility

You can press F1 for on-line help at any time. If context-specific help is available, it appears. Pressing F1 a second time accesses the general Help Index. If no context-specific help is available, the Help Index appears immediately. Use the up and down arrow keys or Page-Up and Page-Down to scroll through the Help Index. Press Enter to select a topic.

There are also HELP buttons at the bottom of many forms. Use Tab or Back-Tab to move to the HELP button, and press Enter to obtain general help on the form you are viewing. Some help topics have more than one page. To move from page to page of each help topic, use Page-Up or Page-Down. You can exit from the help system at any time by pressing Escape. The following table summarizes how to use the help system:

Using the CUI Help System

Action	Key
Display context-specific help	F1
Display the Help Index	F1 when already in the help system F1-F1 when you are *not* in the help system
Scroll through the Help Index	Up and down arrow keys
Page through the Help Index	Page-Up or Page-Down
Select a Help Index topic	Enter
Move to the HELP button	Tab or Back-Tab
Activate the HELP button	Enter
Page through help topics	Page-Up or Page-Down
Exit the help system	Escape

Hypertext

The help in this program is a hypertext system. The system provides one or more pages of help on a number of topics. Topics are connected to each other by means of links. Links are words that appear on-line in **bold** text. Three standard links appear at the bottom of each help window:

```
Exit    Go Back    Index
```

`Exit`
: Always takes you out of the help system.

`Go Back`
: Always takes you to the previous topic, if one exists.

`Index`
: Always takes you to the Help Index.

Links in the text provide a path to additional information about a topic. Press Tab or Back-Tab to move from link to link. When the one you want is highlighted (active), press Enter to follow it.

Links

When you follow a link, a new topic appears. If there is more than one page in the topic, use Page-Up or Page-Down to move through it. To return to the previous topic, press Tab or Back-Tab to move to `Go Back` at the bottom of the help window, and then press Enter. You can follow any number of links through this system. Using `Go Back` will retrace your steps exactly to the original point at which you entered the system.

Using CUI Menus and Forms

The CUI System is designed to make administration of the INTERACTIVE UNIX Operating System as easy as possible. Its major components are the bar menu at the top of your screen, pull-down menus, and forms.

The Bar Menu

Use the left and right arrow keys ← → to move from title to title of the bar menu. Press F1 for on-line help on the bar menu titles. Press Enter to pull down a menu. To exit from the bar menu and return to the system prompt, move to `Quit` and press Enter.

Pull-Down Menus

Use the up and down arrow keys ↑ ↓ to move from option to option, or type the first letter(s) of an option name. If there is an arrow (→) following the option, another menu appears when you press Enter. If there is no arrow, a form appears when you press Enter. Press F1 for on-line help on the menu items. To exit from a pull-down menu, move to `Quit` (or type `q`) and press Enter, or simply press Escape.

Forms

Forms are used to display or gather information needed to perform the task you selected from a pull-down menu. Forms contain several different kinds of fields. Some fields are used only to display information; you cannot press Tab or Back-Tab to move to them or change them. Other fields allow you to edit them by making choices or typing information. Help is available for fields that you can edit. Press F1 for on-line help on these fields. To move from field to field and button to button, press Tab or Back-Tab.

All forms have at least one button at the bottom of the form. Most forms have these buttons:

```
OK        CANCEL       HELP
```

`OK`
: Confirms that you are satisfied with the information on the form and want your changes to take effect.

`CANCEL`
: Exits from the form without any changes taking effect.

`HELP`
: Provides help for the form.

To activate a button, press Tab or Back-Tab to move to it and press Enter.

Fields

Forms in this system contain a number of different kinds of fields. Some fields are lists that you can scroll through using the up and down arrow keys. Some lists are always visible, while others require you to press the spacebar to pop them up. Press Enter while an item is highlighted to select it from the list. Help is available for fields that you can edit. Press F1 for on-line help on these fields.

`< >`
: Fields within angle brackets < > allow you to use the spacebar to cycle through the choices for that field or to press the spacebar to pop up a list of values. Use the up and down arrow keys to scroll through the items in a list. Press Enter to select one.

`[ ]`

Fields within square brackets `[ ]` are check boxes. Press the spacebar to "check" the box (an `X` appears). You can press the spacebar to toggle the `X` in or out of a field.

`( )`

Fields within parentheses `( )` are radio buttons. They usually occur in sets where a choice must be made between two or more mutually exclusive options. Press the spacebar to turn the radio button "on" and an asterisk (`*`) appears. If you turn on one radio button, the other choices automatically go "off."

The System Administration Program 2

This chapter provides an overview of the system administration program, a standard feature of the INTERACTIVE UNIX Operating System which is used to perform commonly required system administration tasks. It is designed to be as easy to use and self-explanatory as possible. The `sysadm` program uses the CUI System to display menus and forms. You may want to read the brief introduction to the CUI System in the previous chapter (or on-line) before attempting to use `sysadm`. Experienced UNIX System users can perform the system administration tasks from the command line rather than using the menu interface, if desired.

More detailed information about using these menus can be found in the appropriate sections of this document and in the *INTERACTIVE UNIX System User's Guide*.

Note – This document does not attempt to describe every possible maintenance procedure available through the system administration program. The system is easy to follow, and help is available on-line at any time by pressing F1.

`sysadm` can be used as a login account or as a utility program. A password should be assigned to `sysadm` when the system is installed.

You can also run `sysadm` in a demonstration mode by typing:

```
$ sysdemo
```

In this mode, all sysadm actions are disabled, and you can safely explore the system.

The following conventions apply when you are using the sysadm login, sysadm command, or any sysadm menu:

- You must be at the console terminal to log in as sysadm, and you must know its password.
- If you do not understand the options or what to put in to a field in a form displayed by the system, press F1 for help at any time. If context-specific help is available, it appears. Pressing F1 a second time accesses the general Help Index. If no context-specific help is available, the Help Index appears immediately. You can exit from the help system at any time by pressing Escape.
- To exit from the bar menu and return to the system prompt, or to exit from a pull-down menu, use the arrow keys to move to `Quit` (or type `q`) and press Enter. You can also exit from a pull-down menu by simply pressing Escape. To exit from a form without any changes taking effect, move to the `CANCEL` button and press Enter. To exit from a form with changes taking effect, move to the `OK` button and press Enter.

`sysadm` *Menu Bypass*

You can access a sysadm pull-down menu directly by typing sysadm, followed by the appropriate command name. For example, type:

```
# sysadm harddisk
```

to access the sysadm `Fixed Disk Management` menu.

Each sysadm pull-down menu contains a list of options and/or other menus. You can bypass the menus and access a known option directly by typing the option name as an argument to sysadm. For example, the `Diskette Management` menu contains an option for formatting a disk, called `Format a Diskette`. Access that option directly by typing:

```
# sysadm fmtdisk
```

If you use sysadm as a login, you can also supply the sysadm options described above, such as fmtdisk, on the command line. The following tables list all the menus and options and their abbreviations:

Menu or Form	Command
Disk	disk
Current Disk Usage	diskuse
Diskette Management	diskettes
Copy a Diskette	cpdisk
Erase a Diskette	eradisk
Format a Diskette	fmtdisk
Verify a Diskette	verdisk
File System Management	filesys
Check a Diskette File System	chkfdfs
Create a Diskette File System	mkfdfs
Mount a Diskette File System	mntfdfs
Unmount a Diskette File System	umntfdfs
Fixed Disk Management	harddisk
Remap Disk Defects	badblks
Add a Fixed Disk to the System	addhd
Check a Fixed Disk File System	chkhdfs
Display Fixed Disk Information	lshd
Mount a Fixed Disk File System	mnthdfs
Unmount a Fixed Disk File System	umnthdfs

Menu or Form	Command
File	file
Back Up or Restore Files	backups
Back Up Selected Files	backup
Archival/Incremental Backup	fsback
Restore Files	restor
File Information	fileinfo
Owner/Group	filuse
Age	filage
Size	filsiz

Menu or Form	Command
Machine	machine
Current Disk Usage	diskuse
Modem Initialization	setmodm
lp Printer Management	lpmgmt
Add a Printer	addprt
Delete a Printer	delprt
Modify a Printer	chgprt
Change Printer Status	statprt
List Printers	lsprt
Spooler Management	statspl
lpr Printer Management	lprmgmt
Add a Printer	addlpr
Delete a Printer	dellpr
Modify a Printer	chglpr
Create Spool Directories	crdirlpd
List Printers	lslpr
Spooler Management	statlpd
System Setup	syssetup
Display/Change System Configuration	setconf
Set System Passwords	syspass
Tty Management	ttys
Virtual Terminals	chgvts
Modify Tty Parameters	chgtty
Getty Line Management	gettys
Add	addgetty
Delete	delgetty
List	lsgetty
Modify	chgetty

Menu or Form	Command
Software	software
Install a Package	addpkg
Remove a Package	delpkg
List Installed Packages	lspkg
Mail System Setup	setmail
Basic Networking	UUCP
Test a UUCP Connection	callsys
Modem Initialization	setmodm
Dialcode Management	dcode
Add	adddcode
Delete	deldcode
List	lsdcode
Modify	chgdcode
Outgoing Connections	device
Add	adddev
Delete	deldev
List	lsdev
Modify	chgdev
Permissions Management	perm
Add	addperm
Delete	delperm
List	lsperm
Modify	chgperm
Poll Management	poll
Add	addpoll
Delete	delpoll
List	lspoll
Modify	chgpoll

Menu or Form	Command
Incoming Connections	`port`
Add	`addport`
Delete	`delport`
List	`lsport`
Modify	`chgport`
System Management	`system`
Add	`addsys`
Delete	`delsys`
List	`lssys`
Modify	`chgsys`

Menu or Form	Command
User (includes **Group**)	`user`
List Active Users	`whoson`
Group Management	`groups`
Add a New Group	`addgroup`
Remove a Group	`delgroup`
Change a Group Name	`chgroup`
List Groups	`lsgrps`
List All Groups	`lsgroup`
List System Groups	`lsSgroup`
List User Groups	`lsUgroup`
User Management	`users`
Add a New User	`adduser`
Remove a User	`deluser`
Modify User Information	`chguser`
Set 'Add User' Defaults	`usrdefs`
Set Available Shells	`shdefs`
Supplementary Groups	`supgrp`
List Users	`lsusrs`
List All Users	`lsuser`
List System Users	`lsSuser`
List Regular Users	`lsUuser`

The information supplied is subject to change. SunSoft reserves the right to modify its software product at any time.

The sysadm *Menus*

When you type the sysadm command, the bar menu is displayed. If you set a password for the sysadm login when you installed the system, you will be prompted for a password in the usual manner. The system displays a screen similar to this:

```
   Disk       File       Machine    Software   User       Help       Quit
  Manage disks and diskettes
┌──────────────────────SYSTEM ADMINISTRATION──────────────────────────────┐
│                                                                          │
│                                                                          │
└──────────────────────────────────────────────────────────────────────────┘
```

Each title on the bar menu represents a category of system administration tasks. Use the left and right arrow keys (or type the first letter(s) of a title) to move to a title. A short summary of that title is displayed under the bar menu. When you move to a title and press Enter to select it, the system pulls down a menu that lists tasks that are specific to that option. When you select an option from a pull-down menu, the system either provides a form to view or fill in to perform the specified task, or the system displays another menu. If an arrow follows an option name, for example:

```
Diskette Management →
```

another menu appears when you press Enter.

If you want to return to the previous menu at any point, type `q` or use the arrow keys to move to `Quit` on the menu and then press Enter.

The `Disk` *Menu*

The `Disk` menu looks similar to this:

```
 Disk       File       Machine    Software   User       Help       Quit
Statistics on fixed disk(s) usage
                            SYSTEM ADMINISTRATION
 Current Disk Usage
 Diskette Management     ->
 Fixed Disk Management   ->
 Quit
```

The `Diskette Management` option displays a menu that looks similar to this:

```
 Disk       File       Machine    Software   User       Help       Quit
Copy the contents of a diskette
                                SYSTEM ADMINISTRATION
Current Disk Usage
Diskette Management    ->
Fi
Qu  Copy a Diskette
    Erase a Diskette
    Format a Diskette
    Verify a Diskette
    File System Management  ->
    Quit
```

`File System Management` includes:

```
 Disk       File       Machine    Software   User       Help       Quit
Check a diskette file system for problems or errors
┌──────────────────────────────SYSTEM ADMINISTRATION──────────────────────┐
│Current Disk Usage           │
│Diskette Management    ->    │
│Fi┌────────────────────────────┐
│Qu│Copy a Diskette             │
└──│Erase a Diskette            │
│  │Format a Diskette           │
│  │Verify a Diskette           │
│  │File System Management  ->  │
│  │Qu┌─────────────────────────────────┐
│  └──│Check a Diskette File System     │
│     │Create a Diskette File System    │
│     │Mount a Diskette File System     │
│     │Unmount a Diskette File System   │
│     │Quit                             │
│     └─────────────────────────────────┘
└─────────────────────────────────────────────────────────────────────────┘
```

The `Fixed Disk Management` option displays a menu that looks similar to this:

```
 Disk      File      Machine    Software   User      Help      Quit
Add information about bad sectors on the fixed disk
                            SYSTEM ADMINISTRATION
 Current Disk Usage
 Diskette Management    ->
 Fixed Disk Management  ->
 Qu
     Remap Disk Defects
     Add a Fixed Disk to the System
     Check a Fixed Disk File System
     Display File System Information
     Mount a Fixed Disk File System
     Unmount a Fixed Disk File System
     Quit
```

The `File` *Menu*

The `File` menu looks similar to this:

```
 Disk       File       Machine    Software   User       Help       Quit
Back up files, restore files, or set reminder schedules for backups
          Back Up or Restore Files  ->TRATION
          File Information          ->
          Quit
```

The `Back Up or Restore Files` menu displays the following:

```
 Disk       File       Machine    Software   User       Help       Quit
Select certain files, directories, and/or subtrees for backup
                                         TRATION
          Back Up or Restore Files  ->
          Fi
          Qu  Back Up Selected Files
              Archival/Incremental Backup
              Restore Files
              Quit
```

The `File Information` menu under `File` on the bar menu displays the following:

```
 Disk       File       Machine    Software   User       Help       Quit
List files based on the owner or group
┌─────────┌─────────────────────────────┐TRATION─────────────────────────────┐
│         │Back Up or Restore Files  -> │                                    │
│         │File Information          -> │                                    │
│         │Qu┌─────────────┐            │                                    │
│         └──│Owner/Group  │────────────┘                                    │
│            │Age          │                                                 │
│            │Size         │                                                 │
│            │Quit         │                                                 │
│            └─────────────┘                                                 │
│                                                                            │
└────────────────────────────────────────────────────────────────────────────┘
```

The Machine *Menu*

The Machine menu looks similar to this:

```
 Disk      File      Machine    Software   User      Help      Quit
Statistics on fixed disk(s) usage
                     Current Disk Usage
                     Modem Initialization
                     Printer Management    ->
                     System Setup          ->
                     Tty Management        ->
                     Quit
```

Note – The Printer Management option refers to either "System V" lp Printer Management or "Berkeley" lpr Printer Management, depending on which subset was installed.

The `Printer Management` option displays a screen similar to this:

```
 Disk       File       Machine    Software   User       Help       Quit
Add a printer to your system

                    Current Disk Usage
                    Modem Initialization
                    Printer Management    ->
                    Sys
                    Tty  Add a Printer
                    Qui  Delete a Printer
                         Modify a Printer
                         Change Printer Status
                         List Printers
                         Spooler Management
                         Quit
```

The `System Setup` option displays a screen similar to this:

```
 Disk       File       Machine    Software   User       Help       Quit
Display or change CMOS system parameters, such as time, date, and time zone

                    Current Disk Usage
                    Modem Initialization
                    Printer Management    ->
                    System Setup          ->
                    Tt
                    Qu  Display/Change System Configuration
                        Set System Passwords
                        Quit
```

The `Tty Management` option displays a screen similar to this:

```
 Disk       File       Machine    Software   User       Help       Quit
Change the number of active virtual terminals
                      Current Disk Usage
                      Modem Initialization
                      Printer Management      ->
                      System Setup            ->
                      Tty Management          ->
                      Qu
                          Virtual Terminals
                          Modify Tty Parameters
                          Getty Line Management  ->
                          Quit
```

The `Getty Line Management` menu displays a screen similar to this:

```
 Disk       File       Machine    Software   User       Help       Quit
Add getty entries or hunt sequences
                      Current Disk Usage
                      Modem Initialization
                      Printer Management      ->
                      System Setup            ->
                      Tty Management          ->
                      Qu
                          Virtual Terminals
                          Modify Tty Parameters
                          Getty Line Management  ->
                          Qu
                              Add
                              Delete
                              List
                              Modify
                              Quit
```

The `Software` *Menu*

The `Software` menu looks similar to this:

```
 Disk        File        Machine     Software   User        Help        Quit
Install a software package
                                       Install a package
                                       Remove a package
                                       List installed packages
                                       Mail System Setup
                                       Basic Networking         ->
                                       Quit
```

The `Basic Networking` menu looks similar to this:

```
  Disk       File       Machine                 User       Help       Quit
 Access and Check a UUCP system
                                   SYST
                                       Install a package
                                       Remove a package
                                       List installed packages
                                       Mail System Setup
                                       Basic Networking        ->
                                       Qu
                                           Test a UUCP Connection
                                           Modem Initialization
                                           Dialcode Management      ->
                                           Outgoing Connections     ->
                                           Permissions Management   ->
                                           Poll Management          ->
                                           Incoming Connections     ->
                                           System Management        ->
                                           Quit
```

The `User` *Menu*

The `User` menu looks similar to this:

```
 Disk       File       Machine    Software   User       Help       Quit
List users currently on this machine
----------------------------SYSTEM ADMINIST
                                          List Active Users
                                          Group Management   ->
                                          User Management    ->
                                          Quit
```

The `Group Management` menu looks similar to this:

```
 Disk      File      Machine    Software   User      Help      Quit
Add a new group
                          SYSTEM ADMINIST
                                           List Active Users
                                           Group Management   ->
                                           Us
                                           Qu  Add a New Group
                                               Remove a Group
                                               Change a Group Name
                                               List Groups          ->
                                               Quit
```

The `User Management` menu looks similar to this:

```
 Disk      File      Machine    Software   User      Help      Quit
Add a new user to the system
                          SYSTEM ADMINIST
                                           List Active Users
                                           Group Management   ->
                                           User Management    ->
                                           Qu
                                               Add a New User
                                               Remove a User
                                               Modify User Information
                                               Set `Add User' Defaults
                                               Set Available Shells
                                               Supplementary Groups
                                               List Users                 ->
                                               Quit
```

Creating and Using INTERACTIVE UNIX System File Systems 3

This chapter describes the procedures for creating and maintaining file systems from the command line (without using `sysadm`). (For file systems other than S51K, S5L, or S52K, refer to Chapter 4, "File System Maintenance.") For basic information on file system structure, file system and device naming conventions, disk partitioning, and using the `sysadm` commands to create and maintain INTERACTIVE UNIX System file systems, refer to the *INTERACTIVE UNIX System User's Guide.*

Creating a File System and Making It Available

After formatting a disk, you can define a file system on it using the `mkfs` command. (Note that you must have the 2 Kilobyte File System Utility Package installed to make a 2K file system.)

The `mkfs` command is used to create all file systems. One of its optional arguments is the rotational gap. For the INTERACTIVE UNIX System, the gap should always be the default value, `2`, which puts the blocks in ascending order. Thus, new files are more likely to be in sequence and are read faster. Because of this ordering, another optional argument to `mkfs` (cylinder size) is unimportant since you get the same order in all cases.

Using `mkfs`

The `mkfs` command has two formats:

```
mkfs [-L] special blocks[:i-nodes] [gap blocks/cyl] [-b blocksize]
mkfs [-L] special prototype [gap blocks/cyl] [-b blocksize]
```

Notice that the file system is not given a name in either format; it is identified by the file name of the special device file on which it will reside. The special device file, traditionally located in the directory `/dev`, is tied to the identifying controller and unit numbers (major and minor, respectively) for the physical device.

In the first format, the only other information that must be furnished on the `mkfs` command line is the number of 512-byte blocks the file system is to occupy. The second format lets you include that information in a prototype file that can also define a directory and file structure for the new file system, and it even allows for reading in the contents of files from an existing file system. The `-L` option specifies that an S5L long-name file system be created instead of the standard S51K file system. (The underlying file system formats are largely compatible with each other, and a file system may later be converted from one type to the other using the `-l` and `-L` options to `fsck.`)

Note – The 2K file system does not support the long-name S5L option.

Both formats let you specify information about the interrecord gap and the blocks per cylinder. If this information is not given on the command line, default values are used. The recommendations depend on the logical block size of the file system (see the discussion of the `-b` option at the end of this section). The recommended values are different from the defaults used by the command. In the first `mkfs` format, even though the number of blocks in the file is required, the number of inodes may be omitted. If the number of inodes is omitted, the command uses a default value of one inode for every four logical storage blocks.

If you use the first format of `mkfs`, the file system is created with a single directory. If you use a prototype file, as noted above, it can include information that causes the command to build and initialize a directory and file structure for the file system. The format of a prototype file is described in *mkfs*(1M).

The final option to `mkfs` lets you specify the logical block size to be used for the file system. By default, the file system has a logical block size of 1024 bytes. With the `-b` option, you can specify a logical block size of 512 bytes, 1024 bytes, or 2048 bytes, as explained in the next section.

Choosing Logical Block Size

Logical block size is the size of the blocks the INTERACTIVE UNIX System kernel uses to read or write files. The logical block size is usually different from the physical block size, which is the size of the smallest block that the disk controller can read or write, usually 512 bytes.

An administrator who uses the `mkfs` command to make a file system may specify the logical block size of the file system. By default, the logical block size is 1024 bytes (1K). All file systems created during system installation are 1K file systems. Besides 1K file systems, the INTERACTIVE UNIX System also supports 512-byte file systems and 2048-byte (2K) file systems. To use a 2K file system, you must install the 2 Kilobyte File System Utility Package optional subset.

To choose a reasonable logical block size for your system, you must consider performance and space. Since the 1K file systems use the INTERACTIVE UNIX Operating System's Fast File System software, it is almost always the best choice for logical block size.

Creating a File System on a Diskette

You can create your own file system on a diskette by using the `mkfs` and `labelit` commands. You will actually be specifying the file system that you want on the diskette and then mounting the file system as a directory under the INTERACTIVE UNIX System. Refer to *labelit*(1M) for additional information on `labelit`.

Creating a file system on a diskette can be very useful because the file systems are portable and it saves room on the fixed disk. The maximum size of a file system that can be created on a diskette is 720 blocks (512-byte blocks) for a 360K diskette, 2400 blocks (512-byte blocks) for a 1.2 MB diskette, 2880 blocks for a 1.44 MB (high-density, 3.5-inch) diskette, and 5760 blocks for a 2.88 MB (extra-density, 3.5-inch) diskette.

The following steps are used to create and identify a file system on a diskette:

1. Log in as `root` to ensure that you have the proper read/write permissions.
2. Insert a formatted diskette into the diskette drive. Type the format command lines appropriate for the type of diskette you are using:
   a. If you have a 360K diskette, type:

```
# /bin/format /dev/rdsk/f0d9dt
```

   and proceed to the next step.

   b. If you have a 1.2 MB diskette, type:

```
# /bin/format /dev/rdsk/f0q15dt
```

   and skip to step 4.

   c. If you have a 1.44 MB diskette, type:

```
# /bin/format /dev/rdsk/f0q18dt
```

   and skip to step 5.

   d. If you have a 2.88 MB diskette, type:

```
# /bin/format /dev/rdsk/f0q36dt
```

   and skip to step 6

3. For a 360K diskette, make a file system of 720 blocks and 160 inodes using the following:

```
# /etc/mkfs /dev/dsk/f0d9dt 720:160 1 18
```

   The rotational gap is 1 and the blocks per cylinder is 18. A 360K diskette has 18 512-byte blocks per cylinder. Skip to step 7.

4. If you have a 1.2 MB diskette, make a file system of 2400 blocks and 592 inodes using the following:

```
# /etc/mkfs /dev/dsk/f0q15dt 2400:592 1 30
```

The rotational gap is 1 and the blocks per cylinder is 30. The 1.2 MB diskette has 30 512-byte blocks per cylinder. Skip to step 7.

5. If you have a 1.44 MB diskette, make a file system of 2880 blocks and 600 inodes using the following:

```
# /etc/mkfs /dev/dsk/f0q18dt 2880:600 1 36
```

The rotational gap is 1 and the blocks per cylinder is 36. A 1.44 MB diskette has 36 512-byte blocks per cylinder. Skip to step 7.

6. If you have a 2.88 MB diskette, make a file system of 5,760 blocks and 900 inodes using the following:

```
# /etc/mkfs /dev/dsk/f0q36dt 5760:900 1 72
```

The rotational gap is 1 and the blocks per cylinder is 72. A 2.88 MB diskette has 72 512-byte blocks per cylinder. Skip to step 7.

7. Assume that for the rest of the example you are using a 1.2 MB diskette. Regardless of the size of your diskette, your screen will look similar to this:

```
# /etc/mkfs /dev/dsk/f0q15dt 2400:592 1 30
Mkfs: /dev/dsk/f0q15dt?
(DEL if wrong)
bytes per logical block = 1024
total logical blocks = 1200
total inodes = 592
gap (physical blocks) = 1
cylinder size (physical blocks) = 30
```

If the command output in this screen is not what you want, press Delete to cancel the command.

8. Label the diskette file system using the `labelit` command. For this example, assume the file system will be called `memo`. The volume name will be `memo2.0`. Type:

```
# /etc/labelit /dev/dsk/f0q15dt memo memo2.0
```

Your screen will look similar to this:

```
# /etc/labelit /dev/dsk/f0q15dt memo memo2.0
Current fsname: , Current volname: Blocks: 2400, Inodes: 592
FS Units: 1Kb, Date last modified: Thu Sep 10 13:24:03 1987
NEW fsname = memo, NEW volname = memo2.0 --Del if wrong!!
```

9. File systems are usually mounted in `root` (/) as directories. Make a directory in the `root` directory with the same name as the file system you are mounting:

```
# mkdir /memo
```

You must be `root` to `mount` or `umount` an INTERACTIVE UNIX Operating System file system.

10. Mount the file system as follows:

```
# /etc/mount /dev/dsk/f0q15dt /memo
```

The file system `/memo` is now associated with a directory in the `root` file system. As long as the `memo` file system is mounted on `/memo`, you can create and modify files on it as if it were an extension to the fixed disk.

11. A `lost+found` directory should be created on the file system for use by `fsck`:

```
# mkdir /memo/lost+found
# mklost+found /memo/lost+found
```

Mounting a file system at a directory that does not match the file system name produces a warning message defining what has been mounted. For example, to mount /dev/dsk/f0q15dt (file system name is memo) as directory /mnt, enter the following command line:

```
# /etc/mount /dev/dsk/f0q15dt /mnt
```

The following warning message will be displayed:

```
/etc/mount: warning: <memo> mounted as </mnt>
```

Unmounting a File System

When you have finished using the file system, you should unmount it. This is done with the umount command. All files in the file system to be unmounted must be closed, and you must change to a directory not in this file system. For example, if your current directory (pwd) is in the file system you want to unmount, you must change out of the file system before executing the umount command. Otherwise, you will get the following message:

```
/etc/umount: /memo busy
```

To unmount a file system from a 1.2 MB diskette, type the following:

```
# /etc/umount /dev/dsk/f0q15dt
```

If the file system is unmounted cleanly, there will be no need to run fsck the next time it is mounted. If it does not unmount cleanly, the next attempt to mount it will produce the following error message:

```
mount: possibly damaged file system
```

If this should happen, you can do one of two things:

- Run `fsck` on the file system and mount it again by typing:

```
# /etc/fsck /dev/dsk/f0q15dt
```

- Mount the file system with read permission only as follows:

```
# /etc/mount /dev/dsk/f0q15dt /memo -r
```

`root` *File System Free Space*

A predetermined and finite amount of disk space is allocated for the `root` file system. The unoccupied disk space within this area, called free space, allows for additional and temporary files and often serves as a scratch pad for certain system programs. System administration and other types of programs require `root` file system free space to run. It is recommended that you try to avoid using all the space in the `root` file system. If you should run out of space in `root`, a message similar to this is displayed:

```
NOTICE: no space on ESDI/IDE/MFM/RLL Controller <athd> DISK Target 0, LUN 0 partition 1
```

If you see this message, you should manually remove the files you do not need from the `root` file system. Since the system creates the file `/etc/mnttab` during startup time, it is recommended that you save at least 10 free blocks in the `root` file system before shutting down the machine. The `df` command can be used to find out how many free blocks are in your file systems. Refer to *df*(1M) for more information.

File System Maintenance 4

When the INTERACTIVE UNIX Operating System is first installed and initialized, a `root` (/) file system is automatically created. Depending on the size of your fixed disk, one or more file systems, named `/usr`, `/home`, `/home2`, and so on, may also be created. See the *INTERACTIVE UNIX System Installation Guide* for more information about system partitions.

Refer to the *INTERACTIVE UNIX System User's Guide* for information about using the `sysadm` command for:

- Formatting a diskette
- Copying files to a diskette
- Creating a file system on a diskette
- Mounting a file system
- Unmounting a file system
- Checking and repairing a file system

Checking and Repairing a File System

Every time a file is modified, the INTERACTIVE UNIX Operating System does a series of file system updates. These updates, when written to the disk, produce a consistent file system. The components of a file system are:

superblock
: The superblock defines the internal structure and size of a file system. There is one superblock for each file system.

inodes

An information node (or inode) is the internal definition of a file or directory. An inode contains information about the type of file, the number of directory entries linked to the file, a list of blocks claimed by the file, and the size of the file.

data blocks

A data block can contain either directory entries or file data. Each directory entry consists of a file name and an inode number. Each data block contains 1024 bytes.

indirect blocks

Indirect blocks are needed to reference the data blocks of large files (more than 10 blocks long). There are three types of indirect blocks: single-indirect, double-indirect, and triple-indirect.

first free-list block

The free-list blocks are lists of all the blocks not allocated to the superblock, inodes, or existing files. The superblock points to the first free-list block.

File System Reliability Features

The INTERACTIVE UNIX System is always checking to see if your file systems are in working order. Your system has several reliability features built in. For example:

- When a file is written to the fixed disk, its inode and blocks are written in an order that ensures maximum reliability. This is known as ordered writes.
- System buffers are periodically written to the fixed disk to keep the file contents up to date. This is known as automatic update.
- If the file system becomes corrupted, you will be required to run the `fsck` program to clean up the file system before mounting it. This ensures the reliability of all computer-mounted file systems.

The `fsck` Program

The `fsck` program is a file system check-and-repair program that uses information that is stored in the file system to check for inconsistencies. When the INTERACTIVE UNIX Operating System is booted, your computer runs a consistency check on the status of the root file system. If a potential problem exists, or if the information it checks does not match, the fsck program automatically detects the problem or inconsistency and attempts to repair it. Because fsck runs automatically on the root file system when the system is booted, you should not have to run fsck manually for the root file system. The fsck command is usually located in /etc.

The fsck program can also be run manually to check diskettes that have INTERACTIVE UNIX System file systems on them; or if you suspect something is wrong with your file system, you may want to check it. This should *only* be attempted by expert users.

Refer to the *INTERACTIVE UNIX System User's Guide* for information about using the Check a Diskette File System option from the File System Management menu from the Diskette Management menu under the Disk menu to run fsck on damaged diskettes and using the Check a Fixed Disk File System option on the Fixed Disk Management menu under the Disk menu to run fsck on a damaged file system on a fixed disk.

If the system experiences a power failure, it is usually possible to recover from it without suffering extensive file system damage. However, if the system experiences a hardware failure, the file system may be damaged beyond repair.

If file system damage is relatively minor, fsck will automatically repair many of the problems without assistance from you. In some cases the system asks you to confirm the action it plans to take. If fsck cannot repair the problem, you may have to reinstall a backup version of the file system.

After the initial setup, fsck performs successive phases of tests over the file system, including cleanup and checking blocks and sizes, path names, connectivity, reference counts, and the free-block list (possibly rebuilding it).

When an inconsistency is detected, fsck reports the error condition to the user. If a response is required, fsck prints a prompt message and waits for a response. The following paragraphs explain the meaning of an error condition, the possible responses, and the related error conditions.

The error conditions are organized by the "phase" of the fsck program in which they can occur.

For a list and explanation of the error messages you could encounter when using fsck, refer to "fsck Error Messages" later in this chapter.

Phase 1: Check Blocks and Sizes

This phase concerns itself with the inode list. Activities include checking inode types, setting up the zero-link-count table, examining inode block numbers for bad or duplicate blocks, checking inode size, and checking inode format.

Phase 1B: Rescan for More DUPS

When a duplicate block is found in the file system, the file system is rescanned to find the inode that previously claimed that block.

Phase 2: Check Path Names

This phase concerns itself with removing directory entries pointing to error-conditioned inodes from Phase 1 and Phase 1B. Checks are run for `root` inode mode and status, directory inode pointers in range, and directory entries pointing to bad inodes.

Phase 3: Check Connectivity

This phase concerns itself with the directory connectivity seen in Phase 2. This part lists error conditions resulting from unreferenced directories and missing or full `lost+found` directories.

Phase 4: Check Reference Counts

This phase concerns itself with the link count information seen in Phase 2 and Phase 3. This part lists error conditions resulting from unreferenced files; missing or full `lost+found` directories; incorrect link count for files, directories, and special files; unreferenced files and directories; bad and duplicate blocks in files and directories; and incorrect total free inode counts.

Phase 5: Check Free List

This phase concerns itself with the free-block list. This part lists error conditions resulting from bad blocks in the free-block list, bad free-block count, duplicate blocks in the free-block list, unused blocks from the file system not in the free-block list, and an incorrect total free-block count.

Phase 6: Salvage Free List

This phase concerns itself with the free-block list reconstruction. This part lists error conditions resulting from the blocks-to-skip and blocks-per-cylinder values.

Cleanup

Once a file system has been checked, a few cleanup functions are performed. This lists the following advisory messages about the file system and modifies the status of the file system:

```
***** FILE SYSTEM STATE SET TO OKAY *****
```

A flag in the superblock will be set to indicate that the file system is not corrupted and can be mounted.

```
X files Y blocks Z free
```

This advisory message indicates that the file system checked contained *X* files using *Y* blocks leaving *Z* blocks free in the file system.

`fsck` *and the* `root` *File System*

`root` is the only file system that can (and must) be checked while mounted. Automated mechanisms are provided for checking the `root` file system. These mechanisms handle a dirty `root` when booting and periodic checks during shutdown. You can also force a check on shutdown. These mechanisms hide the messages from `fsck`. If they were not hidden, you would see the error message described next.

```
***** ROOT FILE SYSTEM WAS MODIFIED *****
```

This advisory message indicates that the `root` file system was modified by `fsck`. If a system reboot is necessary, `fsck` with the `-b` option forces an automatic reboot and prints the following message:

```
**** SYSTEM WILL REBOOT AUTOMATICALLY ****
```

If you decide not to use the automated mechanisms, the `-b` option is not used, and a system reboot is necessary. Press the Reset button on your computer; if your computer doesn't have a reset button, turn the computer off and then on again.

The automated procedures establish the proper environment (no processes accessing files) for checking `root`.

Running `fsck` *Manually*

To run the `fsck` program manually, the file system must be unmounted (with the exception of the `root` file system). The legal `fsck` options are `-b`, `-f`, `-y`, `-n`, `-s`, `-S`, `-t`, `-q`, and `-D`. The `-y` option is recommended for `fsck`. This option answers yes to all questions prompted by `fsck` and requires no user intervention. Use the following command line for `fsck`:

```
# /etc/fsck -y special
```

Your screen will look similar to this:

```
/dev/dsk/c0t0s3
File System: /usr   Volume: c0t0

** Phase 1 - Check Blocks and Sizes
POSSIBLE FILE SIZE ERROR I=321

POSSIBLE FILE SIZE ERROR I=394

** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Free List
 411 files 4394 blocks 8880 free
```

Refer to *fsck*(1M) for additional information. If you attempt to use the `fsck` command on a mounted file system other than the `root` file system, the following message is displayed:

```
/dev/dsk/?? is a mounted file system, ignored.
```

?? is the special device name.

Bad Block Handling

The requirements of bad block handling fall into six categories:

- Dynamic handling of bad blocks
- Maintaining a bad block mapping table
- Detecting bad blocks
- Detecting unreadable blocks
- Reporting bad blocks
- Remapping bad blocks

Dynamic Handling of Bad Blocks

The basic requirement for the bad block handling feature is that it must be done dynamically, without user intervention. Dynamic handling provides immediate attention to the problem and thus minimizes data loss. It also avoids errors that may be introduced by the user. The current implementation reports problems to the console as they are found without retaining the messages in a log.

Maintaining a Bad Block Mapping Table

The bad block mapping table is created and stored on the fixed disk when the disk is first formatted. It consists of a bad block list and a list of alternate blocks (commonly called the "alternate sector list" or "surrogate images"). These two lists are in a one-to-one correspondence.

The bad block list is used to record the address, on disk, of the blocks that are bad. The alternate sector list is used to record the address, on disk, of all the reserved sectors to be used as alternates for bad blocks.

Detecting Bad Blocks

The bad block handling feature should be able to detect two different types of problems:

- Marginal blocks
- Unreadable blocks

A marginal block is a block that is readable, but with some difficulty. That is, the fixed disk controller's Error Correction Code (ECC) algorithm has to be used to successfully read the block.

Detecting Unreadable Blocks

An unreadable block is a harder problem to solve. There are two possible solutions. One method deals with the possible reconstruction of data to minimize data loss; the other simply accepts that the data is lost.

Reconstruction of data requires that a thorough and extensive analysis of the block in question is done before any kind or form of data repair can be attempted.

While this method offers higher data conservation, its design and implementation require a considerable amount of time and effort. Implementation of this method did not occur at this time; however, it is being considered as a future extension to the bad block handling feature.

It should be noted that by having an implementation in place that detects and takes care of marginal blocks, the incidence of potentially unreadable blocks is greatly reduced.

Reporting Bad Blocks

The system displays an error message when bad blocks are encountered. The following message may appear on the console:

```
*** DEVICE ERROR: Data Address Mark not found ***
*** Controller 0 (Primary AT Hard Disk), Disk Drive 0,
*** Absolute Sector # 168328
***
```

The system reports that absolute sector number 168328 on disk drive 0, controller 0 is bad. Note that the driver string, `Primary AT Hard Disk`, may be different depending on the controller configuration.

Remapping Bad Blocks

Bad blocks may be remapped into the reserved alternate sectors using the `-A` option of the `mkpart` command. (Refer to *mkpart*(1) for more information.) For example, to remap the bad sector in the above example, type:

```
# mkpart -A 168328 disk_c0t0
```

The `mkpart` command can also remap multiple bad blocks on a single command line. Thus, to remap bad sectors 100000 and 200000, type:

```
# mkpart -A 100000 -A 200000 disk_c0t0
```

Recovery of the INTERACTIVE UNIX System

If the message `/unix is missing or corrupted` appears when booting your computer or when the file system becomes corrupted to the point where the system is inoperable, you will need to replace `/unix` with the default `/unix`. Try the following:

1. Turn the machine off.
2. Insert the *Boot/Install* diskette into the diskette drive and turn the machine back on.
3. When `Booting the INTERACTIVE UNIX Operating System` appears, press the spacebar.
4. When prompted for the name of a kernel to boot, type:

```
/maint
```

and press Enter. The INTERACTIVE UNIX System will boot up and, after several messages, the shell prompt appears. You are now running off of a RAM disk.

5. Run `fsck` on the `root` file system on your fixed disk (from the `root` prompt) by typing:

```
# /etc/fsck /dev/dsk/0s1
```

   The `fsck` command will either run with `no` errors or request action from the user on repairing the file system. Most of the time answering `yes` to the questions asked by `fsck` will be sufficient, but be aware that this could remove some files.

6. Mount the fixed disk (device `0s1`) by typing:

```
# mount /dev/dsk/0s1 /mnt
```

7. Mount the *Boot/Install* diskette:

```
# /etc/mount /dev/dsk/f0 /mnt/mnt -r
```

8. Copy the kernel (`/unix`) from the *Boot/Install* diskette to the fixed disk by typing:

```
# cp /mnt/mnt/unix /mnt/unix
```

9. Unmount the diskette by typing:

```
# umount /dev/dsk/f0
```

10. Remove the *Boot/Install* diskette and type:

```
# shutdown
```

This unmounts the fixed disk and shuts down your system cleanly. Press any key to reboot when prompted.

Note – When you get the INTERACTIVE UNIX System login prompt, make sure you log in as `root`. You should then reinstall all drivers previously installed using the `kconfig` command. Refer to *kconfig*(1) for more information.

`fsck` *Error Messages*

This section describes the error messages you might receive when using `fsck`.

Initialization

Before a file system check can be performed, certain tables have to be set up and certain files opened. This section describes the opening of files and the initialization of tables. Error conditions resulting from command line options, memory requests, opening of files, status of files, file system size checks, and creation of the scratch file are listed below. The `fsck` program terminates on initialization errors.

Error Messages

`Bad -t option Illegal scratch file` *device_name*
: The `-t` option is not followed by a file name. The `fsck` program terminates on this error condition.

`Invalid -s argument, defaults assumed`
: The `-s` option is not suffixed by 3, 4, or blocks-per-cylinder:blocks-to-skip. The `fsck` program assumes a default of 400 blocks-per-cylinder and 7 blocks-to-skip.

`Incompatible options: -n and -q`
: It is not possible to remove FIFOs without modifying the file system. The `fsck` program terminates on this error condition.

`Incompatible options: -n and -s`
: It is not possible to salvage the free-block list without modifying the file system. The `fsck` program terminates on this error condition.

`Cannot fstat standard input`
: The attempt to `fstat` standard input failed. This error condition indicates a serious problem that may require additional assistance. The `fsck` program terminates on this error condition.

`Can't get memory`
: The request for memory for virtual memory tables failed. This error condition indicates a serious problem that may require additional assistance. The `fsck` program terminates on this error condition.

`Cannot open checklist file:` *F*
: The default file system *checklist* file *F* (usually `/etc/checklist`) cannot be opened for reading. The `fsck` program terminates on this error condition. Check access modes of *F*, or explicitly invoke `fsck` with the name of a device to check.

`Cannot stat root`
: The request for statistics about the `root` directory failed. This error condition indicates a serious problem that may require additional assistance. The `fsck` program terminates on this error condition.

`Cannot stat` *F*
: The request for statistics about the file system *F* failed. The `fsck` program ignores this file system and continues checking the next file system given. Check access modes of *F*.

F `is not a block or character device`
: The `fsck` program has been given a regular file name by mistake. It ignores this argument and continues checking the next file system given. Check the file type of *F*.

`Can't open` *F*
: The file system *F* cannot be opened for reading. The `fsck` program ignores this and continues checking the next file system given. Check the access modes of *F*.

`Size check: fsize` *X* `isize` *Y*
: More blocks are used for the inode list *Y* than there are blocks in the file system *X*, or there are more than 65,535 inodes in the file system. The `fsck` program ignores this file system and continues checking the next file system given.

`Can't create` *F*

The request to create a scratch file *F* failed. The `fsck` program ignores this file system and continues checking the next file system given. Check the access modes of *F*.

`CAN NOT SEEK: BLK` *B* `(CONTINUE)`

The request for moving to a specified block number *B* in the file system failed. The occurrence of this error condition indicates a serious problem that may require additional assistance.

Possible responses to the `CONTINUE` prompt are:

`YES`	Attempt to continue to run file system check. Often, however, the problem persists. This error condition does not allow a complete check of the file system. A second run of `fsck` should be made to recheck this file system. If the block was part of the virtual memory buffer cache, `fsck` will terminate with the message `Fatal I/O error`.
`NO`	Terminate program.

`CAN NOT READ: BLK` *B* `(CONTINUE)`

The request for reading a specified block number *B* in the file system failed. The occurrence of this error condition indicates a serious problem that may require additional assistance.

Possible responses to the `CONTINUE` prompt are:

`YES`	Attempt to continue to run file system check. Often, however, the problem persists. This error condition does not allow a complete check of the file system. A second run of `fsck` should be made to recheck this file system. If the block was part of the virtual memory buffer cache, `fsck` will terminate with the message `Fatal I/O error`.
`NO`	Terminate program.

`CAN NOT WRITE: BLK` *B* `(CONTINUE)`

The request for writing a specified block number *B* in the file system failed. The file system should not be opened for writing.

Possible responses to the CONTINUE prompt are:

- YES — Attempt to continue to run file system check. Often, however, the problem persists. This error condition does not allow a complete check of the file system. A second run of fsck should be made to recheck this file system. If the block was part of the virtual memory buffer cache, fsck terminates with the message Fatal I/O error.
- NO — Terminate program.

Phase 1: Check Blocks and Sizes

This phase concerns itself with the inode list. This part lists error conditions resulting from checking inode types, setting up the zero-link-count table, examining inode block numbers for bad or duplicate blocks, checking inode size, and checking inode format.

UNKNOWN FILE TYPE I=*I* (CLEAR)

The mode word of the inode *I* indicates that the inode is not a special character inode, regular inode, or directory inode.

Possible responses to the CLEAR prompt are:

- YES — Deallocate inode *I* by zeroing its contents. This invokes the UNALLOCATED error condition in Phase 2 for each directory entry pointing to this inode.
- NO — Ignore this error condition.

LINK COUNT TABLE OVERFLOW (CONTINUE)

An internal table for fsck containing allocated inodes with a link count of zero has no more room.

Possible responses to the CONTINUE prompt are:

- YES — Continue with program. This error condition does not allow a complete check of the file system. A second run of fsck should be made to recheck this file system. If another allocated inode with a zero link count is found, this error condition will be repeated.
- NO — Terminate the program.

B `BAD I=`*I*

Inode *I* contains block number *B* with a number lower than the number of the first data block in the file system or greater than the number of the last block in the file system. This error condition may invoke the `EXCESSIVE BAD BLKS` error condition in Phase 1 if inode *I* has too many block numbers outside the file system range. This error condition invokes the `BAD/DUP` error condition in Phase 2 and Phase 4.

`EXCESSIVE BAD BLKS I=`*I* `(CONTINUE)`

There is more than a tolerable number (usually 10) of blocks claimed by other inodes.

Possible responses to the `CONTINUE` prompt are:

`YES`	Ignore the rest of the blocks in this inode and continue to check using the next inode in the file system. This error condition does not allow a complete check of the file system. A second run of `fsck` should be made to recheck this file system.
`NO`	Terminate the program.

B `DUP I=`*I*

Inode *I* contains block number *B* that is already claimed by another inode. This error condition may invoke the `EXCESSIVE DUP BLKS` error condition in Phase 1 if inode *I* has too many block numbers claimed by other inodes. This error condition invokes Phase 1B and the `BAD/DUP` error condition in Phase 2 and Phase 4.

`EXCESSIVE DUPS BLKS I=`*I* `(CONTINUE)`

There is more than a tolerable number (usually 10) of blocks claimed by other inodes.

Possible responses to the `CONTINUE` prompt are:

`YES`	Ignore the rest of the blocks in this inode and continue to check using the next inode in the file system. This error condition does not allow a complete check of the file system. A second run of `fsck` should be made to recheck this file system.
`NO`	Terminate the program.

`DUP TABLE OVERFLOW (CONTINUE)`

An internal table in `fsck` containing duplicate block numbers has no more room.

Possible responses to the `CONTINUE` prompt are:

- `YES` Continue with program. This error condition does not allow a complete check of the file system. A second run of `fsck` should be made to recheck this file system. If another duplicate block is found, this error condition will repeat.
- `NO` Terminate the program.

`POSSIBLE FILE SIZE ERROR I=`*I*

The inode *I* size does not match the actual number of blocks used by the inode. This is only a warning. If the `-q` option is used, this message will not print.

`DIRECTORY MISALIGNED I=`*I*

The size of a directory inode is not a multiple of the size of a directory entry (usually 16). This is only a warning. If the `-q` option is used, this message will not print.

`PARTIALLY ALLOCATED INODE I=`*I* `(CLEAR)`

Inode *I* is neither allocated nor unallocated.

Possible responses to the `CLEAR` prompt are:

- `YES` Deallocate inode *I* by zeroing its contents.
- `NO` Ignore this error condition.

Phase 1B: Rescan for More DUPS

When a duplicate block is found in the file system, the file system is rescanned to find the inode that previously claimed that block. This part lists the error condition when the duplicate block is found.

B `DUP I=`*I*

Inode *I* contains block number *B* that is already claimed by another inode. This error condition invokes the `BAD/DUP` error condition in Phase 2. Inodes with overlapping blocks may be determined by examining this error condition and the `DUP` error condition in Phase 1.

Phase 2: Check Path Names

This phase concerns itself with removing directory entries pointing to inodes from Phase 1 and Phase 1B that have error conditions. This part lists error conditions resulting from `root` inode mode and status, directory inode pointers in range, and directory entries pointing to bad inodes.

`ROOT INODE UNALLOCATED. TERMINATING`

The `root` inode (always inode number 2) has no allocated mode bits. The occurrence of this error condition indicates a serious problem that may require additional assistance. The program stops.

`ROOT INODE NOT DIRECTORY (FIX)`

The root inode (usually inode number 2) is not a directory inode type.

Possible responses to the `FIX` prompt are:

`YES` Replace the `root` inode type to be a directory. If the `root` inode data blocks are not directory blocks, a large number of error conditions will be produced.

`NO` Terminate the program.

`DUPS/BAD IN ROOT INODE (CONTINUE)`

Phase 1 or Phase 1B has found duplicate blocks or bad blocks in the `root` inode (usually inode number 2) for the file system.

Possible responses to the `CONTINUE` prompt are:

`YES` Ignore `DUPS/BAD` error condition in `root` inode and attempt to continue to run the file system check. If `root` inode is not correct, then this may result in a large number of other error conditions.

`NO` Terminate the program.

I `OUT OF RANGE I=`*I* `NAME=`*F* `(REMOVE)`

A directory entry *F* has an inode number *I* that is greater than the end of the inode list.

Possible responses to the `REMOVE` prompt are:

`YES` The directory entry *F* is removed.

`NO` Ignore this error condition.

UNALLOCATED I=*I* OWNER=*O* MODE=*M* SIZE=*S* MTIME=*T* NAME=*F* (REMOVE)

A directory entry *F* has an inode *I* without allocate mode bits. The owner *0*, mode *M*, size *S*, modify time *T*, and file name *F* are printed. If the file system is not mounted and the -n option is not specified, the entry will be removed automatically if the inode it points to is size 0.

Possible responses to the REMOVE prompt are:

YES The directory entry *F* is removed.

NO Ignore this error condition.

DUP/BAD I=*I* OWNER=*O* MODE=*M* SIZE=*S* MTIME=*T* DIR=*F* (REMOVE)

Phase 1 or Phase 1B has found duplicate blocks or bad blocks associated with directory entry *F*, inode *I*. The owner *O*, mode *M*, size *S*, modify time *T*, and directory entry *F* are printed.

Possible responses to the REMOVE prompt are:

YES The directory entry *F* is removed.

NO Ignore this error condition.

DUP/BAD I=*I* OWNER=*O* MODE=*M* SIZE=*S* MTIME=*T* FILE=*F* (REMOVE)

Phase 1 or Phase 1B has found duplicate blocks or bad blocks associated with directory entry *F*, inode *I*. The owner *O*, mode *M*, size *S*, modify time *T*, and file name *F* are printed.

Possible responses to the REMOVE prompt are:

YES The directory entry *F* is removed.

NO Ignore this error condition.

BAD BLK *B* IN DIR I=*I* OWNER=*O* MODE=*M* SIZE=*S* MTIME=*T*

This message only occurs when the -q option is used. A bad block was found in DIR inode *I*. Error conditions looked for in directory blocks are nonzero padded entries, inconsistent "." and ".." entries, and embedded slashes in the name field. This error message indicates that the user should at a later time either remove the directory inode if the entire block looks bad or change (or remove) those directory entries that look bad.

Phase 3: Check Connectivity

This phase concerns itself with the directory connectivity seen in Phase 2. This part lists error conditions resulting from unreferenced directories and missing or full `lost+found` directories.

`UNREF DIR I=`*I* `OWNER=`*O* `MODE=`*M* `SIZE=`*S* `MTIME=`*T* `(RECONNECT)`

The directory inode *I* was not connected to a directory entry when the file system was traversed. The owner *O*, mode *M*, size *S*, and modify time *T* of directory inode *I* are printed. The fsck program forces the reconnection of a nonempty directory.

Possible responses to the `RECONNECT` prompt are:

- `YES` — Reconnect directory inode *I* to the file system in the directory for lost files (usually `lost+found`). This may invoke a `lost+found` error condition in Phase 3 if there are problems connecting directory inode *I* to `lost+found`. This may also invoke a `CONNECTED` error condition in Phase 3 if the link was successful.
- `NO` — Ignore this error condition. This invokes the `UNREF` error condition in Phase 4.

`SORRY, NO lost+found DIRECTORY`

There is no `lost+found` directory in the `root` directory of the file system; `fsck` ignores the request to link a directory in `lost+found`. This invokes the `UNREF` error condition in Phase 4. Check the access modes of `lost+found`.

`SORRY, NO SPACE IN lost+found DIRECTORY`

There is no space to add another entry to the `lost+found` directory in the `root` directory of the file system; `fsck` ignores the request to link a directory in `lost+found`. This invokes the `UNREF` error condition in Phase 4. Clean out unnecessary entries in `lost+found` or make `lost+found` larger.

`DIR I=`*I1* `CONNECTED, PARENT WAS I=`*I2*

This is an advisory message indicating a directory inode *I1* was successfully connected to the `lost+found` directory. The parent inode *I2* of the directory inode *I1* is replaced by the inode number of the `lost+found` directory.

Phase 4: Check Reference Counts

This phase concerns itself with the link count information seen in Phase 2 and Phase 3. This part lists error conditions resulting from unreferenced files; a missing or full `lost+found` directory; an incorrect link count for files, directories, or special files; unreferenced files and directories; bad and duplicate blocks in files and directories; and incorrect total free inode counts.

`UNREF FILE I=`*I* `OWNER=`*O* `MODE=`*M* `SIZE=`*S* `MTIME=`*T* `(RECONNECT)`

Inode *I* was not connected to a directory entry when the file system was traversed. The owner *O*, mode *M*, size *S*, and modify time *T* of inode *I* are printed. If the `-n` option is omitted and the file system is not mounted, empty files will be cleared automatically. Nonempty directories are not cleared.

Possible responses to the `RECONNECT` prompt are:

`YES` Reconnect inode *I* to file system in the directory for the lost files (usually `lost+found`). This can cause a `lost+found` error condition in Phase 4 if there are problems connecting inode *I* to `lost+found`.

`NO` Ignore this error condition. This invokes the `CLEAR` error condition in Phase 4.

`SORRY. NO lost+found DIRECTORY`

There is no `lost+found` directory in the `root` directory of the file system; `fsck` ignores the request to link a file in `lost+found`. This invokes the `CLEAR` error condition in Phase 4. Check access modes of `lost+found`.

`SORRY. NO SPACE IN lost+found DIRECTORY`

There is no space to add another entry to the `lost+found` directory in the `root` directory of the file system; `fsck` ignores the request to link a file in `lost+found`. This invokes the `CLEAR` error condition in Phase 4. Check size and contents of `lost+found`.

`(CLEAR)`

The inode mentioned in the error condition just prior to this one cannot be reconnected.

Possible responses to the `CLEAR` prompt are:

`YES` Deallocate the inode mentioned in the error condition just prior to this one by zeroing its contents.

`NO` Ignore this error condition.

`LINK COUNT FILE I=`*I* `OWNER=`*O* `MODE=`*M* `SIZE=`*S* `MTIME=`*T* `COUNT=`*X* `SHOULD BE` *Y* `(ADJUST)`

The link count for inode *I*, which is a file, is *X* but should be *Y*. The owner *O*, mode *M*, size *S*, and modify time *T* are printed.

Possible responses to the `ADJUST` prompt are:

`YES` Replace link count of file inode *I* with *Y*.

`NO` Ignore this error condition.

`LINK COUNT DIR I=`*I* `OWNER=`*O* `MODE=`*M* `SIZE=`*S* `MTIME=`*T* `COUNT=`*X* `SHOULD BE` *Y* `(ADJUST)`

The link count for inode *I*, which is a directory, is *X* but should be *Y*. The owner *O*, mode *M*, size *S*, and modify time *T* of directory inode *I* are printed.

Possible responses to the `ADJUST` prompt are:

`YES` Replace link count of directory inode *I* with *Y*.

`NO` Ignore this error condition.

`LINK COUNT` *F* `I=`*I* `OWNER=`*O* `MODE=`*M* `SIZE=`*S* `MTIME=`*T* `COUNT=`*X* `SHOULD BE` *Y* `(ADJUST)`

The link count of *F* inode *I* is *X* but should be *Y*. The file name *F*, owner *O*, mode *M*, size *S*, and modify time *T* are printed.

Possible responses to the `ADJUST` prompt are:

`YES` Replace link count of inode *I* with *Y*.

`NO` Ignore this error condition.

`UNREF FILE I=`*I* `OWNER=`*O* `MODE=`*M* `SIZE=`*S* `MTIME=`*T* `(CLEAR)`

Inode *I*, which is a file, was not connected to a directory entry when the file system was traversed. The owner *O*, mode *M*, size *S*, and modify time *T* of inode *I* are printed. If the `-n` option is omitted and the file system is not mounted, empty files will be cleared automatically. Nonempty directories are not cleared.

Possible responses to the `CLEAR` prompt are:

`YES`	Deallocate inode *I* by zeroing its contents.
`NO`	Ignore this error condition.

`UNREF DIR I=`*I* `OWNER=`*O* `MODE=`*M* `SIZE=`*S* `MTIME=`*T* `(CLEAR)`

Inode *I*, which is a directory, was not connected to a directory entry when the file system was traversed. The owner *O*, mode *M*, size *S*, and modify time *T* of inode *I* are printed. If the `-n` option is omitted and the file system is not mounted, empty files will be cleared automatically. Nonempty directories are not cleared.

Possible responses to the `CLEAR` prompt are:

`YES`	Deallocate inode *I* by zeroing its contents.
`NO`	Ignore this error condition.

`BAD/DUP FILE I=`*I* `OWNER=`*O* `MODE=`*M* `SIZE=`*S* `MTIME=`*T* `(CLEAR)`

Phase 1 or Phase 1B has found duplicate blocks or bad blocks associated with the file inode *I*. The owner *O*, mode *M*, size *S*, and modify time *T* of inode *I* are printed.

Possible responses to the `CLEAR` prompt are:

`YES`	Deallocate inode *I* by zeroing its contents.
`NO`	Ignore this error condition.

`BAD/DUP DIR I=`*I* `OWNER=`*O* `MODE=`*M* `SIZE=`*S* `MTIME=`*T* `(CLEAR)`

Phase 1 or Phase 1B has found duplicate blocks or bad blocks associated with directory inode *I*. The owner *O*, mode *M*, size *S*, and modify time *T* of inode *I* are printed.

Possible responses to the `CLEAR` prompt are:

`YES`	Deallocate inode *I* by zeroing its contents.
`NO`	Ignore this error condition.

`FREE INODE COUNT WRONG IN SUPERBLK (FIX)`

The actual count of the free inodes does not match the count in the superblock of the file system. If the `-q` option is specified, the count will be fixed automatically in the superblock.

Possible responses to the `FIX` prompt are:

`YES`	Replace count in superblock by actual count.
`NO`	Ignore this error condition.

Phase 5: Check Free List

This phase concerns itself with the free-block list. This part lists error conditions resulting from bad blocks in the free-block list, bad free-block count, duplicate blocks in the free-block list, unused blocks from the file system not in the free-block list, and an incorrect total free-block count.

`EXCESSIVE BAD BLKS IN FREE LIST (CONTINUE)`

The free-block list contains more than a tolerable number (usually 10) of blocks with a value less than the first data block in the file system or greater than the last block in the file system.

Possible responses to the `CONTINUE` prompt are:

`YES`	Ignore the rest of the free-block list and continue execution of `fsck`. This error condition will always invoke `BAD BLKS IN FREE LIST` error condition in Phase 5.
`NO`	Terminate the program.

`EXCESSIVE DUP BLKS IN FREE LIST (CONTINUE)`

The free-block list contains more than a tolerable number (usually 10) of blocks claimed by inodes or earlier parts of the free-block list.

Possible responses to the `CONTINUE` prompt are:

`YES`	Ignore the rest of the free-block list and continue execution of `fsck`. This error condition will always invoke `DUP BLKS IN FREE LIST` error condition in Phase 5.
`NO`	Terminate the program.

`BAD FREE BLK COUNT`
The count of free blocks in a free-list block is greater than 50 or less than 0. This error condition will always invoke the `BAD FREE LIST` condition in Phase 5.

X `BAD BLKS IN FREE LIST`
X blocks in the free-block list have a block number less than the first data block in the file system or greater than the last block in the file system. This error condition will always invoke the `BAD FREE LIST` condition in Phase 5.

X `DUP BLKS IN FREE LIST`
X blocks claimed by inodes or earlier parts of the free-list block were found in the free-block list. This error condition will always invoke the `BAD FREE LIST` condition in Phase 5.

X `BLK(S) MISSING`
X blocks unused by the file system were not found in the free-block list. This error condition will always invoke the `BAD FREE LIST` condition in Phase 5.

`FREE BLK COUNT WRONG IN SUPERBLOCK (FIX)`
The actual count of free blocks does not match the count in the superblock of the file system. When a file system is mounted, the free list is converted to a bit map and the blocks are removed from the free list (a small number of blocks are reserved on the free list). If the system goes down before the bit map is reconverted to the free list, then this message is displayed. Answering `yes` to the fix prompt will restore the blocks to the free list.

Possible responses to the `FIX` prompt are:

`YES`	Replace count in superblock by actual count.
`NO`	Ignore this error condition.

`BAD FREE LIST (SALVAGE)`

Phase 5 has found bad blocks in the free-block list, duplicate blocks in the free-block list, or blocks missing from the file system. If the `-q` option is specified, the free-block list will be salvaged automatically.

Possible responses to the `SALVAGE` prompt are:

`YES`	Replace actual free-block list with a new free-block list. The new free-block list will be ordered to reduce the time spent by the disk rotating into position.
`NO`	Ignore this error condition.

Phase 6: Salvage Free List

This phase concerns itself with the free-block list reconstruction. This part lists error conditions resulting from the blocks-to-skip and blocks-per-cylinder values.

`Default free-block list spacing assumed`

This is an advisory message indicating that blocks-to-skip (gap size) is greater than blocks-per-cylinder, blocks-to-skip is less than 1, blocks-per-cylinder is less than 1, or blocks-per-cylinder is greater than 500. The default values of 7 blocks-to-skip and 400 blocks-per-cylinder are used. These values were set previously when the `mkfs` (make file system) command was used to make the file system.

Using DOS-FSS and Mounting Diskette-Based File Systems

If you are using DOS-FSS and mounting an MS-DOS® (DOS) file system diskette, you should know how DOS-FSS applies the UNIX System concept of permissions to a DOS file system.

Under DOS-FSS, the user who mounts a file system (usually the system administrator) is considered to be the owner of the `root` of the mounted file system. The initial permissions are 777, that is, all users have read, write, and execute privileges. If the file system is mounted read-only, then the initial permissions are 555, that is, all users have read and execute privileges. Once the file system is mounted, the owner can change the permissions to restrict other users' access to it.

When an attempt is made to access an individual file or directory, the permissions for that file or directory are copied from the current permissions of the `root` of the mounted directory. For example, if the permissions of the `root` of the mounted directory are set to 777 and a user accesses a file in that directory for the first time, the file's permissions will also be set to 777. When a file system is unmounted, all permissions are effectively lost and must be reinstated the next time the file system is mounted.

To mount a DOS file system on a diskette, change to or log in as the superuser and type the command line:

```
# mount -f DOS /dev/dsk/f0q15dt /mnt
```

The file system is then mounted.

Note also that only one link is permitted to a DOS file or directory, and that any changes to the permissions or ownerships of files will be lost when the file system is unmounted.

File System Types

The INTERACTIVE UNIX Operating System is shipped with support for a variety of different file system types. Some are supplied with the base operating system while others are included with optional subsets and extensions. Each file system type has a unique structure. The UNIX System provides a consistent interface to the various file systems so that applications will be able to access files on a wide variety of local and remote storage devices.

Different file system types include network file systems such as NFS, the DOS file system type, and two standard UNIX System Laboratories, Inc. (USL) UNIX System file system types. File system types often vary in the size of their smallest logical unit. A 1K file system type makes a logical block size of 1K. On a typical device with a physical sector size of 512 bytes, this means that one logical block is equivalent to two physical blocks. With a 2K file system type, each disk allocation or access would take place on four physical blocks.

Note – The names of file system types that are available in your system are listed, one per file, in the directory `/etc/conf/sfsys.d`. Those files containing the file system type followed by a `Y` are enabled, and those type names followed by an `N` are disabled. The `kconfig Facilities` option under `Configure` on the bar menu will enable or disable the support for different file system types, respectively. When using the `-f` (type) flag of the `mount` command, use the file system type name exactly as it appears in these files. If the file contains a type name in uppercase letters, be sure to specify it in uppercase.

VF File System (VF)

The VF File System (VF) is designed for high-speed sequential reading and writing of large files, such as optical image files of a megabyte or more in size.

A VF file system can be very large, up to 2 terabytes (2 to the 41st power bytes), though individual files are subject to the standard 2 gigabyte UNIX System limit. The logical block size of the VF file system is 1K bytes.

When the VF file system is used as intended, its transfer rate to and from the disk approaches the maximum transfer rate of the drive itself. The programming interfaces of the S5 and VF file systems are identical, and thus when appropriate the VF file system can transparently replace the S5 file system. The on-disk structure of the VF file system is not, however, S5-compatible, which means it cannot be used when interoperability with S5 file systems on other UNIX System platforms is required.

The improved performance of the VF file system is achieved through four techniques:

- Files are allocated in large extents. An extent is a set of 1K blocks that are contiguous on the disk.
- Large data transfers are performed directly to and from the user program address space, rather than through the buffer cache.
- Small and partial-block transfers are buffered and cached to permit read-ahead and to avoid redundant disk operations.
- The data storage area on the disk is separated into directory and data zones to reduce seek times.

Data cannot be transferred directly to or from the user address space if the first block of a read or write does not start at a block boundary and/or the last block of a read or write does not end on a block boundary. Intermediate blocks are transferred directly. The highest transfer rates are achieved when the programmer follows these rules:

- Read and write in large units.
- Read and write in block multiples on block boundaries.
- Align buffers on page boundaries.

VF file systems are created by the *mkvffs*(1M) command. Other file system administration and maintenance commands, *fsck*(1M), *fsstat*(1M), *fstyp*(1M), and *mount*(1M), recognize and correctly process VF file systems. *fsdb*(1M) cannot be used on a VF file system.

When a file in the VF file system is first written, an initial primary extent is allocated. After all the blocks in the primary extent have been written, additional secondary extents are allocated as needed. To minimize file discontinuities, a new secondary extent is allocated contiguous to the preceding extent whenever possible.

Each file has primary and secondary extent size attributes (expressed in 1K logical blocks) that determine how many blocks the operating system will attempt to allocate when a new extent is needed. If a free extent of the specified size is not available, a smaller extent will be allocated. When a VF file system is created using *mkvffs*(1M), it is given default primary and secondary extent size attributes. A file inherits these default extent size attributes when it is created. The extent sizes of a file can be changed with an *fcntl*(2) system call as described in *vf*(4), which allows a program that knows the size of the file it will write to specify a primary extent size that can contain the entire file.

Each file in a VF file system is represented by an inode structure that contains information about the file. A VF inode has entries for up to 12 extent descriptors. When a newly allocated extent is contiguous to the preceding extent, it is simply added to the last extent descriptor, minimizing the number of extent descriptors used. Thus the limit of 12 extent descriptors per file will only be a problem if the file system is badly fragmented.

High Sierra File System (HS)

The High Sierra file system (HS) is currently the standard format for organizing data on Compact Disc Read-Only Memory (CD-ROM) disks. ISO 9660 is an almost identical international standard. The HS file system support in the INTERACTIVE UNIX System allows a user to mount a CD-ROM disk that is in High Sierra or ISO 9660 format as an extension of the UNIX System file system.

mount(1M) recognizes and correctly mounts HS file systems if HS support is configured into the kernel. Because CD-ROM disks cannot be written, the HS file system must be mounted read-only. Attempts to create, modify, or delete files or directories will fail, and the *mkfs*(1M) and *fsck*(1M) commands fail when applied to a CD-ROM. *fsdb*(1M) cannot be used on an HS file system.

The High Sierra and ISO 9660 standards specify that alphabetic characters in file and directory names are stored on the disk in upper case. Lowercase characters in file names presented to the HS support system are converted to uppercase, allowing users to enter file names in lower case. See also *mount*(1M).

The System V 1 Kilobyte File System (S51K and S5L)

The default file system is the System V 1 Kilobyte file system (S51K). The INTERACTIVE UNIX System will automatically create file systems of this type if you use any of the `sysadm addhd`, `mkfdfs`, or `mkfs` facilities. The normal installation will create file systems of this type for you automatically.

The structure of this file system follows the standards for UNIX System V on personal computers. All system utilities and applications for the INTERACTIVE UNIX System are compatible with this file system type. SunSoft, Inc. has added enhancements to the UNIX System kernel to improve the performance of the S51K file system type. This high performance Fast File System is structurally the same as the unenhanced S51K file system. However, the UNIX System kernel manages this file system in a more intelligent fashion while remaining compatible with USL file systems.

The S5L ("L" for long) file system is similar to the S51K file system but is enhanced to support file names of up to 512 characters.

The 2 Kilobyte File System Utility Package (S52K)

The optional 2 Kilobyte File System Utility Package (S52K) from USL accesses the disk in fixed, 4-block units. If it is installed and enabled using the `kconfig` program, it will:

- Decrease the available disk space
- Increase the memory requirements for the UNIX System kernel

There will be no other effects unless S52K file systems are actually created manually using the `mkfs2K` command (see *mkfs*(1M), *fsck*(1M), *fsdb*(1M), and *mount*(1M)). Since the standard versions of certain other commands and utilities are not compatible with the S52K file system type, alternate versions of these are supplied. These typically have names ending in `2K`. Not all user applications are compatible with the S52K. When using the 2K file system:

- Performance is several times slower than with the Fast File System.
- Even the smallest files require 2048 bytes of storage.
- Mounting and unmounting times are reduced.

If you are using media with a physical block size of 2K, you must use the S52K file system type. This file system is provided with the INTERACTIVE UNIX System mainly for consistency with USL releases.

The XENIX File System Type

The XENIX® file system type (`XX`) is used by the XENIX operating system. No provision is made in the INTERACTIVE UNIX Operating System for creating these file systems. The INTERACTIVE UNIX System does not support the mounting of any XENIX disk partitions that already exist on the fixed disk. Only diskette-based XENIX file systems can be mounted. The `xfsck` program is also provided for checking and repairing these file systems.

DOS File Systems

DOS file systems can be created on diskettes using the `dossette` command. Alternatively, the `mkdosfs` command can be used to create DOS file systems on both removable and fixed media. (However, you must use native DOS to create any DOS fixed disk partitions.) Both the `dosformat` command in `dossette` and the `FORMAT` utility in the optional VP/ix™ Environment extension can perform a low-level diskette format and then create the DOS file system in a single operation. The `mkdosfs` command requires that the diskette be preformatted by the `format` command.

Optical and Removable Media

Drives that support writable optical media can be prepared using the normal procedure and `sysadm addhd`. You can use `addhd` even if your writable media is contained in a removable cartridge. In some cases cartridges may need to be prepared or formatted using special utilities, although the INTERACTIVE UNIX System formatting procedure will correctly prepare most 512 and 1024 byte-per-sector optical cartridges. If the medium has 2K sectors, you must use the S52K file system type. If you are using a WORM drive, the fast file systems conversion from bitmap to free list may use up an unacceptable number of blocks, in which case the S52K file system may be preferable.

CD-ROM and other unwritable storage devices employ their own individual file system types. The most popular file system type for CD-ROM devices is the High Sierra file system type. Since writing on them is impossible, `mkfs` and `fsck` utilities are not supplied. These devices should always be mounted read-only. To prevent the UNIX System from becoming confused, it is important to unmount CD-ROM cartridges before removing them, just as you would any other UNIX System file system.

Using `kconfig` *to Tailor Your System Kernel* 5

Your system kernel is initially configured to support a basic hardware and software configuration. The default kernel includes drivers for the following configuration:

Hardware:

- A keyboard
- A fixed disk and diskette controller
- A monochrome, color, EGA, or VGA display adapter
- One serial communications port (`COM1`)
- Up to three parallel ports
- A real-time clock
- CMOS RAM
- Intel® 80387 floating point coprocessor (80287 supported on COMPAQ 386™)

UNIX System V Release 3.2 Software, including:

- MS-DOS file system service
- UNIX System V file system service (S51K and S5L file system types)

- Common interprocess communication routines, including:
  - Interprocess communication message facility
  - Interprocess communication semaphore facility
  - Shared memory
- 287 or 387 floating point coprocessor emulator

Your vendor may deliver a different default configuration. If your software did not come directly from SunSoft, check the documentation supplied with your system to determine its default configuration.

The kernel initially installed on the INTERACTIVE UNIX Operating System supports the standard default configuration shown above. You may have already reconfigured and rebuilt your kernel if you installed certain optional subsets and extensions provided with your system. The kernel will need to be reconfigured and rebuilt to support further changes you may want to make to your system if such changes include:

- Changing the configuration in a way that involves either the addition or removal of a device driver or the reconfiguration of the High Performance Device Driver (HPDD)
- Adding or removing certain software packages
- Changing the system memory size
- Adding or modifying tunable system parameters

Note that simply *installing* a driver or software package that contains a driver does *not add* that driver to your system configuration. After installing the driver you will still need to reconfigure and rebuild the kernel so that it will recognize and use the driver.

The `kconfig` program is a menu-driven interface used to configure, build, and install new kernels. The general uses of `kconfig` are discussed here. The information on specific hardware configurations is found in Chapter 6, "Hardware Compatibility and Configuration." For more specific technical information on the underlying programs and files used to configure, build, and install kernels, see *idbuild*(1M) and *inskern*(1).

The `kconfig` program is supplied with the Kernel Configuration subset, which must be installed before you can access it. To install this subset, follow the instructions for installing optional software in the *INTERACTIVE UNIX System Installation Guide*.

The displays in this chapter are examples only. Your screens may differ, depending on your software release and the drivers or facilities available on your machine.

The `kconfig` *Interface*

The kernel that is shipped with the INTERACTIVE UNIX System is built in such a way to allow you to install many different hardware configurations. This kernel may be used successfully without alteration, but you can substantially improve your system's performance by reconfiguring, building, and installing a kernel specifically tailored to your hardware configuration. The `kconfig` program makes it easy for you to reconfigure, build, and install a kernel that reflects your system needs. You will probably want to reconfigure the HPDD, system tunable parameters, and the individual driver configuration.

To find out how to use the `kconfig` interface program, refer to Chapter 1, "The Character User Interface," or use the `kconfig` program to access `Help` on the on-line bar menu and read `Using Help` and `Using the CUI System`.

Using the `kconfig` *Interface*

You must be the superuser (`root`) to run the `kconfig` program. The kernel configuration files are located in the `etc/conf` directory tree under the `root` directory (`/` by default). The `root` directory can be changed either by setting the environment variable `$ROOT` at the system prompt or by using the `-r` *root_directory* option on the `kconfig` command line. The configuration directory tree is referred to here as `$ROOT/etc/conf`. When running the `kconfig` program, the actual `root` directory will be substituted.

If the `UIcolor` environment variable has not been set, the system asks if you want to continue in color mode. Answer `n` to invoke the `kconfig` program in monochrome mode. To prevent this question from being asked each time you run `kconfig`, set the `UIcolor` environment variable for `root`. To do this, edit the file `/.profile` to add either:

```
UIcolor=TRUE
export UIcolor
```

for color monitors or

```
UIcolor=FALSE
export UIcolor
```

for monochrome monitors.

Log in as or `su` to `root` and type:

```
# kconfig
```

The system displays the `kconfig` bar menu:

```
 Configure   Build      Install    Manage     Help       Quit
Configure a kernel
┌──────────────────────────Kernel Configuration──────────────────────────┐
│                                                                         │
│                                                                         │
│                                                                         │
│                                                                         │
└─────────────────────────────────────────────────────────────────────────┘
```

Select `Configure` from the bar menu to access the `Configure` menu:

```
 Configure  Build      Install    Manage     Help       Quit
Use a previously built kernel as a baseline for building a new kernel
                        Kernel Configuration
 Clone Kernel
 Drivers
 Facilities
 HPDD         ->
 Tunables     ->
```

The options on the `Configure` menu allow you to use a previously built kernel as a baseline for building a new kernel and to configure drivers, facilities, the High Performance Device Driver, and kernel tunable parameters.

The following diagrams show the possible menu choices. To configure the High Performance Device Driver:

```
Configure
    |
+-------------------+
| Clone Kernel      |
| Drivers           |
| Facilities        |
| HPDD           -> |----+
| Tunables          |    |
| Quit              |    |
+-------------------+    |
          +--------------------------------+
          | View Current Settings          |
          | Reconfigure HPDD            -> |----+
          | Quit                           |    |
          +--------------------------------+    |
                          +----------------------------+
                          | Boot Controller            |
                          | Other Controllers       -> | . . . .
                          | RAM Disk                   |       .
                          | Remove HPDD                |       .
                          | Quit                       |       .
                          +----------------------------+       .
                                              +-------------------+
                                              | Second Controller |
                                              | Third Controller  |
                                              | Fourth Controller |
                                              | Fifth Controller  |
                                              | Sixth Controller  |
                                              | Quit              |
                                              +-------------------+
```

To configure the tunable parameters:

```
Configure
    |
 Clone Kernel
 Drivers
 Facilities
 HPDD
 Tunables    ->
 Quit

     Memory Size Defaults
     Parameters            ->
     Quit

          Kernel
          Device Driver
          Paging
          Streams
          Message
          Semaphore
          Shared Memory
          XENIX
          DMA
          2K File System
          All Parameters
          Quit
```

Each option is discussed in subsequent sections. When you exit the `Configure` menu, `kconfig` asks if you want to save the changes that have been made during the session. If you have changed your mind about the new configuration and do not want it to take effect, press Enter while the cursor is on the `NO` button. If you want to save the changes, press Enter while the cursor is on the `YES` button and the values that you configured will be saved.

Configuring the Kernel

The INTERACTIVE UNIX System is delivered with a number of device drivers that support a wide variety of devices. More information about these drivers can be found in Chapter 6, "Hardware Compatibility and Configuration." Refer to the *INTERACTIVE UNIX System Hardware Compatibility List* for a list of the hardware devices that are supported.

The kernel also contains the High Performance Device Driver, a system of device and controller drivers that together provide fast, consistent support for many disk and tape devices. Refer to Chapter 6, "Hardware Compatibility and

Configuration" for more information about the devices supported by the HPDD and to the *INTERACTIVE UNIX System Installation Guide* for information on the default configuration supported by the HPDD.

Many optional subsets, extensions, and facilities of the INTERACTIVE UNIX System contain device drivers. (Facilities may be software extensions to the kernel that provide special capabilities, such as interprocess communication.) and shared memory, or they may be packages that support specific hardware.) Depending on the type of driver, different mechanisms are used to configure the kernel to support it. However, you will always begin by selecting `Configure` from the `kconfig` bar menu. Each procedure is discussed in subsequent sections.

- If the device or driver is supported by the HPDD, select `Reconfigure HPDD` from the `HPDD` menu under `Configure`.
- If the driver is in the list of drivers provided with the INTERACTIVE UNIX System as noted in the *INTERACTIVE UNIX System Hardware Compatibility List*, select `Drivers` from the `Configure` menu. This option provides a list of the drivers available for addition or removal.
- If the driver is part of a facility provided with the INTERACTIVE UNIX System, it can be added or removed by selecting `Facilities` under the `Configure` menu. This option provides a list of the facilities available for addition or removal.
- To configure the system for a driver provided by a third party (that is, not SunSoft), follow the installation instructions provided by the vendor. If the driver is not part of the HPDD and the vendor used the integration scheme set up by SunSoft for the INTERACTIVE UNIX System, then after installation you should be able to remove the driver by selecting `Drivers` under the `Configure` menu.

To configure the kernel, do the following:

1. Log in as or `su` to `root`, type `kconfig`, and select `Configure` from the bar menu.

   Each option provides step-by-step procedures for the indicated task. If your configuration changes result in modifications to system files, then after quitting the `Configure` menu, `kconfig` asks if you want to save your changes.

2. If you have changed your mind about the new configuration and do not want it to take effect, select the NO button. If you want the new configuration, press Enter to accept the default (the OK button). You must then build and install the new kernel for this configuration to take effect.

Cloning a Kernel

Select `Clone Kernel` under `Configure` on the bar menu to duplicate the configuration of an existing kernel. This procedure changes the entries in the existing kernel configuration files:

```
/etc/conf/sdevice.d
/etc/conf/cf.d/mdevice
/etc/conf/cf.d/mtune
/etc/conf/cf.d/stune
```

This kernel can then be modified and used as a basis for building and installing a new kernel.

A list of the available kernels will be displayed. Select the one you want and press Enter while the `DUPLICATE` button is highlighted. Use the `Drivers`, `Facilities`, and `Tunables` options to modify this kernel to suit your needs. If no kernels are available, a message that there are no kernels to choose from is displayed.

Configuring Drivers

Individual drivers may need to be configured, depending on the configuration of system hardware. Select `Drivers` under `Configure` on the bar menu to include kernel device driver modules in the set of configured modules, which may also include third-party drivers that were previously installed and configured into the system. The system displays the drivers currently available on your machine and whether each is configured in to or out of your software. To review or modify a driver, move to the one you want and press Enter while it is highlighted. Another screen appears, describing the configuration of the driver. You can change the status of the driver (either include it or exclude it from the configuration of the next kernel you build), and you can modify individual driver parameters. The parameters that can be modified include the interrupt (IRQ) level, Direct Memory Access (DMA) channel, I/O address space, controller memory address space, and interrupt priority level. To configure a driver into the kernel configuration, move to the `Driver Status` radio button and select `on`; select `off` to configure this driver out of the kernel configuration.

Note – Certain drivers require that both driver entries be configured `on` for the driver to be fully functional. Both must also be configured out to remove them from the kernel configuration. Consult the driver documentation for any driver that you are unsure about configuring.

Configuring Facilities

Select `Facilities` under `Configure` on the bar menu to add or remove optional facilities that require kernel modification. Facilities may be software extensions to the kernel that provide special capabilities, such as interprocess communication and shared memory, or they may be packages that support specific hardware, such as INTERACTIVE TCP/IP.

The system displays the facilities currently available on your machine and whether each is configured in to or out of your software. To add or remove a facility, move to the one you want and press Enter while it is highlighted. The system will ask you to confirm the addition or removal of the facility selected. Configuring a facility configures all the drivers associated with it. In some cases, INTERACTIVE TCP/IP, for example, you will also need to configure the facility's driver(s) using the `sysadm` (system administration) program.

Configuring the High Performance Device Driver

The devices supported by the HPDD are fixed disk controllers and SCSI tape drives (see the *INTERACTIVE UNIX System Hardware Compatibility List* for a list of supported devices). The HPDD also supports a RAM disk, created by reserving a portion of the computer's available memory that is treated as if it were a disk storage device. (Refer to "Configuring a RAM Disk" later in this section for more information about RAM disks.) To support a configuration of these devices that differs from the default (documented in the *INTERACTIVE UNIX System Installation Guide*), you must reconfigure the HPDD.

Even if your hardware does not require you to reconfigure the HPDD, reconfiguring it will result in faster startup times. The kernel used to install the system and placed on the fixed disk supports many different fixed disk controllers. During system startup, the software attempts to initialize all of the supported controllers in order. If the controller is not found (because it is not part of your hardware configuration), the next one is tried until the entire list is finished. This process takes time because the system waits a certain amount of time to be sure that each controller is not responding. Reconfiguring the HPDD eliminates the time spent attempting to initialize all the supported controllers.

The `Reconfigure HPDD` menu presents options that enable you to:

- Add, remove, change, or specify the type of fixed disk controller
- Change the interrupt vector, DMA addresses, and I/O memory addresses used by a SCSI host adapter
- Add, remove, or change the type of tape drive connected to a SCSI host adapter
- Add, remove, or change the size of a RAM disk

After reconfiguring the HPDD, if you make any additional changes to your system involving these devices (other than replacing a standard AT® controller of one type with a different standard AT controller or adding or removing fixed disk drives), you will need to reconfigure the HPDD. For more complete information on the HPDD, see Chapter 6, "Hardware Compatibility and Configuration."

The High Performance Device Driver can control up to 32 devices per controller. There are four options when configuring the HPDD: selecting a boot controller, selecting a secondary controller, configuring a RAM disk, if desired, and removing the HPDD from the kernel configuration.

The Boot Controller

Select `Boot Controller` from the `Reconfigure HPDD` menu on the `HPDD` menu under `Configure`. The available controllers are displayed; move to the controller you want, and press Enter to select it. A form then appears with the pertinent information about the selected controller. In most cases it is not necessary to change any of the fields in this form; however, the fields may be altered to match your hardware specifications. After you are satisfied with the values in the form and have pressed Enter while on the `OK` button, you will be asked whether you want to use the basic or advanced configuration screen for the controller. The advanced configuration form is required *only* if there is more than one Logical Unit Number (LUN) for any particular target (SCSI ID number). Only seven available targets are presented on the basic configuration form. For the standard IDE/ESDI/RLL/MFM-type controller, select the basic form. Disks attached to these controllers correspond to target 0 for the first disk and target 1 for the second disk. Select the appropriate device for the target number on the form you selected and save your changes. The devices supported by the HPDD are magnetic disk, SCSI (Small Computer Systems Interface) tape devices, CD-ROM devices, and WORM (Write Once, Read Many) drives.

Target numbers and Logical Unit Numbers are adapted from the standards used by the Small Computer Systems Interface to describe the physical layout of the devices connected to the SCSI host bus adapter. In this layout, each adapter has 0 through 7 target numbers, one of which is reserved for the adapter itself. Each of the remaining seven target numbers can have up to eight devices attached to it. Each device is assigned a LUN number on the target. Devices are accessed by the controller through their target and LUN numbers.

Secondary Controllers

This procedure is essentially the same as configuring a boot controller. The same forms appear, in the same order. Up to five secondary controllers may be installed and configured. The only limitations are hardware considerations, such as IRQ and address space conflicts.

Configuring a RAM Disk

To configure a RAM disk, select `RAM Disk` from the `Reconfigure HPDD` menu on the `HPDD` menu under `Configure`. Press the spacebar to pop up a list of sizes for the RAM disk. Move to the one you want, and press Enter to select it. If the size you want is not on the list, select `other` and a new form appears. You can type in any size from 256 to 3840 4K pages.

Removing the HPDD From the Kernel Configuration

This option is available for third-party vendors who require that the HPDD not be configured into the system.

Caution – This option should *not* be used under any other circumstances.

Select `Remove HPDD` from the `Reconfigure HPDD` menu on the `HPDD` menu under `Configure` and follow the instructions.

Configuring Tunable System Parameters

You can adjust default system parameters to improve performance. For example, you may have installed a number of applications on your system and run out of a particular resource (such as Stream queues) when running them concurrently. You can select `Parameters` from the `Tunables` menu under `Configure` to increase the parameter being exceeded; however, you should be familiar with adjusting tunable parameters before using this option.

The system tunable parameters are used to set various table sizes and system thresholds to handle the expected system load. They can have a profound effect on system performance. The initial tunable parameter values delivered for your system are acceptable for most configurations and applications. If an application has special performance needs, you may have to experiment with different combinations of parameter values to find an optimal set, or a new tunable parameter may need to be defined for add-on drivers. The intended use of your computer and your observations on how well it is performing should be used as a guide in determining the need to adjust tunable parameters. To modify kernel parameters, the INTERACTIVE UNIX System kernel will have to be reconfigured, built, and installed.

There are two `kconfig` options that pertain to tunable parameters. If you have added more memory, refer to "Memory Size Defaults" (later in this section) to change a group of parameters to preset values based on the memory size. To add a new tunable parameter or to modify the value of an existing one, see "Parameters" below. After parameters or memory size have been modified, the kernel must be rebuilt and installed and the system rebooted for the new values to take effect.

Parameters

To alter the values of a kernel tunable parameter, select the `Parameters` menu from the `Tunables` menu, under `Configure`. If you are unsure of the classification of the tunable parameter you want to change, select `Miscellaneous`, which displays *all* the available tunable parameters. Move to the tunable parameter you want to alter. If you want to see a description of the parameter, press F1. If you want to change the current setting and default values, Tab to the field you want to change and type in the new value. "Parameter Descriptions," later in this section, provides a breakdown of system tunable parameters.

The tunable parameters are contained in the `mtune` and `stune` files in `$ROOT/etc/conf/cf.d`, the kernel configuration directory. Each parameter is assigned a default value, a minimum value, and a maximum value in the `mtune` file. To override the default value of a tunable parameter, the parameter is added to the `stune` file with a value that falls between the minimum and maximum values in `mtune`. These files can be examined to determine the tunable parameter settings for your computer. See *mtune*(4) and *stune*(4) for more information about these files.

Although it is not recommended that a parameter be set outside the `mtune` limits, if it is determined that a parameter must be set higher than permitted in the `mtune` file, you can edit the limits directly. Extreme care must be taken when modifying `mtune` to ensure that other values are not modified or deleted.

You should also be aware that the INTERACTIVE UNIX System kernel forces some parameters to be within preset limits. For example, the parameter `NOFILES` (number of open files per user process) is forced to fall within the 20/100 limit regardless of how the values in `mtune` and `stune` are adjusted. You should never modify an `mtune` value unless you have a full understanding of how the parameter is used in the INTERACTIVE UNIX System.

Memory Size Defaults

The standard kernel supplied with the INTERACTIVE UNIX System is optimized for a system with 4 MB of RAM. The INTERACTIVE UNIX System will use all of the available memory that is detected when the system is initially booted. However, if you have more memory than the standard configuration, you can substantially increase system performance by using the set of default parameters that have been optimized for the amount of memory closest to the amount on your system. This will dedicate more memory to system buffers and other kernel structures as well as increase certain process-related parameters.

You should configure the kernel for the largest memory size option that does not exceed the total amount available on your system. The INTERACTIVE UNIX System will operate unreliably if the system does not have as much memory as the kernel expects. If you later increase or decrease the amount of memory on your system so that it falls into a different size category, you should reconfigure the kernel.

If the memory size chosen is an increase over the size for which the system was previously configured, and if an existing value for a parameter is larger than the value for that parameter in the defaults for the chosen memory size, the parameter retains its present higher value.

Thus, if you have increased a parameter prior to using this option, the effects of your increase will not be reversed. However, if the memory size chosen is a decrease in size, all of the default values are used.

Select `Memory Size Defaults` from the `Tunables` menu under `Configure` and follow the instructions on-line.

Special Case Needs

At some point you will probably need to tune certain parameters for particular circumstances. One common need is to create very large files. If you are the superuser, you can modify the `ulimit` for the particular shell process that you are running as superuser. You can also modify the system ULIMIT parameter for all users. This parameter and other commonly encountered limits are summarized in the following table.

Improvement	Parameters
Improve system performance[1] when additional memory installed	NBUF[2], NHBUF[2]
Other performance-related system parameters	NAUTOUP[2], MAXSLICE, BDFLUSHR, AGEINTERVAL (also see paging parameters)
Increase system limits when additional memory installed (support more users, reduce chances of system problems at times of heavy load, etc.)	NCALL, NINODE[2], NS5INODE[2], NFILE[2], NPROC[2], NREGION[2], NCLIST[2] (also see message, semaphore, and shared memory parameters)
Users need to create bigger files	ULIMIT
Each user needs to open more files	NOFILES
Each user needs to run more processes	MAXUP[2]
Increase the size limit on arguments to the exec call	NCARGS
Increase the amount of user memory available to a single process	MAXUMEM
Other system limits that may be encountered	SHLBMAX[2], FLCKREC, SPTMAP, NUMXT, NUMSXT, PRFMAX (also see Streams parameters)
Miscellaneous	PUTBUFSZ, DO387CR3

1. Note that increasing the size of the buffer cache will increase the chances that data frequently read will be found in memory rather than on the disk. Depending on your system usage, an increase in the chance of reusing a data block may not yield an overall system performance improvement. In some system usage scenarios, it can provide a significant performance improvement.

2. This parameter is automatically adjusted when using the Memory Size Defaults option of Tunables under Configure on the kconfig bar menu. Refer to "Memory Size Defaults" earlier in this section.

Exceeding System Limits

There are cases when the INTERACTIVE UNIX System kernel will print a message on the console that system limits are being exceeded. Some of the messages are only to advise you of the problem. Others precede a system "panic," in which case additional diagnostic messages are printed, and the system hangs until you reboot it. If you encounter any of the messages in the next table, refer to the appropriate tunable parameter for additional information.

Kernel Messages and Associated Tunable Parameters	
Kernel Console Message	**Parameter**
`iget - inode table overflow`	`NINODE`
`Timeout table overflow`	`NCALL`
`File table overflow`	`NFILE`
`mfree map overflow` *n*[1]	`SPTMAP`
`Region table overflow`	`NREGION`
`Configured value of NOFILES` (*n*)[1] `is less than minimum (greater than maximum)`	`NOFILES`
`swapdel - too few free pages`	`MINASMEM`
`stropen: out of streams`	`NSTREAM`
`stropen: out of queues`	`NQUEUE`

1. The value *n* indicates the actual value encountered by the kernel.

Parameter Descriptions

The following sections provide a breakdown of system tunable parameters defined in `$ROOT/etc/conf/cf.d/mtune`. The parameter categories are as follows:

- Kernel Parameters
- Device Driver Parameters
- Paging Parameters
- Streams Parameters
- Message Parameters
- Semaphore Parameters
- Shared Memory Parameters
- XENIX Tunable Parameters
- Direct Memory Access (DMA) Parameters
- S52K (2K File System) Parameters

Note that the Streams parameters determine configuration of the STREAMS Facilities optional subset, and the S52K (2K File System) parameters control configuration of the 2 Kilobyte File System Utility Package optional subset. If these packages are not installed, adjusting the parameter values will have no effect upon your system's configuration.

General Kernel Parameters

The following parameters are associated with the kernel.

`NBUF`

The `NBUF` parameter specifies how many 1K system buffers to allocate. The INTERACTIVE UNIX System buffers form a data cache—a pool of memory in which recently accessed disk data is stored. Data stored in the cache is found faster when requested than data that has to be accessed on the disk, so "cache hits" reduce the number of disk accesses and thus may improve overall performance. The cache hit rate increases with the number of buffers. The entries usually range from 500 to 2000. Each buffer contains 1076 bytes. For optimal performance, 1K hash buffers (`NHBUF`) should be increased along with system buffers (`NBUF`).

`NCALL`

The `NCALL` parameter specifies how many call-out table entries to allocate. Each entry represents a function to be invoked at a later time by the clock handler portion of the kernel. This value must be greater than 2 and usually ranges from 30 to 70, with a default value of 50. Each entry contains 16 bytes. Software drivers may use call entries to check hardware device status. When the call-out table overflows, the system crashes and displays the following on the console:

```
PANIC: Timeout table overflow
```

`NCARGS`

The `NCARGS` parameter specifies the amount of stack space passed from one process to another during an `exec` system call. The message `argument list too long` will display when stack space is exceeded. Values to 128000 have been tested.

`NINODE`

The `NINODE` parameter specifies how many inode table entries to allocate. Each table entry represents an in-core inode that is associated with an active file, such as a current directory, an open file, or a mount point. Changing this variable modifies the file control structure. The number of entries used depends on the number of opened files. Entries usually range from 400 to 800. The value for `NINODE` is directly related to `NFILE`, as `NINODE` must be equal to or greater than `NFILE`. `NINODE` should always be equal to `NS5INODE` and to `NDOSINODE`. If `NINODE` is greater than `NS5INODE`, the system will be unusable. When the inode table overflows, the following message is sent to the console:

```
WARNING: i-node table overflow
```

`NS5INODE`

`NS5INODE` should always be equal to the value `NINODE` and to `NDOSINODE`.

`NFILE`

The `NFILE` parameter specifies how many open file table entries to allocate. Each entry represents an open file. Entries usually range from 400 to 800. Each entry contains 12 bytes. `NFILE` relates to the `NINODE` entry and should be less than or equal to `NINODE`. The `NFILE` control structure operates in the same manner as the `NINODE` structure. When the file table overflows, the following warning message is displayed on the system console:

```
NOTICE: file table overflow
```

As a reminder, this parameter does not affect the number of open files per process (refer to the `NOFILES` parameter).

`NMOUNT`

The `NMOUNT` parameter specifies how many mount table entries to allocate. Each entry represents a mounted file system. The `root` (`/`) file system is always the first entry. When the table is full, the `mount` command fails with an error message. Since the mount table is searched linearly, this value should be as low as possible.

NPROC

The NPROC parameter specifies how many process table entries to allocate. Each table entry represents an active process. The swapper is always the first entry, and /etc/init is always the second entry. The number of entries is determined by the number of terminal lines available and the number of processes spawned by each user. The average number of processes per user ranges from two to five (refer to MAXUP, default value 30). When the table is full, the fork system call fails and displays an error message and you will be unable to execute additional commands. The NPROC entry usually ranges from 100 to 200. The maximum value is 1000.

NREGION

The NREGION parameter specifies how many region table entries to allocate. Each NREGION entry contains 36 bytes. Most processes have three regions: text, data, and stack. Additional regions are needed for each shared memory segment and shared library (text and data) attached. However, the region table entry for the text of a "shared text" program is shared by all processes executing that program. Each shared memory segment attached to one or more processes uses another region table entry. A good starting value for this parameter is about 3.5 times NPROC. If the system runs out of region table entries, the following message is displayed on the console:

```
Region table overflow
```

NCLIST

The NCLIST parameter specifies how many character list buffers to allocate. Each buffer contains up to 64 bytes. The buffers are dynamically linked to form input and output queues for the terminal lines and other slow-speed devices. The average number of buffers needed per terminal ranges from 5 to 10. Each entry (buffer space plus header) contains 72 bytes. When all clist buffers are full, input and output characters dealing with terminals are lost, although echoing continues.

MAXUP

The MAXUP parameter specifies how many concurrent processes an ordinary user is allowed to run, usually from 30 to 50. This value should not exceed the value of NPROC. NPROC should be at least 10 percent more than MAXUP. This value is a per user identification number, not per terminal. For example, if 12 people are logged in on the same user identification, each with an average of 3 processes, the default limit would be reached very quickly. The maximum value is 1000.

`NOFILES`

The `NOFILES` parameter specifies the maximum number of open files per process. The default is 60. Unless an application package recommends that `NOFILES` be changed, the default setting of 60 should be used.

The Bourne shell uses three file table entries: standard input, standard output, and standard error (0, 1, and 2 are usually reserved for `stdin`, `stdout`, and `stderr`, respectively). Thus, the usable value of `NOFILES` is three less than shown as the number of other open files available per process. If a process requires up to three more than this number, the standard files must be closed. *This should not be done.*

If the configured value of `NOFILES` is greater than the maximum (100) or less than the minimum (20), the configured value is set to 100 in the first case and 20 in the second, and a `NOTICE` message is sent to the console.

`NHBUF`

The `NHBUF` parameter specifies how many "hash buckets" to allocate for 1K buffers. These are used to search for a buffer by a device number and block number rather than making a linear search through the entire list of buffers. *This value must be a power of 2.* Each entry contains 12 bytes. The `NHBUF` value must be chosen so that the value `NBUF` divided by `NHBUF` is approximately equal to 4.

`NPBUF`

The `NPBUF` parameter specifies how many physical I/O buffers to allocate. One I/O buffer is needed for each physical read or write currently active. Each entry contains 52 bytes. The default value is 20.

`NAUTOUP`

The `NAUTOUP` parameter specifies the number of seconds between automatic file system updates. A system buffer is written to the fixed disk when it has been memory-resident for the interval specified by the `NAUTOUP` parameter. Specifying a smaller limit increases system reliability by writing the buffers to disk more frequently and decreases system performance. Specifying a larger limit increases system performance at the expense of reliability. This parameter, along with BDFLUSHR, controls the behavior of the bdflush daemon process.

BDFLUSHR
: The BDFLUSHR parameter specifies the rate in seconds for checking whether the file system buffers need to be written to the disk. The default is 1 second. This parameter, along with NAUTOUP, controls the behavior of the bdflush daemon process.

MAXPMEM
: The MAXPMEM parameter specifies the maximum amount of physical memory (in pages). The default value of 0 specifies that all available physical memory be used.

SHLBMAX
: The SHLBMAX parameter specifies the maximum number of shared libraries that can be attached to a process at one time.

FLCKREC
: The FLCKREC parameter specifies the number of records (areas in a file) that can be locked by the system. The default value is 100. Each entry contains 28 bytes.

PUTBUFSZ
: The PUTBUFSZ parameter specifies the size of a circular buffer, putbuf, that is used to contain a copy of the last PUTBUFSZ characters written to the console by the operating system. The contents of putbuf can be viewed using the crash command.

MAXSLICE
: The MAXSLICE parameter specifies in clock ticks the maximum time slice for user processes. After a process executes for its allocated time slice, that process is suspended. The operating system then dispatches the highest priority process and allocates to it MAXSLICE clock ticks. MAXSLICE is usually the number of clock ticks in 1 second.

ULIMIT
: The ULIMIT parameter specifies, in 512-byte blocks, the size of the largest file that an ordinary user may create or write. The default value is 4194303; that is, the largest file an ordinary user may write is

2 megabytes. The superuser may write a file as large as the file system can hold. The ULIMIT parameter does not apply to reads; any user may read a file of any size.

UAREAUS and UAREARW

The UAREAUS and UAREARW parameters control the ability of a user process to read and write its user area in the kernel. These parameters can be set to 0 or 1 (the default), with the following effects:

UAREAUS	UAREARW	**User Area Accessibility**
0	0	Cannot be read or written
0	1	Cannot be read or written
1	0	Entire area can be read, but not written
1	1	Entire area can be read, first page can be written

As far as is known, no combination of these parameters compromises fundamental system security. The first page of the user area contains only the kernel stack and floating point state save area. However, some administrators may prefer to configure their systems with the user area inaccessible to the user. Note the following constraints:

- If the system does not have a 387 floating point coprocessor, the user area must be readable and writable. UAREARW and UAREAUS must be set to 1 (default).
- If the system has a 387 floating point coprocessor and the older B1 stepping version of the i386™ or compatible CPU, the U-area must be readable. Set UAREAUS to 1 and UAREARW to 0.
- If the system has an i486™ DX, Pentium™, or compatible CPU, or later stepping version of a 387 floating point coprocessor, the user area need not be accessible. Set UAREAUS and UAREARW to 0.

SPTMAP

The SPTMAP parameter determines the size of the map entry array used for managing kernel virtual address space. *You should not modify this parameter.*

`PIOMAP`

The `PIOMAP` parameter determines the size of the map entry array used by the kernel programmed I/O (PIO) breakup routine. This routine allows device drivers to do programmed I/O of large data blocks at interrupt level by breaking the data blocks into smaller data units. *You should not modify this parameter.*

`PIOMAXSZ`

The `PIOMAXSZ` parameter determines the maximum number of pages to use at one time for programmed I/O. *You should not modify this parameter.*

`DO387CR3`

The `DO387CR3` parameter controls the setting of high-order bits of Control Register 3 (CR3) when an 80387 math coprocessor is installed.

Device Driver Parameters

The following parameters control various data structure sizes and other limits in base system device drivers.

`NUMXT`

The `NUMXT` parameter determines the number of layers a subdevice can configure to support bitmapped display devices such as the AT&T BLIT or the AT&T 5620 terminal.

`NUMSXT`

The `NUMSXT` parameter determines the number of shell layers a subdevice can configure.

`NCPYRIGHT`

The `NCPYRIGHT` parameter defines the size of a kernel data structure used to print console initialization messages. *You should not modify this parameter.*

`NKDVTTY`

The `NKDVTTY` parameter determines the number of virtual terminals (ttys) supported by the console keyboard driver. *You should not modify this parameter.*

`PRFMAX`

The `PRFMAX` parameter determines the maximum number of text symbols that the kernel profiler (`/dev/prf`) will be able to properly process.

`TTHOG`

The `TTHOG` parameter determines the terminal input buffer threshold. It sets the number of characters that can be entered into a tty line buffer and remain unprocessed, before the data is discarded.

`RAMDSIZE`

The `RAMDSIZE` parameter determines the maximum size of a RAM disk. The default maximum is 3840 4K pages or 15 MB.

Paging Parameters

A paging daemon, `vhand`, is responsible for freeing up memory as the need arises. It uses a "least recently used" algorithm to approximate process working sets, and it writes out those pages that have not been modified during some period of time to the disk. The page size is 4096 bytes. When memory is exceptionally tight, the working sets of entire processes may be swapped out.

The following tunable parameters determine how often `vhand` and `bmapflush` run and under what conditions. The default values should be adequate for most applications.

`AGEINTERVAL`

The `AGEINTERVAL` parameter specifies the number of clock ticks a process runs before its pages are aged.

`VHNDFRAC`

The `VHNDFRAC` parameter determines the initial value for the system variable `VHANDL`. `VHANDL` is set to the maximum user-available memory divided by `VHNDFRAC`, or the value of `GPGSHI`, whichever is larger. The value of `VHANDL` determines when the paging daemon `vhand` runs. The amount of available free memory is compared with the value of `VHANDL`. If free memory is less than `VHANDL`, the paging daemon `vhand` is awakened.

The default is 16. Decreasing the value makes the daemon more active. The value must be between 0 and 25 percent of available memory.

`GPGSLO`
: The `GPGSLO` parameter specifies the low-water mark (in pages) of free memory before `vhand` will start to steal pages from processes. The default is 25. Increase the value to make the daemon more active; decrease the value to make the daemon less active (this value must be an integer greater than or equal to 0 and less than `GPGSHI`).

`GPGSHI`
: The `GPGSHI` parameter specifies the high-water mark (in pages) of free memory before `vhand` will stop stealing pages from processes. The default is 40. Increase the value to make the daemon more active; decrease the value to make the daemon less active. The value must be an integer greater than `GPGSLO`, and less than 25 percent of the number of pages of available memory.

`GPGSMSK`
: The `GPGSMSK` parameter is a mask used by the paging daemon. The default is 1056. *This value should not be changed.*

`MAXSC`
: The `MAXSC` parameter specifies the maximum number of pages that will be swapped out in a single operation. The default is 1.

`MAXFC`
: The `MAXFC` parameter specifies the maximum number of pages that will be added to the free list in a single operation. The default is 1.

`MAXUMEM`
: The `MAXUMEM` parameter specifies the maximum size of a user's virtual address space in pages. This value cannot be greater than 8192. The default is 4096.

`MINARMEM`
: The `MINARMEM` parameter specifies the minimum number of memory pages reserved for the text and data segments of user processes.

`MINASMEM`
: The `MINASMEM` parameter is a threshold value that specifies the number of memory and swap pages reserved for system purposes (unavailable for the text and data segments of user processes).

MINHIDUSTK

The MINHIDUSTK parameter specifies the minimum data relocation value that allows the user stack and data to share a page table. *This value should not be changed.*

MINUSTKGAP

The MINUSTKGAP parameter specifies the minimum data relocation value that allows the user stack and data to share a page table. *This value should not be changed.*

Streams Parameters

The following tunable parameters are associated with Streams processing. The values have no effect on the system unless the STREAMS Facilities optional subset is installed.

NQUEUE

The NQUEUE parameter determines the number of Streams queues to be configured. Queues are always allocated in pairs, so this number should be even. A minimal Stream contains four queues (two for the Stream head, two for the driver). Each module pushed on a Stream requires an additional two queues. A typical configuration value is 4*NSTREAM.

NSTREAM

The NSTREAM parameter determines the number of "Stream-head" (stdata) structures to be configured. One is needed for each Stream opened, including both Streams currently open from user processes and Streams linked under multiplexers. The recommended configuration value is highly application-dependent, but a value ranging from 32 to 40 usually suffices for running a single transport provider with moderate traffic.

NSTRPUSH

The NSTRPUSH parameter determines the maximum number of modules that may be pushed onto a Stream. This is used to prevent an errant user process from consuming all of the available queues on a single Stream. By default this value is 9, but in practice, existing applications usually push at most four modules on a Stream.

NSTREVENT

The NSTREVENT parameter determines the initial number of Stream event cells to be configured. Stream event cells are used for recording process-specific information in the poll system call, in the implementation of the STREAMS I_SETSIG ioctl, and in the kernel bufcall() mechanism. A minimum value to configure is the expected number of processes simultaneously using the poll system call times the expected number of Streams being polled per process, plus the expected number of processes expected to use Streams concurrently. The default is 256. Note that this number is not necessarily a hard upper limit on the number of event cells that will be available on the system (see MAXSEPGCNT).

MAXSEPGCNT

The MAXSEPGCNT parameter determines the number of additional pages of memory that can be dynamically allocated for event cells. If this value is 0, only the allocation defined by NSTREVENT is available for use. If the value is not 0 and if the kernel runs out of event cells, under some circumstances it will attempt to allocate an extra page of memory from which new event cells can be created. MAXSEPGCNT places a limit on the number of pages that can be allocated for this purpose. Note that once a page has been allocated for event cells, it cannot be recovered later for use elsewhere. It is recommended that the NSTREVENT value be set to accommodate most load conditions and that MAXSEPGCNT be set to 1 to handle exceptional load cases.

NMUXLINK

The NMUXLINK parameter determines the maximum number of multiplexer links to be configured. One link structure is required for each active multiplexer link (STREAMS I_LINK ioctl). This number is application-dependent; the default allocation guarantees availability of links.

STRMSGSZ

The STRMSGSZ parameter determines the maximum allowable size of the data portion of any Streams message. This should usually be set just large enough to accommodate the maximum packet size restrictions of the configured Streams modules. If it is larger than necessary, a single write or putmsg can consume an inordinate number of message blocks. The recommended value of 4096 is sufficient for existing applications.

STRCTLSZ

The STRCTLSZ parameter determines the maximum allowable size of the control portion of any Streams message. The control portion of a putmsg message is not subject to the constraints of the minimum/maximum packet size, so the value entered here is the only way of providing a limit for the control part of a message. The recommended value of 1024 is more than sufficient for existing applications.

NBLK*n*

NBLK4 through NBLK4096 control the number of Streams data blocks and buffers to be allocated for each size class. Message block headers are also allocated based on these numbers; the number of message blocks is 1.25 times the total of all data block allocations. This provides a message block for each data block, plus some extras for duplicating messages (kernel functions dupb() and dupmsg()). The optimal configuration depends on both the amount of primary memory available and the intended application.

STRLOFRAC

The STRLOFRAC parameter determines the percentage of data blocks of a given class at which low priority block allocation requests are automatically failed. For example, if STRLOFRAC is 40 and there are 48 256-byte blocks, a low priority allocation request will fail when more than 19 256-byte blocks are already allocated. The parameter is used to help prevent deadlock situations by starving out low priority activity. A recommended value of 80 works well for current applications. STRLOFRAC must always be in the range greater than or equal to 0 and less than or equal to STRMEDFRAC.

STRMEDFRAC

The STRMEDFRAC parameter determines the percentage cutoff at which medium priority block allocations are failed (see STRLOFRAC above). The recommended value of 90 works well for current applications. STRMEDFRAC must always be in the range greater than or equal to STRLOFRAC and less than or equal to 100.

Note that there is no cutoff fraction for high priority allocation requests; it is effectively 100.

NLOG

The NLOG parameter determines the number of minor devices to be configured for the log driver; the active minor devices will be 0 through (NLOG–1). The recommended value of 3 services an error logger (strerr) and a trace command (strace), with one left over for miscellaneous usage. If only an error logger and a tracer are to be supported, this number can be set to 2. If there are several daemons for an application that may be submitting log messages, this number can be increased to accommodate the extra users.

NUMSP

The NUMSP parameter determines the number of Streams pipe devices (/dev/sp) supported by the system. The maximum value is 256.

NUMTIM

The NUMTIM parameter determines the maximum number of Streams modules that can be pushed by the Transport Library Interface (TLI). This value controls the number of data structures used to hold pushed Streams modules configuration data. Each entry contains 12 bytes.

NUMTRW

The NUMTRW parameter determines the number of Transport Library Interface read/write data structures to allocate in kernel data space. Each entry contains 12 bytes.

Message Parameters

The following tunable parameters are associated with interprocess communication messages.

MSGMAP

The MSGMAP parameter specifies the size of the control map used to manage message segments. The default value is 100. Each entry contains 8 bytes.

MSGMAX

The MSGMAX parameter specifies the maximum size of a message. The default value is 2048. Although the maximum possible size the kernel can process is 64 kilobytes –1, the mtune limit is 8192.

MSGMNB

The MSGMNB parameter specifies the maximum length of a message queue. The default value is 4096.

MSGMNI
: The MSGMNI parameter specifies the maximum number of message queues system wide (ID structure). The default value is 50.

MSGSSZ
: The MSGSSZ parameter specifies the size, in bytes, of a message segment. Messages consist of a contiguous set of message segments large enough to fit the text. The default value is 8. The value of MSGSSZ times the value of MSGSEG must be less than or equal to 131,072 bytes (128 kilobytes).

MSGTQL
: The MSGTQL parameter specifies the number of message headers in the system and therefore the number of outstanding messages. The default value is 40. Each entry contains 12 bytes.

MSGSEG
: The MSGSEG parameter specifies the number of message segments in the system. The default value is 1024. The value of MSGSSZ times the value of MSGSEG must be less than or equal to 131,072 bytes (128 kilobytes).

Semaphore Parameters

The following tunable parameters are associated with interprocess communication semaphores.

SEMMAP
: The SEMMAP parameter specifies the size of the control map used to manage semaphore sets. The default value is 10. Each entry contains 8 bytes.

SEMMNI
: The SEMMNI parameter specifies the number of semaphore identifiers in the kernel. This is the number of unique semaphore sets that can be active at any given time. The default value is 10. Each entry contains 32 bytes.

SEMMNS
: The SEMMNS parameter specifies the number of semaphores in the system. The default value is 60. Each entry contains 8 bytes.

SEMMNU

The SEMMNU parameter specifies the number of undo structures in the system. The default value is 30. The size is equal to 8 times (SEMUME + 2) bytes. (Refer also to SEMUME.)

SEMMSL

The SEMMSL parameter specifies the maximum number of semaphores per semaphore identifier. The default value is 25.

SEMOPM

The SEMOPM parameter specifies the maximum number of semaphore operations that can be executed per semop system call. The default value is 10. Each entry contains 8 bytes.

SEMUME

The SEMUME parameter specifies the maximum number of undo entries per undo structure. The default value is 10. The size is equal to 8 times SEMMNU bytes. (Refer also to SEMMNU.)

SEMVMX

The SEMVMX parameter specifies the maximum value a semaphore can have. The default value is 32767, which is the maximum value for this parameter.

SEMAEM

The SEMAEM parameter specifies the adjustment on exit for maximum value, alias semadj. This value is used when a semaphore value becomes greater than or equal to the absolute value of semop, unless the program has set its own value. The default value of 16384 is the maximum value for this parameter.

Shared Memory Parameters

The following tunable parameters are associated with interprocess communication shared memory.

SHMMAX

The SHMMAX parameter specifies the maximum shared memory segment size. The default value is 1048576.

SHMMIN

The SHMMIN parameter specifies the minimum shared memory segment size. The default value is 1.

`SHMMNI`
: The `SHMMNI` parameter specifies the maximum number of shared memory identifiers system wide. The default value is 100. Each entry contains 52 bytes.

`SHMSEG`
: The `SHMSEG` parameter specifies the number of attached shared memory segments per process. The default value is 6. The maximum value is 15.

`SHMALL`
: The `SHMALL` parameter specifies the maximum number of in-use shared memory text segments. The default value is 512.

XENIX Tunable Parameters

The following describes the XENIX tunable parameters.

`DSTFLAG`
: The `DSTFLAG` parameter specifies the `dstflag` described for the XENIX `ftime`(S) system call.

`NEMAP`
: The `NEMAP` parameter specifies the maximum number of I/O translation mappings.

`TIMEZONE`
: The `TIMEZONE` parameter specifies the `timezone` setting referred to in the XENIX `ftime`(S) system call. Note that the time zone value is a system default time zone and not the value of the `TZ` environment variable.

`XSEMMAX`
: The `XSEMMAX` parameter specifies the maximum number of XENIX special semaphores allowed system wide. The minimum value for `XSEMMAX` is 20; the maximum value (and default) is 60.

`XSDSEGS`
: The `XSDSEGS` parameter specifies the maximum number of XENIX special shared data segments allowed system wide. The minimum value for `XSDSEGS` is 1; the maximum value (and default) is 25.

XSDSLOTS

The maximum number of XENIX special shared data segment attachments system wide is XSDSEGS times XSDSLOTS. The minimum value for XSDSLOTS is 1; the maximum value (and default) is 3.

Direct Memory Access (DMA) Parameter

The following parameter is associated with Direct Memory Access.

DMAEXCL

The DMAEXCL parameter specifies whether simultaneous DMA requests are allowed. Some computers have DMA chips that malfunction when more than one allocated channel is used simultaneously. For all installations on these computers, DMAEXCL is set to 1 by default. On computers that do not have this problem, set DMAEXCL to 0 to allow simultaneous DMA on multiple channels.

S52K (2K File System) Parameters

Note that these parameters have no effect on your system unless the 2 Kilobyte File System Utility Package optional subset is installed.

S52KNBUF

The S52KNBUF parameter specifies how many 2K system buffers to allocate. This parameter performs the same function for 2K file systems that NBUF performs for 1K file systems. The entries usually range from 100 to 400. Each buffer contains 2100 bytes. 2K hash buffers (S52KNHBUF) should be increased along with S52KNBUF for optimal performance. If you configure 2K buffers in your system, you should reduce the number of 1K buffers (NBUF) to keep available memory at an acceptable level.

S52KNHBUF

The S52KNHBUF parameter specifies how many hash buckets to allocate for 2K buffers. These are used to search for a buffer by device number and block number rather than making a linear search through the entire list of buffers. *This value must be a power of 2.* Each entry contains 12 bytes. The S52KNHBUF value must be chosen so that the value of S52KNBUF divided by S52KNHBUF is approximately equal to 4.

Building a New Kernel

Select `Build` from the bar menu to review the configuration of the most recently configured kernel or to build that kernel.

After you have made changes to the current kernel configuration, you need to build a new kernel, install it in the appropriate directory, shut down the system, and reboot it.

Log in as or `su` to `root` and type `kconfig`. Select `Build` from the bar menu. The system kernel is automatically built, using the information you have already provided. If the kernel builds successfully, the following message is displayed:

```
Finished building unix system
```

After a successful kernel build, the system prompts you to type in a description of the kernel. Type something that will help you differentiate this kernel from other kernels you build. You may want to mention the drivers and facilities configured into this kernel. If you do not provide a description, the time and date of the build is used.

The system uses the modified configuration files to build a new kernel in a subdirectory under `$ROOT/etc/conf/kconfig.d`. The subdirectory has the name of the new kernel with `.d` appended to the end. In this example, the new kernel is `unix.1` and is located in `$ROOT/etc/conf/kconfig.d/unix.1.d`. The new kernel is linked to the `root` directory (`/`) if possible; otherwise it is copied there. When it is installed, it is linked from the `root` directory to the `/unix` file, overwriting the current copy of the kernel contained in `/unix`. Thus, `/unix` always reflects the latest installed version of the kernel. When the system is rebooted, the new kernel is in effect.

If the kernel fails to build, error messages are displayed. You can scroll through them using the up and down arrow keys. Correct the errors and build the kernel again.

Installing a Previously Built Kernel

The kconfig program keeps a copy of every kernel you build in a subdirectory under $ROOT/etc/conf/kconfig.d. You may have configured and built a new kernel but not yet installed it, or, at times, you may want to install one of the kernels that you have used in the past. You might want to install a previous version of the kernel if, for example:

- A new device driver has been installed, but the new kernel proves to be unreliable
- You remove a facility or driver and want to return to a kernel that is not configured for that facility or driver

After a kernel is built, it must be installed in order to have the configuration values take effect. To install a previously built kernel, type kconfig to access the bar menu and select Install. The system displays all the kernels contained in the $ROOT/etc/conf/kconfig.d directory. Move to the one you want and press Enter to select it. If the description line is insufficient to identify the kernel you want, you can review the precise configuration of any available kernel by highlighting the one you want to review in the list and then selecting the REVIEW button. The exact configuration of the selected kernel is then displayed. The description of this kernel can also be altered at this time.

If you are running kconfig from the console, the system will ask if you want to run shutdown. You will have the option to type in the time in seconds before the system will shut down and reboot with the newly installed kernel.

Managing Kernels

Select Manage from the bar menu to remove kernels from the system or to generate a report on a specific kernel. When a kernel is generated, it is placed in a file called $ROOT/etc/conf/kconfig.d/unix.*n*.d, along with all the configuration files that were used to create it. Use the Remove a Kernel option to remove an old kernel, the configuration files, and this directory, freeing file system space. The configuration of the kernel can be reviewed before removal.

If the changes to the system include changes in the hardware configuration, such as the addition of a new device, then after configuring, building, and installing the kernel and shutting down the system, you should:

- Power down the system.

- Physically install the hardware device (unless it is already present as a component of your basic system hardware) and set any jumpers or switches that are necessary.
- Use the manufacturer's *setup* program to configure your system, if necessary.
- Reboot the system.

Removing a Previously Built Kernel

Kernels use a large amount of space on your system. If you build several kernels, you may want to periodically delete those you do not anticipate using in the future, in order to free up space on your system. Select `Remove a Kernel` under `Manage` on the bar menu to remove previously built kernels.

Reporting On a Previously Built Kernel

A report generated from the configuration files of a previously built kernel may be produced by selecting `Report on Kernels` under `Manage` on the bar menu. The report contains a description of the kernel, the status of the drivers and facilities, HPDD information, and all the tunable system parameter information. This report can be directed to the screen, to a file, or to a printer (using the `/usr/ucb/lpr` program). To direct it to a file, select `File` in the form and a new form appears, asking for the full path name for the generated report. After completing the form, select the `GENERATE` button to obtain the report.

What to Do If the System Doesn't Boot

There is a remote possibility that a new kernel you have installed will not boot properly. For example, this can happen if a system parameter or some combination of parameters you have modified has built a kernel that will not initialize properly. In such an event, do the following:

1. Reboot the system.
2. When the message

```
Booting the INTERACTIVE UNIX Operating System...
```

is displayed, quickly press the spacebar and type `/OLD.unix` when prompted for the name of the kernel.

3. After the system is up, move the file named `/OLD.unix` to `/unix` and then move `/etc/conf/cf.d/OLD/*` to `/etc/conf/cf.d`. See *inskern*(1) for further information about which files were moved or copied during the installation of the new kernel.

Hardware Compatibility and Configuration 6

The hardware mentioned in this chapter is not meant to be an exhaustive list of the systems that are believed to be compatible with the INTERACTIVE UNIX Operating System or an endorsement of these systems. It is supplied for purposes of information only. No guarantees are expressed or implied. The definitive list of supported hardware can be found in the *INTERACTIVE UNIX System Hardware Compatibility List*. Additional information about driver configuration files can be found in the manual entries *mdevice*(4) and *sdevice*(4).

High Performance Device Driver

The High Performance Device Driver (HPDD) is a system of device and controller drivers that together provide fast, consistent support for many disk and tape devices. Underneath the HPDD "surface" are tape and disk device drivers that support operations for each of these device types. These drivers communicate via an HPDD-supplied interface to the controllers that actually drive their devices.

Tape devices on Small Computer System Interface (SCSI) host bus adapters are supported by the `tape` driver in the HPDD, except for the COMPAQ® SCSI tape adapter, which is *not* handled by the HPDD (refer to "COMPAQ SCSI Tape" later in this chapter). Non-SCSI tape devices are handled by other drivers, which are independent of the HPDD. Diskettes are also handled separately. (Refer to "The Diskette Driver" later in this chapter for more information about diskette devices.) MFM, RLL, ESDI, IDE, and SCSI disks, on either AT or Micro Channel® Architecture (MCA) buses, are handled by the disk driver, `dsk`, which also provides a RAM disk software driver.

Disk controllers for MFM, RLL, IDE, and ESDI drives on AT bus machines are handled by the `athd` module. On an MCA bus machine, MFM and RLL disk controllers use `mcst`, and ESDI controllers use `mcesdi`. SCSI host bus adapter modules are specific to their respective cards. Refer to the next table to find out how the adapter module names correlate to supported adapters. RAM disks are controlled by `gramd`.

The HPDD uses the SCSI naming convention to address cascaded peripheral systems with multiple devices. In earlier releases of the HPDD, only the first four available disks and first four available tape drives were accessible by users. This was insufficient to handle the 32 devices that can be connected to a SCSI bus running off a host bus adapter (HBA). Each HBA can connect up to seven target controllers, each of which can have eight devices (logical units). Each logical unit is assigned a number, which is known as a LUN or *l*ogical *u*nit *n*umber for short.

Under the new scheme, the HPDD can access up to 56 devices per HBA. Physical devices are addressed by the same HBA/target/LUN combination defined in the SCSI standards. The same scheme is applied to the AT attachment systems. An AT disk controller can be attached to a SCSI HBA and a direct couple disk to an embedded SCSI device with a fixed target/LUN combination. For example, the secondary disk on the primary AT fixed disk controller can be represented using the SCSI convention as HBA 0, target 1, LUN 0.

Compatibility

The HPDD supports the following classes of controllers.

Controller Module	Manufacturer	Controller
aha1520	Adaptec	AHA-1520, AHA-1522
aha1540	Adaptec	AHA-154*x*, AHA-164*x*, AHA-1740 in compatibility mode
aha1740	Adaptec	AHA-1740 in extended mode
aha274x	Adaptec	AHA-274*x* EISA
aha284x	Adaptec	AHA-284*x* VLB
ami	American Megatrends, Inc.	SCSI controller
athd	Various	All ST-506, ESDI, RLL, and IDE controllers on ISA or EISA machines
	COMPAQ	IDA in compatibility mode
bt742	BusTek	BT-742 in extended mode
dpt20xx	DPT	PM 20*xx* in extended mode
escx	Olivetti	BT-742 in extended mode
ida	COMPAQ	IDA in extended mode
mcesdi	Various	All ESDI controllers on MCA machines
mcis	IBM	IBM PS/2 MCA SCSI adapter
mcst	Various	All ST-506 controllers on MCA machines
ncr700	NCR	NCR 53C700 SCSI I/O processor
ncr810	NCR	NCR 53C810 NCR 53C820 NCR 53C825
tmc8x0	Future Domain	Various Future Domain SCSI adapters
wdasc	Western Digital	WD-7000ASC – FASST, WD-7000EX in compatibility mode
wdex	Western Digital	WD-7000EX in extended mode

Note that the `ncr700` driver is a single-threaded driver that was developed to support the Intel L-series systems with an on-board 32-bit NCR® 53C700 SCSI controller (see *ncr700*(7)). The NCR 53C700's I/O base address is at 0xcc0.

Hardware Configuration

The installation kernel is configured to recognize all of the Industry Standard Architecture (ISA) and Micro Channel Architecture (MCA) adapters listed in this section. The Extended Industry Standard Architecture (EISA) modules, `ida`, `aha1740`, `bt742`, and `wdex`, are not configured into the installation kernel because these adapters can be operated in compatibility mode. In this mode, they behave like their ISA counterparts. To use their extended mode capabilities, the extended mode module must be loaded using the `kconfig` program. In the case of `aha1740` and `wdex`, EISA configuration is also needed.

The Adaptec® AHA-1520 and Western Digital WD-7000ASC have an I/O address conflict. The Western Digital WD-7000ASC can only boot at base address 0x350. The standard Adaptec AHA-1520 can only boot at base address 0x340, and the on-board registers extend into the address space of the WD-7000ASC. If the boot adapter is the standard Adaptec AHA-1520, a secondary WD-7000ASC may only exist at a base address other than 0x350. If the boot adapter is a WD-7000ASC, you cannot use a standard Adaptec AHA-1520 as a secondary adapter. To remove the conflict, Adaptec can supply a nonstandard AHA-1520 adapter with an I/O base address of 0x140. The driver in the installation kernel will automatically find the adapter at the new address.

The Adaptec AHA-154*x* and Western Digital WD-7000ASC adapters can only access physical memory smaller than 16 MB, which is a problem when using these adapters in EISA machines with more than 16 MB of memory. To circumvent this problem, the `aha1540` and `wdasc` drivers can restrict the use of more than 16 MB of memory. If either driver is loaded and found to have devices attached, memory greater than 16 MB is disabled. These drivers are loaded in the installation kernel and may be used by EISA adapters in their compatibility mode. Memory truncation can be avoided by replacing Adaptec

AHA-154*x* and Western Digital WD-7000ASC adapters with their EISA counterparts and by running EISA adapters in their extended mode. This usually requires that the kernel be reconfigured using the `kconfig` program.

EISA Adapter	Has a compatability mode for
Adaptec AHA-1740	Adaptec AHA-1540
BusTek 742	Adaptec AHA-1540
AMI	Adaptec AHA-1540
WD-7000EX	WD-7000ASC
IDA	ATHD
DPT	ATHD

The default configuration is shown below.

Installation Kernel Configuration

Controller Module	I/O Addresses	IRQ	DMA Channel
`athd`	0x1f0-0x1f7,0x3f6	14	None
`tmc8x0`	None	5	None
`aha1520`	0x340	11	None
`aha1540`	0x330	11	5
`mcis`	0x3510	14	None
`mcst`	0x320	14	3
`mcesdi`	0x3510	14	5
`wdasc`	0x350	15	6
`dpt20xx`	0x230 (ISA only)	14	5 (ISA only)
`ncr700`	0xcc0	11	None (EISA only)

The HPDD can deal with as many adapters as there are slots available in the machine. An arbitrary limit of six is imposed by the `kconfig` program. There are limits on the number of adapters of a particular type that are allowed in any configuration. Adapters under the control of the `athd`, `mcst`, and `msesdi` drivers are limited to two per configuration. Other than that, the only limiting factors are hardware resources such as interrupt vectors, DMA channels, and I/O address space.

While the theoretical limit on the number of devices on a SCSI adapter is 56, 32 is the limit imposed by table sizes within the HPDD. Non-SCSI adapters are limited to two disks.

Configuring the High Performance Device Driver

If you add or remove a device that is supported by the HPDD, you will need to reconfigure it. Regardless of adding or removing devices, it is a good idea to configure the HPDD after you install your system because it will significantly increase the speed of booting the system.

Note – You must already have installed the Kernel Configuration optional subset in order to configure the HPDD.

To configure the HPDD, read the preceding sections carefully to determine any special requirements. Log in to the system as `root` (or use `su`), and type `kconfig` to invoke the `kconfig` program.

Press Enter to accept the default. The system displays the `kconfig` bar menu:

```
 Confi ure  Build     Install    Manage     Help       Quit
Confi ure a kernel
┌───────────────────────Kernel Configuration───────────────────────┐
│                                                                  │
│                                                                  │
│                                                                  │
│                                                                  │
│                                                                  │
│                                                                  │
└──────────────────────────────────────────────────────────────────┘
```

Select `Configure` from the bar menu to access the `Configure` menu. Select `HPDD`, then `Reconfigure HPDD`, and follow the instructions on-line for the changes you want to make. Refer to Chapter 5, "Using kconfig to Tailor Your System Kernel," for more information about reconfiguring the HPDD.

```
 Configure  Build      Install    Manage     Help       Quit
Select and configure the boot controller
                                Kernel Configuration
  Clone Kernel
  Drivers
  Facilities
  HPDD         ->
  Tunables     ->

   View Current Settings
   Reconfigure HPDD       ->
   Quit

    Boot Controller
    Other Controllers ->
    RAM Disk
    Remove HPDD
    Quit
```

When you exit the `Configure` menu, `kconfig` asks if you want to save the changes that have been made during the session. If you have changed your mind about the new configuration and do not want it to take effect, press Enter while the cursor is on the `NO` button. If you want to save the changes, press Enter while the cursor is on the `YES` button and the values that you configured will be saved.

You will then need to build and install the new kernel as described in Chapter 5, "Using kconfig to Tailor Your System Kernel." If you also plan to add new drivers for unsupported devices at this time, or otherwise reconfigure the kernel, you may want to do those tasks first and then build and install the kernel after all changes are done.

Autoconfiguration

Some HPDD driver modules can override the values configured in `dsk/space.c` and `tape/space.c` to match values presented by the adapter itself. During initialization, these modules look at the configuration information in the EISA NonVolatile Random Access Memory (NVRAM), Micro Channel POS registers, PCI configuration, or in special registers on the adapter itself. This information is checked against the software configuration in `dsk/space.c` and `tape/space.c`. If the information matches, initialization proceeds as normal. Several passes through the software configuration are made in order to get the best possible match. During pass 1, adapters are assigned positions if the software configuration matches the hardware configuration exactly. During pass 2, adapters are assigned positions if the software I/O base address matches the hardware I/O base address. During pass 3, adapters are assigned positions if the software interrupt vector matches the hardware interrupt vector. During pass 4, unallocated adapters are assigned based on slot order.

Autoconfiguration can alter the software configuration to match the hardware configuration, which can have some detrimental effects on system operation. When autoconfiguration changes an adapter's I/O base address, the change is transparent because many EISA adapters have slot-dependent I/O base addresses; thus, simply moving an adapter to a different slot should not prevent the system from booting. An interrupt vector mismatch, however, is not transparent. The interrupt vector given to the HPDD for an adapter may already be in use by another device. In this case, the interrupt is stolen from the other device for use by the HPDD, and the other device can no longer operate. When the interrupt vectors do not match, a screen that looks similar to this will appear during initialization:

```
--------------- HPDD IDA NOTICE ----------------
Kernel's IDA interrupt vector=15
EISA NV RAM configuration IDA interrupt vector=11
These two interrupts should be the same number!
Run kconfig, reconfigure the HPDD, and rebuild the kernel
-------- AUTO CONFIGURATION TAKING OVER --------
```

This should be resolved quickly by either giving the named adapter (in this case the Intelligent Drive Array (IDA)) a new interrupt using the appropriate configuration utility (or an on-board jumper for non-software-configured boards) or by changing the software configuration to match the hardware configuration using the `kconfig` program, or both.

Warnings

The following drivers provided with the INTERACTIVE UNIX Operating System are known to conflict with the default configuration of the HPDD.

Driver	Conflict
`logi`	Uses interrupt 5, conflicts with `tmc8x0`
`mouse`	Uses interrupt 5, conflicts with `tmc8x0`
`wt`	Uses interrupt 5, conflicts with `tmc8x0`

If any of these drivers are to be used in the same kernel as the HPDD configured with a conflicting controller, the conflict must be resolved by rejumpering one of the boards and changing the corresponding driver configuration files, using `kconfig`. Remember that if you are not actually using the conflicting HPDD controller, configuring the HPDD as described in the previous section will resolve the conflict.

Device Names

Disks

There are two sets of rules for device name construction as it applies to disks. The set you should use depends on the type of pseudo device (target or LUN) to which the real device is connected. If the connection is to a target pseudo device, `dsk0t`, for example, the device names follow the forms `c`*i*`t`*j*`p`*k* and `c`*i*`t`*j*`s`*k*, where *i* is the controller number (`0` in this case because `dsk0t` is being used), *j* is the target on the designated controller, and *k* is the partition number on that drive. All numbers are zero-based.

If the connection is to a LUN pseudo device, `dsk32`, for example, the device names follow the forms `c`*i*`t`*j*`l`*k*`p`*m* and `c`*i*`t`*j*`l`*k*`s`*m*, where *i* is the controller number (`3` in this case because `dsk32` is being used), *j* is the target number on the controller (2 in this case because dsk32 is being used), *k* is the LUN on the designated target on the designated controller, and *m* is the partition number (0-f) on that drive. All numbers are zero-based.

When a LUN pseudo device is being used, the first device on the designated target will have two names—the LUN-based name and the target-based name. This occurs because the target interface always accesses the first LUN on any target.

By default, the first two targets on the primary adapter have additional names constructed of three characters: a digit representing the target number (0 or 1), the letter p for non-remapped fdisk partitions or the letter s for remapped INTERACTIVE UNIX System partitions, and a hexadecimal digit for the partition number (0–9, a–f).

It is recommended that you use the full names for the sake of clarity.

The fdisk partitions 1–f correspond to those shown by the fdisk program. The fdisk table itself can only have four partitions, but DOS supports a sub-partitioning scheme via extended partitions. The first 11 of these are available through the suffixes p5 through pf.

The INTERACTIVE UNIX System partitions 1–f correspond to those in the UNIX System VTOC structure, maintained by the mkpart program (see *mkpart*(1M)). These partitions contain file systems, swap space, alternate sectors, the INTERACTIVE UNIX System bootstrap, and other utility areas.

Partitions p0 and s0 are special cases. The p0 device provides the entire disk device; the s0 device provides the entire INTERACTIVE UNIX System partition, which corresponds to one of the devices p1 through p4, depending on how the disk was partitioned with fdisk, except that bad sectors are mapped out.

Each disk pseudo driver has its own major numbers for the block and character interfaces. Major numbers are normally assigned by the kconfig program and both numbers should be the same. The target interface to the primary adapter (dsk0t) is always major number 0.

The minor numbers associated with device names are constructed of three fields. The high order bit (0x80) is the p bit and is 1 if a p device is being referenced. The next three bits (0x70) have a different meaning, depending on the type of pseudo driver attached. If the pseudo driver is a target driver (for example, dsk4t—adapter 4, first LUN on each target), then the three bits refer to the target number. If the pseudo driver is a LUN driver (for example, `dsk46`—adapter `4`, all LUN s on target `6`), then the three bits refer to the LUN number on the designated target. The last field (`0x0f`) is the partition number and corresponds to the last digit of the device name.

The RAM disk, if configured, is simply another target pseudo driver and has only one minor number, `0`.

The following table contains some examples of device names and their minor numbers. Note that the major numbers are not known for many devices since these are assigned by the `kconfig` program and thus vary from system to system.

Sample HPDD Device Names and Minor Numbers

Device Name	Major No.	Minor No.	Description
`0p0`	0	0x80	The whole disk on the first disk on the primary controller
`0p1`	0	0x81	The first `fdisk` partition on the primary controller
`0p2`	0	0x82	The second `fdisk` partition on the primary controller
`0p3`	0	0x83	The third `fdisk` partition on the primary controller
`c0t0p0`	0	0x80	Full name for `0p0`
`0s1`	0	0x01	Usually the INTERACTIVE UNIX System `root` file system on the first disk on the primary controller
`c0t0s0`	0	0x00	Full name for `0s0`
`1s1`	0	0x11	The first INTERACTIVE UNIX System partition on the second disk on the primary controller
`c0t1s1`	0	0x11	Full name for `1s1`
`c3t5s3`	Varies	0x53	Fourth adapter, target 5, INTERACTIVE UNIX System partition 3
`c1t2l3s4`	Varies	0x34	Second adapter, target 2, LUN 3, INTERACTIVE UNIX System partition 4
`ram`	Varies	0x00	The RAM disk

Special Disk Information

This version of the INTERACTIVE UNIX Operating System supports CD-ROM drives on the SCSI bus.

Note – Consult the *INTERACTIVE UNIX System Hardware Compatibility List* to see what devices are currently supported.

Raw device names appear in the `/dev/rdsk` directory. Device names depend on a host bus adapter number (HBA), SCSI target ID numbers (SCSI ID), and an optional logical unit number (LUN). Device names have the following syntax:

```
cdHBAtSCSI_ID[lLUN]
```

The CD-ROM major number is handled by pseudo device major numbers.

The minor numbers associated with CD-ROM device names are constructed from three fields. The high order bit (`0x80`) is the *p* bit; it is always 1 since with CD-ROM the entire physical disk is always referenced. The next three bits are the device index field (`0x70`). The last field is the partition number field (`0x0f`) and is always set to zero since CD-ROM disks do not use partitions. As with other devices, minor numbers can be set differently by configuring the HPDD manually.

All known SCSI CD-ROM drives have integrated SCSI controllers that only communicate with a single CD-ROM drive. Until controllers that support multiple devices are developed, CD-ROM drives must have a unique SCSI ID for a given HBA, and the logical unit number is generally internally set to 0. This means that the only devices that should be set to CD-ROM on the HPDD SCSI configuration screen are targets 0–6, all with LUN 0.

Tapes

This version of the INTERACTIVE UNIX Operating System supports cartridge tape drives on the SCSI bus (see *tape*(7)).

Note – Consult the *INTERACTIVE UNIX System Hardware Compatibility List* to see what other devices are currently supported.

As with disks, there are two sets of rules for tape device name construction. The set used depends on the type of pseudo device to which the real device is connected.

If the connection is to a target pseudo device, `tp0t`, for example, the device names are of the forms `rmt`*i*`t`*j* and `nrmt`*i*`t`*j*, where *i* is the controller number (`0` in this case because `tp0t` is being used) and *j* is the target on the designated controller. All numbers are zero-based. The initial `n` indicates that the tape will not rewind when closed.

If the device is connected is to a LUN pseudo device, `tp32`, for example, the device names are of the form `rmt`*i*`t`*j*`l`*k* `nrmt`*i*`t`*j*`l`*k*, where *i* is the controller number (`3` in this case because `tp32` is being used), *j* is the target number on the controller (`2` in this case because `tp32` is being used), and *k* is the LUN on the designated target on the designated controller. All numbers are zero-based.

When a LUN pseudo device is being used, the first device on the designated target has two names—the LUN-based name and the target-based name. This occurs because the target interface always accesses the first LUN on any target.

Each tape pseudo driver has its own major numbers for the character interface. Major numbers are normally assigned by the `kconfig` program. The minor numbers associated with device names are constructed from three fields. The three bits (`0x70` or the fifth, sixth, and seventh bits) have a different meaning, depending on the type of pseudo driver attached.

- If the pseudo driver is a target driver (`tp4t`, for example, which is adapter `4`, first LUN on each target), then the three bits refer to the target number.
- If the pseudo driver is a LUN driver (`tp46`, for example, which is adapter `4`, all LUN s on target `6`), then the three bits refer to the LUN number on the designated target.

- The second field (`0x0c`) is the tape behavior field and determines the "rewind on close" (`0x80`, the fourth bit) and "return immediately" (`0x40`, the third bit) characteristics of the unit. The last field (`0x03`) sets the density for the tape and is usually `0`.

HPDD Tape Minor Number Flag Bits	
Bit Mask	**Meaning**
0x08	Rewind on close. If not set, write a filemark or forward to next filemark on close, if opened for writing or read-only, respectively.
0x04	Return immediately. Tape motion *ioctl()* calls and closings should return immediately. If not set, these calls wait for tape motion to complete before returning.
0x03	Tape density. See the next table for the meaning of this field.

The tape density field above is defined below.

HPDD Tape Density Values			
Descriptor	**Value**	**Density Description**	
		9-Track BPI	**Cartridge Tracks/Size**
Default	0x00	Default density (defined by device driver)	
Low	0x01	800 BPI	9 track (60MB)
Medium	0x02	1600 BPI	15 track (125MB)
High	0x03	6250 BPI	18 track (150MB)

The tape device names created when configuring with the `kconfig` program depend on which pseudo devices exist. In any case, one tape must be designated as the default. In addition to its formal names, this tape will also be known by the names in the following table.

SCSI Tape Devices	
Name	**Description**
`/dev/ntape`	No rewind after closing
`/dev/rnmt0`	No rewind after closing
`/dev/tape`	Rewind after closing
`/dev/rmt0`	Rewind after closing
`/dev/rSA/tape`	Rewind after closing

Tape devices are created using the `node.d` file with the same name as the pseudo driver to which the device is connected. If the `kconfig` program is always used to configure the kernel, it should not be necessary to modify these files. In the case of conflict with other tape devices (the Wangtek tape driver, `wt`, for example), it may be necessary to modify a pseudo driver `node.d` file or the `/etc/conf/node.d/wt` file.

For example, if you have a Wangtek (`wt`) tape drive and a SCSI tape drive (for example, target 1 on the first adapter) on your system and you want the SCSI tape drive to be the default drive, edit `/etc/conf/node.d/wt` so that it includes the following:

```
wt      rnmt0   c      0
wt      rmt0    c      1
```

and edit `/etc/conf/node.d/tp0t` so that it includes:

```
tp0t      ntape          c      16
tp0t      nscsitape      c      16
tp0t      tape           c      24
tp0t      scsitape       c      24
tp0t      rSA/tape       c      24
```

Note that the names `scsitape` and `nscsitape` are arbitrary and can be any names that do not conflict with existing node names. Using the `kconfig` program to describe the layout of any devices will automatically create the appropriate node files.

If you are using an Exabyte drive, you must have it on-line before booting the new kernel.

Error Messages

`Command aborted with no apparent cause`
: This indicates an undetermined error that should be reported to your driver and/or controller supplier.

`Command timeout`
: When this error occurs on ESDI devices, it represents a hardware failure that must be investigated and repaired. On SCSI devices this may occur because of long, undisconnected operations competing on the SCSI bus, and should be retried later.

`Controller error or failure`
: This indicates a hardware failure that should be reported to your controller manufacturer.

`Correctable data error occurred`
: This indicates that the sector or block is becoming unreliable, and the data in it should be saved. On a disk device, the sector should be mapped out using the `mkpart` utility before any data is lost.

`Data address mark not found`
: This indicates a format failure on a disk. The sector should be remapped using the `mkpart` program. The data in that sector is lost.

`Data overrun`
: This is a driver error in programming a DMA operation and should be reported to your driver supplier.

`Data underrun`
: This is a driver error in programming a DMA operation and should be reported to your driver supplier.

`Error during Format operation`
: The medium may be damaged and cannot be successfully formatted for use.

`Illegal/erroneous command`
: This indicates a driver error that should be reported to your driver supplier.

`Medium has been changed in drive`
: On a removable device such as a tape or cartridge disk, the medium has been changed without the cooperation of the driver.

`Medium is write-protected`
: On removable disks and tapes, the medium is write-protected and a write was requested.

`Sector not found`
: A request was made for a sector that does not exist (for example, would logically be located beyond the end of the medium). This is a configuration error and shoúld be corrected using `fdisk` or `mkpart`, as appropriate.

`(uncorrectable) Error found in sector data`
: The sector has a defect. The data is lost. The sector should be remapped using `mkpart`.

`Sector or track was marked bad`
: A sector was requested that was formatted as bad. This may occur on physical disk devices (devices with a `p` as the penultimate character in the name) when a marked-bad sector is read or written, and is a normal result. On remapped (`s`-type) disk devices, this can be avoided by using `mkpart` to remap the sector.

`Unable to recalibrate to track 0`
: A fixed disk failure has occurred. The disk needs servicing.

`Undetermined error`
: This indicates an undetermined error that should be reported to your driver and/or controller supplier.

Related information can be found in: *kconfig*(1), *mkpart*(1M), *mdevice*(4), *sdevice*(4), *disk*(7).

The Diskette Driver

Compatibility

The diskette (floppy disk) driver (also referred to as the `fd` driver, after the name of the device driver files) supports the hardware shown here.

Controller Chip	Drive
Intel 8272A, NEC 765	5.25-inch double-sided double density 5.25-inch double-sided quad density 3.5-inch double-sided double density 3.5-inch double-sided quad density 3.5-inch double-sided extra density

Hardware Configuration

The following display shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX System.

Maximum number of devices	2
Number of devices enabled	2
DMA channel	2
Interrupt priority level	4
Interrupt vector	6
Sharable interrupt	Yes
I/O address range	0x3f0–0x3f7
Controller memory address range	None

Software Setup

1. The `fd` driver is configured into the kernel by default. However, to reinstall the driver if it has been removed, run the `kconfig` program. Select `Drivers` under `Configure` from the `kconfig` bar menu. The system displays the drivers currently available on your machine and whether each is configured in to or out of your software.

2. To review or modify the `fd` driver, move to it and press Enter. Another screen appears, describing the configuration of the driver. You can change the status of the driver (either include it or exclude it from the configuration of the next kernel you build), and you can modify individual driver parameters. The parameters that can be modified are IRQ level, DMA channel, I/O address space, controller memory address space, and interrupt priority level.

3. To configure a driver into the kernel configuration, move to the `Driver Status` radio button and select `on`; select `off` to configure this driver out of the kernel configuration.

4. Be sure to rebuild and reinstall the kernel using `kconfig` after adding the driver.

Device Names

The fd driver provides access to diskettes as both block and raw (character) devices. For example, to access a 1.2 MB diskette as a raw device, use the device name /dev/rdsk/f0q15dt. However, to access the same diskette type as a block device, specify the device name /dev/dsk/f0q15dt. Both raw and block devices use the same major and minor numbers.

The driver controls up to four diskette drives. To access a 1.2 MB diskette on the second diskette drive as a raw device, use the device name /dev/rdsk/f1q15dt. The second diskette drive has the same major number as the first diskette drive. However, the minor number is 1 plus the minor number of the first diskette drive. For example, the major and minor numbers for both the first and second diskette drives supporting 1.2 MB diskettes are:

Device Name	Major No.	Minor No.
/dev/dsk/f0q15dt	1	0
/dev/dsk/f1q15dt	1	1
/dev/dsk/f2q15dt	1	2
/dev/dsk/f3q15dt	1	3
/dev/rdsk/f0q15dt	1	0
/dev/rdsk/f1q15dt	1	1
/dev/rdsk/f2q15dt	1	2
/dev/rdsk/f3q15dt	1	3

One diskette device configuration can have several device names. For example, the device names /dev/{r}dsk/f0q15dt, /dev/{r}dsk/f05ht, /dev/{r}fd096, /dev/{r}fd096ds15, /dev/{r}install, and /dev/{r}SA/disk0_1.2M refer to a diskette configuration that supports 1.2 MB diskettes in the first diskette drive.

Note that the /dev/{r}SA/disk*X* device names only exist for the first two diskette drives.

The following shows device names and their corresponding major and minor numbers and the supporting diskette geometry.

Device Name	Major No.	Minor No.	Cylinder	Sectors Per Track	Sector Size
/dev/{r}dsk/f0d4d, /dev/{r}dsk/f05d4	1	52	39	4	1024
/dev/{r}dsk/f0d4dt, /dev/{r}dsk/f05d4t	1	48	40	4	1024
/dev/{r}dsk/f0d8d, /dev/{r}dsk/f05d8	1	36	39	8	512
/dev/{r}dsk/f0d8dt, /dev/{r}dsk/f05d8t, /dev/{r}fd048ds8	1	32	40	8	512
/dev/{r}dsk/f0d9d, /dev/{r}dsk/f05d9	1	20	39	9	512
/dev/{r}dsk/f0d9dt, /dev/{r}dsk/f05d9t, /dev/{r}fd048, /dev/{r}fd048ds9, /dev/{r}SA/disk0_360k	1	16	40	9	512
/dev/{r}dsk/f0q15d, /dev/{r}dsk/f05h	1	4	79	15	512
/dev/{r}dsk/f0q15dt, /dev/{r}dsk/f05ht, /dev/{r}fd096, /dev/{r}fd096ds15, /dev/{r}install, /dev/{r}SA/disk0_1.2M	1	0	80	15	512
/dev/{r}dsk/f05d16	1	68	39	16	256
/dev/{r}dsk/f05d16t	1	64	40	16	256
/dev/{r}dsk/f05q, /dev/{r}dsk/f0q18d	1	84	79	18	512
/dev/{r}dsk/f05qt, /dev/{r}dsk/f0q18dt, /dev/{r}SA/disk0_1.44M	1	80	80	18	512
/dev/{r}dsk/f03d, /dev/{r}dsk/f0q9d	1	100	79	9	512

Device Name	Major No.	Minor No.	Cylinder	Sectors Per Track	Sector Size
/dev/{r}dsk/f03dt, /dev/{r}dsk/f0q9dt, /dev/{r}fd0135ds9, /dev/{r}SA/disk0_720k	1	96	80	9	512
/dev/{r}dsk/f0q36d	1	148	79	36	512
/dev/{r}dsk/f0q36dt, /dev/{r}SA/disk0_2.88M	1	144	80	36	512
/dev/{r}dsk/f03h	1	116	79	16	256
/dev/{r}dsk/f03ht	1	112	80	16	256
/dev/{r}fd0	1	128	auto-sense	auto-sense	auto-sense

Error Messages

The `fd` driver displays three types of error messages:

- `FD(`*n*`): diskette not present — please insert`
- `FD drv` *n* `blk` *b*`: drive error message`
- `FD controller: controller error message`

The driver displays the first message at 5-second intervals if the diskette in drive *n* is removed prematurely or is not inserted quickly enough. Insert the correct diskette in the diskette drive and close the drive door.

The driver displays the second message as a drive error message. It specifies the driver number *n* and block number *b* when an error occurs after a transfer has begun. The `drive error message` may be one of the following:

`Missing data address mark`
: The diskette may not be formatted properly.

`Cylinder marked bad`
: The accessed cylinder has been marked bad during formatting.

`Seek error (wrong cylinder)`
: The drive positioned itself at the wrong cylinder when attempting to set up for the requested transfer.

`Uncorrectable data read error`
: A cyclic redundancy check (CRC) error was detected when attempting to read the requested block from the drive.

`Sector marked bad`
: The accessed sector has been marked bad during formatting.

`Missing header address mark`
: The diskette may not be formatted properly.

`Write protected`
: A write was attempted to a diskette that is currently write protected.

`Sector not found`
: The diskette may not be formatted properly.

`Data overrun`
: The system could not keep up with the requested transfer of data.

`Header read error`
: The diskette may not be formatted properly.

`Illegal sector specified`
: The driver is confused by the format of the inserted diskette.

The third type of message occurs when there is a controller error during the setup for, or actual transfer of, a block. A controller error message is one of following:

`command timeout`
: The controller failed to complete the requested command in a reasonable length of time.

`status timeout`
: The controller failed to return its status after a command was completed.

`busy`
: During an attempt to access the controller, a timeout occurred.

Related information can be found in: *kconfig*(1), *fd*(7).

The Asynchronous Port Driver

Compatibility

The asynchronous port driver (also referred to as the `asy` driver, after the name of the device driver files) supports any IBM® AT-compatible serial card based on the National Semiconductor 8250, 16450, or 16550 chips.

Hardware Configuration

The following shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX Operating System.

Maximum number of devices	32
Interrupt priority level	7
Interrupt vector	4
I/O address range	3f8–3ff

Software Setup

The `asy` driver is configured into the kernel by default to support one serial port. Refer to Chapter 7, "Adding Modems, Printers, and Other Serial Devices," for more information.

Bi-Directional Capabilities

The INTERACTIVE UNIX System `asy` driver allows each serial port to be configured for dial-in and dial-out use at the same time. New devices are created for each port, which interlock to prevent simultaneous access to the port. This scheme negates the need for `uugetty`. Each hardware port tty0*N* (that is, `tty01`) has the following devices associated with it:

`ttyd`*N*
: Used for dial-in. `/etc/getty` should be placed on this line for incoming calls.

`acu`*N*
: Used for dial-out. This is the device name that should be used when setting up the file `/usr/lib/uucp/Devices` for `cu` and `uucico`.

`tty0N`
: Used for directly connected devices. This is intended for hardwired terminals, printers, and mice. It does not use the carrier-detect handshake signal and should *not* be used for modems.

`ttyhdN`
: Used for dial-in. `/etc/getty` should be placed on this line for incoming calls. Uses RTS/CTS hardware flow control handshaking.

`acuhN`
: Used for dial-out. This is the device name that should be used when setting up the file `/usr/lib/uucp/Devices` for `cu` and `uucico`. Uses RTS/CTS hardware flow control handshaking.

`ttyh0N`
: Used for directly connected devices. This is intended for hardwired terminals, printers, and mice. It does not use the carrier-detect handshake signal and should *not* be used for modems. Uses RTS/CTS hardware flow control handshaking.

Kernel Configuration

The device is configured by default for one port. The I/O address and IRQ for this port is:

Port	**I/O Address**	**IRQ**
COM1	3f8	4

If you wanted to use four `asy` ports, for example, one possible way to configure `/etc/conf/sdevice.d/asy` is as follows:

```
asy    Y    1    7    1    4    3f8    3ff    0    0
asy    Y    1    7    1    3    2f8    2ff    0    0
asy    Y    1    7    1    9    338    33f    0    0
asy    Y    1    7    1    5    238    23f    0    0
```

(See *sdevice*(4) for a complete explanation of each field.)

Note the sixth, seventh, and eighth fields of the /etc/conf/sdevice.d/asy file, which represent the interrupt vector (IRQ line), beginning I/O address, and ending I/O address, respectively. The second field defines whether or not that minor device should be activated in the kernel (Y to include). These values can be tuned for your specific serial port board. Once these extra ports are added manually, their configuration information can be changed through the kconfig program.

This will set up /etc/conf/sdevice.d/asy to configure the extra three ports to the following addresses.

Port	I/O Address	IRQ
Alternate COM2	2f8	3
Alternate COM3	338	2
Alternate COM4	238	5

Note – Due to the AT-bus interrupt design, the IRQ 2 signal is seen by the processor as IRQ 9. Thus, you must configure the hardware for IRQ 2, but must configure the software to use IRQ 9. To configure the hardware, see the following section on sharable interrupts.

Use of IBM standard COM3 and COM4 ports requires that the system use sharable interrupts. The actual hardware used must also support shared interrupts. The hardware configuration used is:

Port	I/O Address	IRQ
IBM COM3	3E8	4
IBM COM4	2E8	3

Using Sharable Interrupts

Serial port boards of Micro Channel Architecture design and certain AT serial port boards can use sharable interrupts. The INTERACTIVE UNIX System checks for possible interrupt conflicts when building a kernel.

Device Names

The following device names are used.

Device Name	Major No.	Minor No.	Description
/dev/tty00	3	00	Port 0 direct connect
.	.	.	.
.	.	.	.
/dev/tty03	3	03	Port 3 direct connect
/dev/ttyd0	3	16	Port 0 dial-in
.	.	.	.
.	.	.	.
/dev/ttyd3	3	19	Port 3 dial-in
/dev/acu0	3	32	Port 0 dial-out
.	.	.	.
.	.	.	.
/dev/acu3	3	35	Port 3 dial-out

Related information can be found in: *kconfig*(1), *mdevice*(4), *sdevice*(4), *asy*(7).

The Line Printer Driver

Compatibility

The line printer driver (also referred to as the `lp` driver, after the name of the device driver files) supports use of a line printer on the following parallel ports:

- Monochrome adapter (printer 0, port `LPT1`)
- Parallel adapter 1 (printer 1, port `LPT2`)
- Parallel adapter 2 (printer 2, port `LPT3`)

Hardware Configuration

The following display shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX Operating System. To support other configurations, the driver's configuration files must be modified using the kconfig program. If any of the configuration files are modified, the kernel must be rebuilt and reinstalled with the kconfig program. See *mdevice*(4) and *sdevice*(4) for details about the configuration files.

Maximum number of devices	3
Number of devices enabled	1
Interrupt priority level	3
Interrupt vector	7
LPT2 I/O address range	378 – 37f parallel port on adapter 1

Note that if more than one parallel printer port is configured into the system, each must use a separate interrupt. These are the I/O address ranges for the other two parallel printer ports.

LPT3 I/O address range	278 – 27f parallel port on adapter 2
LPT1 I/O address range	3bc – 3bf parallel port on monochrome adapter

Software Setup

The lp driver is configured into the kernel by default. Use the sysadm lpmgmt utility to add a line printer to your system. Refer to the *INTERACTIVE UNIX System User's Guide* for information about using sysadm to configure line printers into your system.

Device Names

Device Name	Minor No.	Description
/dev/lp	1	LPT1
/dev/lp[0–2]	0–2	LPT 1-3

If you have an lp port on your monochrome adapter, that port is LPT1 under DOS and /dev/lp0 under the INTERACTIVE UNIX Operating System. The first lp port you have on the motherboard or on an add-in I/O adapter is LPT1 or /dev/lp1. However, if you have both a mono display card with a parallel port *and* an add-in card, the add-in card becomes LPT2 under DOS but remains /dev/lp1 under the INTERACTIVE UNIX System.

If you have a second motherboard or add-in parallel port, it is usually `LPT2` under DOS and `/dev/lp2` under the INTERACTIVE UNIX System. The name varies under DOS because as DOS boots it looks for an adapter addressed one way, then another way, and then another. It calls the first parallel port it finds `LPT1`. Note that `/dev/lp0` corresponds to I/O port 0x3BC, `/dev/lp1` to 0x378, and `/dev/lp2` to 0x278. By default, `/dev/lp` is linked to `/dev/lp1`.

Tunable Parameters

The line printer driver allows users to determine how often the system checks whether a printer job is done, and how often it will send a message to the console alerting the operator that a printer needs attention.

Parameter	Default Value
`LP_POLLINT`	(HZ/20)
`LP_WARNINT`	(HZ*120)

where the system parameter HZ (hertz) is the number of ticks per second of the system clock (100). For example, a value of HZ*120 specifies 120 seconds.

This improves performance when using certain monochrome adapter ports. Some parallel port interfaces, such as the parallel port on many monochrome display adapters, do not latch (retain) their interrupt signals. This results in a loss of expected completion (`READY`) interrupts.

When the printer driver detects a condition requiring operator intervention (such as paper-out), it writes a message on the console. `LP_WARNINT` (default 2 minutes) defines (in minutes) the interval between these warnings.

These two parameters may be added to your system by editing the file `/etc/conf/pack.d/lp/space.c` to add the following:

```
# define LP_POLLINT (HZ/20)
# define LP_WARNINT (HZ*120)
```

Error Messages

The `lp` driver sends all error codes back to the user through the `errno` variable (see *intro*(2)); no other error messages are produced.

Related information can be found in: *kconfig*(1).

The Wangtek Cartridge Tape Driver

Compatibility

The Wangtek cartridge tape driver (also referred to as the `wt` driver, after the name of the device driver files) supports the following hardware.

Manufacturer	Controller	Drive
Bell Technologies	XTC-60-IC XTC-60-I XTC-125-I	Teac MT-2ST/45D (Cassette) Wangtek 5099 Wangtek 5125
COMPAQ	COMPAQ	Wangtek-compatible
Everex	EV-831, EV-833	Wangtek 5099
Wangtek	PC-36, PC-36II, 30631, 5099EK, 5150PK, 5150MK	Wangtek 5099, 512, 5150, 5150EQ, 5099EK, 5150PK, 5150MK
Cipher	5400	Wangtek-compatible
Gigatrend	Everex EV-811 Rev B	DAT Model 1236

The information supplied is subject to change. SunSoft reserves the right to modify its software product at any time.

Consult the *INTERACTIVE UNIX System Hardware Compatibility List* to find out which other devices are currently supported.

Note – There are three different types of cartridge tapes, and you should not attempt to use lower density cartridges than are supported by the drive. In particular, QIC-24 (60 MB) tapes may not work properly in drives that use QIC-120 or QIC-150 (120 MB and 150 MB) tapes.

Hardware Configuration

The following display shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX System. To support other configurations, the `kconfig` program must be used (see the next section).

Maximum number of devices	1
Number of devices enabled	1
DMA channel	1
Interrupt priority level	5
Interrupt vector	5
Sharable interrupt	No
I/O address range	0x300–0x301
Controller memory address range	None

Note that this driver supports only one device. You should change only the DMA channel, interrupt vector, and I/O address range.

Software Setup

1. Since the `wt` driver is not configured into the kernel by default, access the `kconfig` program.
2. Select `Drivers` under `Configure` from the `kconfig` bar menu. The system displays the drivers currently available on your machine and whether each is configured in to or out of your software.
3. To review or modify the `wt` driver, move to it and press Enter. Another screen appears, describing the configuration of the driver. You can change the status of the driver (either include it or exclude it from the configuration of the next kernel you build), and you can modify individual driver parameters. The parameters that can be modified are IRQ level, DMA channel, I/O address space, controller memory address space, and interrupt priority level.
4. To configure a driver into the kernel configuration, move to the `Driver Status` radio button and select `on`; select `off` to configure this driver out of the kernel configuration.

5. Be sure to rebuild and reinstall the kernel using `kconfig` after adding the driver.

Warnings

The following drivers provided with the INTERACTIVE UNIX Operating System are known to conflict with the default configuration of the `wt` driver.

3Com Ethernet (`ec`)	I/O address range 0x300–0x30f
3Com 3C503 Etherlink (`el`)	I/O address range 0x300–0x30f
Future Domain TMC-830/841 SCSI host adapter (`tmc 8x0`) (located on Boot-Loadable Drivers diskette)	Interrupt vector 5
LOGITECH Bus Mouse (`logi`)	Interrupt vector 5
Microsoft Bus Mouse (`mouse`)	Interrupt vector 5
Second parallel port (`lp`)	Interrupt vector 5

(Note that the INTERACTIVE UNIX Operating System is shipped without the second parallel port configured.) If any of these drivers are to be used in the same kernel as the `wt` driver, the conflict must be resolved by rejumpering one of the boards and using `kconfig` to change the appropriate driver parameters. For example, if the second serial port is not in use, the `wt` driver and controller board can be reconfigured to use interrupt vector 3 to avoid a conflict with the LOGITECH™ Bus Mouse. Any board that is "soft-strapped" must be reconfigured using the setup utilities supplied by its manufacturer.

Do not use DMA channel 2 with the `wt` driver, since this DMA channel is used by the diskette drive. To make the cartridge tape stream, read or write from the drive in clusters of 256 blocks or more.

Device Names

Device Name	Minor No.	Comment
`/dev/ntape`	0	No rewind on close
`/dev/rnmt0`	0	No rewind on close
`/dev/tape`	1	Rewind on close
`/dev/rmt0`	1	Rewind on close
`/dev/rSA/tape`	1	Rewind on close

Error Messages

`Streamer: Bad DMA channel, cannot init driver`
: The driver was configured with an illegal DMA channel number. The DMA channel must be 1 or 3.

`Streamer: Beginning of tape`
: The beginning of the tape was encountered unexpectedly. This usually indicates an attempt to read from an uninitialized tape.

`Streamer: Block not located`
: A specific block does not exist on the tape. This error message usually results from an application or utility attempting to read a block that would be beyond the logical end-of-tape. It may also result from attempting to read an uninitialized tape. Verify that the correct tape is in the drive and that the application or utility was invoked properly.

`Streamer: Drive not online`
: The controller cannot communicate with the tape drive, or the tape is broken. Check to see that the tape cartridge is not defective and recheck the hardware connections between the controller and the drive.

`Streamer: End of tape`
: The end of the tape was encountered. The usual cause is attempting to write more data to the tape than it is capable of holding.

`Streamer: Illegal command`
: The driver issued an illegal command to the controller. This can be caused by a bug in the driver or by faulty hardware.

`Streamer: No cartridge`
No tape cartridge is in the drive, or the tape is not firmly seated in the drive. Insert a tape or reseat the tape, if one is already in the drive.

`Streamer: No data detected`
No data was found when reading the tape. This indicates that the tape is uninitialized. Initialize the tape by writing to it before attempting to read.

`Streamer: Tape is write protected`
An attempt was made to write to a write-protected tape. Disable the tape's write protection or use a different tape.

`Streamer: Unrecoverable data error`
Data on the tape has been damaged. This error message may also result from attempting to read an uninitialized tape. Rewrite the data to the tape, if possible, or replace the tape.

Related information can be found in: *kconfig*(1), *wt*(7).

The Archive Cartridge Tape Driver

Compatibility

The Archive™ cartridge tape driver (also referred to as the `ct` driver, after the name of the device driver files) supports the following hardware.

Manufacturer	Controller	Drive
Archive	VP409A	VP150i, VP150e, VP60i, VP60e, ST600

Hardware Configuration

The following display shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX System. To support other configurations, the `kconfig` program must be used (see the next section).

Maximum number of devices	2
Number of devices enabled	1
DMA channel	1

Interrupt priority level	5
Interrupt vector	3
Sharable interrupt	No
I/O address range	0x200-0x201
Controller memory address range	None

Software Setup

1. Unless it is already loaded on your hard disk, install the driver from the *Additional Drivers* diskette. (Refer to the *INTERACTIVE UNIX System Installation Guide* for information about installing optional subsets.)
2. Since this driver is not configured into the kernel by default, use the `kconfig` utility to configure it into the kernel. Select `Drivers` under `Configure` from the `kconfig` bar menu. The system displays the drivers currently available on your machine and whether each is configured in to or out of your software.
3. To review or modify the `ct` driver, move to it and press Enter. Another screen appears, describing the configuration of the driver. You can change the status of the driver (either include it or exclude it from the configuration of the next kernel you build), and you can modify individual driver parameters. The parameters that can be modified are IRQ level, SM DMA channel, I/O address space, controller memory address space, and interrupt priority level.
4. To configure a driver into the kernel configuration, move to the `Driver Status` radio button and select `on`; select `off` to configure this driver out of the kernel configuration.
5. Be sure to rebuild and reinstall the kernel using `kconfig` after adding the driver.

Warnings

The following driver provided with the INTERACTIVE UNIX System is known to conflict with the default configuration of the `ct` driver:

Serial I/O Driver (`asy`)	Interrupt vector 3

If this driver is to be used in the same kernel as the `ct` driver, the conflict must be resolved by rejumpering one of the boards and manually changing the corresponding driver configuration files. See *mdevice*(4) and *sdevice*(4) for details about the configuration files.

Device Names

Device Name	Minor No.	Comment
`/dev/ntape`	0	No rewind on close
`/dev/rnmt0`	0	No rewind on close
`/dev/tape`	1	Rewind on close
`/dev/rmt0`	1	Rewind on close
`/dev/rSA/tape`	1	Rewind on close

Error Messages

`NO SC409A ADAPTER FOUND (getslot)`
: The driver cannot locate the controller during initialization.

`NOTICE: Cartridge Tape Controller Not Found`
: The driver cannot locate the controller during driver open time.

Related information can be found in: *kconfig*(1), *mdevice*(4), *sdevice*(4), *ct*(7).

COMPAQ SCSI Tape

Compatibility

The COMPAQ SCSI tape driver (also referred to as the `cpqs` driver, after the name of the device driver files) provides support for the COMPAQ 320/525MB 1/4-inch cartridge and associated adapter. It also supports Digital Audio Tape (DAT) units connected to the same adapter and enables hardware data compression on adapters so equipped.

Hardware Configuration

The following display shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX System. To support other configurations, the `kconfig` program must be used (see the next section).

Maximum number of devices	1
Number of devices enabled	1
DMA channel	7
Interrupt priority level	5
Interrupt vector	5
Sharable interrupt	No
I/O address range	130 – 133
Controller memory address range	None

The default hardware setup is achieved by placing all switches on the controller (SW 1) in the OFF position.

Software Setup

1. To configure the `cpqs` driver into the kernel, access the `kconfig` program. Select `Drivers` under `Configure` from the `kconfig` bar menu. The system displays the drivers currently available on your machine and whether each is configured in to or out of your software.

2. To review or modify the `cpqs` driver, move to it and press Enter. Another screen appears, describing the configuration of the driver. You can change the status of the driver (either include it or exclude it from the configuration of the next kernel you build), and you can modify individual driver parameters. The parameters that can be modified are IRQ level, DMA channel, I/O address space, controller memory address space, and interrupt priority level.

3. To configure a driver into the kernel configuration, move to the `Driver Status` radio button and select `on`; select `off` to configure this driver out of the kernel configuration.

4. Be sure to rebuild and reinstall the kernel using `kconfig` after adding the driver.

Warnings

The following drivers provided with the INTERACTIVE UNIX System are known to conflict with the default configuration of the `cpqs` driver:

Future Domain TMC-830/841 SCSI host adapter (`tmc8x0`)	Interrupt vector 5
LOGITECH Bus mouse (`logi`)	Interrupt vector 5
Microsoft Bus mouse (`mouse`)	Interrupt vector 5
Second parallel port (`lp`)	Interrupt vector 5

Note that the INTERACTIVE UNIX Operating System is shipped without the second parallel port configured. If any of these drivers is to be used in the same kernel as the `cpqs` driver, the conflict must be resolved by rejumpering one of the boards and using the `kconfig` program to change the appropriate driver parameters. For example, if the second serial port is not in use, the `cpqs` driver and controller board can be reconfigured to use interrupt vector `3` to avoid a conflict with the LOGITECH Bus Mouse. Any board that is "soft-strapped" must be reconfigured using the setup utilities supplied by its manufacturer.

The 320/525 MB tape adapter may be operated at any combination of the following:

I/O address 0x330 or 0x130
Interrupt vector 3 or 5
DMA channel 5 or 7

Set the on-board switch (SW1) in accordance with the COMPAQ documentation and modify the driver's configuration files.

In addition, if the target machine is an EISA machine, the new configuration must be placed in the NonVolatile Memory by the EISA Configuration Utility *before* the new kernel is booted. Failure to run the EISA Configuration Utility at this point will cause the driver to assume the old configuration parameters in deference to the on-board switch settings. When this happens, warning messages are printed as the new kernel boots.

Device Names

The tape names provided by default were chosen so as not to conflict with existing tape devices and are constructed as follows for the first unit:

```
/dev/[i][n][c]rct0[-150]
```

where

`i`	means return immediately in when performing actions that take a long time, such as rewind and retension.
`n`	means no rewind on close.
`c`	means enable compression.
`-150`	means write using QIC-150 format (512 byte blocks).

Only the first tape unit is configured. If more tapes are added, the file `/etc/conf/node.d/cpqs` must be edited manually to include new entries. Minor numbers are constructed using the following bit definitions.

Bit	Meaning
0x00	0=Rewind on close 1=No rewind on close
0x01	0=No compression 1=Compression enabled
0x02	0=Wait for command completion 1=Return immediately after command begins executing
0x03	0=Use tape media format and block size 1=Use QIC-150 format and block size

Once a tape has been written as a 150 MB tape, it cannot be used as a 525 MB tape until the beginning portion has been erased using the `erase` option of the `mt` command.

Error Messages

`Warning: Compaq SCSI Tape Adapter not configured`

   There is no EISA record of the adapter (EISA machines only). Run the EISA Configuration Utility and manually add the adapter definition. The system will work but will not use the automatic configuration facilities.

`Warning: Compaq SCSI Tape EISA Config Port mismatch (Setting to 0xXX)`

   The port address for this adapter does not match the address reported by the EISA NVRAM (EISA machines only). The driver will switch to the address as reported by the EISA NVRAM. This will work unless it causes a conflict. The discrepancy should be resolved.

`Warning: Compaq SCSI Tape EISA Config IRQ mismatch (Setting to X)`

   The interrupt vector for this adapter does not match the interrupt vector reported by the EISA NVRAM (EISA machines only). The driver will switch the interrupt vector used by the adapter but will not allocate the new interrupt vector in the kernel. Under these circumstances, the tape system will not work and the discrepancy must be resolved.

`Warning: Compaq SCSI Tape EISA Config DMA mismatch (Setting to X)`

   The DMA channel for this adapter does not match the DMA channel reported by the EISA NVRAM (EISA machines only). The driver will switch to the DMA channel as reported by the EISA NVRAM. This will work unless it causes a conflict. The discrepancy should be resolved.

`Compaq SCSI Tape Adapter not responding...`
`cpqs_scsi_reset: SCSI Adapter timed out, Status=X`

   The last command to the adapter timed out. The adapter status byte is returned.

`Adapter self test failed`

   The adapter failed to initialize.

Related information can be found in *kconfig*(1), *cpqs*(7).

The Mini-Cartridge Floppy Tape Driver

Compatibility

The mini-cartridge floppy tape driver (also referred to as the `mc` driver, after the name of the device driver files) supports the following Irwin® floppy drives.

Class	Drive Model
110	110, 310, 410
120[XL]	120, 220, 320, 420, 720, 2020
125	125, 225, 325, 425, 725
145[XL]	145, 245, 345, 445, 745, 2040
165	165, 265, 465, 765
285XL	285, 485, 785, 2080
287XL	287, 487, 787, 2120

Note that the letters "XL" in the 120[XL] and 145[XL] may or may not be present. When XL is present, the drive is capable of servo writing extra long (that is, 307.5-foot DC 2120) tapes.

In addition, the driver supports the following controllers.

Mnemonic	Description
SYSFDC	System floppy controller
ALTFDC	Alternate floppy controller
4100MC	Irwin 4100MC Micro Channel controller
4100	Irwin 4100 PC Bus controller

Hardware Configuration

The following display shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX System. To support other configurations, the `kconfig` program must be used (see the next section).

Maximum number of devices	2
Number of devices enabled	1
DMA channel	None
Interrupt priority level	None
Interrupt vector	None
Sharable interrupt	Yes
I/O address range	None
Controller memory address range	None

Software Setup

1. Install the driver from the *Additional Drivers* diskette. (Refer to the *INTERACTIVE UNIX System Installation Guide* for information about installing optional subsets.)

2. This driver is not configured into the kernel by default. Use the `kconfig` utility to configure the driver into the kernel. Select `Drivers` under `Configure` from the `kconfig` bar menu. The system displays the drivers currently available on your machine and whether each is configured in to or out of your software.

3. To review or modify the `mc` driver, move to it and press Enter. Another screen appears, describing the configuration of the driver. You can change the status of the driver (either include it or exclude it from the configuration of the next kernel you build), and you can modify individual driver parameters. The parameters that can be modified are IRQ level, DMA channel, I/O address space, controller memory address space, and interrupt priority level.

4. To configure a driver into the kernel configuration, move to the `Driver Status` radio button and select `on`; select `off` to configure this driver out of the kernel configuration.

5. Be sure to rebuild and reinstall the kernel using `kconfig` after adding the driver.

Warnings

The mini-cartridge floppy tape driver does not support the no-rewind tape option.

Device Names

Device Name	Minor No.	Comment
`/dev/mc/rmc0`	0	Mapped mode

This is the normal mode for tape read/write access. This mode recognizes tapes with either Irwin headers or older style XENIX headers in physical block 0. For both headers, defects are mapped out according to a defect list. ECC is encoded during write operations. ECC is decoded, when necessary, for read data recovery. When a Irwin header is present, (for example, AccuTrak® tapes or tapes formatted with the MC, the SCO® UNIX System, or SCO SLS drivers), the `mc` driver reserves the first three good blocks for header and relocation table information. The user data area starts in the fourth good tape block. A relocation table is written on the first device write for a tape that has none. When a medium error is encountered during a write operation, the block is relocated to a spare area at the end of the tape. When an older style XENIX header is present, the user data area starts in the second good tape block. No relocation table is written.

Device Name	Minor No.	Comment
`/dev/mc/rmc0p`	16	Physical block mapping

This mode is intended for diagnostic use. All data (not ECC) sectors are accessible starting with the first sector in physical block 0. Defects are not mapped out. ECC is encoded for writes, and decoded, when necessary, for read data recovery.

Device Name	Minor No.	Comment
/dev/mc/rmc0pl	32	Physical long block mapping

This mode is intended for diagnostic use. All data and ECC sectors are accessible starting with the first sector in physical tape block 0. Defects are not mapped out. ECC is not encoded for writes, but ECC is decoded, when necessary, for read data recovery.

Device Name	Minor No.	Comment
/dev/mc/rmc0a	64	All sectors data mode

This is a compatibility mode for tape backups created with early INTERACTIVE Systems tape drivers. Since all sectors contain data starting with the first sector of physical tape block 0, there is no defect mapping and no ECC. Writing tapes with this mapping mode is discouraged.

Device Name	Minor No.	Comment
/dev/mc/rmc0nr	80	Mapped mode with no relocation table

This device file is used for compatibility with the SCO SLS and UNIX System drivers. It is similar to map mode 0 with the exception that no relocation table is written on the first write. If, however, a relocation table is present, it is used.

Device Name	Minor No.	Comment
/dev/mc/mcdaemon	112	Daemon special mode

This is a special file used by the daemon process.

Related information can be found in: *kconfig*(1), *mt*(1) *mdevice*(4), *sdevice*(4), *mc*(7).

The Keyboard and Display Driver

Compatibility

The keyboard and display driver (also referred to as the `kd` driver, after the name of the device driver files) supports most IBM-compatible monochrome, CGA, EGA, and VGA adapters. In addition, most IBM-compatible keyboards (84 key, 101 key, and 102 key) are supported.

Consult the *INTERACTIVE UNIX System Hardware Compatibility List* to find out what other devices are currently supported.

Hardware Configuration

The following display shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX Operating System. Other configurations are not supported.

Maximum number of devices	1
Number of devices enabled	1
DMA channel	No
Interrupt priority level	6
Interrupt vector	1 (keyboard)
Sharable interrupt	No
I/O address range	0x60,0x64,0x3b0-0x3df
Controller memory address range	0xa0000-0xbffff

Software Setup

The `kd` driver is configured into the kernel by default. Note that the `kconfig` program *cannot* be used to configure this driver.

Enabling Virtual Terminals

Use the command `sysadm chgvts` or access the `sysadm` program and select `Virtual Terminals` on the `Tty Management` menu under `Machine` to change the number of virtual terminals available at the console terminal.

Follow the instructions on the form that appears. You can enable up to seven virtual terminals. The eighth terminal is the console terminal; it is always enabled.

Device Names

Device Name	Minor No.
/dev/console	0
/dev/vt00	0
/dev/vt01	32
/dev/vt02	64
/dev/vt03	96
/dev/vt04	128
/dev/vt05	160
/dev/vt06	192
/dev/vt07	224

Error Messages

Occasionally, while running programs such as the VP/ix Environment or INTERACTIVE X11, the system will beep when you attempt to switch to a new virtual terminal. You should attempt to switch again. Note that virtual terminal switching is disabled while using the kernel debugger.

Related information can be found in: *display*(7), *keyboard*(7).

The Keyboard Mouse Driver

Compatibility

The Keyboard Mouse driver (also referred to as the `kdmouse` driver, after the name of the device driver files) supports IBM-compatible keyboard mice, including the LOGITECH series and Microsoft® Mouse, when connected through a keyboard port on a Micro Channel (PS/2®) or AT architecture machine.

Hardware Configuration

The following display shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX System. To support other configurations, the `kconfig` program must be used (see the next section).

Maximum number of devices	1
Number of devices enabled	1
DMA channel	Not used
Interrupt priority level	6
Interrupt vector	12
Sharable interrupt	No
I/O address range	None (uses `kd`'s ports)
Controller memory address range	None

Software Setup

Unless it is already loaded on your system, install the driver from the *Additional Drivers* diskette. The `kdmouse` driver is not configured into the kernel by default. To install the driver, run the `kconfig` program. Select `Drivers` under `Configure` in the bar menu. Select `Keyboard Mouse` from the list of available drivers. When the form appears, Tab to the `Driver Status` radio button and press the spacebar to turn the driver `On`. Exit the form, and be sure to rebuild and reinstall the kernel to have this change take effect.

Device Names

Device Name	Minor No.	Comment
`/dev/kdmouse`	0	The mouse

Related information can be found in: *kconfig*(1), *kdmouse*(7).

The Microsoft Bus Mouse Driver

Compatibility

The Microsoft Bus Mouse driver (also referred to as the `mouse` driver, after the name of the device driver files) supports all three current models of the Microsoft Bus Mouse, also referred to as the InPort Mouse.

Hardware Configuration

The following display shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX Operating System. To support alternate configurations, the driver's configuration files must be modified and the kernel rebuilt and reinstalled using the `kconfig` program.

Maximum number of devices	1
Number of devices enabled	1
DMA channel	Not used
Interrupt priority level	6
Interrupt vector	5
Sharable interrupt	No
I/O address range	0x23c – 0x23f
Controller memory address range	None

Software Setup

1. Install the driver from the *Additional Drivers* diskette. (Refer to the *INTERACTIVE UNIX System Installation Guide* for information about installing optional subsets.)

2. Use the `kconfig` program to configure the `mouse` driver into the kernel, since it isn't configured by default,. Select `Drivers` under `Configure` from the `kconfig` bar menu. The system displays the drivers currently available on your machine and whether each is configured in to or out of your software.

3. To review or modify the `mouse` driver, move to it and press Enter. Another screen appears, describing the configuration of the driver. You can change the status of the driver (either include it or exclude it from the configuration of the next kernel you build), and you can modify individual driver parameters. The parameters that can be modified are IRQ level, DMA channel, I/O address space, controller memory address space, and interrupt priority level.
4. To configure a driver into the kernel configuration, move to the `Driver Status` radio button and select `on`; select `off` to configure this driver out of the kernel configuration.
5. Be sure to rebuild and reinstall the kernel using `kconfig` after adding the driver.

No further configuration is necessary if you are using the standard hardware configuration described above.

Warnings

The following drivers provided with the INTERACTIVE UNIX System are known to conflict with the default configuration of the `mouse` driver.

LOGITECH Bus Mouse (`logi`)	Interrupt vector 5, I/O address range 0x23c-0x23f
Wangtek Tape Drive (`wt`)	Interrupt vector 5
Future Domain TMC-830/841 SCSI host adapter (`tmc 8x0`)	Interrupt vector 5

If any of these drivers are to be used in the same kernel as the `mouse` driver, the conflict must be resolved by rejumpering one of the boards and using `kconfig` to change the corresponding driver configuration files.

For example, the I/O address range can be changed to 0x238-0x23b by jumpering the mouse card for the "Secondary InPort" setting. If a secondary serial port is not installed, the interrupt vector can be changed to IRQ 3 (4 if no serial ports are installed), and the card can also be set for IRQ 2 as long as the `mouse` driver is configured for IRQ 9 (due to the AT bus design).

Device Names

Device Name	Minor No.	Comment
`/dev/mouse`	0	The mouse

Related information can be found in *kconfig*(1), *mouse*(7), and the Microsoft Mouse owner's manual.

The LOGITECH Bus Mouse Driver

Compatibility

The LOGITECH Bus Mouse driver (also referred to as the `logi` driver, after the name of the device driver files) supports most models built to date of the LOGITECH Bus Mouse.

Hardware Configuration

The following display shows the hardware configuration that is supported in the standard distribution of the INTERACTIVE UNIX Operating System. To support alternate configurations, the driver's configuration files must be modified and the kernel rebuilt and reinstalled using the `kconfig` program.

Maximum number of devices	1
Number of devices enabled	1
DMA channel	Not used
Interrupt priority level	6
Interrupt vector	5
Sharable interrupt	No
I/O address range	0x23c – 0x23f
Controller memory address range	None

Software Setup

1. Install the driver from the *Additional Drivers* diskette. (Refer to the *INTERACTIVE UNIX System Installation Guide* for information about installing optional subsets.)

2. Use the `kconfig` program to configure the `logi` driver into the kernel since it isn't configured by default. Select `Drivers` under `Configure` from the `kconfig` bar menu. The system displays the drivers currently available on your machine and whether each is configured in to or out of your software.

3. To review or modify the `logi` driver, move to it and press Enter. Another screen appears, describing the configuration of the driver. You can change the status of the driver (either include it or exclude it from the configuration of the next kernel you build), and you can modify individual driver parameters. The parameters that can be modified are IRQ level, DMA channel, I/O address space, controller memory address space, and interrupt priority level.

4. To configure a driver into the kernel configuration, move to the `Driver Status` radio button and select `on`; select `off` to configure this driver out of the kernel configuration.

5. Be sure to rebuild and reinstall the kernel using `kconfig` after adding the driver.

Warnings

The following drivers provided with the INTERACTIVE UNIX System are known to conflict with the default configuration of the `logi` driver.

Microsoft Bus Mouse (`mouse`)	Interrupt vector 5, I/O address range 0x23c-0x23f
Wangtek Tape Drive (`wt`)	Interrupt vector 5
Future Domain TMC-830/841 SCSI host adapter (`tmc 8x0`)	Interrupt vector 5

If any of these drivers are to be used in the same kernel as the `logi` driver, the conflict must be resolved by rejumpering one of the boards and using `kconfig` to change the corresponding driver configuration files.

The I/O address on the LOGITECH mouse card cannot be changed, requiring conflicts to be resolved by changing the other card. However, if a secondary serial port is not installed, the interrupt vector can be changed to IRQ 3 (4 if no serial ports are installed), and the card can also be set for IRQ 2 as long as the `logi` driver is configured for IRQ 9 (due to the AT bus design).

Device Names

Device Name	Minor No.	Comment
`/dev/logi`	0	The mouse

Related information can be found in *kconfig*(1), *mouse*(7), and the LOGITECH Bus Mouse owner's manual.

Adding Modems, Printers, and Other Serial Devices 7

Basic Networking Procedures

Adding Basic Networking involves the following:

- Choosing to physically connect your computer to another computer terminal, or other remote device using one of the following:

  A direct link
  : Physically connect a null-modem cable from a serial port on your computer to a port on another computer. See "Direct Links and Modems" for details.

  A modem
  : Install and configure your modem as outlined in "Direct Links and Modems" and "Setting Up Modems."

- Establishing the logical connection between the INTERACTIVE UNIX Operating System and your modem or direct link. This involves updating the appropriate support files to reflect the presence of a direct link or modem. Refer to "Direct Links and Modems" and Chapter 8, "Basic Networking Administration," for details.

Direct Links and Modems

This section discusses the following configurations:

- Computer to Data Terminal Equipment (DTE) direct link, such as another computer or terminal
- A computer to Data Communications Equipment (DCE), such as a modem

Your computer will connect to any other machine with an RS-232 port. Your computer will support any kind of auto dial modem.

An advantage of using a direct link is that the link is always available and the time required to access the link is short. Direct links are beneficial when:

- The two machines transfer large amounts of data on a regular basis
- The two machines are located no more than several hundred cable feet apart

The amount of cable used to link two machines depends on the environment in which the cable is run. The standard for RS-232 connections is 50 feet or less. As the cable length is increased, noise on the lines may become a problem. This means that the transmission rate must be decreased or limited distance modems should be placed on each end of the line. Generally, you should not use more than 1000 cable feet to connect the two machines, and even this usually requires special cabling and line signal boosters.

The advantage of using a modem is that a port is not dedicated to only one computer. You can also be networked to a remote computer located anywhere in the world where the telephone network exists. The disadvantages are that the port of the remote computer may be busy, the transmission rate is slower, and line quality is not guaranteed.

The following table shows a subset of the RS-232C standard pins for a DB9 connector, their names, and whether they function as input or output pins for a particular device (DTE or DCE). In the table, "in" means the signal on the line is generated by an external source and treated as input by the device; "out" means the device generates the signal on the pin. Note that 25-pin DB25 connectors have a different pin assignment. Logical pin functions, however, are identical.

Pin	Description	Name	DTE	DCE
1	Frame Ground	–	–	–
2	Transmitted Data	TD	out	in
3	Received Data	RD	in	out
4	Request to Send	RTS	out	in
5	Clear to Send	CTS	in	out
6	Data Set Ready	DSR	in	out
7	Signal Ground	–	–	–
8	Data Carrier Detect	DCD	in	out
9	Data Terminal Ready	DTR	out	in

Physical Connection of a Computer to a DTE Direct Link

Connecting a computer to another RS-232C device (DTE) (for example, another computer) requires the use of a null-modem cable that must be constructed as follows:

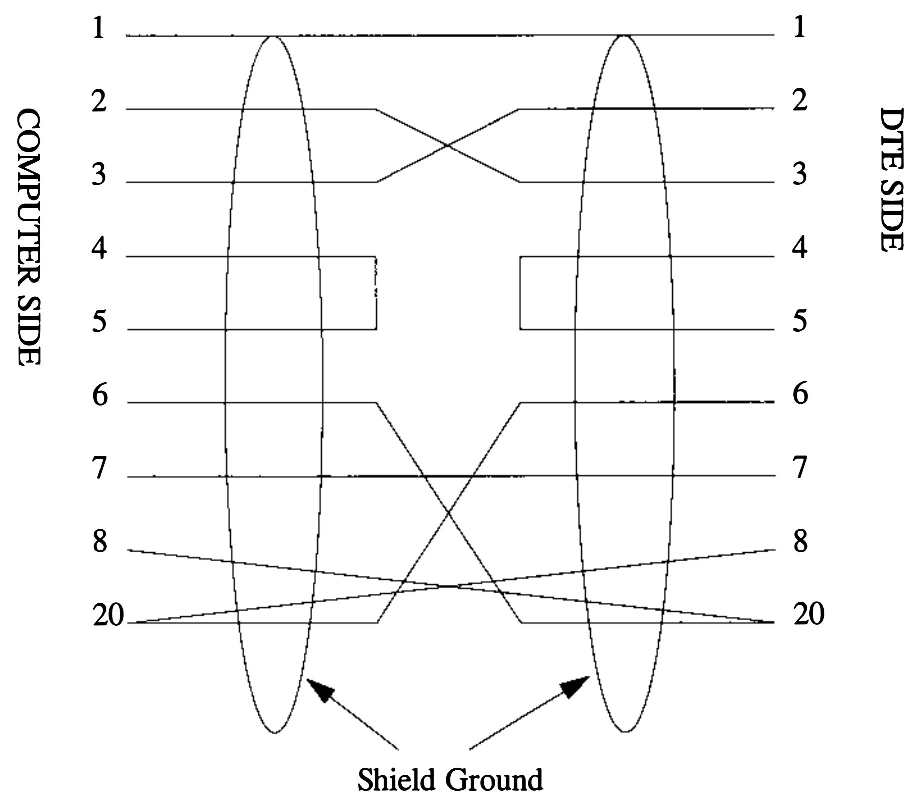
Pin 1 to 1
Pin 2 to 3
Pin 3 to 2
Strap pin 4 to 5 in the same plug*
Pin 6 to 20
Pin 7 to 7
Pin 8 to 20
Pin 20 to 6
Pin 20 to 8

* If you want to use hardware flow control, pin 4 to 5 and pin 5 to 4 (not in the same plug).

In a null-modem cable, input pins are connected to output pins to simulate the existence of an intermediate modem pair in the connection.

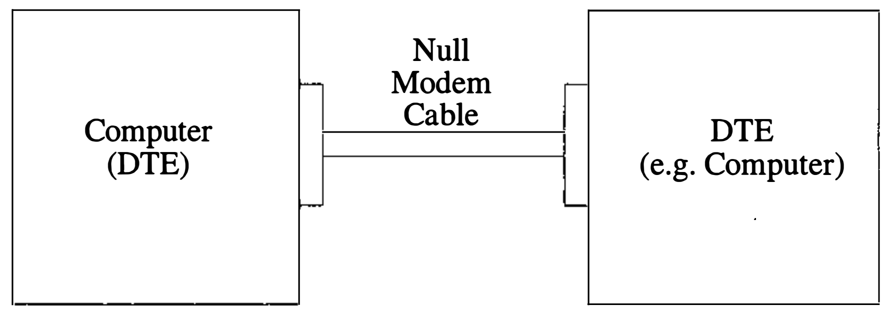
Wiring for Direct Link

Null-modem cables are commercially available for the direct link. If you want to customize your own null-modem cable, nine leads must be wired for a connection to be made as shown in the following figure. Do not attach wiring to unused signals.



DTE—Data Terminal Equipment

The next figure shows a simple illustration of how a computer connects to a DTE direct link.



DTE—Data Terminal Equipment

Basic Networking Software and Direct Links

The following support files must be updated to reflect the presence of a direct link. Consult the documentation provided with your machine if you are linking directly to a remote machine other than a computer.

- `/usr/lib/uucp/Devices`
- `/etc/inittab`
- `/usr/lib/uucp/Systems`

Physical Connection of a Computer to a Modem (DCE)

A DCE device such as a modem can connect to your computer with an RS-232 cable. The computer's serial connector usually has a DTE configuration, and the modem is required to have a DCE configuration. The pin connections for modem cable are shown in "Physical Connection of a Computer to a DTE Direct Link" earlier in this chapter. The pin connections for the RS-232 modem cable are:

Pin 1 to 1
Pin 2 to 2
Pin 3 to 3*
Pin 6 to 6
Pin 7 to 7
Pin 8 to 8
Pin 20 to 20

* If you want to use hardware flow control, pin 4 to 5 and pin 5 to 4 (not in the same plug).

DB-9 (DTE) to DB-25 (DCE)		
DB9 Pin (DTE)	**Signal Name**	**DB25 Pin (DCE)**
1	Carrier Detect (DCD)	8
2	Received Data (RD)	3
3	Transmitted Data (TD)	2
4	Data Terminal Ready (DTR)	20
5	Signal Ground	7
6	Data Set Ready (DSR)	6
7	Request to Send (RTS)	4
8	Clear to Send (CTS)	5
9	Ring Indicator (RI)	22 (not used)

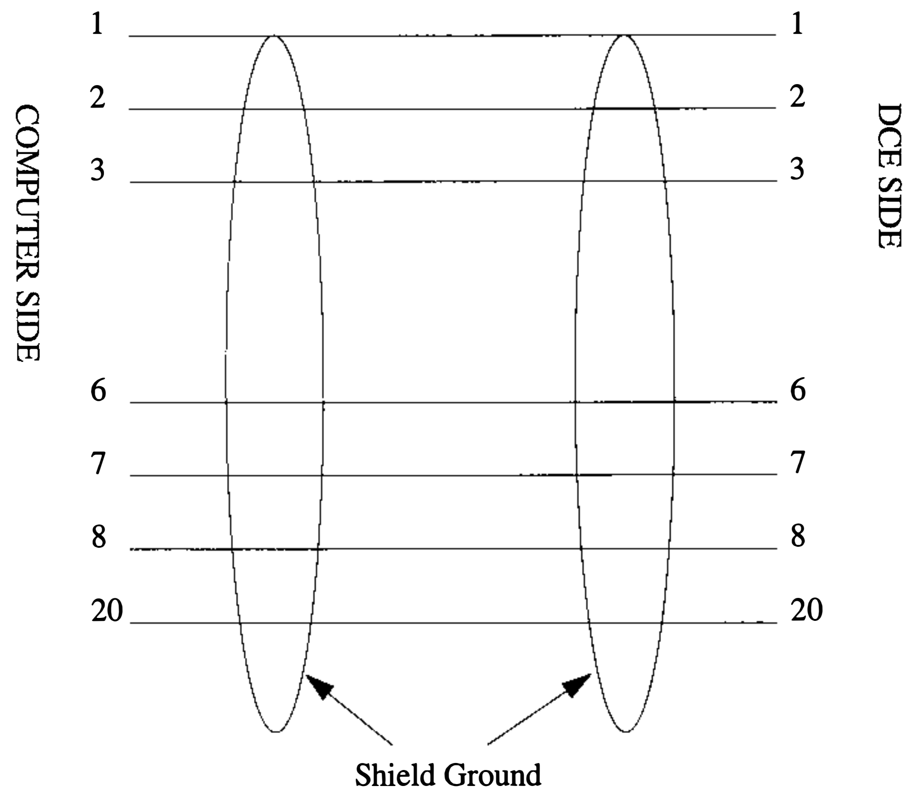
Note that the following signals must be connected for a direct line: TD, RD, ground; and for a modem: TD, RD, ground, CD, DTR.

DB-25 (DTE) to DB-25 (DCE)		
DB25 Pin (DTE)	**Signal Name**	**DB25 Pin (DCE)**
1	–	Chassis Ground
2 (out)	TD	2
3 (in)	RD	3
4 (out)	RTS	4
5 (in)	CTS	5
6 (in)	DSR	6
7	Signal Ground	7
8 (in)	DCD	8
20 (out)	DTR	20
22 (in)	RI	22

Note that since this cable is wired straight through, a ribbon cable and a pair of Insulation Displacement Connectors (IDC) can be used to make an instant cable. Ribbon cable is not shielded, though, so be sure to keep the length short—a few feet at most.

Wiring for Modems

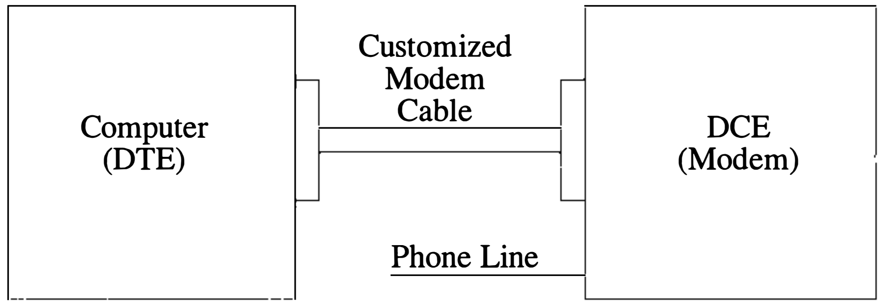
Wire to pins only used at both ends. Do not attach wiring to an unused signal. Seven leads must be wired to customize modem cable as shown in the following figure.



DCE—Data Communication Equipment

Note that pins 4 and 5 (RTS/CTS) can be used for hardware control.

The following figure shows a simple connection of a computer to a DCE device such as a modem.



DTE—Data Terminal Equipment

DCE—Data Communication Equipment

Bidirectional Capabilities

The INTERACTIVE UNIX System `asy` driver allows a single serial port to be configured for simultaneous dial-in and dial-out use. To accomplish this, devices that interlock to prevent simultaneous access to the port are created for each `asy` port. `uugetty` is no longer necessary.

Each port has the following devices associated with it. For each hardware port `tty0`*N* (that is, `tty01`):

`ttyd`*N*
: Used for dial-in. `/etc/getty` should be placed on this line for incoming calls.

`acu`*N*
: Used for dial-out. This is the device name that should be used when setting up `/usr/lib/uucp/Devices` for `cu` and `uucico`.

`tty0`*N*
: Used for directly connected devices. This is intended for hardwired terminals, printers, and mice. This device does not use the carrier-detect handshaking line. This device should *not* be used for modems.

For more information about the `asy` driver and its minor numbers, refer to the description of the `asy` driver and the asynchronous port driver information in Chapter 6, "Hardware Compatibility and Configuration."

Setting Up Modems

Modems can be used to connect your computer to remote terminals and computers using ordinary telephone lines. Terminals attached to a modem can dial in to your system from a remote location, and the INTERACTIVE UNIX Operating System and other UNIX System machines with a modem can use `uucp` to exchange information with your computer. Configuring your system to support a modem is similar to configuring a terminal, but more complex.

An entry must be made for each modem in the `/etc/conf/init.d/asy` file, but the entry can differ, depending on the intended use of the modem.

A modem can be configured to support either single-directional or bidirectional communications. Single-directional communication means that only incoming or outgoing calls are allowed through the modem, while bidirectional communications allow your system to receive incoming and initiate outgoing calls using the modem. Therefore, you must determine the desired type of communications.

Note – Your modem must have auto-dial capability if you want outgoing or bidirectional communications.

If only incoming communication is desired, the calling terminal or computer must be issued a `getty` process. Therefore, an entry similar to the following would be added to `/etc/inittab`:

```
00:23:respawn:/etc/getty ttyd0 1200
```

Single-directional, outgoing communication requires that neither `getty` nor `uugetty` be attached to the port. In this case, the *action* field (the third field) should be set to `off` instead of `respawn`.

In addition to the `/etc/inittab` entry, the `/usr/lib/uucp/Devices` file must be updated to configure the system to support modem dialing. Refer to Chapter 8, "Basic Networking Administration," for a description of the `/usr/lib/uucp/Devices` file.

Initial Modem Installation

Initial modem setup occurs the first time the modem is installed. You should:

1. Make sure the modem power switch is off.
2. Set up the modem option switches, if any.
3. Physically connect the modem to the computer and the telephone line (that is, attach the cables).
4. Plug the modem in.
5. Turn on the power to the modem.
6. Configure the kernel as described in Chapter 5, "Using kconfig to Tailor Your System Kernel."
7. Return to this section after you have set up the serial port.

The software commands necessary to configure the modem are then executed. Since these commands are sent to the modem, the modem must be connected to the computer and the power to the modem must be on. In addition, a number of configuration files on the computer must be modified. These steps may take a few moments.

The following sections contain switch and register settings for some popular modems. Consult the appropriate section for your particular type of modem.

If there is no entry for your specific modem, use another type as a guide, and refer to the modem manufacturer's documentation for your specific settings.

If you follow the instructions in "Configuring Dial-In Modem Lines" and "Configuring Dial-Out Modem Lines" (later in this chapter) to configure a COM port, your port will have bidirectional capabilities.

If your modem is configured using software registers instead of DIP switches, follow the directions in "Configuring Dial-Out Modem Lines" on dial-out use to set up a "direct" outgoing connection. Then use `cu` to connect to the modem (that is, `cu -lttyOn`). You can then communicate directly to the modem and send the appropriate `AT` codes to set the modem registers.

If you have a DOS communications program that you are familiar with, such as "kermit" or the BASICA "comm" program, you may prefer to set the modem software registers using that.

Configuring Dial-In Modem Lines

To set up dial-in lines, you need to inform the `init` process which devices should run `gettys`. Edit `/etc/conf/init.d/asy` to add the correct `getty` lines for each modem. For example:

```
# Hayes-compatible 1200 baud modem on ttyd0
00:2345:respawn:/etc/getty -t 60 ttyd0 d1200
# Hayes-compatible 2400 baud modem on ttyd1
01:2345:respawn:/etc/getty -t 60 ttyd1 D2400
# Telebit Trailblazer on ttyd2
02:2345:respawn:/etc/getty -t 60 ttyd2 TB19200
```

The `getty` line contains information on the status of the port, the timeout value for the modems, the port to spawn the `getty` on, and the type of `getty` to be used, as named in `/etc/gettydefs`.

If the field marked `respawn` is changed to `off`, the port will be disabled and will not allow logins. Refer to "Commented gettydefs Listing" later in this chapter to find the proper `gettydef` for the modem being installed. See *getty*(1M) for a further explanation of the options to `/etc/getty`.

After `/etc/conf/init.d/asy` has been configured, a new `/etc/inittab` file must be built. Do the following:

1. Log in as or `su` to `root` and type:

```
# /etc/conf/bin/idmkinit
```

   to build a new `inittab` file from the files in `/etc/conf/init.d`.

2. Type:

```
# mv /etc/inittab /etc/inittab.save
```

to save your old `/etc/inittab` file.

3. Type:

```
# mv /etc/conf/cf.d/inittab /etc/inittab
```

to install the new `inittab` file.

4. When this is done, the changes can be tested by running:

```
# telinit q
```

Editing `/etc/conf/init.d/asy` preserves the changes through future kernel builds. Changes made by directly editing `/etc/inittab` will be lost, as this file is overwritten the first time a new kernel is booted.

If you are installing an external modem with a DTR indicator light, it should turn on. This indicates that `/etc/init` has spawned a `getty` on the modem and is waiting for a caller. If the RD (Receive Data) and/or SD (Send Data) lights on the modem panel start flashing quickly at this point, it indicates that the modem has not been configured properly or the cable is not wired correctly. If this occurs, restore the old `inittab` file and cause `init` to reread the `inittab` file by typing:

```
# mv /etc/inittab.save /etc/inittab
# telinit q
```

Configuring Dial-Out Modem Lines

To configure a dial-out line, the `Dialers` and `Devices` files in `/usr/lib/uucp` must be modified.

1. Edit /usr/lib/uucp/Dialers to add the Dialers entry for your particular modem as indicated in the appropriate modem section in Chapter 8, "Basic Networking Administration." There may already be an entry for your modem in the file. The entries in that chapter have been tested with the INTERACTIVE UNIX System asy drivers and are known to work.

2. For each device you are configuring for a dial-out line, edit /usr/lib/uucp/Devices to add one direct entry for configuring (for example, changing the modem software registers) and testing, and one or more acu-style entries.

   For example:

```
# Hayes SmartModem 1200 on acu0 (direct access via tty00)
Direct tty00   -     Any    direct \D
ACU    acu0    -     300    hayes1200  \T
ACU    acu0    -     1200   hayes1200  \T

# Hayes SmartModem 2400 on acu1 (direct access via tty01)
Direct tty01   -     Any    direct  \D
ACU    acu1    -     300    hayes2400 \T
ACU    acu1    -     1200   hayes2400 \T
ACU    acu1    -     2400   hayes2400 \T

# Telebit Trailblazer on acu2 (direct access via tty02)
Direct tty02   -     Any    direct  \D
ACU    acu2    -     300    tb300   \T
ACU    acu2    -     1200   tb1200  \T
ACU    acu2    -     2400   tb2400  \T
ACU    acu2    -     9600   tbfast  \T
ACU    acu2    -     9200   tbfast  \T
```

Troubleshooting Modem Installation

This section discusses problems that may occur during installation and suggests solutions.

Modem Transmit and Receive lines are flashing—no connection established.
Modem is in "verbose" mode. Check the "verbose" switch setting or register value, as documented for your particular modem.

Modem transmits data as soon as `getty` is enabled.

Modem has DCD forced high, so the system thinks a caller has dialed in. If the command `ps -eaf` shows the `TTY` as anything except `?` before someone calls in, then DCD is somehow forced high. This is typically configured in the modem, or the cable may have the DCD pin tied high.

`/etc/init` prints the message `getty respawning too rapidly`.

Several problems can trigger this message. You may not have specified the correct device name in `inittab` or `/etc/conf/init.d/asy`, the device may not be configured into kernel properly (check `/etc/conf/sdevice.d/asy`), or the device may not be found by the kernel (check jumper settings for I/O address or possible addressing conflict, for example, overlapping addresses with some other board).

The modem answers the phone but gives no `login` prompt.

`getty` is either somehow misconfigured or the kernel is configured for the wrong interrupt line. Check to see that the file `/etc/conf/sdevice.d/asy` specifies the correct IRQ (or `/etc/conf/pack.d/asy/space.c` has been properly edited for shared interrupts). Another possible cause is that you are attempting to share interrupts but the hardware cannot handle it. Try again with only one device on the interrupt line.

The modem answers the phone, but the system prints `Password:` immediately.

You may have called in using a modem with error correction enabled (such as MNP™ or v.42), but the remote system's modem doesn't support error correction. Either disable error correction on the calling modem, or add the `-f` option to the `/etc/getty` command line in the remote system's `/etc/inittab` file.

The modem answers the phone, but garbage characters come out.

Try pressing Break (or `~%b` if using `cu`) or Enter several times to get `/etc/getty` to cycle through baud rates. This is normal behavior if you connected at other than the primary baud rate. If that does not work, check that the `gettydefs` entry specified in the `inittab` file cycles through the baud rate at which you connected.

If some characters appear to be correct, check that your character size (data bits) and parity agree. If the standard `gettydefs` entries are being used, you need 7 data bits, even parity. If you are using the 8-bit `gettydefs` entries, be sure to use 8 bits and no parity.

The modem answers the phone, the user can log in and work, but the modem does not hang up when the user logs off.

There are several possibilities.

- The modem may be ignoring DTR. Check the modem switches or registers. The modem should be set to drop the line and reset itself when DTR is dropped.
- The cable may be forcing DTR to always be high. In this case, the modem may also auto-answer even when the system is not in multi-user mode. Check that DTR goes low when `/etc/getty` is not running (that is, change `respawn` to `off` in `/etc/inittab`).
- The "hang up on close" field is not set. Add the `HUPCL` flag to the associated `/etc/gettydefs` entry.

UUCP gives `no devices available` message.

- `getty` could be getting the device open without carrier. Check the cable and registers.
- A direct line could be open.
- Is there a mouse attached using this line?
- The entry in `/usr/lib/uucp/Devices` could be incorrect.

UUCP or `cu` will not dial out properly.

- The modem may be configured incorrectly. Check the switch settings and modem register settings.
- `/usr/lib/uucp/Devices` may specify an incorrect `Dialers` entry. See the examples in Chapter 8, "Basic Networking Administration."

  As a debugging tool, try using the following command, which attempts to set up a `uucp` connection to a designated site. The command is set at a debugging level (`-xN`) that shows the actions of the process:

```
# /usr/lib/uucp/uucico -r1 -x6 -shostname
```

To check the modem connection and modem operation, use the following command:

```
# cu -sspeed -ltty0N
```

This gives you direct access to the modem and the ability to give it "AT" codes directly.

Recommended Modem Settings

Some modems, such as the Hayes® SMARTMODEM 1200™, are hardware-configured. They have switch settings that must be manually set in order for the modem to work correctly. Other modems are configured via software or nonvolatile memory.

Hardware-configured modems that have a carrier detect (CD) switch usually must have that switch set low or off.

Using a Hayes SMARTMODEM 1200

The switch setting values may be different on Hayes-compatible modems. Check the documentation that accompanied your modem.

Recommended Switch Settings		
1	UP	Modem obeys DTR
2	UP	English response codes
3	UP DN	(Dialin) no result codes unless enabled via ATQ0 (Dialout only) send result codes
4	UP	Echo commands (optional)
5	UP DN	(Dialin) enable auto-answer (Dialout) disable auto-answer
6	UP	Carrier detect shows true carrier state
7	DN	Single-line telephone
8	UP DN	"Smart mode" (command recognition) (HAYES) Enable 1200 baud operation (QUBIE) (This setting varies from modem to modem. You may have to find the proper setting for your modem.)

Add the following entry to the /usr/lib/uucp/Dialers file:

```
#########################
#
# Hayes Smartmodem 1200
#
#########################
hayes2400 =,-, "" \M\pAT\r\dATQ0V1\r\c OK\r ATDT\T\r\m\c CONNECT
```

Add the following entries to the /usr/lib/uucp/Devices file:

```
# Hayes SmartModem 1200 on acuN (direct access via tty0N)
Direct  tty0N   -      Any     direct \D
ACU     acuN    -      300     hayes1200  \T
ACU     acuN    -      1200    hayes1200  \T
```

Using a Hayes SMARTMODEM 2400 (Nonvolatile Settings)

These settings should be set using the DOS “comm” program or the cu –ltty0*N* command. Issue the commands listed in the first column of the following table:

Recommended Settings	
AT&F	Reset to factory settings
ATQ1	No result codes (unless explicitly enabled using ATQ0)
ATM0	[Optional—speaker always off]
ATE1	Echo command characters
ATV1	English result codes
ATX4	Extended status (allows Hayes to detect dial tone and busy signal)
AT&C1	Indicate true carrier state
AT&D3	Hangup and reset to saved values when DTR is dropped
ATS0=1	[set this for dial-in use only] Answer phone after 1 ring. You may want to increase this value if the phone line will be used for both voice and computer communications.
AT&W	Save settings to nonvolatile memory

Add the following entry to /usr/lib/uucp/Dialers:

```
########################
#
# Hayes Smartmodem 2400
#
########################
hayes2400 =,-, "" \M\pAT\r\dATQ0V1\r\c OK\r ATDT\T\r\m\c CONNECT
```

Add the following entries to /usr/lib/uucp/Devices:

```
# Hayes SmartModem 2400 on acu1 (direct access via tty0N)
Direct  tty0N   -       Any     direct  \D
ACU     acuN    -       300     hayes2400       \T
ACU     acuN    -       1200    hayes2400       \T
ACU     acuN    -       2400    hayes2400       \T
```

Telebit TrailBlazer Suggested Configurations (Nonvolatile Settings)

The following register settings are known to work for the Telebit™ TrailBlazer™ Plus at ROM revision level "BA4.00," as reported by the ATN? and ATI3 commands.

ATE1
: Enable modem to echo characters while in command mode (default).

ATF1
: Disable echoing of data stream (default).

ATM1
: Enable speaker during dial and connecting steps only (default). (You may wish to completely disable the speaker via "M0" instead.)

ATQ4
: Modem will not report result codes unless connection was initiated locally via an ATD or ATA command. This is an important register setting.

ATV1
: Send result codes as English words instead of numeric codes (default).

ATX3

Enable PEP and MNP extended result codes.

"S" register settings (only non-default settings are listed here):

ATS00=001

Answer on the first ring. Set this to `0` if dial-in mode is not desired. You may want to increase this value if the phone line will be used for both voice and computer communications.

ATS51=254

Allow the interface speed to automatically match that of the modem connection. Also configures incoming PEP-mode calls to use 19200 baud. (Note that early versions of Telebit's documentation overlooked this value for s51. This setting behaves exactly as the documented value of 255 except that 254 uses an interface of 19200 baud instead of 9600 baud.)

ATS52=002

Disconnect modem and reset to stored parameters when DTR is dropped by the host system.

ATS53=002

DCD and DSR are on only when a carrier is detected.

ATS54=003

Break signals are passed in-line in the data stream.

ATS58=000

No flow control is used between modem and host when in PEP mode. (Because the interface speed should match the modem speed, this setting turning off flow control should not matter.)

ATS92=001

When answering the phone, issue PEP tones *after* the normal 300/1200/2400 sequence so that non-PEP callers do not get confused.

ATS110=001

Use data compression in PEP mode if remote also has it enabled.

ATS111=030

Use UUCP "g" protocol spoofing in PEP mode.

AT&W

Save the EEPROM configuration. (Remember to issue this command to save the settings.)

Other registers that may be of interest are:

S61

Modem volume control

S95

MNP operating mode

Add the following entries to `/usr/lib/uucp/Dialers`:

```
########################
#
# Telebit TrailBlazer
#
########################
tb300    =W-,  "" A\pA\pA\pA\pA\pT OK ATx3s0=0s64=1s50=1DT\T NNECT
tb1200   =W-,  "" A\pA\pA\pA\pA\pT OK ATx3s0=0s64=1s50=2DT\T NNECT\s1200
tb2400   =W-,  "" A\pA\pA\pA\pA\pT OK ATx3s0=0s64=1s50=3DT\T NNECT\s2400
tbfast   =W-,  "" A\pA\pA\pA\pA\pT OK ATx3s0=0s64=1s50=255DT\T FAST-\c-FAST
```

Certain revisions of the TrailBlazer will do better with the following:

```
tb300    =W-,  "" A\pA\pA\pA\pA\pT OK ATx3s0=0s64=1s50=0DT\T NNECT
tb1200   =W-,  "" A\pA\pA\pA\pA\pT OK ATx3s0=0s64=1s50=0DT\T NNECT\s1200
tb2400   =W-,  "" A\pA\pA\pA\pA\pT OK ATx3s0=0s64=1s50=0DT\T NNECT\s2400
tbfast   =W-,  "" A\pA\pA\pA\pA\pT OK ATx3s0=0s64=1s50=255DT\T FAST-\c-FAST
```

Add the following entries to `/usr/lib/uucp/Devices`:

```
# Telebit TrailBlazer on acuN (direct access via tty0N)
Direct tty0N   -      Any    direct \D
ACU    acuN    -      300    tb300  \T
ACU    acuN    -      1200   tb1200 \T
ACU    acuN    -      2400   tb2400 \T
ACU    acuN    -      9600   tbfast \T
ACU    acuN    -      19200  tbfast \T
```

Commented gettydefs *Listing*

This section shows some sample gettydefs entries. Rather than directly editing the /etc/gettydefs file, you can use the sysadm program to initialize a modem.

```
# This sequence is for a Telebit TrailBlazer modem.  It cycles
# through 19200, 2400, and 1200 baud.  It sets the line for the
# "traditional" UNIX System values of 7 data bits, even parity.
#
# To enable a getty on ttyd0 using this sequence, specify:
#
# answering initially at 19200 -- "/etc/getty -t 60 /dev/ttyd0 TB19200"
# answering initially at 2400  -- "/etc/getty -t 60 /dev/ttyd0 TB2400"
# answering initially at 1200  -- "/etc/getty -t 60 /dev/ttyd0 TB1200"
#

TB19200# B19200 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
PARENB ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD # B19200
HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY PARENB ISTRIP
ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD #login: #TB2400

TB2400# B2400 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
PARENB ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD # B2400
HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY PARENB ISTRIP
ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD #login: #TB1200

TB1200# B1200 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
PARENB ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD # B1200
HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY PARENB ISTRIP
ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD #login: #TB19200


#
#  This sequence is also for the Telebit, the only difference
#  being that each entry specifies 8 data bits, no parity, for
#  use by most PC users.  (INTERNATIONAL USERS: remove the
#  "ISTRIP" keyword to allow 8-bit input.)
#

TB8-19200# B19200 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS8 CREAD # B19200 HUPCL
OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY ISTRIP ECHO ECHOE
ECHOK ICANON ISIG CS8 CREAD #login: #TB8-2400
```

```
TB8-2400# B2400 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS8 CREAD # B2400 HUPCL
OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY ISTRIP ECHO ECHOE
ECHOK ICANON ISIG CS8 CREAD #login: #TB8-1200


TB8-1200# B1200 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS8 CREAD # B1200 HUPCL
OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY ISTRIP ECHO ECHOE
ECHOK ICANON ISIG CS8 CREAD #login: #TB8-19200


#
#  This sequence is for 2400 baud modems.  It cycles through
#  2400, 1200, and 300 baud.  It sets the line to 7 data bits
#  and even parity.
#
# To enable a getty on ttyd0 using this sequence, specify
# answering initially at 2400 -- "/etc/getty -t 60 /dev/ttyd0 D2400"
# answering initially at 1200 -- "/etc/getty -t 60 /dev/ttyd0 D1200"
# answering initially at  300 -- "/etc/getty -t 60 /dev/ttyd0 D300"
#

D2400# B2400 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
PARENB ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD # B2400
HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY PARENB ISTRIP
ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD #login: #D1200

D1200# B1200 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
PARENB ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD # B1200
HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY PARENB ISTRIP
ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD #login: #D300

D300# B300 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
PARENB ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD # B300
HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY PARENB ISTRIP
ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD #login: #D2400


#
#  This is the eight data bits, no parity version of the 2400
#  baud sequence.
```

```
#  (INTERNATIONAL USERS: remove the "ISTRIP" keyword to allow
#  8-bit input.)
#

D8-2400# B2400 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON
IXANY ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS8 CREAD # B2400
HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY ISTRIP ECHO
ECHOE ECHOK ICANON ISIG CS8 CREAD #login: #D8-1200


D8-1200# B1200 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS8 CREAD # B1200 HUPCL
OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY ISTRIP ECHO ECHOE
ECHOK ICANON ISIG CS8 CREAD #login: #D8-300


D8-300# B300 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS8 CREAD # B300 HUPCL
OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY ISTRIP ECHO ECHOE
ECHOK ICANON ISIG CS8 CREAD #login: #D8-2400


#
# This sequence is for 1200 baud modems.  It switches between
# 1200 and 300 baud only, using 7 data bits, even parity.
#
# To enable a getty on ttyd0 using this sequence, specify
# answering initially at 1200  -- "/etc/getty -t 60 /dev/ttyd0 d1200"
# answering initially at  300  -- "/etc/getty -t 60 /dev/ttyd0 d300"
#

d1200# B1200 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
PARENB ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD # B1200
HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY PARENB ISTRIP
ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD #login: #d300

d300# B300 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
PARENB ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD # B300
HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY PARENB ISTRIP

ECHO ECHOE ECHOK ICANON ISIG CS7 CREAD #login: #d1200
#
#  This is the eight data bits, no parity version of the
#  1200/300 baud sequence.
```

```
#  (INTERNATIONAL USERS: remove the "ISTRIP" keyword to allow
#  8-bit input.)
#

d8-1200# B1200 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS8 CREAD # B1200 HUPCL
OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY ISTRIP ECHO ECHOE
ECHOK ICANON ISIG CS8 CREAD #login: #d8-300

d8-300# B300 HUPCL OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY
ISTRIP ECHO ECHOE ECHOK ICANON ISIG CS8 CREAD # B300 HUPCL
OPOST ONLCR TAB3 BRKINT IGNPAR IXON IXANY ISTRIP ECHO ECHOE
ECHOK ICANON ISIG CS8 CREAD #login: #d8-1200
```

The Default Hardware Configuration

The INTERACTIVE UNIX Operating System supports systems based on the ISA (AT), EISA, and Micro Channel (PS/2) architectures. The kernel delivered with the INTERACTIVE UNIX System is capable of supporting the following default hardware configurations:

ISA or EISA systems assume:

- An AT keyboard
- An IBM AT or AT-compatible fixed disk controller card or an AT SCSI host and floppy disk controller card
- A monochrome, color, EGA, or VGA display controller
- One or two serial communications ports (`COM1`, `COM2`)
- One parallel printer port
- 4 MB AT system RAM

Micro Channel (MCA) systems assume:
(for example, PS/2 Model 70 and PS/2 Model 80)

- An MCA keyboard
- An MCA floppy controller
- An MCA fixed disk controller
- A VGA display controller
- One parallel printer port
- One or two serial communications ports

The default configuration of your machine may be different. Consult the documentation provided by your vendor to determine your default hardware configuration.

Peripheral Devices

One of the tasks a system administrator is responsible for is adding peripheral devices such as terminals, line printers, or a second fixed disk to the system. These peripheral devices are used to increase the number of users or disk space or to expand the overall capabilities of the system.

Many of the computers supported by the INTERACTIVE UNIX System are configured with one or two serial (RS-232) ports and one parallel (CENTRONICS) interface port. While the serial ports can be used to connect any RS-232 device to your computer, the parallel port is usually only used with printers. Typical RS-232 devices include terminals, modems, line printers, and other computers.

A port can be either serial or parallel. A serial port, such as `COM1` or `COM2`, allows the transmission of data 1 bit at a time; a parallel port, such as `LPT1` or `LPT2`, allows the transmission of data 8 bits at a time.

Terminals must be connected to the system through serial ports. When the computer is powered up or rebooted, it is automatically initialized to support both the console and the remote terminals if the serial ports are appropriately set up. Printers, in contrast, can be either serial or parallel, depending on how they have been configured by the manufacturer.

The INTERACTIVE UNIX Operating System has a hardware configuration that typically has at least one parallel and two serial ports. With such a configuration you might, for example, connect two terminals and a parallel printer or one terminal, one serial mouse, and a parallel printer.

This section explains how to add RS-232 devices and parallel line printers to your system.

Setting Up an RS-232 Connection

When connecting an RS-232 device to your system, a connection must be made from your computer to the device either directly (hardwired) or through a modem. Once the connection is made, the system must be told what type of connection exists and what type of device is being connected. You will configure the RS-232 port when adding terminals, modems, and serial line printers.

Connecting an RS-232 Device Directly to Your Computer

The direct, hardwired connection between your computer and an RS-232 device depends on the type of device. Consult the hardware manufacturer's documentation to determine the specific type of cable and pin assignments needed. The following hints generally apply when connecting an RS-232 device to your computer:

- RS-232 cables should be less than 50 feet long. Cables longer than 50 feet may introduce line noise. In some cases, you can use a longer cable by reducing the line speed (baud rate) or by using low-capacitance cables in conjunction with signal boosters.
- An in-line adapter, called a null-modem adapter, is usually required when connecting a terminal or computer directly to the serial port on your computer. Alternately, the cable itself may have been modified to function as a null modem.
- A null-modem cable or adapter should not be used when connecting a modem to your computer.

Configuring Your Computer With Additional Terminals

A terminal is connected to the system through a *port* (where the terminal is actually plugged in) on the motherboard (main board) or on a multi-port adapter board on your system. This allows more than one person to use the computer at one time.

Asynchronous serial ports are also called *tty lines*. A tty line is a line that connects a terminal, modem, or other device to the main system and is associated with a specific port. A tty line is also referred to as a *com* (communication) line by some manufacturers.

Tty lines are either alphabetically ordered or sequentially numbered. For example, some of the tty lines included on your system are `tty00` and `tty01`.

Each time you add a new terminal to your system, you will need to configure it for your system, using the `sysadm` `Tty Management` menu under `Machine` on the bar menu. However, you will not need to install a device driver or reconfigure and rebuild your kernel if you are using a terminal configuration that is already supported. Most terminals that use one of the two standard serial ports of your computer are supported by the default configuration supplied by SunSoft. Check the documentation supplied by your vendor to determine whether your terminal is supported by this configuration. If it is, you can proceed immediately to step 1 of the procedure outlined in this section to add your new terminal.

If you plan to add a terminal that is not supported by the serial ports on the main board of your CPU, you must first install the terminal adapter (possibly with a special device driver), then configure, build, and install a new kernel (see Chapter 5, "Using kconfig to Tailor Your System Kernel"). You should then follow the steps outlined below.

If you plan to use a multi-port serial expansion card to connect your terminals, you may need to install additional device drivers. Refer to the manufacturer's installation instructions supplied with your hardware.

If you are unsure about whether your terminal type is supported on this system, check the `/usr/lib/terminfo/`*?* directory, where *?* stands for the first character of the name of your terminal type. For example, the directory `/usr/lib/terminfo/a` will contain a file for each supported terminal which has a name that begins with the letter `a`. Alternatively, if you have installed the Terminal Utilities subset, you can use *infocmp*(1M) to check the system support for your terminal.

To install a new terminal, follow these procedures:

1. Refer to the manufacturer's information you received with your new terminal to determine the appropriate *baud rate* and how to set it. The baud rate is the speed at which data is to be transmitted to and from your terminal. Follow the manufacturer's instructions to set the baud rate on your terminal.

2. Refer to your hardware manufacturer's information to determine how the ports on your board are labeled and where to plug in the terminal. Plug the new terminal into the appropriate port on your main board or other serial board.

3. Log in as `sysadm` and select `Modify Tty Parameters` from the `Tty Management` menu under `Machine`, or type `sysadm chgtty` to access the `Modify Tty Parameters` form directly. The system displays a screen similar to this:

```
Disk       File       Machine     Software   User       Help       Quit
Modify parameters or change status of tty lines

                   Current Disk Usage
                     Modify Tty Parameters

                    Current Tty Parameters

   Device:       <          >        Timeout:

   Gettydefs speed parameter:
   <                                                    >

   Can this line be used for input?        <   >

   Comment:

          OK              CANCEL             HELP
```

Complete all fields on the form.

Multi-User Operation Requirements

The following are requirements for the computer to support a remote terminal:

- The `TERM` shell variable must match the type of terminal attached to your computer. This ensures correct operation of screen-oriented applications, such as the `vi` editor. The *login*(1) program automatically sets TERM to the terminal name listed in `/etc/ttytype` for the particular tty device, for example, AT386 for the system console. At each remote terminal, you can set `TERM` from the command line by typing:

```
$ TERM=terminal_name
$ export TERM
```

Whenever you add a directly attached terminal, you should add a line to the `/etc/ttytype` file, of the form *terminal_name device_name*. For example:

```
wy60 tty01
```

This will set the terminal to one particular type for anyone logging in on that tty line (device name). This is particularly useful for terminals that are directly connected; it probably should not be used for modem connections since you cannot necessarily predict the type of terminal used with a modem.

- The terminal must have the correct baud rate set.
- The remote terminal will probably require a null-modem cable or adapter if directly linked to your computer (rather than through a modem over a telephone line).

Installing Software Support for Additional Terminals

Some of the software included in the Base operating system defines the terminals you can add to your computer. This group of terminal characteristics is known as the Terminal Information Library (`terminfo`). Each entry in the library represents one supported terminal.

By default, the only supported terminals are: the console (`AT386`), the DEC® VT100™ (`vt100`), and all ANSI-compatible terminals (`ansi`).

If you are adding a different terminal to your computer, you must add the additional terminal information. These entries are packaged separately in the Terminal Utilities optional subset, which is used to install the `terminfo` database.

If adding a terminal type for which no `terminfo` entry is supplied, ask the terminal vendor for a `terminfo` description for the terminal. If this is not available, see if the terminal has an emulation mode that uses a supported terminal (for example, `vt100` or `wyse60`). Otherwise, a `terminfo` must be created. See *terminfo*(4).

Configuring and Maintaining a Serial Line Printer

A printer is added to the system using the `sysadm lpmgmt` command. Printers can be either serial or parallel. A serial printer must be plugged into a serial port, such as `COM1` or `COM2`; a parallel printer must be plugged into a parallel port, such as `LPT1` or `LPT2`. Printers can be associated with tty lines, just as terminals are.

The system supports a set of default printer types. Your system may be configured to support additional or different printers. Before you attempt to install a new printer, refer to the documentation provided by your vendor or to your hardware manufacturer's instructions. Refer to the *INTERACTIVE UNIX System User's Guide* for information about using `sysadm lpmgmt` or `sysadm lprmgmt` to add a printer.

Configuring Other Directly Connected Devices

Other devices, such as another computer, can be connected to your computer using an RS-232 cable. Configuring your system to support such a connection is similar to configuring your system to support a modem. The same type of entry must be made in the `/etc/inittab` file to allow bidirectional or single-directional communication.

The difference lies in the `/usr/lib/uucp/Devices` file entry. Chapter 8, "Basic Networking Administration," describes the `/usr/lib/uucp/Devices` file and the proper entries for directly connected devices.

Basic Networking Administration 8

This chapter describes the administration of the Basic Networking Utilities (BNU). It allows you to communicate with other INTERACTIVE UNIX System computers using either dial-up or hardwired communication lines. The BNU comes as part of the base INTERACTIVE UNIX Operating System in the Basic Networking optional subset.

The installation script for the Base system is normally used to select the type(s) of modems that you will be using. However, if you have not made a selection, refer to Chapter 7, "Adding Modems, Printers, and Other Serial Devices." The basic instructions for setting up your computer to communicate with other computers is provided there. This includes connecting your computer to a modem so you can send electronic mail. Once a particular modem has been selected, refer to Chapter 7, "Adding Modems, Printers, and Other Serial Devices," for additional details.

Basic Networking is complex; the documentation included in this chapter covers only the information most important for you.

Terms You Need to Know

The following list contains some terms used in Basic Networking and a brief description of each item:

local machine
: Refers to the machine on the "near" end of a communication link, normally your computer.

remote machine
Refers to a machine on the "far" end of a communication link, normally a machine to which your computer talks.

active machine
A machine that has Basic Networking and the hardware required to establish communication links (for example, Auto Dial Modem).

passive machine
A machine that has Basic Networking but does not have the hardware required to establish communication links.

network
A group of machines set up to exchange information and resources.

node
A terminating point (machine) on a network.

UUCP
Indicates a group of programs and files that allow systems to send or copy information from one system to another. UUCP means "UNIX-to-UNIX copy." In general, it refers to Basic Networking with the exception of the `cu` and `ct` programs. If "uucp" (written in lowercase) is used in text with constant width type (`uucp`), it refers specifically to the `uucp` program or login ID.

Overview of Basic Networking

Basic Networking allows machines using the INTERACTIVE UNIX Operating System or any UNIX System to communicate with one another. In general, Basic Networking allows you to do the following:

- Transfer files and send electronic mail to other UNIX System machines as background processes
- Interactively communicate with other UNIX System machines and, in some cases, non-UNIX System machines
- Execute restrictive subset commands on a remote machine without logging in
- Call a remote terminal and allow the user of that terminal to log in on your system

Hardware Requirements

Before your computer can communicate with a remote machine, a communication link must be established to the remote machine. There are two types of hardware used to establish a communication link to another machine.

The first is a direct link from a serial port on the computer to a serial port on the other machine. This type of connection is useful when two machines communicate with each other on a regular basis. Even though the RS-232 standard recommends that direct links be limited to 50 feet or less, two machines can be separated by several hundred feet provided that noise on the direct link does not become a problem. If noise becomes a problem or greater distance is needed between the two machines, the transfer rate may need to be decreased or limited distance modems placed at each end of the connection.

The second type of communication link uses the telephone network. In this type of link, the machine that establishes the connection (local machine) must have an Automatic Call Unit (ACU). The ACU dials the specified telephone number upon request from Basic Networking. The called (remote) machine must have a telephone modem capable of answering incoming calls so that other machines can contact it through the telephone network. The INTERACTIVE UNIX System supports a number of automatic dial modems as ACUs. Refer to Chapter 7, "Adding Modems, Printers, and Other Serial Devices," for details.

The Basic Networking Software

Basic Networking is composed of software programs, daemons (background routines), and a supporting database. The supporting database contains support files that store information such as telephone numbers, location of the devices (hardware) used to establish links, and security restrictions. The software programs and a skeleton database are supplied in Basic Networking.

The BNU Directories and Their Purpose

There are several directories that contain the programs and support files of Basic Networking. Some of these directories are unique to Basic Networking, while others are also common to the INTERACTIVE UNIX Operating System. The directories used by Basic Networking are:

`/usr/bin`

This directory is used by the INTERACTIVE UNIX Operating System and by Basic Networking to store executable programs.

`/usr/lib/uucp`

This directory is the "home" directory for the `uucp` administrative login. It contains the files of the supporting database and some executable programs.

`/usr/spool/locks`

This directory contains the lock (LCK) files for the Basic Networking hardware devices. Lock files prevent duplicate conversations and multiple attempts to use the same device.

`/usr/spool/uucp`

This directory is the "spool directory" for work that is to be processed by Basic Networking. It contains a tree-like structure of subdirectories associated with remote machines that your computer communicates with or has communicated with recently. These subdirectories are also used for administrative purposes such as storing log and status information.

`/usr/spool/uucppublic`

This directory is the "public" directory for UUCP transfers. The public directory is used to store files that have been sent to your computer. Some remote machines may be restricted to placing files in this directory, while others may have permission to place files elsewhere.

The BNU Software Programs and Their Purpose

There are several types of programs associated with Basic Networking. Some are used by regular users to transfer data and obtain status information, while others are used for administration purposes or are executed internally. A brief description of the programs and their purpose follows.

User Programs

`cu`

This program connects your computer to a remote machine and lets you be logged in on both machines at the same time. This allows you to transfer files or execute commands on either machine.

`ct`

This program connects your computer to a remote terminal and allows you to log in from that terminal. The user of the remote terminal may call into the computer and request that the computer call the remote terminal back. In this case, the computer drops the initial link so that the modem will be available when it is called back.

`uucp`

This program allows you to transfer files between UNIX Systems and performs all of the preliminary work to allow you to send files to remote machines. It creates work files that contain the instructions for transferring the queued file(s). Depending on the options specified, it may make a copy of the file to be transferred in the spool directory. These files are called data files.

Once the work and data files have been created, `uucp` calls the `uucico` daemon that attempts to contact the remote machine to deliver the files.

`uuto`

This program works very similarly to the `uucp` program. In fact, it calls the `uucp` program to create work and data files. The main difference between `uuto` and `uucp` is the way the transferred files are placed on the remote machine. With `uucp`, you can specify a path name on the remote machine where you want the files to be placed. With `uuto`, all transferred files are placed in the directory `/usr/spool/uucppublic/receive`. See *uuto*(1C) for additional information.

`uupick`

When files are transferred to a machine using `uuto`, `uupick` can be used to retrieve the files under `/usr/spool/uucppublic/receive`.

`uux`

This program creates work files, data files, and execute files for running commands on a remote machine. The work file contains the same information as files created by `uucp` and `uuto`. The execute files contain the command string to be run on the remote machine and a list of the data files. The data files are those files required for the command execution.

`uustat`

This program displays status information for requested transfers (`uucp`, `uuto`, or `uux`). It also provides a means of controlling queued transfers.

Administrative Programs

`uulog`

This program displays the contents of a specified machine's log file. Individual log files are created for each remote machine that your computer communicates with using the `uucp`, `uuto`, and `uux` programs.

`uucleanup`

This program has several functions associated with the cleanup of the spool directory. It is usually executed out of the `uudemon.clean` shell script that is started by `cron`. See *cron*(1M) for additional information.

`Uutry`

This program is a shell script used to test call processing capabilities with a moderate amount of debugging. It invokes the `uucico` daemon to establish the communication link between your computer and the specified machine.

`uucheck`

This program checks for the presence of Basic Networking directories, programs, and support files. It is also capable of checking certain parts of the `Permissions` file.

Internal Programs

`uugetty`

This program is not needed for most purposes. It is very similar to the `getty` program except it permits a line (port) to be used in both directions. The `uugetty` program allows users to log in on your computer; if the line is not in use, it will allow `uucico`, `cu`, or `ct` to use it for dialing out. If one of these programs attempts to dial out when the line is busy, `uugetty` denies the requester permission and echoes a message indicating that the device is unavailable. The `uugetty` is executed by the `init` program.

UUCP Daemons and Their Purpose

Basic Networking contains three daemons (routines that run as background processes to handle file transfers and command executions).

`uucico`
: This daemon is referred to as the transport program for UUCP requests. It selects the device used for the link, establishes the link to the remote machine, performs the required login sequence, and performs permission checks. It also transfers data and execute files, logs results, and notifies specified users of transfer completions via `mail`. When the local `uucico` daemon calls a remote machine, it "talks" to the `uucico` daemon on the remote machine during the session. The `uucico` daemon is executed by several methods. It is started by the `uucp`, `uuto`, and `uux` programs to contact the remote machine after all the required data, work, and/or execute files have been created. It is also started by the `uusched` and `Uutry` programs.

`uuxqt`
: This daemon is the execution program for remote execution requests. It searches the spool directory for execute files (`X.`) that have been sent from a remote machine. When an `X.` file is found, `uuxqt` opens it to get the list of data files required for the execution. It then checks to see if the required data files are available and accessible. If the files are present and can be accessed, `uuxqt` checks the `Permissions` file to verify that it has permission to execute the requested command. The `uuxqt` daemon is executed out of the `uudemon.hour` shell script that is started by `cron`.

`uusched`
: This daemon schedules the queued work in the spool directory. Before starting the `uucico` daemon, `uusched` randomizes the order in which remote machines will be called. The `uusched` is executed out of a shell script called `uudemon.hour` that is started by `cron`.

The BNU Supporting Database Files and Their Purpose

As mentioned earlier, several of the Basic Networking programs require information contained in support files. These support files are located in the `/usr/lib/uucp` directory. The `cu`, `ct`, `uucico`, and `uuxqt` programs require supporting information from the following files:

`Devices`
: This file contains information concerning the location and line speed of the automatic call unit, direct links, and possibly network devices.

`Dialers`
: This file contains character strings required to negotiate with network devices (automatic calling devices) in the establishment of connections to remote computers (non-801-type dialers).

`Systems`
: This file contains information needed by the `uucico` daemon (and possibly the `cu` program) to establish a link to a remote machine. It contains information such as the name of the remote machine, the name of the connecting device associated with the remote machine, when the machine can be reached, the telephone number, the login ID, the password, and so on.

`Dialcodes`
: This file contains dial-code abbreviations that may be used in the phone number field of `Systems` file entries.

`Permissions`
: This file defines the level of access granted to machines when they attempt to transfer files or remotely execute commands on your computer.

There are several other files that may be considered part of the supporting database, but these files are not directly related to the process of establishing a link and transferring files. For this reason, discussion of these files is reserved for the "Administrative Files" section later in this chapter.

How Basic Networking Operates

There are five programs that allow your computer to communicate with remote machines. The following sections briefly describe what happens when you execute these programs.

`ct` *Program—Connect a Terminal*

The `ct` program instructs your computer to initiate a call to a remote terminal and issue a `getty` to that remote terminal. The `ct` command line must contain the telephone number of the remote terminal. Of course, the remote terminal must be attached to a modem that will automatically answer the call.

When the `ct` command line is issued, the `ct` program searches for an automatic dialer in the `Devices` file with a transfer rate that matches what was specified in the command line. If no transfer rate was specified, it defaults to 1200 bps. When `ct` finds the dialer to be used, it attempts to dial the telephone number specified in the command line. If no dialer is available, `ct` asks if it should wait for an available dialer and, if so, how many minutes it should wait. An option is available to override this dialogue. When the modem at the remote terminal answers the call from your computer, it is issued a `getty` (login) process. At this point, the user at the remote terminal may attempt to log in.

The user at a remote terminal can call your computer, log in, and request that the computer call the remote terminal back using the `ct` command. If this scenario is used, the remote user issues a `ct` command, and the link from the remote terminal is dropped. After `ct` finds an available dialer in the `Devices` file, it calls the remote terminal back.

`cu` *Program—Call a UNIX System*

The `cu` program enables you to call another machine and log in as a remote user. The telephone number or node name of the remote machine is required in the command line. If the telephone number is specified, it is passed on to the automatic dial modem. If a system name is specified, the telephone number is obtained from the associated `Systems` file entry. If an automatic dial modem is not used to establish the connection, the line (port) associated with the direct link to the remote machine can be specified in the command line.

If an automatic dial modem is used, the `cu` program will search for an automatic dialer in the `Devices` file with a transfer rate that matches what was specified in the command line. If no speed is specified, the first dialer listed (if available) will be used regardless of its transfer rate. After the link has been established and you have successfully completed the login process, you will be logged in on both computers. This allows you to execute commands on either computer and/or transfer ASCII coded files from one computer to another.

After you have terminated the connection, you will still be logged in on your computer (calling computer). This command can only be executed by an active computer.

uucp *Program—UNIX-to-UNIX Copy*

The uucp command allows you to transfer file(s) to a remote computer without knowing any details of the connection. All that you are required to know is the name of the remote computer and possibly the login ID of the remote user to whom the file(s) is being sent. The details of the connection are kept in the Systems file.

When you enter a uucp command, the uucp program creates a work file and possibly a data file for the requested transfer. The work file contains information required for transferring the file(s). The data file is simply a copy of the specified source file. After these files have been created in the spool directory, the uucico daemon will start.

The uucico daemon attempts to establish a connection to the remote machine that is to receive the file(s). It first gathers the information required for establishing a link to the remote machine from the Systems file so that it knows what type of device to use in establishing the link. uucico then searches the Devices file for devices that match the requirements listed in the Systems file. After uucico has found an available device, it attempts to establish the link and log in on the remote machine.

When uucico logs in on the remote machine, it starts the uucico daemon on the remote machine. The two uucico daemons then negotiate the line protocol to be used in the file transfer(s). The local uucico daemon then transfers the file(s) to the remote machine, and the remote uucico places the file in the specified path name(s) on the remote machine. After your computer completes the transfer(s), the remote machine may send files that are queued for your computer. The remote machine can be denied permission to transfer these files with an entry in the Permissions file. If this is done, the remote machine must establish a link to your computer to perform the transfers. If the remote machine or the device selected to make the connection to the remote machine is unavailable, the request remains queued in the spool directory. Each hour, cron starts uudemon.hour, which in turn starts the uusched daemon.When the uusched daemon starts, it searches the spool directory for the remaining work files, generates the random order in which these requests are to be processed, and then starts the transfer process (uucico) described in the previous paragraphs.

The transfer process described generally applies to an active machine. An active machine (one with calling hardware and Basic Networking software) can be set up to "poll" a passive machine. A passive machine can queue file transfers (because it has Basic Networking software), but it cannot call the remote machine because it does not have the required hardware. The Poll file (/usr/lib/uucp/Poll) contains a list of machines that are to be polled in this manner. For additional information, refer to the discussion of the Poll file in the "Supporting Database" section and the discussion of uudemon.poll in the "UUCP and cron" section, both later in this chapter.

uuto *Program—Public UNIX-to-UNIX Copy*

The uuto program uses the uucp program to build work files and data files in the spool directory for requested transfers. The difference is that the uuto command does not allow you to specify a path name as a destination for the file. The uuto command automatically puts the file in a directory under /usr/spool/uucppublic/receive. Once the transfer is complete, mail is sent to the appropriate user indicating that a file has arrived and has been placed in the public area. That user can then use the uupick command to retrieve that file. The uupick command searches the public area for files destined for the user and allows the user to interactively delete, print, or move the file to a named directory.

uux *Program—UNIX-to-UNIX Execution*

The uux command allows commands to be executed on a remote machine. It gathers files from various computers, executes the specified command on these files, and sends the standard output to a file on the specified computer. Remote mail is implemented using the uux program, but its execution is embedded in the standard mail command. For security, many machines limit the list of commands that can be executed via uux to the default (receipt of mail).

When the uux command is issued, the uux program creates an execute (X.) file that contains the names of the files required for execution, your login name, the destination of the standard output, and the command to be executed. The uux command also creates work (`C.`) files that are used to gather the files required for execution. These files are then sent to the remote machine, along with the execute file, by the `uucico` daemon and placed in the remote spool directory.

Periodically, the `uuxqt` daemon on the remote machine is started to search for `X.` files in the spool directory. When it finds an `X.` file, the `uuxqt` daemon checks to see if all the required data files are available and accessible. It then checks the `Permissions` file to verify that the command(s) listed can be performed. After execution, `uuxqt` sends the standard output to a file on the specified computer.

Administrative Files

The files and tasks associated with the operation of Basic Networking are discussed here. The amount of effort required to administer Basic Networking depends on the amount of traffic that enters or leaves your computer. For an average computer, little, if any, intervention is required by the automatic cleanup functions. A computer with a large amount of traffic may require more attention as problems arise.

As should be evident, the UUCP facilities make up the bulk of Basic Networking. These can generally be defined as all of the programs and support files in Basic Networking with the exception of the `ct` and `cu` programs.

`TM.`—*Temporary Data File*

This data file is created under the spool directory, `/usr/spool/uucp/`*XXXX*, when receiving a file from another machine. The directory *XXXX* has the same name as the remote machine that is sending the file. The temporary data file name has the following format:

```
TM.pid.ddd
```

pid is a process ID and *ddd* is a sequential 3-digit number starting at zero.

After the entire file is received, the `TM.` file is moved to the path name specified in the command line. If the file was sent via the `uuto` program, the file is automatically moved to the public area. If processing is abnormally terminated, the `TM.` file may remain in the *XXXX* directory. This file should be periodically removed.

LCK—*Lock File*

The lock file is created in the /usr/spool/locks directory for each device in use. A lock file prevents duplicate conversations and multiple attempts to use the same calling device. The file name has the following format:

```
LCK..str
```

where *str* is either a device or computer name. The file may be left in the spool directory if runs abort (usually on computer crashes). The lock file will be ignored (reused) after the parent process is no longer active.

Work (C.) File

The work file is created in a spool directory when work (transfers or remote command executions) has been queued for a remote computer. The name has the following format:

```
C.sysnxxxx
```

where *sys* is the name of the remote computer, *n* is the ASCII character representing the grade (priority) of the work, and *xxxx* is the 4-character job sequence number assigned by UUCP. A work file contains the following:

- Full path name of the file to be sent or requested
- Full path name of the destination or ~user/filename

Note – The ~ is shorthand for /usr/spool/uucppublic and must be included if the full path name is not used.

- User login name
- List of options
- Name of the associated data file in the spool directory (If the -c or -p option was specified, a dummy name [D.0] will be used.)
- Mode bits of the source file
- Remote user's login name to be notified upon completion of the transfer

Data (`D.`) *File*

The data file is created when it is specified in the command line to copy the source file to the spool directory. The file name has the following format:

```
D.sysnxxxx
```

where *sys* is the name of the remote computer, *n* is the character representing the grade (priority) of the work, and *xxxx* is the 4-character job sequence number assigned by UUCP. The 4-character job sequence number may be followed by a subjob number that is used when there are several `D.` files created for a work (`C.`) file.

Execute (`X.`) *File*

The execute file is created in the spool directory prior to remote command executions. The file name has the following format:

```
X.sysnxxxx
```

where *sys* is the name of the remote computer, *n* is the character representing the grade (priority) of the work, and *xxxx* is the 4-character sequence number assigned by UUCP.

The execute file contains the following information:

- Requester's login and computer name
- Name of file(s) required for execution
- Input to be used as the standard input to the command string
- Computer and file name to receive standard output from the command execution
- Command string
- Option lines for return status requests

Machine Log File

A log file is created for each remote machine with which your computer communicates. Each machine may have four log files, one for `uucico`, `uuxqt`, `uux`, and/or `uucp` requests, depending on the type of communication that has taken place. The log files are kept in the directory `/usr/spool/uucp/.Log`. Each day, these log files are combined and stored in the directory `/usr/spool/uucp/.Old` when `uudemon.clean` is executed. The combined files are kept 3 days before they are removed. If space is a problem, the administrator may consider reducing the number of days the files are kept by modifying the `uudemon.clean` shell file.

Supporting Database

The database that supports Basic Networking is composed of several support files. These support files contain information required by the `uucico` and `uuxqt` daemons during file transfers or remote command executions. All of the support files are located in the `/usr/lib/uucp` directory.

`Devices` File

The `Devices` file (`/usr/lib/uucp/Devices`) contains the information for all of the devices that can be used to establish a link to a remote machine. It contains information for both automatic call units, direct links, and network connections. Although provisions are made for several types of devices, only modems and direct links are supported.

This file works very closely with the `Dialers`, `Systems`, and `Dialcodes` files. It may be beneficial to become familiar with these files before attempting to understand the `Devices` file.

Each entry in the `Devices` file has the following format:

Type Line Line2 Class Dialer-Token-Pairs (DTP)

Each field (separated by a space) is defined as follows.

- *Type*

  This field may contain one of five keywords:

`Direct`
: This keyword indicates a direct link to another computer (for `cu` connections only).

`ACU`
: This keyword indicates that the link to a remote computer is made through an automatic call unit (Automatic Dial Modem). This modem may be connected either directly to the computer or indirectly through a Local Area Network (LAN) switch.

Network
: This keyword indicates that the link is established through a LAN switch where *Network* is replaced with either `micom` or `develcon`. These two LAN switches are the only ones that contain caller scripts in the `Dialers` file. Other switches may be used if caller scripts are constructed and placed in the `Dialers` file.

`Modem Control`
: This keyword causes the device to be opened with `O_NDELAY` set (so the open does not hang waiting for carrier). After the open, `O_NDELAY` is cleared.

System-Name
: This keyword indicates a direct link to a particular machine where *System-Name* is replaced by the name of the particular computer. This naming scheme is used to convey the fact that the line associated with this `Devices` entry is for a particular machine.

The keyword used in the *Type* field is matched against the third field of `Systems` file entries as follows:

Devices: `ACU tty01,M - 1200 penril`

Systems: `eagle Any ACU 1200 3-2-5-1 ogin: nuucp ssword: Oakgrass`

- *Line*

This field contains the device name of the line (port) associated with the `Devices` entry. For instance, if the Automatic Dial Modem for a particular entry was attached to the `/dev/tty01` line, the device name would be `tty01`. The `,M` indicates that modem control is being used.

- *Line2*

  If the ACU keyword was used in the *Type* field and the ACU is an 801-type dialer, this field would contain the device name of the 801 dialer. It should be noted that 801-type ACUs do not contain a modem. Therefore, a separate modem is required and would be connected to a different line (defined in the *Line* field). This means that one line would be allocated to the modem and another to the dialer. Since the computer will not normally use this type of configuration, this field is ignored but must contain a pseudo entry as a placeholder (use a "-" as a placeholder).

- *Class*

  If an ACU keyword is used, this may be just the speed of the device. It may contain a letter and speed (for example, C1200, D1200, and so on) to differentiate between classes of dialers (centrex or DIMENSION PBX). This is necessary because many larger offices may have more than one type of telephone network. One network may be dedicated to serving only internal office communications, while another handles the external communications. Therefore, it is necessary to distinguish which line(s) should be used for internal communications and which should be used for external communications. The same distinction must be made in the Systems file because a match is made against the fourth field of Systems file entries as follows:

```
Devices: ACU tty01,M - D1200 penril

Systems: eagle Any ACU D1200 3-2-5-1 ogin: nuucp ssword: Oakgrass
```

  Some devices can be used at any speed, so the keyword Any may be used in the *Class* field. If Any is used, the line will match any speed requested in a Systems entry. If this field is Any\ and the Systems *Class* field is Any, the speed defaults to 1200 bps.

- *Dialer-Token-Pairs*

  This field contains pairs of dialers and tokens. The dialer portion may be an automatic dial modem or direct for Direct Link devices. The token portion may be supplied immediately following the dialer; or if not present, it can be taken from the Systems file.

This field has the following format:

dialer-token dialer-token

where the last pair may or may not be present, depending on the associated device (dialer). In most cases, the last pair will contain only a dialer and the token is retrieved from the `Phone` field of the `Systems` entry. The *DTP* field may be structured in four ways, depending on the device associated with the entry:

1. If a direct link is established to a particular computer, the *DTP* field of the associated entry will contain the keyword `direct`. This is true for both types of direct link entries, `Direct` and *System-Name* (refer to discussion on the *Type* field).
2. If an automatic dialing modem is connected directly to a computer port, the *DTP* field of the associated `Devices` entry will only have one pair, normally the name of the modem. This name is used to match the particular `Devices` entry with an entry in the `Dialers` file. Therefore, this dialer must match the first field of a `Dialers` file entry as follows:

```
Devices:  ACU acu1,M - 1200 ventel

Dialers:  hayes2400 =&-% "" \M\r\p\r\c $ <K\T%%\r>\c ONLINE!\m
```

Notice that only the dialer (`ventel`) is present in the *DTP* field of the `Devices` entry. This means that the token to be passed on to the dialer (in this case the telephone number) is taken from the *Phone* field of a `Systems` file entry.

3. If an automatic dialing modem is connected to a local area network (LAN), the computer must first access the switch, and the switch will make the connection to the automatic dialing modem. This type of entry would have two pairs. The dialer portion of each pair (fifth and seventh fields of entry) is used to match entries in the `Dialers` file as follows:

```
Devices:  ACU tty01 - 1200 develcon vent ventel

Dialers:  ventel =&-% "" \M\r\p\r\c $ <K\T%%\r>\c ONLINE!\m

Dialers:  develcon "" "" \pr\ps\c est:\077 \E\D\e \007
```

   In the first pair, `develcon` is the dialer and `vent` is the token that is passed to the Develcon switch to tell it which device (ventel modem) to connect to the computer. This token would be unique for each LAN switch since each switch may be set up differently. Once the ventel modem has been connected, the second pair is accessed where ventel is the dialer and the token is retrieved from the `Systems` file.

4. If a machine that you want to communicate with is on the same local network switch as your computer, your computer must first access the switch, and then the switch can make the connection to the other machine. In this type of entry, there is only one pair. The dialer portion is used to match a `Dialers` entry as follows:

```
Devices:  develcon tty01 - 1200 develcon \D

Dialers:  develcon "" "" \pr\ps\c est:\007 \E\D\e \007
```

   As shown, the token is left blank to indicate that it is retrieved from the `Systems` file. The `Systems` file entry for this particular machine will contain the token in the *Phone* field that is normally reserved for the telephone number of the machine. This type of *DTP* contains an escape character (`\D`) that ensures that the contents of the *Phone* field will not be interpreted as a valid entry in the `Dialcodes` file.

There are two escape characters that may appear at the end of a *DTP* field:

\T

Indicates that the *Phone* (token) field should be translated using the Dialcodes file. This escape character is normally placed in the Dialers file for each caller script associated with an automatic dial modem (penril, ventel, and so on). Therefore, the translation will not take place until the caller script is accessed.

\D

Indicates that the *Phone* (token) field should not be translated using the Dialcodes file. If no escape character is specified at the end of a Devices entry, the \D is assumed (default). A \D is also used in the Dialers file with entries associated with network switches (develcon and micom).

Dialers *File*

The Dialers file (/usr/lib/uucp/Dialers) is used to specify the initial handshaking that must take place on a line before it can be made available for transferring data. This initial handshaking is usually a sequence of ASCII strings that are transmitted and expected and is often used to dial a telephone number using an ASCII dialer (such as the AT&T 2212C Modem). As shown in the above examples, the fifth field in a Devices file entry is used as an index into the Dialers file. Here an attempt is made to match the *Devices* field with the first field of each Dialers entry. In addition, each odd numbered *Devices* field starting with the seventh position is used as an index into the Dialers file. Changes must be made using a text editor, such as ed or vi.

If the match succeeds, the Dialers entry is interpreted to perform the dialer negotiations. The first field matches the fifth and additional odd numbered fields in the Devices file. The second field is used as a translate string (the first of each pair of characters is mapped to the second character in the pair).

This is usually used to translate = and - into whatever the dialer requires for "wait for dial tone" and "pause." The remaining fields are "expect-send" strings. The following `Dialers` file entries are typical examples:

```
att4000  =,-,  "" \M\dat\r\c OK\r \EATDT\T\r\c CONNECT \m\c
ventel   =&-%  "" \M\r\p\r\c $ <K\T%%\r>\c ONLINE!\m
hayes    =,-,  "" \M\dAT\r\c OK\r \EATDT\T\r\c CONNECT\m\c
vadic    =K-K  "" \005\p *-\005\p-*\005\p-* D\p BER? \E\T \r\c LINE
develcon       ""     "" \pr\ps\c est:\007 \E\D \007
micom    ""    "" \s\c NAME? \D\r\c GO
direct
hayes2400 =,-, "" \M\pAT\r\dATQ0V1\v\c OK\r  ATDT\T\r\m\c  CONNECT
```

The meaning of some of the escape characters (those beginning with `\`) used in the `Dialers` file are shown in the following list:

`\p`
: Pauses (approximately 1/4 to 1/2 second).

`\d`
: Delays (approximately 2 seconds).

`\D`
: Phone number or token without `Dialcodes` translation.

`\M`
: Sets no modem control.

`\T`
: Phone number or token with `Dialcodes` translation.

`\K`
: Inserts a `BREAK`.

`\E`
: Enables echo checking (for slow devices).

`\e`
: Disables echo checking.

`\0`*digit(s)*
: Sets timeout for send/expect.

`\r`
: Carriage return.

`\c`
: No newline.

`\m`
: Restores modem control.

`\n`
: Sends newline.

nnn
: Sends octal number.

Additional escape characters that may be used are listed in the next section, "Systems File."

The `penril` entry in the `Dialers` file is executed as follows. First, the telephone number argument is translated, replacing any = with a `W` (wait for dialtone) and replacing any - with a `P` (pause). The handshake given by the remainder of the line works as follows:

`""`
: Waits for nothing.

`\d`
: Delays for 2 seconds.

`>`
: Waits for a >.

`s\p9\c`
: Sends an `s`, pauses for 1/2 second, sends a `9`, sends no terminating newline.

`)-W\p\r\ds\p9\c-)`
: Waits for a `)`. If it is not received, processes the string between the - characters as follows. Sends a `W`, pauses, sends a carriage return, delays, sends an `s`, pauses, sends a `9` without a newline and then waits for the `)`.

`y\c`
: Sends a `y` without a newline.

`:`
Waits for a `:`.

`\M`
Sets no modem control (CLOCAL).

`\m`
Restores modem control. Typically, CLOCAL is set for the duration of the dialer chat, then cleared (so `uucico`, `cu`, or `ct` will detect dropped lines) once connected to the remote system.

`\E\TP`
Enables echo checking. (From this point on, whenever a character is transmitted, it will wait for the character to be received before doing anything else.) Then, sends the telephone number followed by a pause character (`P`). The `\T` means take the telephone number passed as an argument and apply the `Dialcodes` translation and the modem function translation specified by field number 2 of this entry.

`>`
Waits for a >.

`9\c`
Sends a `9` without a newline.

`OK`
Waits for the string `OK`.

`Systems` *File*

The `Systems` file (`/usr/lib/uucp/Systems`) contains the information needed by the `uucico` daemon to establish a communication link to a remote machine. Each entry in the file represents a machine that can be called by the computer. Furthermore, only those machines listed in the `Systems` file will be permitted to communicate with your computer via Basic Networking (UUCP) unless the execute permissions for `remote.unknown` are changed to permit communications with other machines. More than one entry may be present for a particular machine. The additional entries represent alternate communication paths that will be tried in sequential order.

Each entry in the `Systems` file has the following format:

```
System-Name Time Type Class Phone Login
```

Each field is defined as follows:

- *System-Name*

  This field contains the node name of the remote machine.

- *Time*

  This field is a string that indicates the day of week and time of day when the remote machine can be called. The day portion may be a list containing some of the following:

```
Su Mo Tu We Th Fr Sa
```

`Wk:`
: For any weekday.

`Any:`
: For any day.

`Never:`
: For a passive arrangement with the remote machine. In this case, the computer will never initiate a call to the remote machine. The call must be initiated by the remote machine. The computer is in a passive mode in respect to the remote machine. (See "Permissions File" later in this chapter.)

The time should be a range of times such as 0800–1230. If no time portion is specified, any time of day is assumed to be allowed for the call. Note that a time range that spans 0000 is permitted. For example, 0800–0600 means all times are allowed other than times between 6 A.M. and 8 A.M. An optional subfield is available to specify the minimum time (in minutes) before a retry following a failed attempt. The subfield separator is a semicolon (;). For example, "`\Any ;9`" is interpreted as call any time, but wait at least 9 minutes before retrying if a failure occurs.

- *Type*

  This field contains the device type that should be used to establish the communication link to the remote machine. The `Devices` file is searched for the device type listed, and the device found is used to establish the connection (if available). The following keywords may appear in this field:

  `ACU`
  : This keyword indicates that the link to a remote computer is made through an automatic call unit (Automatic Dial Modem). This modem may be connected either directly to the computer or indirectly through a Local Area Network (LAN) switch.

  Network
  : This keyword indicates that the link is established through a LAN switch, where `Network` is replaced with either `micom` or `develcon`. These two LAN switches are the only ones that contain caller scripts in the `Dialers` file. Other switches may be used if caller scripts are constructed and placed in the `Dialers` file.

  System-Name
  : This keyword indicates a direct link to a particular machine where *System-Name* is replaced by the name of the particular computer (should be the same as field one).

  The keyword used in this field is matched against the first field of `Devices` file entries as follows:

Systems:	`eagle Any ACU D1200 3-2-5-1 ogin: nuucp ssword: Oakgrass`
Devices:	`ACU tty01 - D1200 penril`

- *Class*

  This field is used to indicate the transfer speed of the device used in establishing the communication link. It may contain a letter and speed (for example, C1200, D1200, etc.) to differentiate between classes of dialers (refer to the discussion on the "Devices File," *Class* field). Some devices can be used at any speed, so the keyword `Any` may be used. This field must match the *Class* field in the associated `Devices` entry as follows:

```
Systems: eagle Any ACU D1200 3-2-5-1 ogin: nuucp ssword: Oakgrass

Devices: ACU tty01 - D1200 penril
```

- *Phone*

  This field is used to provide the telephone number (token) of the remote machine for automatic dialers (LAN switches). The telephone number is made up of an optional alphabetic abbreviation and a numeric part. The abbreviation must be one that is listed in the `Dialcodes` file. In this string, an equals sign (=) tells the ACU to wait for a secondary dial tone before dialing the remaining digits. A dash in the string (-) instructs the ACU to pause 4 seconds before dialing the next digit.

  If your computer is connected to a LAN switch, you may access other machines that are connected to that switch. The *Systems* entries for these machines will not have a telephone number in the *Phone* field. Instead, this field will contain the "token" that must be passed on to the switch so it will know which machine the computer wishes to communicate with. The associated `Devices` entry should have a `\D` at the end of the entry to ensure that this field is not translated using the `Dialcodes` file. For direct connections, the telephone field is ignored. A dash (-) should be used as a place holder.

- *Login*

  This field contains the login information given as a series of fields and subfields of the following format:

```
[expect send] ...
```

where *expect* is the string that is received and *send* is the string that is sent when the *expect* string is received. The *expect* field may be made up of subfields of the following form:

```
expect[-send-expect]...
```

where the *send* is sent if the prior *expect* is not successfully read and the *expect* following the *send* is the next expected string. For example, with `login--login`, UUCP will expect `login`. If UUCP gets `login`, it will go on to the next field. If it does not get `login`, it will send nothing followed by a newline, then look for `login` again. If no characters are initially expected from the remote machine, the characters `""` (null string) should be used in the first *expect* field. Note that all *send* fields will be sent followed by a newline unless the *send* string is terminated with a `\c`.

There are several escape characters that cause specific actions when they are a part of a string sent during the login sequence. The following escape characters are useful in UUCP communications:

`\N`
: Sends a null character.

`\b`
: Sends a backspace character.

`\c`
: If at the end of a string, suppresses the newline that is normally sent. Ignored otherwise.

`\d`
: Delays 2 seconds before sending or reading more characters.

`\p`
: Pauses for approximately 1/4 to 1/2 second.

`\n`
: Sends a newline character.

`\r`
: Sends a carriage return.

`\s`
: Sends a space character.

`\t`
Sends a Tab character.

`\\`
Sends a \ character.

`\Onnn\`
Set timeout for expect-send.

`EOT`
Sends an EOT character (actually EOT newline is sent twice).

`BREAK`
Sends a break character.

`\ddd`
Collapses the octal digits (*ddd*) into a single character and sends that character.

`Dialcodes` *File*

The `Dialcodes` file (`/usr/lib/uucp/Dialcodes`) contains the dial-code abbreviations used in the *Phone* field of the `Systems` file. Each entry has the following format:

```
abb dial-seq
```

where *abb* is the abbreviation used in the `Systems` file (*Phone* field), and *dial-seq* is the dial sequence that is passed to the dialer when that particular `Systems` entry is accessed.

The entry

```
jt 9=847-
```

would be set up to work with a *Phone* field in the `Systems` file such as `jt7867`. When the entry containing `jt7867` is encountered, the sequence 9=847-7867 would be sent to the dialer.

`Permissions` *File*

The `Permissions` file (`/usr/lib/uucp/Permissions`) is used to specify the permissions that remote machines have with respect to login, file access, and command execution. Options are provided for restricting the ability to request files and the ability to receive files queued by the local site. In addition, an option is available to specify the commands that a remote site can execute on the local machine. Changes must be made using a text editor, such as `ed` or `vi`.

How Entries Are Structured

Each entry is a logical line with physical lines terminated with a \ to indicate continuation. Entries are made up of options delimited by white space. Each option is a name/value pair. These are constructed by an option name followed by an = and the value. Note that no white space is allowed within an option assignment.

Comment lines begin with a pound sign (#), and they occupy the entire line up to a newline character. Blank lines are ignored (even within multi-line entries). There are two types of `Permissions` entries:

`LOGNAME`
: Specifies permissions that take effect when a remote machine logs in on (calls) your computer.

`MACHINE`
: Specifies permissions that take effect when your computer logs in on (calls) a remote machine.

`LOGNAME` entries contain a `LOGNAME` option, and `MACHINE` entries contain a `MACHINE` option.

Considerations

The following items should be considered when using the `Permissions` file to restrict the level of access granted to remote machines:

1. All login IDs used by remote machines to log in for UUCP-type communications must appear in one and only one `LOGNAME` entry.
2. Any site that is called whose name does not appear in a `MACHINE` entry will have the following default permissions/restrictions:

- Local send and receive requests will be executed.
- The remote machine can send files to your computer's `/usr/spool/uucppublic` directory.
- The commands sent by remote machine for execution on your computer must be one of the default commands, usually `rmail`.

Options

This section provides the details of each option, specifying how they are used and their default values.

- `REQUEST`

  When a remote machine calls your computer and requests to receive a file, this request can be granted or denied. The `REQUEST` option specifies whether or not the remote machine can request to set up file transfers from your computer. The string:

```
REQUEST=yes
```

  specifies that the remote machine can request to transfer files from your computer. The string:

```
REQUEST=no
```

  specifies that the remote machine cannot request to receive files from your computer. The `no` string is the default value. It will be used if the `REQUEST` option is not specified. The `REQUEST` option can appear in either a `LOGNAME` (remote calls you) entry or a `MACHINE` (you call remote) entry.

- `SENDFILES`

  When a remote machine calls your computer and completes its work, it may attempt to take work that your computer has queued for it. The `SENDFILES` option specifies whether or not your computer can send the work queued for the remote machine. The string:

```
SENDFILES=yes
```

specifies that the computer may send the work that is queued for the remote machine as long as it logged in as one of the names in the LOGNAME option. This string is mandatory if the computer is in a "passive mode" with respect to the remote machine. The string:

```
SENDFILES=call
```

specifies that files queued in your computer will only be sent when the computer calls the remote machine. The call value is the default for the SENDFILE option. This option is only significant in LOGNAME entries since MACHINE entries apply when calls are made out to remote machines. If the option is used with a MACHINE entry, it will be ignored.

- READ and WRITE

  These options specify the various parts of the file system that uucico can read from or write to. The READ and WRITE options can be used with either MACHINE or LOGNAME entries.

  The default for both the READ and WRITE options is the uucppublic directory, as shown in the following strings:

```
READ=/usr/spool/uucppublic WRITE=/usr/spool/uucppublic
```

  The strings:

```
READ=/ WRITE=/
```

  specify permission to access any file that can be accessed by a local user with "other" permissions.

  The value of these entries is a list of path names separated by colons. The READ option is for requesting files, and the WRITE option is for depositing files. One of the values must be the prefix of any full path name of a file coming in or going out. To grant permission to deposit files in /usr/news as well as the public directory, the following values should be used with the WRITE option:

```
WRITE=/usr/spool/uucppublic:/usr/news
```

It should be pointed out that if the READ and WRITE options are used, all path names must be specified because the path names are not added to the default list. For instance, if the /usr/news path name was the only one specified in a WRITE option, permission to deposit files in the public directory would be denied.

- NOREAD and NOWRITE

The NOREAD and NOWRITE options specify exceptions to the READ and WRITE options or defaults. The strings:

```
READ=/ NOREAD=/etc WRITE=/usr/spool/uucppublic
```

would permit reading any file except those in the /etc directory (and its subdirectories—remember, these are prefixes) and writing only to the default /usr/spool/uucppublic directory. NOWRITE works in the same manner as the NOREAD option. NOREAD and NOWRITE can be used in both LOGNAME and MACHINE entries.

- CALLBACK

The CALLBACK option is used in LOGNAME entries to specify that no transaction will take place until the calling system is called back. The string:

```
CALLBACK=yes
```

specifies that your computer must call the remote machine back before any file transfers will take place.

The default for the CALLBACK option is:

```
CALLBACK=no
```

The CALLBACK option is very rarely used. Note that if two sites have this option set to yes for each other, a conversation will never get started.

- COMMANDS

The COMMANDS option can be hazardous to the security of your system. Use it with extreme care.

The uux program will generate remote execution requests and queue them to be transferred to the remote machine. Files and a command are sent to the target machine for remote execution. The COMMANDS option can be used in MACHINE entries to specify the commands that a remote machine can execute on your computer. The string:

```
COMMANDS=rmail
```

indicates the default commands a remote machine can run on your computer. If a command string is used in a MACHINE entry, the default commands will be overridden. For instance, the entry:

```
MACHINE=owl:raven:hawk:dove \
COMMANDS=rmail:rnews:lp
```

overrides the COMMAND default such that the command list for machines owl, raven, hawk, and dove now consists of rmail, rnews, and lp. In addition to the names as specified above, there can be full path names of commands. For example:

```
COMMANDS=rmail:/usr/lbin/rnews:/usr/local/lp
```

specifies that command rmail uses the default path. The default paths for the computer are /bin, /usr/bin, and /usr/lbin. When the remote machine specifies rnews or /usr/lbin/rnews for the command to be executed, /usr/lbin/rnews will be executed regardless of the default path. Likewise, /usr/local/lp is the lp command that will be executed.

Including the ALL value in the list means that any command from the remote machine(s) specified in the entry will be executed. If you use this value, you give the remote machine full access to your computer.

The string:

```
COMMANDS=/usr/lbin/rnews:ALL:/usr/local/lp
```

illustrates two points. The ALL value can appear anywhere in the string, and the path names specified for rnews and lp will be used (instead of the default) if the requested command does not contain the full path names for rnews or lp.

The VALIDATE option should be used with the COMMANDS option whenever potentially dangerous commands like cat and uucp are specified with the COMMANDS option. Any command that reads or writes files is potentially dangerous to local security when executed by the UUCP remote execution daemon (uuxqt).

- VALIDATE

  The VALIDATE option is used with the COMMANDS option when specifying potentially dangerous commands. It is used to provide a certain degree of verification of the caller's identity. The use of the VALIDATE option requires that privileged machines have a unique login/password for UUCP transactions. An important aspect of this validation is that the login/password associated with this entry be protected. If an outsider gets that information, that particular VALIDATE option can no longer be considered secure.

  A great deal of consideration should be given to providing a remote machine with a privileged login and password for UUCP transactions. Giving a remote machine a special login and password with file access and remote execution capability is like giving anyone on that machine a normal login and password on your computer. Therefore, if you cannot trust someone on the remote machine, do not provide that machine with a privileged login and password.

  The LOGNAME entry:

```
LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

specifies that if one of the remote machines that claims to be `eagle`, `owl`, or `hawk` logs in on your computer, it must have used the login `uucpfriend`. As can be seen, if an outsider gets the `uucpfriend` login/password, masquerading is trivial. But what does this have to do with the `COMMANDS` option that only appears in `MACHINE` entries? It links the `MACHINE` entry (and `COMMANDS` option) with a `LOGNAME` entry associated with a privileged login. This link is needed because the execution daemon is not running while the remote machine is logged in. In fact, it is an asynchronous process with no knowledge of what machine sent the execution request. Therefore, the real question is how does your computer know where the execution files came from?

Each remote machine has its own "spool" directory on your computer. These spool directories have write permission given only to the UUCP programs. The execution files from the remote machine are put in its spool directory after being transferred to your computer. When the `uuxqt` daemon runs, it can use the spool directory name to find the `MACHINE` entry in the `Permissions` file and get the `COMMANDS` list, or if the machine name does not appear in the `Permissions` file, the default list will be used.

The following example shows the relationship between the `MACHINE` and `LOGNAME` entries:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
COMMANDS=ALL \
READ=/ WRITE=/

LOGNAME=uucpz VALIDATE=eagle:owl:hawk \
REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

These entries provide unlimited read, write, and command execution for the remote machines `eagle`, `owl`, and `hawk`. The `ALL` value in the `COMMANDS` option means that any command can be executed by either of these machines. Using the `ALL` value gives the remote machine unlimited access to your computer. In fact, files that are only readable or writable by user `uucp` (like `Systems` or `Devices`) can be accessed using commands like `ed`. This means a user on one of the privileged machines can write in the `Systems` file as well as read it.

In the first entry, you must make the assumption that when you want to call one of the machines listed, you are really calling either `eagle`, `owl`, or `hawk`. Therefore, any files put into one of the `eagle`, `owl`, or `hawk` spool directories are put there by one of those machines. If a remote machine logs in and says that it is one of these three machines, its execution files will also be put in the privileged spool directory. You, therefore, have to validate that the machine has the privileged login `uucpz`.

MACHINE *Entry for "Other" Systems*

You may want to specify different option values for the machines your computer calls that are not mentioned in specific `MACHINE` entries. This may occur when there are many machines calling in, and the command set changes from time to time. The name `OTHER` for the machine name is used for this entry as follows:

```
MACHINE=OTHER \
COMMANDS=rmail:rnews:/usr/lbin/Photo:/usr/lbin/xp
```

All other options available for the `MACHINE` entry may also be set for the machines that are not mentioned in other `MACHINE` entries.

Combining MACHINE *and* LOGNAME *Entries*

It is possible to combine `MACHINE` and `LOGNAME` entries into a single entry where the common options are the same. For example, the two entries:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
READ=/ WRITE=/

LOGNAME=uucpz REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

share the same REQUEST, READ, and WRITE options. These two entries can be merged into one entry as follows:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
LOGNAME=uucpz SENDFILES=yes \
READ=/ WRITE=/
```

Sample Permissions *Files*

Example 1:

This first example represents the most restrictive access to your computer.

```
LOGNAME=nuucp
```

It states that login nuucp has all the default permissions/restrictions:

- The remote machine can only send files to uucppublic.
- The remote machine cannot request to receive files (REQUEST option).
- No files that are queued for the remote machine will be transferred during the current session (SENDFILES option).
- The only commands that can be executed are the defaults.

This entry alone is sufficient to start communications with remote machines, permitting files to be transferred only to the /usr/spool/uucppublic directory.

Example 2:

The next example is for remote machines that log in but have fewer restrictions. The login and password corresponding to this entry should not be distributed to the general public; it is usually reserved for closely coupled systems where the Systems file information can be tightly controlled.

```
LOGNAME=uucpz REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

This entry places the following permissions/restrictions on a machine that logs in as `uucpz`:

- Files can be requested from your computer (`REQUEST` option).
- Files can be transferred to any directory or any file that is writable by user "other," that is, a file/directory that is writable by a local user with neither owner nor group permissions (`WRITE` option).
- Any files readable by user "other" can be requested (`READ` option).
- Any requests queued for the remote machine will be executed during the current session. These are files destined for the machine that has called in (`SENDFILES` option).
- The commands sent for execution on the local machine must be in the default set.

Example 3:

The two previous examples show entries that refer to remote machines when they log in to your computer. This example is an entry used when calling remote machines:

```
MACHINE=eagle:owl:hawk:raven \
REQUEST=yes READ=/ WRITE=/
```

When calling any of the systems given in the `MACHINE` list, the following permissions prevail:

- The remote machine can both request and send files (`REQUEST` option).
- The source or destination of the files on the local machine can be anywhere in the file system (with read/write option).
- The only commands that will be executed for the remote machine are those in the default set.

Any site that is called that does not have its name in a `MACHINE` entry will have the default permissions as stated in Example 1, with the exception that files queued for that machine will be sent. (The `SENDFILES` option is only interpreted in the `LOGNAME` entry.)

`Poll` *File*

The `Poll` file (`/usr/lib/uucp/Poll`) contains information for polling specified machines. Each entry in the `Poll` file contains the name of the remote machine to call, followed by a Tab character, and finally the hours the machine should be called. The entry:

```
eagle   0 4 8 12 16 20
```

will provide polling of machine `eagle` every 4 hours.

Note – It should be understood that `uudemon.poll` does not actually perform the poll; it merely sets up a polling work (`C.`) file in the spool directory that will be seen by the scheduler, started by `uudemon.hour`. Refer to the "uudemon.poll" section later in this chapter.

`Maxuuxqts` *File*

The `Maxuuxqts` (`/usr/lib/uucp/Maxuuxqts`) file contains an ASCII number to limit the number of simultaneous `uuxqt` programs running. This file is delivered with a default entry of 2. This can be changed to meet local needs. If there is a lot of traffic from `mail`, it may be advisable to increase the number of `uuxqt` programs that will run to reduce the time it takes for the mail to leave your system. However, keep in mind that the load on the system increases with the number of `uuxqt` programs running.

`Maxuuscheds` *File*

The `Maxuuscheds` (`/usr/lib/uucp/Maxuuscheds`) file contains an ASCII number to limit the number of simultaneous `uusched` programs running. Each `uusched` running will have one `uucico` associated with it; limiting the number will directly affect the load on the system. The limit should be less than the number of outgoing lines used by UUCP (a smaller number is often desirable). This file is delivered with a default entry of 2. Again, this can be changed to meet the needs of the local system. However, keep in mind that the load on the system increases with the number of `uusched` programs running.

remote.unknown Program

usr/lib/uucp/remote.unknown, which is the remote.unknown program, is a shell file that is executed when a remote site that is not in the Systems file calls in to start a conversation. The shell script will append the name and time information to the file /usr/spool/uucp/.Admin/Foreign. Since it is a shell, it can be easily modified. For example, it can be set up to send mail to the administrator. The contents of this file, as delivered, is as follows:

```
FOREIGN=/usr/spool/uucp/.Admin/Foreign
echo "`date`: call from system $1" >>$FOREIGN
```

If you want to permit machines that are not listed in your Systems file to communicate via Basic Networking, remove the execute permissions from the remote.unknown file. For example:

```
# chmod 444 /usr/lib/uucp/remote.unknown
```

When remote.unknown is executable, your computer will hang up if a machine that is not in your Systems file calls in (to UUCP) on your system.

Administrative Tasks

There is a minimum amount of maintenance that must be done to your computer to keep the files updated, to ensure that the network is running properly, and to track down line problems. When more than one remote machine is involved, the job becomes more difficult because there are more files to update and because users are much less patient when failures occur between machines that are under local control. The uustat program provides you with information about the latest attempts to contact various machines and the age and number of jobs in the queue for remote machines. The following sections describe the routine administrative tasks that must be performed by someone acting as the UUCP administrator or are automatically performed by the UUCP daemons.

The biggest problem in a dialup network like UUCP is dealing with the backlog of jobs that cannot be transmitted to other machines. The following cleanup activities should be routinely performed.

Cleaning Up Undeliverable Jobs

The `uustat` program should be invoked regularly to provide information about the status of connections to various machines and the size and age of the queued requests. The `uudemon.admin` shell should be started by `cron` at least once each day. This will send the administrator the current status. Of particular interest is the age (in days) of the oldest request in each queue, the number of times a failure has occurred when attempting to reach that machine, and the reason for failure. In addition, the age of the oldest execution request (`X.` file) is also given.

The `uudemon.clean` shell file is set up to remove any jobs that have been queued for several days and cannot be sent. Leftover data (`D.`) and work (`C.`) files are removed after 7 days, and execute (`X.`) files are removed after 2 days. It also provides feedback to the user indicating when jobs are not being accomplished and when these jobs are being deleted.

Cleaning Up the Public Area

To keep the local file system from overflowing when files are sent to the public area, the `uudemon.clean` procedure is set up with a `find` command to remove any files that are older than 7 days and directories that are empty. This interval may need to be shortened by changing the `uudemon.clean` shell file if there is not sufficient space to devote to the public area.

Since the spool directory is very dynamic, it may grow large before transfers take place. Therefore, it is a good idea to reorganize its structure. The best way to do this on your computer is to use the `crontab` command to clean out the spool directory at a specified time.

First, specify the file in which you want to have the cleanup code, as follows:

```
# crontab clean.wk
```

The *clean.wk* file will contain the code for all files cleaned at a specified time (every Monday, for example), based on the time specified in the `crontab` file. You may already have entries in *clean.wk*, which means you will also have the cleanup time specified. See *crontab*(1) for additional information. If you want to specify a new cleanup time, first, make a new file with the `crontab` command as above. Then edit the `crontab` file to specify the time of cleanup. For example:

```
0 0 1 15 * 1
```

in the `crontab` file would indicate cleanup on the first and fifteenth of each month, as well as on every Monday. In the file you specified with the `crontab` command, enter the following code (# represents comment lines):

```
#   Clean up /usr/spool/uucp
#   Most cleanup is now done by uudemon.clean
#   so just copy out and back.
#
echo "UUCP SPOOL DIRECTORIES CLEANUP STARTED"
#
cd /usr/spool/uucp
mkdir ../nuucp
chown uucp ../nuucp
chgrp uucp ../nuucp
find . -print|cpio -pdml ../nuucp
cd ..
mv uucp ouucp
mv nuucp uucp
rm -rf ouucp
rm -f /usr/spool/locks/LCK*
#
# Note:
# Change the tty?? device to the
# device you are using for UUCP.
# For example change tty?? to tty01.
#
chown uucp /dev/tty??
chgrp uucp /dev/tty??
chmod 0644 /dev/tty??
chmod 0222 /dev/tty??
echo "UUCP SPOOL DIRECTORIES CLEANUP FINISHED"
```

Compacting Log Files

This version of Basic Networking has individual log files for each machine and each program. For example, machine `eagle` has a log file for `uucico` requests and a log file for `uuxqt` execution requests. The `uulog` program gives the user access to the information in these files by machine name. These files are combined and stored in directory `/usr/lib/uucp/.Old` whenever `uudemon.clean` is executed. This shell script saves files that are 2 days old. The 2 days can be easily changed by changing the appropriate line in the `uudemon.clean` shell.

If space is a problem, the administrator might consider reducing the number of days the files are kept.

Cleaning Up `sulog` *and* `cron/log`

The `/usr/adm/sulog` and `/usr/lib/cron/log` files are both indirectly related to UUCP transactions. The `sulog` file contains a history of the `su` command usage. Since each `uudemon` entry in the file `/usr/spool/cron/crontabs/root` uses the `su` command, the `sulog` could become rather large over a period of time. The `sulog` should be purged periodically to keep the file at a reasonable size.

Similarly, a history of all processes spawned by `/etc/cron` are recorded in `/usr/lib/cron/log`. The `cron/log` file will also become large over a period of time and should be purged periodically to limit its size.

UUCP and `cron`

The `cron` daemon is a tool that is very useful in the administration of UNIX Systems. When the computer is in run state 2 (multi-user), the `cron` daemon scans the `/usr/spool/cron/crontabs/root` file every minute for entries that contain "work" scheduled to be executed at that time. It is recommended that the UUCP administrator make use of `cron` to aid in the administration of Basic Networking.

As delivered, Basic Networking contains four entries in the `root` crontab file. Each one of these entries executes shell scripts that are used for various administrative purposes. These shell scripts can be easily modified to meet the needs of your system.

uudemon.admin

The uudemon.admin shell script mails status information to the UUCP administrative login (uucp) using uustat commands with the -p and -q options. Refer to *uustat*(1C) for descriptions of these options.

The uudemon.admin shell script should be executed daily by an entry in the root crontab file. The default root crontab entry for uudemon.admin is as follows:

```
48 11,14 ** 1-5 /bin/su uucp -c "/usr/lib/uucp/uudemon.admin" >
/dev/null 2>&1
```

uudemon.clean

The uudemon.clean shell script cleans up the Basic Networking log files and directories. Archived log files are updated so that no log information over 3 days old is kept. Log files for individual machines are taken from the /usr/spool/uucp/.Log directory, merged, and placed in the /usr/spool/uucp/.Old directory along with the older log information. Files and directories that are no longer needed in the spool directories are removed. After cleanup is performed, the UUCP administrative login (uucp) is mailed a summary of the status information gathered during the current day.

The uudemon.clean shell script should be executed by an entry in the root crontab file. It can be run daily, weekly, or whenever, depending on the amount of UUCP traffic that enters and leaves your computer.

The default root crontab entry for uudemon.clean is as follows:

```
45 23 * * * ulimit 5000; /bin/su uucp -c
"/usr/lib/uucp/uudemon.clean" > /dev/null 2>&1
```

If log files get very large, the ulimit may need to be increased.

uudemon.hour

The uudemon.hour shell script is used to call UUCP programs on an hourly basis. The uusched program is called to search the spool directory for work files (C.) that have not been processed and to schedule these files for transfer to a remote machine. The uuxqt daemon is called to search the spool directory for execute files (X/C.) that have been transferred to your computer and were not processed at the time they were transferred.

The uudemon.hour shell script should be executed by an entry in the root crontab file. If the amount of traffic leaving and entering your computer is large, it can be started once or twice an hour. If it is small, it can be started once every 4 hours or so. The default root crontab entry for uudemon.hour is as follows:

```
26,56 * * * * /bin/su uucp -c
"/usr/lib/uucp/uudemon.hour" > /dev/null 2>&1
```

uudemon.poll

The uudemon.poll shell script is used to poll the remote machines listed in the Poll file (/usr/lib/uucp/Poll). It creates work files (C.) for machines according to the entries listed in the Poll file. It should be set up to run once an hour just prior to uudemon.hour so that the work files will be present when uudemon.hour is called.

The uudemon.poll script should be executed by an entry in the root crontab file. The exact times it runs is dependent on the scheduling of uudemon.hour. The default root crontab entry for uudemon.poll is as follows:

```
40 * * * * /bin/su uucp -c "/usr/lib/uucp/uudemon.poll" >
/dev/null 2>&1
```

Notice how uudemon.poll is scheduled to run 14 minutes before and after uudemon.hour runs.

UUCP Logins and Passwords

There are two login IDs associated with Basic Networking: one is the UUCP and the other is an access login, nuucp, used by remote computers to access your computer. These logins should not be changed from their default settings of uucp and nuucp.

The uucp administrative login is the owner of all the UUCP object and spooled data files. The following is a sample entry in the /etc/passwd file for the administrative login:

```
uucp:x:5:1:UUCP.Admin:/usr/lib/uucp:
```

The nuucp access login allows remote machines to log in on your computer. The following is a sample entry in the /etc/passwd file for the access login:

```
nuucp:x:6:1:UUCP.Admin:/usr/spool/uucppublic:
/usr/lib/uucp/uucico
```

Notice that the standard shell is not given to the nuucp login. The shell that nuucp receives is the uucico daemon that controls the conversation when a remote machine logs in to your machine.

The assigning of passwords for the uucp and nuucp logins is left up to the administrator. The passwords should be at least six to eight characters. Only the first eight characters of the passwords are significant. If the password for the access login is changed for security reasons, make certain that the remote machines that are a part of your network are properly notified of the change.

Electronic Mail 9

Electronic mail is a form of communication used on computers to send messages to and receive messages from other users. You can exchange mail locally with other users on your computer, with users on other computers within your company over a network, or with users almost anywhere in the world, if your computer is tied in to one of the international networks.

Electronic mail has two major components:

The user interface
: The user interface is the part of the mail system with which the user interacts to send and receive mail. It picks up the mail in a known location and transfers it to the user's mailbox.

The mail transport
: The mail transport is that part of the mail system which puts the mail received in a known location and sends the mail given by the user interface program to the desired destination.

Users interact directly with the user interface program and in most cases are not even aware of the existence and function of the mail transport program.

The User Interface

Two different user interfaces for the mail system are available with the INTERACTIVE UNIX Operating System. One is a simple "character command" interface called `mailx`; the other is a "menu driven" interface called the TEN/PLUS® Mail System.

The Mail Transport Program (sendmail)

The mail transport program used in the INTERACTIVE UNIX System is `sendmail`. For more information regarding `sendmail`, see the *INTERACTIVE UNIX System User's Guide.*

Installation

Before mail can be exchanged, the following tasks must be performed:

- Establish a physical connection between the various computers. These connections can be established using modems or direct connections.
- Assign system names to your computers.
- Assign a password to the `nuucp` login on each computer for other computers to use when sending mail. This will ensure that only friendly computers can log in to your system (when using `uucp`).
- Set up the `uucp` program, if you plan to use it, or install and set up TCP/IP, if you plan to use that transport protocol with `sendmail`.
- Provide your system name, login and password, data telephone number, and communication attributes to the system administrators of computers with which you will be exchanging mail.
- Obtain the system names, logins and passwords, telephone numbers, and attributes for their computers from the other system administrators.

When you install the Basic Networking optional subset on versions 2.2 and later of the INTERACTIVE UNIX Operating System, the `sendmail` mail transport program and the character-based user interface, `mailx`, are installed. You should then configure the `sendmail` mail transport using the `sysadm mailmgmt` command. Refer to *mailx*(1) for more information about using that user interface to mail.

To use the TEN/PLUS Mail System as the user interface for electronic mail, you must install the TEN/PLUS Environment subset.

The Berkeley Line Printer Spooler (lpr) System 10

This chapter describes the structure and installation procedure for the line printer spooling system. The Berkeley `lpr` system is different from the System V `lp` printer in several ways. The `lpr` system supports a network environment. It is recommended for use with the INTERACTIVE UNIX System and is also used in the Solaris™ computing environment. The `lp` printer is a System V utility and comes with filters for several printers. Refer to Appendix A, "Setting Up Your Printer," for more information about the `lp` printer.

The line printer system supports:

- Multiple printers and spooling queues
- Both local and remote printers
- Printers attached via serial lines that require line initialization, such as the baud rate

Raster output devices, such as a Varian or Versatec®, and laser printers, such as an HP®LaserJet®, are also supported by the line printer system.

This chapter is reprinted (with format and editorial changes) from the University of California, Berkeley document entitled *4.3BSD Line Printer Spooler Manual*, by Ralph Campbell.

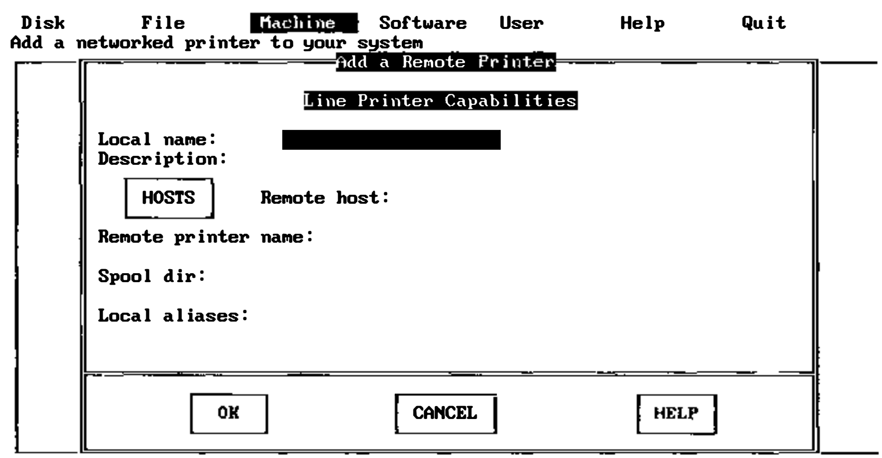
Configuring a Remote Printer

If your system has INTERACTIVE TCP/IP installed and there are printers already configured elsewhere on your network, you may configure your INTERACTIVE UNIX System to recognize and use those printers.

Adding a Printer

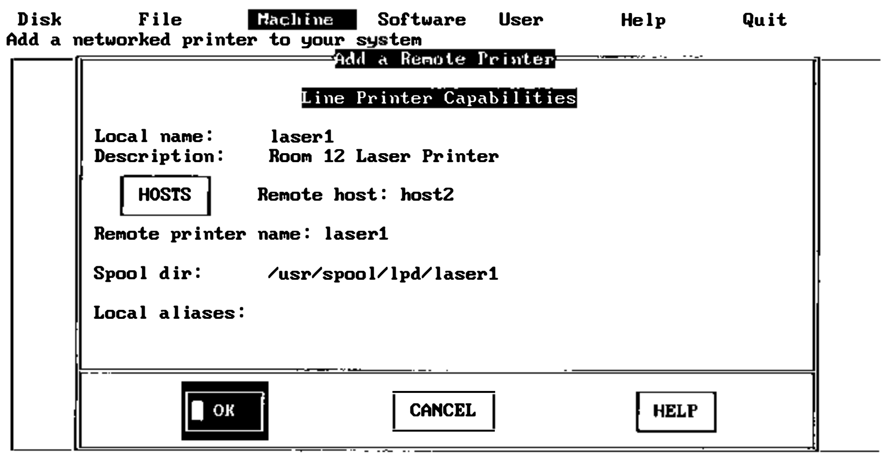
To add a printer to the system:

1. Start `sysadm`. Select the `lpr Printer Management` option under `Machine`. Then select the `Add a Printer` submenu. Your screen will look similar to this:



2. Enter the name that the printer will be called on the current host and a description of it. You can select `HOSTS` to choose a host from `/etc/hosts`, or you can type the remote host name in the `Remote host:` field. Enter the remote printer name if different from the local printer name, the spool directory (where the printer jobs should be stored), and any local aliases for

the printer. Press Enter to accept any default values. Press Escape to move to the `OK` button from the `Local aliases:` field. Sample answers are provided in the following screen. Your screen will look similar to this:

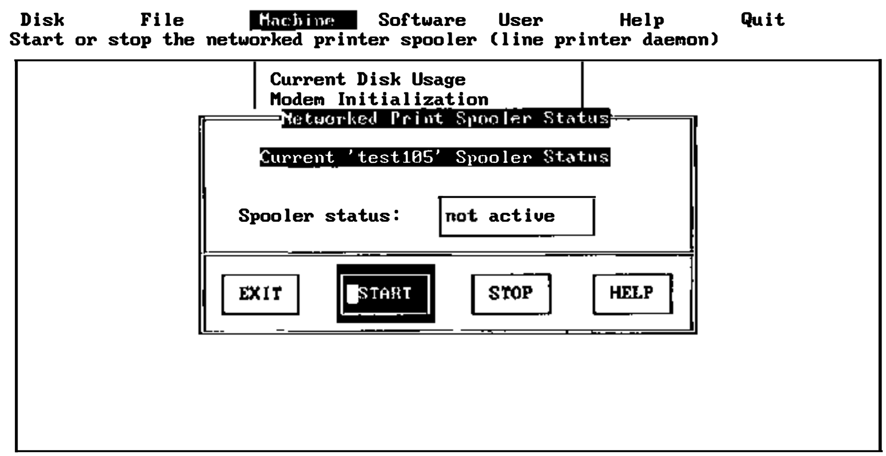


3. Select `OK` when you have finished entering the printer information.

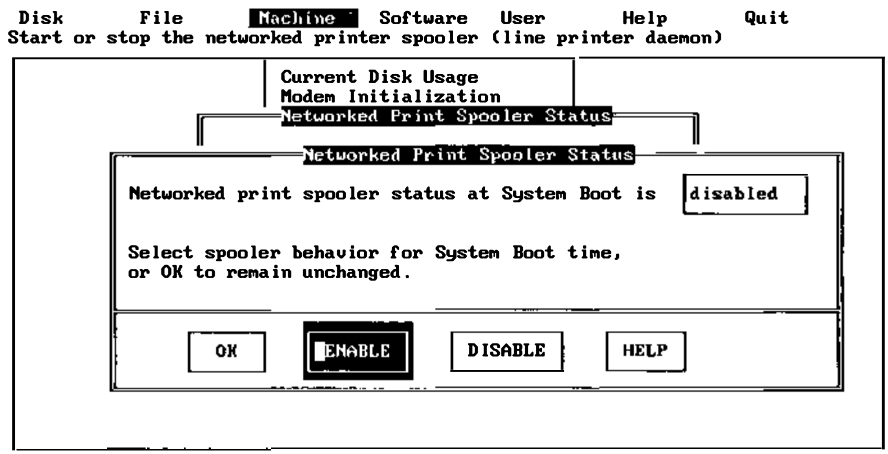
Starting the Printer and Enabling Automatic Startup

To start the printer and enable the print system to start when the system is booted, follow this procedure:

1. Once a remote printer has been added, you can start the printer system. Select `Spooler Management` from the `lpr Printer Management` menu under `Machine`. Your screen will look similar to this:



2. Select `START` to enable the line printer system. Your screen will look similar to this:

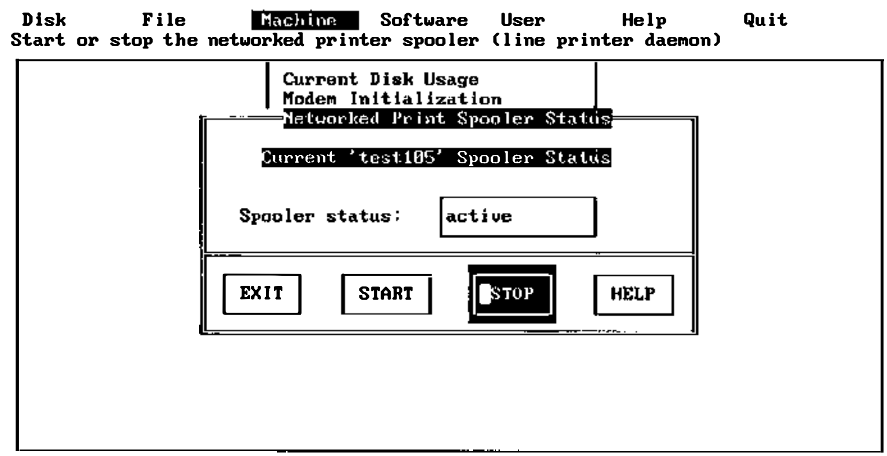


3. Select `ENABLE` to enable the line printer system to start automatically when the system is booted. Then select `EXIT` from the `Networked Print Spooler Status` menu.

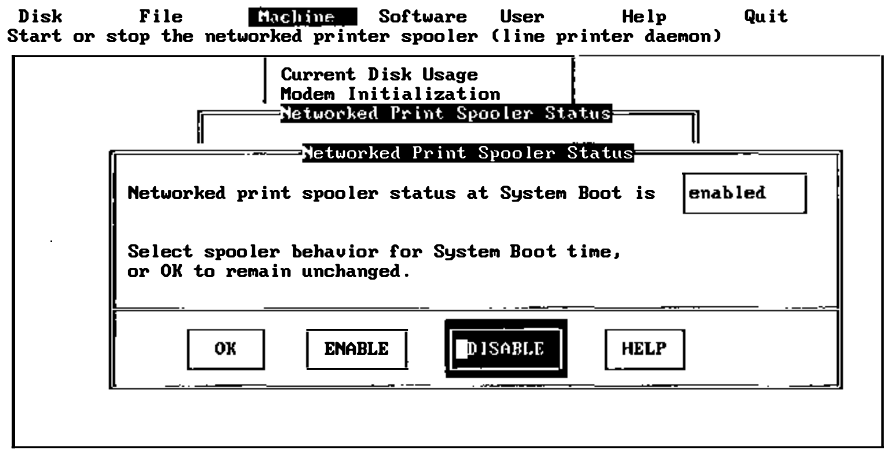
Stopping the Printer and Disabling Automatic Startup

To stop the printer and stop the print system from starting when the system is booted, do the following:

1. Select `Spooler Management` from the `lpr Printer Management` menu under `Machine`. Your screen will look similar to this:



2. Select STOP. Your screen will look similar to this:

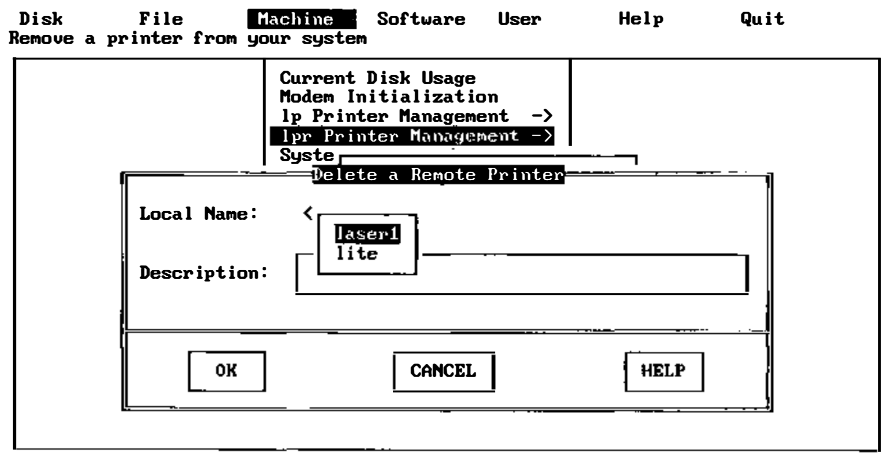


3. If you want to disable the line printer system from starting automatically when the system is booted, choose DISABLE. Otherwise, you can choose OK to leave the printer startup status as is. Then select EXIT from the Networked Print Spooler Status menu.

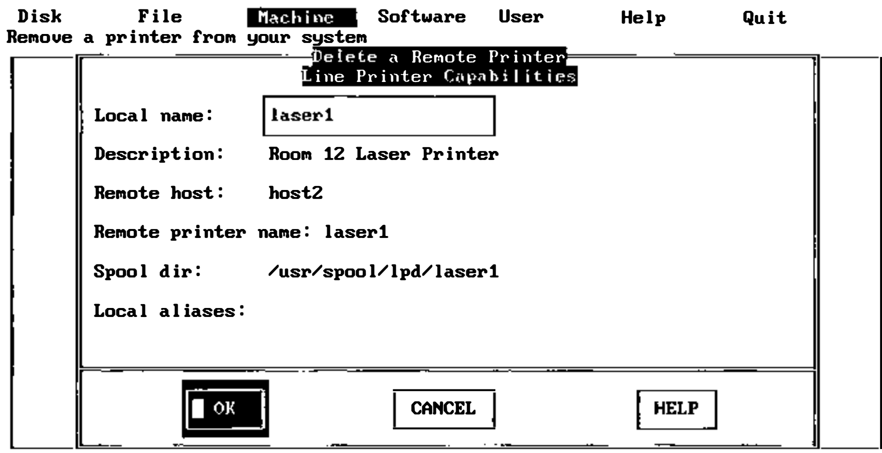
Deleting a Printer

To remove a printer from the system:

1. Select `Delete a Printer` from the `lpr Printer Management` menu under `Machine`. Your screen will look similar to this:



2. Press the spacebar and use the up and down arrow keys to select the printer to delete. Press Enter, then select `OK`. Your screen will look similar to this:

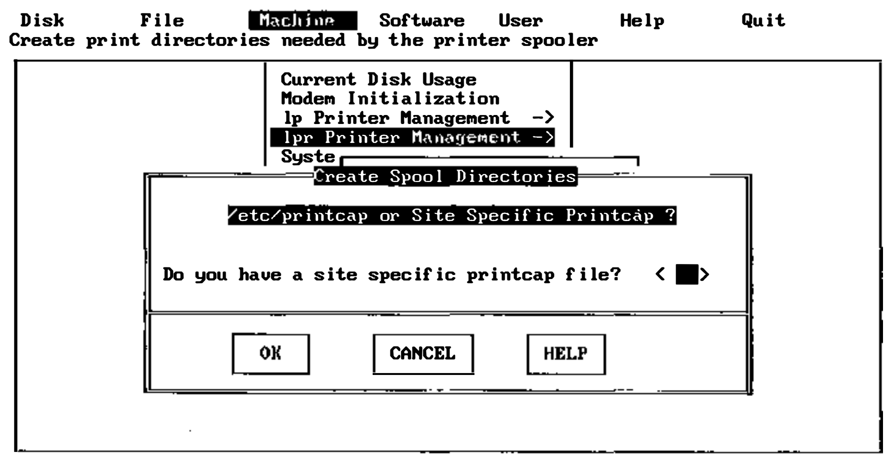


3. Select `OK` to delete the printer or `CANCEL` to cancel deleting the printer.

Creating Spool Directories

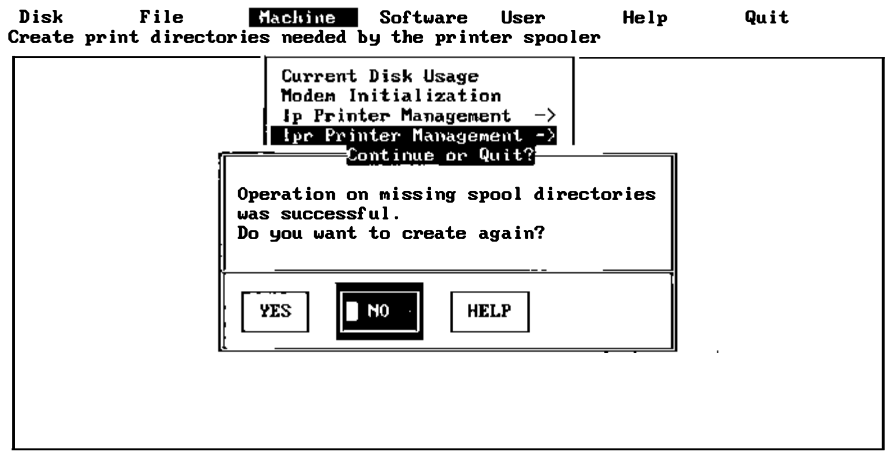
Spool directories are made automatically if you use the `sysadm` procedure described in "Adding a Printer." However, if you have added entries to the `printcap` file with a text editor, or a spool directory has been inadvertently deleted, use the following procedure to create a new spool directory:

1. Select `Create Spool Directories` from the `lpr Printer Management` menu under `Machine`. Your screen will look similar to this:



2. Press the spacebar to toggle between `yes` and `no`. Choose `no` if you are using the standard `printcap` database file, `/etc/printcap`. Choose `yes` if your `printcap` file is in a different directory, or has a different name. After

choosing yes or no, select OK. If you choose yes, you must type the full path name of the printcap file. If you choose no, your screen will look similar to this:



3. Select NO to return to the lpr Printer Management menu.

Commands

The line printer system consists mainly of the following files and commands:

/etc/printcap	Printer configuration and capability database
/usr/lib/lpd	Line printer daemon which does all the real work
/usr/ucb/lpr	Program to enter a job in a printer queue
/usr/ucb/lpq	Spooling queue examination program
/usr/ucb/lprm	Program to delete jobs from a queue
/etc/lpc	Program to administer printers and spooling queues
/dev/printer	Streams node on which lpd listens

The file `/etc/printcap` is a master database which describes line printers that are directly attached to a machine as well as printers that are accessible across a network. The manual entry *printcap*(5) provides the authoritative definition of the format of this database and specifies default values, such as the directory in which spooling is performed. This document introduces some of the information that can be placed in `printcap`.

`lpd`—*Line Printer Daemon*

The program *lpd*(1M), usually invoked at boot time from the `/etc/rc3.d/s11lpd` file, acts as a master server for coordinating and controlling the spooling queues configured in the `printcap` file. When `lpd` is started, it makes a single pass through the `printcap` database, restarting any printers that have jobs. In normal operation `lpd` listens for service requests on multiple sockets, one in the local domain (named `/dev/printer`) for local requests and one in the Internet domain (under the "printer" service specification) for requests for printer access from off machine; see *socket*(7) and *services*(5) for more information on sockets and service specifications, respectively. `lpd` spawns a copy of itself to process the request; the master daemon continues to listen for new requests.

Clients communicate with `lpd` using a simple transaction-oriented protocol, as specified in RFC1179. Authentication of remote clients is done based on the "privilege port" scheme employed by *rshd*(1M) and *rcmd*(3I). The following table shows the requests understood by `lpd`. In each request the first byte indicates the "meaning" of the request, followed by the name of the printer to which it should be applied. Additional qualifiers may follow, depending on the request.

Request	**Interpretation**
^Aprinter\n	Check the queue for jobs and print any found
^Bprinter\n	Receive and queue a job from another machine
^Cprinter [users . . .][jobs . . .]\n	Return a short list of current queue state
^Dprinter [users . . .][jobs . . .]\n	Return a long list of current queue state
^Eprinter person [users . . .][jobs . . .]\n	Remove jobs from a queue

`lpr`—*Enter Jobs in a Queue*

The *lpr*(1) command is used by users to enter a print job in a local queue and to notify the local `lpd` that there are new jobs in the spooling area. `lpd` either schedules the job to be printed locally, or if printing remotely, attempts to forward the job to the appropriate machine. If the printer cannot be opened or the destination machine is unreachable, the job will remain queued until it is possible to complete the work.

`lpq`—*Show Line Printer Queue*

The *lpq*(1) program works recursively backwards, displaying the queue of the machine with the printer and then the queue(s) of the machine(s) that lead to it. `lpq` has two forms of output: in the default (short) format it gives a single line of output per queued job; in the long format it shows the list of files (and their sizes) that comprise a job.

`lprm`—*Remove Jobs From a Queue*

The *lprm*(1) command deletes jobs from a spooling queue. If necessary, `lprm` will first kill off a running daemon that is servicing the queue and restart it after the required files are removed. When removing jobs destined for a remote printer, `lprm` acts similarly to `lpq` except it first checks locally for jobs to remove and then tries to remove files in queues off-machine.

`lpc`—*Line Printer Control Program*

The *lpc*(1M) program is used by the system administrator to control the operation of the line printer system. For each line printer configured in `/etc/printcap`, `lpc` may be used to:

- Disable or enable a printer
- Disable or enable a printer's spooling queue
- Rearrange the order of jobs in a spooling queue
- Find the status of printers and their associated spooling queues and printer daemons

Access Control

The printer system maintains protected spooling areas so that users cannot circumvent printer accounting or remove files other than their own. The strategy used to maintain protected spooling areas is as follows:

- The spooling area is writable only by a `daemon` user and `daemon` group.
- The `lpr` program runs set-user-ID to `root` and set-group-ID to group `daemon`. The `root` access permits reading any file required. Accessibility is verified with an *access*(2) call. The group ID is used in setting up proper ownership of files in the spooling area for `lprm`.
- Control files in a spooling area are made with `daemon` ownership and group ownership `daemon`. Their mode is 0660. This ensures control files are not modified by a user and that no user can remove files except through `lprm`.
- The spooling programs, `lpd`, `lpq`, and `lprm` run set-user-ID to `root` and set-group-ID to group `daemon` to access spool files and printers.
- The printer server, `lpd`, uses the same verification procedures as *rshd*(1M) in authenticating remote clients. The host on which a client resides must be present in the `/etc/hosts.equiv` or `/etc/hosts.lpd` file, and the request message must come from a reserved port number.

In practice, `lpd`, `lpq`, or `lprm` would not have to run as user `root` if remote spooling were not supported. In previous versions of the printer system, `lpd` ran set-user-ID to `daemon` and set-group-ID to group `spooling`, and `lpq` and `lprm` ran set-group-ID to group `spooling`.

Setting Up

The Berkeley line printer system is a separate subset that should be installed using `sysadm installpkg`. The real work in setting up is to create the `printcap` file and the appropriate filters for the printers you intend to use.

Creating a `printcap` File

The `printcap` database contains one or more entries per printer. A printer should have a separate spooling directory; otherwise, jobs will be printed on different printers depending on which printer daemon starts first. This section describes how to create basic printer entries.

Printers on Serial Lines

When a printer is connected via a serial communication line, it must have the proper baud rate and terminal modes set. The following example is for a simple line printer connected locally via a 1200 baud serial line:

```
lp|Line Printer at 1200 baud:\
        :lp=/dev/tty01:tt=B1200 CS8OPOST ONLCR:\
        :tr=\f:of=/usr/lib/lpf:lf=/usr/adm/lpd-errs:
```

The `lp` entry specifies the file name to open for output. If not specified, `/dev/lp` is the default. The `tt` entry specifies that the baud rate is 1200 using 8-bit data and that newline characters should be mapped to the carriage return, line feed pair (see *termio*(7)). The `tr` entry indicates that a form-feed should be printed when the queue empties so the paper can be torn off without turning the printer off-line and pressing form-feed. The `of` entry specifies that the filter program `lpf` should be used for printing the files; more will be said about filters later. The last entry causes errors to be written to the file `/usr/adm/lpd-errs` instead of to the console. Most errors from `lpd` are logged using *syslogd*(1M) and will not be logged in the specified file. The filters should use `syslogd` to report errors; only those that write to standard error output will end up with errors in the `lf` file. (Occasionally errors sent to standard error output have not appeared in the log file; the use of `syslogd` is highly recommended.)

Remote Printers

Printers that reside on remote hosts should have an empty `lp` entry. For example, the following `printcap` entry would send output to the printer named `lp` on the machine `printhost`:

```
lp|default line printer:\
        :lp=:rm=printhost:rp=lp:sd=/usr/spool/lpd/lp:
```

The `rm` entry is the name of the remote machine to connect to; this name must be a known host name for a machine on the network. The `rp` capability indicates that the name of the printer on the remote machine is `lp`; here it could be left out since this is the default value. The `sd` entry specifies `/usr/spool/lpd/lp` as the spooling directory instead of the default value `/usr/spool/lpd`.

Output Filters

Filters are used to handle device dependencies and to do accounting functions. The output filtering of `of` is used when accounting is not being done or when all text data must be passed through a filter. It is not intended to do accounting since it is started only once, all text files are filtered through it, and no provision is made for passing owners' login names, identifying the beginning and ending of jobs, and so forth. The other filters (if specified) are started for each file printed and do accounting if there is an `af` entry. If entries for both `of` and other filters are specified, the output filter is used only to print the banner page; it is then stopped to allow other filters access to the printer. (Refer to "Output Filter Specifications" for information on how this is done.) An example of a printer that requires output filters is the Benson-Varian:

```
va|varian|Benson-Varian:\
        :lp=/dev/va0:sd=/usr/spool/lpd/vad:of=/usr/lib/vpf:\
        :tf=/usr/lib/rvcat:mx#2000:pl#58:px=2112:py=1700:tr=\f:
```

The `tf` entry specifies `/usr/lib/rvcat` as the filter to be used in printing *troff*(1) output. This filter is needed to set the device into print mode for text and into plot mode for printing `troff` files and raster images. Note that the page length is set to 58 lines by the `pl` entry for 8 1/2 x 11-inch fanfold paper. To enable accounting, the Varian entry would be augmented with an `af` file and an `if` filter as shown below:

```
va|varian|Benson-Varian:\
        :lp=/dev/va0:sd=/usr/spool/lpd/vad:of=/usr/lib/vpf:\
        :if=/usr/lib/vpf:tf=/usr/lib/rvcat:af=/usr/adm/vaacct:\
        :mx#2000:pl#58:px=2112:py=1700:tr=\f:
```

Access Control

Local access to printer queues is controlled with the `rg` `printcap` entry:

```
:rg=lprgroup:
```

Users must be in the group `lprgroup` to submit jobs to the specified printer. The default is to allow all users access. Note that once the files are in the local queue, they may be printed locally or forwarded to another host depending on the configuration.

Remote access is controlled by listing the hosts in either the file `/etc/hosts.equiv` or `/etc/hosts.lpd`, one host per line. Note that *rsh*(1C) and *rlogin*(1C) use the file `/etc/hosts.equiv` to determine which hosts are equivalent for allowing logins without passwords. The `/etc/hosts.lpd` file is only used to control which hosts have line printer access. Remote access can be further restricted to only allow remote users with accounts on the local host to print jobs by using the `rs` `printcap` entry:

```
:rs:
```

Output Filter Specifications

The filters supplied handle printing and accounting for most simple line printers. For other devices or accounting methods, it may be necessary to create a new filter.

Filters are spawned by `lpd` with the data to be printed as their standard input and with the printer as the standard output. The standard error is attached to the `lf` file for logging errors, or `syslogd` may be used for logging errors. A filter must return a 0 exit code if there are no errors, 1 if the job should be reprinted, and 2 if the job should be thrown away. When `lprm` sends a kill signal to the `lpd` process controlling printing, it sends a `SIGINT` signal to all filters and descendents of filters. This signal can be trapped by filters that need to do cleanup operations, such as deleting temporary files.

Arguments passed to a filter depend on the type of filter. The `of` filter is called with the following arguments:

```
filter -wwidth -llength
```

The *width* and *length* values come from the `pw` and `pl` entries in the `printcap` database. The `if` filter is passed the following parameters:

```
filter [ -c ] -wwidth -llength -iindent -n login -h host accounting_file
```

The `-c` flag is optional and only supplied when control characters are to be passed uninterpreted to the printer (when using the `-l` option of `lpr` to print the file). The `-w` and `-l` parameters are the same as for the `of` filter. The `-n` and `-h` parameters specify the login name and host name of the job owner. The last argument is the name of the accounting file from `printcap`.

All other filters are called with the following arguments:

```
filter -xwidth -ylength -n login -h host accounting_file
```

The `-x` and `-y` options specify the horizontal and vertical page size in pixels (from the `px` and `py` entries in the `printcap` file). The rest of the arguments are the same as for the `if` filter.

Additionally, when `lpd` needs to stop the `of` filter in order to use another filter (for example, the `if` filter), it will write the two-character string Control-y Control-a (octal `\031 \001`) to the `of` filter's input. It is the responsibility of the `of` filter to suspend its own execution upon receiving this string, using the C language construct `kill (getpid(), SIGSTOP);`. `lpd` will resume execution of the `of` filter when it is necessary.

Line Printer Administration

The `lpc` program provides local control over line printer activity. The major commands and their intended uses are described here. The command format and remaining commands are described in *lpc*(1M).

`abort` and `start`

`abort` terminates an active spooling daemon on the local host immediately and then disables printing (preventing new daemons from being started by `lpr`). This is normally used to forcibly restart a hung line printer daemon, that is, `lpq` reports that there is a daemon present but nothing is happening. It does not remove any jobs from the queue; use the `lprm` command instead. `start` enables printing and requests `lpd` to start printing jobs.

`enable` and `disable`

`enable` and `disable` allow spooling in the local queue to be turned on/off. This allows/prevents `lpr` from putting new jobs in the spool queue. It is frequently convenient to turn spooling off while testing new line printer filters since the `root` user can still use `lpr` to put jobs in the queue but no one else can. The other main use is to prevent users from putting jobs in the queue when the printer is expected to be unavailable for a long time.

`restart`

`restart` allows ordinary users to restart printer daemons when `lpq` reports that there is no daemon present.

`stop`

`stop` halts a spooling daemon after the current job completes; this also disables printing. This is a clean way to shut down a printer to do maintenance, and so forth. Note that users can still enter jobs in a spool queue while a printer is stopped.

`topq`

`topq` moves jobs to the top of a printer queue. This can be used to reorder high priority jobs since `lpr` only provides first-come-first-serve ordering of jobs.

Diagnostics

There are several messages that may be generated by the line printer system. This section categorizes the most common and explains the causes for their generation. Where the message implies a failure, directions are given to remedy the problem.

In the examples below, the name *printer* is the name of the printer from the `printcap` database.

lpr *Messages*

lpr: *printer* : unknown printer
The *printer* was not found in the printcap database. Usually this is a typing mistake; however, it may indicate a missing or incorrect entry in the file /etc/printcap.

lpr: *printer* : jobs queued, but cannot start daemon
The connection to lpd on the local machine failed. This usually means that the printer server started at boot time has died or is hung. Check the local Streams node /dev/printer to be sure it still exists (if it does not exist, there is no lpd process running). Usually it is enough to have the superuser type the following to restart lpd:

```
# /usr/lib/lpd
```

You can also check the state of the master printer daemon with the following:

```
$ ps -p `cat /usr/spool/lpd.lock`
```

Another possibility is that the lpr program is not set-user-ID to root and set-group-ID to group daemon. This can be checked with:

```
$ ls -l /usr/ucb/lpr
```

lpr: *printer* : printer queue is disabled
This means the queue was turned off with:

```
# lpc disable printer
```

to prevent lpr from putting files in the queue. This is normally done by the system administrator when a printer is going to be down for a long time. The printer can be turned back on by the superuser with lpc.

`lpq` *Messages*

`waiting for` *printer* `to become ready (offline?)`

The printer device could not be opened by the daemon. This can happen for several reasons, the most common being that the printer is turned off-line. This message is also be generated if the printer is out of paper, the paper is jammed, and so forth. The actual reason is dependent on the meaning of the error codes returned by the system device driver. Not all printers supply enough information to distinguish when a printer is off-line or having trouble, for example, a printer is connected through a serial line. Another possible cause of this message is that some other process, such as an output filter, has an exclusive open on the device. In this instance, your only recourse is to kill the offending program(s) and restart the printer with `lpc`.

printer `is ready and printing`

The `lpq` program checks to see if a daemon process exists for *printer* and prints the file *status* located in the spooling directory. If the daemon is hung, the superuser can use `lpc` to abort the current daemon and start a new one.

`waiting for` *host* `to come up`

This implies there is a daemon trying to connect to the remote machine named *host* to send the files in the local queue. If the remote machine is up, `lpd` on the remote machine is probably dead or hung and should be restarted as mentioned for `lpr`.

`sending to` *host*

The files should be in the process of being transferred to the remote *host*. If not, the local daemon should be aborted and started with `lpc`.

`Warning:` *printer* `is down`

The printer has been marked as being unavailable with `lpc`.

`Warning: no daemon present`

The `lpd` process overseeing the spooling queue, as specified in the "lock" file in that directory, does not exist. This normally occurs only when the daemon has unexpectedly died. The error log file for the printer and the `syslogd` logs should be checked for a diagnostic from the deceased process. To restart an `lpd`, use:

```
$ lpc restart printer
```

`no space on remote; waiting for queue to drain`
This implies that there is insufficient disk space on the remote. If the file is large enough, there will never be enough space on the remote (even after the queue on the remote is empty). The solution is to move the spooling queue or make more free space on the remote.

`lprm` *Messages*

`lprm:` *printer* `: cannot restart printer daemon`
This case is the same as when `lpr` prints that the daemon cannot be started.

`lpd` *Messages*

The `lpd` program can log many different messages using *syslogd*(1M). Most of these messages are about files that cannot be opened and usually imply that the `printcap` file or the protection modes of the files are incorrect. Files may also be inaccessible if users manually manipulate the line printer system, that is, they bypass the `lpr` program.

In addition to messages generated by `lpd`, any of the filters that `lpd` spawns may log messages using `syslogd` or log them to the error log file (the file specified in the `lf` entry in `printcap`).

`lpc` *Messages*

`couldn't start printer`
This case is the same as when `lpr` reports that the daemon cannot be started.

`cannot examine spool directory`
Error messages beginning with "cannot ..." are usually due to the incorrect ownership or protection mode of the lock file, spooling directory, or the `lpc` program.

Using a Serial Terminal as the Console 11

The console is the part of the computer configuration that the system uses to display error messages. It is also needed by the system administrator to operate the system when it has to be rebooted. By default, the console on a PC UNIX System is the keyboard and the monitor attached to the computer's video card, also called the graphics controller.

The INTERACTIVE UNIX System is used in many different markets, and some require that a serial terminal (for example, connected via a modem) be used as the console instead. Some environments mandate that there be no console at all (black box). With the INTERACTIVE UNIX System, a different console can be selected using the `chgcon` utility. This chapter describes the various choices.

Selecting a Console

When the `chgcon` utility is invoked, it displays the following choices:

1. Graphics Display Console
2. Serial Console
3. No Console
4. Auto Detect Mode

`Graphics Display Console` *Mode*

If you select the `Graphics Display Console` mode, the bootstrap routine stored on your system's primary fixed disk will check for the presence of a video adapter in your machine the next time the system is rebooted. If one is found, booting continues and the display connected to the video adapter will be the system console. If no video adapter is found, the boot process will terminate.

`Serial Console` *Mode*

If you select `Serial Console` mode, a terminal connected to any serial port (for example, `COM1:`, `COM2:`) can become the system console. The `chgcon` program will ask for the port address (for example, 0x3F for `COM1:`), the baud rate, parity, word length, and the number of stop bits.

`chgcon` also checks the `asy` configuration in the `/etc/conf/sdevice.d` directory for the selected port address. The `Serial Console` mode only works for devices that are supported by the `asy` driver (by default, `COM1:`). This driver supports the standard serial ports of the computer, as well as dumb multi-port cards.

If the selected port address is not configured, a new kernel has to be configured, built, and installed for the new specified port. After creating a new kernel, run the `/etc/chgcon` command again and select the new serial console. The serial terminal connected to that port will now be the console when the system is rebooted.

If the selected port base address is already configured in the `/etc/conf/sdevice.d/asy` file (which is checked for and reported by the `chgcon` program), no kernel configuration or creation is necessary. The system can simply be rebooted.

Immediately after a reboot, the bootstrap routine stored on the system's primary fixed disk checks for the existence of the selected `asy` port base address in your machine. If it is found, the booting process continues and the serial terminal connected to the selected `asy` port becomes the system console. If the specified port address does not exist, the boot process will terminate.

`No Console` *Mode*

If you select `No Console`, the bootstrap stored on the system's primary fixed disk selects the `No Console` mode immediately after a reboot. In this mode none of the boot or kernel messages will appear, since there is no actual physical device that is selected as a console. Before selecting this mode, make sure that at least a virtual terminal, or a serial terminal attached to your system, has a `getty` process running on it. Otherwise, no one will be able to log in to the system.

`Auto Detect` *Mode*

For you to select `Auto Detect` mode, a serial port has to be selected in the same manner as the `Serial Console` mode. In this mode, however, the primary bootstrap routine will first check for the availability of a video adapter in your system. If one is found, the display attached to this video adapter becomes the system console. If no video adapter is found, the bootstrap routine will check your system for the selected serial port base address. If this port is found, the terminal attached to this port becomes the console. The bootstrap routine then sets the port to the previously selected serial port attributes (that is, word length, parity, and baud rate). Therefore, ensure that the terminal attached to this port is correctly set to the previously selected serial port attributes. If the specified port does not exist, the bootstrap routine selects the `No Console` mode and boots your system.

Note that the bootstrap routine on the *Boot/Install* diskette is set to `Auto Detect` mode. The bootstrap routine is also set for serial port base address 0x3F8, 8-bit word length, 1 stop bit, no parity, and 9600 baud rate. To install from this port, you are required to have a VT100 terminal or a terminal emulating a VT100 attached to this port.

Recovering a Nonbootable Kernel After Selecting a New Console

If the kernel does not boot after using `chgcon` to select a new console and after rebooting the system, it probably means that a nonexistent serial port address was selected. Follow these steps to recover:

1. Boot your system using the *Boot/Install* diskette. When the message

```
Booting the INTERACTIVE UNIX Operating System...
```

is displayed, quickly press the spacebar and enter maintenance mode by typing `/maint`.

2. Mount the fixed disk using the following command:

```
# mount /dev/dsk/0s1 /mnt
```

3. Change the root and select a shell using the following command:

```
# /mnt/etc/chroot /mnt bin/sh
```

4. Now run the `chgcon` program to select a new console.
5. After selecting a new console, leave the `chroot` environment by typing `exit`.
6. Unmount the fixed disk using the following command:

```
# umount /mnt
```

7. Remove the *Boot/Install* diskette and reboot the system.

Verifying the Selected Console

A special utility, `constype`, is supplied to verify the identity of the current console. When used without arguments, the program prints the current console information to standard output. The exit code of the program also indicates the type of console. This can be useful when this program is invoked from within a program. The exit codes are as follows:

-1	Error
0	Graphics Console
1	No Console
2	Serial Console

This utility also supports the `-d` flag, which can be used to specify the name of a special file that is a candidate for the current console. See *constype*(1M) for more information.

Setting Up an International Environment 12

This chapter explains how to prepare a properly functioning international environment on an INTERACTIVE UNIX Operating System. It also summarizes the internationalization features and provides tips for C programmers who want to develop internationalized applications. Refer to Appendix D, "Advanced Internationalization Features," for information about advanced programming topics for international environments.

Note – Before reading this chapter, you should have already installed the International Supplement optional subset and read the chapter on internationalization in the *INTERACTIVE UNIX System User's Guide*.

Setting Up the Environment for Users' Terminals

This section describes how a system administrator can configure the terminals on the system to use the appropriate codesets and the keyboards supported by those terminals. It also explains the need for character mapping ability and give tips for establishing the correct mapping from boot time.

Motivation

The original UNIX Operating System and most systems derived from it have been based on the ASCII 7-bit coded character set and American English. The ASCII character set consists of 128 different characters, each represented by a single byte (the eighth bit is not used). The INTERACTIVE UNIX System supports characters represented as a byte with the eighth bit set as well. This means that 256 characters can be supported at the same time. A consistent coding convention needs to be applied, however. In the IBM PC world, an 8-bit coding scheme referred to as IBM extended ASCII has been used for several years. This codeset is currently referred to as IBM codepage 437. In heterogeneous UNIX System environments a different codeset, called ISO 8859-1, is used. Both of these codesets are supersets of ASCII.

Although an 8-bit system meets most of the European requirements (for the major Asian Languages, a 16-bit system is necessary even to support a single language), it should function properly in conjunction with the available hardware and, in particular, with the terminals. To use characters from the French, German, Finnish, and other alphabets, several terminals are available that generate 7-bit codes but display the characters from those alphabets on the screen instead of the ones found on a U.S. terminal. Their keyboards have the same number of keys, but different characters are pictured on the keycaps. Others, like the DEC VT220™, support 256 characters at a time but use their own proprietary codeset and have an extra Compose key.

To illustrate the problems that occur when trying to use such terminals in a mixed language environment, imagine an INTERACTIVE UNIX System with a console and a French 7-bit terminal connected to the serial port. When editing a file on the terminal and using the French character é in text, the terminal (hardware) actually generates the ASCII code 123, which is the code normally used for the left curly brace (`{`). (This example assumes that the terminal uses the French national variant of ASCII called ISO 646f.) If the file that was edited is looked at on the console, the letter actually appears to be a curly brace. Therefore input and output mapping should be supported by the `tty` subsystem to allow consistent use of one single codeset throughout the system.

Implementing character mapping support inside the `tty` subsystem has the advantage that its features are automatically supported by all peripherals that use the standard line discipline, without modifying the device drivers for these peripherals.

Mapping Features

For each `tty` device, character mapping can be done on input as well as on output. The information is stored in a buffer, the size of which should not exceed 1K. The following mapping features are supported:

Input mapping
: On input, any byte can be mapped to any byte. Using the example from the previous section, 123 could be mapped to 130, the code used for é in the IBM extended ASCII codeset, or C9, its equivalent in the ISO 8859-1 codeset.

Output mapping
: On output, any byte can be mapped to either a byte or a string. In the previous example, 130 or C9 would be mapped back to 123 to properly display the character on the screen. If the connected device is a printer that does not support the é character, it can be mapped into the string `e` Backspace `'`.

Deadkeys
: Certain keys on typewriters behave differently from the others, because when these keys are pressed, the carriage of the typewriter does not move. `^` is such a character, for example. When it is followed by an `e`, the letter ê is generated. This is called a deadkey or a non-spacing character. The `tty` subsystem supports the use of deadkeys. Typically, the `^` character and the umlaut character are used as deadkeys.

Compose sequences
: Characters can also be generated using compose sequences. A dedicated character, called the *compose character*, followed by two other keystrokes, generates a single character. As an example, Compose followed by the plus sign and the minus sign could generate the plus/minus sign. Compose sequences can also be used as an alternative for deadkeys, for example, Compose `^ e` instead of `^ e` alone.

Decimal representation
: Rarely used characters can be generated by pressing Compose, followed by three digits (which are the decimal representation of the character). This feature has been added by SunSoft. This should alleviate most of the inconvenience caused by the 1K limitation of the mapping buffer.

Toggle key

An optional toggle key can be defined to temporarily disable the current mapping at any time. This can be useful when a German programmer wants easy access to the curly braces and the brackets. A toggle key is also used by Greek users to switch between ASCII and Greek. The toggle key feature and the `ioctl` calls that implement this are SunSoft enhancements.

The `ttymap` *Program*

`ttymap` is a SunSoft utility that permits a user to activate character mapping for the user's terminal on input and output. This utility can be used for regular terminals as well as for scancode devices such as the AT console. It makes full use of all the features of the terminal (`tty`) driver and the keyboard display driver that support such mapping.

The keyboard of the console differs from the keyboards used with regular terminals in two ways: they contain a number of keys, such as the Alt key, that are not found on regular terminals, and they generate *scancodes* rather than ASCII or extended ASCII codes. Scancodes generated by PC keyboards typically represent the location of the key on the keyboard; the keyboard driver has to properly translate these scancodes. Without changing the scancode translation, if French users type an `A`, they see a `Q` on the screen. Several status keys can influence the translated code as well. The keyboard driver, and thus the `ttymap` program, make a distinction between two sets of key combinations that can be translated:

- *Regular keys*
- *Function keys*

Up to 60 key combinations are recognized as function keys. The first 12 are the 12 function keys of a 101-key PC keyboard.

F13 to F24 are the same keys used in combination with Shift, F25 to F36 when used with Control, and F37 to F48 when used with Control and Shift together. F49 to F60 are the keys on the numeric keypad.

On the console, it is more flexible to change the scancode translation than to use the general mapping features described earlier. It also reduces the risk of reaching the 1K limit of the mapping buffer.

ttymap(1) describes how the desired mapping should be laid out in a `mapfile`.

A Sample **mapfile**

Consider the following input to the `ttymap` program:

```
# sample file
input:
#
toggle:   0x14      # CTRL SHIFT F2
#
dead: '^'           # circumflex
' ' '^'             # <circumflex>
'e'       0x88      # <e-circumflex>
#
# compose key
#
compose:  0x18      # CTRL SHIFT F1
'e' ':'   0x89      # <e-diaeresis>
#
output:
'^U'      'K''I''L''L'
scancodes:
# map CTRL SHIFT F1 to be 0x18 for the compose character key
F37       0x18
# map CTRL SHIFT F2 to be 0x14 for the toggle key
F38       0x14
```

This file defines the compose and toggle keys, two deadkey sequences, one compose sequence, and "KILL" as the string to be displayed whenever `^U` is sent to the output.

Assuming this file is named `mapfile`, this mapping could be activated by typing:

```
$ ttymap mapfile
```

The terminal currently in use will then behave according to the mapping described. This has its drawbacks, however, for users with a French keyboard. For example, if a user with the login name `paul` can only use the keyboard correctly after typing this command, he is then forced to type `pqul` to log in to the system, has to have chosen a password that can still be typed in, and has to type:

```
$ tty;qp ;qpfile
```

to access the `ttymap` command itself.

To avoid this awkward situation, SunSoft has enhanced the `getty` command to activate the mapping prior to login. A new option, `-m`, has been added. Refer to the next section ("Activating Mapping Prior to Login") and *getty*(1M) for details.

Activating Mapping Prior to Login

The System Console

When the INTERACTIVE UNIX System is installed, the system asks for keyboard information. This automatically configures the system for the proper mapping on the console for the keyboard selected (provided that IBM codepage 437 is used).

Changing the Default Font for the Console

When the system is booted, IBM codepage 437 is automatically used on the console. The system can be configured to automatically use a different font, without the need for any additional commands from the user.

To do this, create a shell script with a name that starts with `S` and a number (for example, `S95font`), with the appropriate `loadfont` command replacing the one in this example:

```
# set the appropriate loadfont

/usr/bin/loadfont 8859
```

Place this file in the directory /etc/rc2.d, which contains a number of shell scripts that are automatically executed when the system comes up in multi-user mode. The order of execution depends on the number in the file name. Using a number greater than all the others for the script that changes the font is recommended. The directory also contains files with names that begin with the letter K; these are executed when the system is switched back to single-user mode. For example, this directory might contain:

```
K36sendmail      S06TMPRAMD      S21perf
S01MOUNTFSYS     S11uname        S70uucp
S05RMTMPFILES    S20sysetup      S95font
```

Other Terminals

When the system is booted, a getty program is started on every terminal that is configured in the system. This program prints login: or any other "herald" on the screen and waits until someone types input. It then calls the login program for password verification, which in turn executes the user's login program, which is typically the UNIX System command interpreter, the shell.

Each such terminal is represented by one line in the system file /etc/inittab. By modifying such a line, mapping can be activated prior to logging in on any terminal. For example, a line for the console would be:

```
co:12345:respawn:/etc/getty -m /usr/lib/keyboard/437/en_US console console
```

To activate mapping on another terminal, simply add the -m option, followed by the name of the appropriate mapping file to the getty command on the line representing the terminal. Most terminal devices have a name that contains the string tty. For example:

```
00:2345:off:/etc/getty /dev/tty00 9600
```

represents the first serial port of the computer. To test the new configuration, first kill any existing `getty` processes for the devices with entries that have been changed, then as superuser, type:

```
# telinit q
```

This causes the system to reread the `/etc/inittab` file. This file is recreated each time a new UNIX System kernel is built, using information stored in other files. Therefore, one more step needs to be taken after the terminal setup has been successfully tested. Add the same line with `getty -m` to either `/etc/conf/cf.d/init.base` (the base `inittab` file that contains information about the console) or the file in the directory `/etc/conf/init.d` that corresponds to the device driver of the peripheral to which the terminal is attached (for example, `asy` for the serial port).

User-Specific Configuration

The configuration guidelines given in the previous section assume that all users of a particular terminal use the system in the same fashion. This may not always be the case. A French user using a U.S. terminal may want to see a circumflex defined as a deadkey; an American user would not. If this is the case, you can add the appropriate `loadfont` or `ttymap` commands to the user's `$HOME/.profile` file for Bourne Shell users or to the appropriate user-specific configuration files for other shells. These commands override the system-wide configuration.

General `ttymap` *Guidelines*

SunSoft supplies `ttymap` files for the console to support all major keyboard types. These files are delivered with the INTERACTIVE UNIX Operating System in the `/usr/lib/keyboard` directory and are named `*.map`. A number of other `ttymap` files and font files (which have names with the suffix

.bdf, for example, vga855.bdf), some of which have been supplied to SunSoft by third parties, are distributed with the International Supplement on an as-is basis. The ttymap files include:

Language/ Territory	Codesets						
	437	850	863	865	866	8859-1	8859-5
da_DK	x	x		x		x	
de_CH	x	x				x	
de_DE	x	x				x	
en_UK	x	x				x	
en_US	x	x				x	
es_ES	x	x				x	
fr_CA			x			x	
fr_CH	x	x				x	
fr_FR	x	x				x	
it_IT	x	x				x	
no_NO	x	x		x		x	
ru_RU					x		x
sv_SE	x	x				x	

These files are located in directories under the /usr/lib/keyboard directory that represent the codeset (437, 850, 863, and so on) and are named for the *language_territory*, de_DE, for example. Use these files, as they are the most likely to be updated on a regular basis.

In some cases, the experienced user or the system administrator needs to create or modify an existing mapfile to support a specific terminal or environment. The following determines how the mapping should be configured:

- The type of terminal used
- The codeset used
- The layout of the keyboard used
- The country it is used in, or the language spoken by the user

Each time one of these changes, a different ttymap file is required.

Specifying Date and Time Formats

Date and time formatting consists of rules that define how date and time strings appear. These rules are created by placing specifications in the `LC_TIME` file in a `locale` directory.

The default conventions for the date and time format, as well as the names for the days of the week and the months, follow the U.S. conventions and are rarely applicable in other countries. By defining and using the date and time `locale` category, the dates and times displayed by the system will follow the local convention. Install `locale` files in `/lib/locale/ISC/`*localename* . For example, the date and time file for the `locale` `mylocale` should be installed as `/lib/locale/ISC/mylocale/LC_TIME` .

When to Use the Date and Time `locale` Category

A created and installed definition is not activated until the user specifies that it should be used. To do this, set the `LC_ALL`, `LC_TIME`, or `LANG` environment variable to the directory in which the files are stored. This must be done before a program using the stored definitions is executed. Note that the program must be set up to check and set the international environment (via the `setlocale` function). In the INTERACTIVE UNIX Operating System, the standard utilities that display the date and time, such as `date` and `ls`, have been modified to use the international environment.

Date and Time Formatting

Date and time formatting controls the appearance of date and time strings created by the system. The following aspects of formatting are controlled via the `LC_TIME` `locale` category:

- Format of the combined date and time display
- Format of the time display
- Format of the date display
- Format of the 12-hour time display
- Names of the days of the week
- Abbreviated names of the days of the week
- Names of the months

- Abbreviated names of the months
- Format of the ante meridiem and post meridiem strings used in 12-hour clock time displays

Note that the standard INTERACTIVE UNIX System library routine `strftime` (refer to *ctime*(3P)) is set up to use this information. The System V `cftime` routine, on the other hand, does not use the information created in this manner; it uses a different shell variable and searches in a different directory.

Creating a Date and Time Formatting Definition

The source language for the date and time category in the INTERACTIVE UNIX System is the language defined by the IEEE POSIX 1003.2 standard for the `LC_TIME locale` category.

A date and time editing source definition consists of a header, a date and time editing body, and a trailer. The header consists of the word `LC_TIME`. The trailer consists of the string `END LC_TIME`.

The date and time editing body consists of one or more lines of text. Each line contains a keyword followed by one or more operands. Keywords are separated from the operands by one or more blank characters (space or tab).

Operands are characters, strings of characters, or digits. When a keyword is followed by more than one operand, the operands must be separated by semicolons (`;`). Blanks are allowed before and/or after a semicolon. Strings must be surrounded by quotes. Individual characters may be surrounded by quotes, but it is not required. Blank lines or lines containing a # sign in the first column are ignored. A line can be continued by typing a backslash (\) as the last character on the line.

The following keywords are recognized:

`LC_TIME`
: The header.

`abday`
: Defines the abbreviated names of the weekdays, starting with Sunday.

`day`
: Defines the names of the weekdays, starting with Sunday.

`abmon`
Defines the abbreviated names of the months, starting with January.

`mon`
Defines the names of the months, starting with January.

`t_fmt`
Defines the format of the time string.

`d_fmt`
Defines the format of the date string.

`d_t_fmt`
Defines the format of the combined date and time string.

`am_pm`
Defines the strings used to specify ante meridiem and post meridiem in a time string according to the 12-hour clock.

`t_fmt_ampm`
Defines the format of the 12-hour time display.

`END LC_TIME`
The trailer.

Refer to *date*(1) for more information about date field descriptors.

`abday` *Keyword*

This keyword defines the abbreviated weekday names, corresponding to the `date %a` field descriptor. The operand must consist of seven strings, separated by semicolons. The first string must be the abbreviated name of the first day of the week (Sunday), the second string must be the abbreviated name of the second day, and so on. For example:

```
abday   "Sun";"Mon";"Tue";"Wed";"Thu";"Fri";"Sat"
```

day *Keyword*

This keyword is used to define the full weekday names, corresponding to the `date %A` field descriptor. The operand must consist of seven strings, separated by semicolons. The first string must be the full name of the first day of the week (Sunday), the second string must be the full name of the second day, and so on. For example:

```
day        "Sonntag";"Montag";"Dienstag";\
           "Mittwoch";"Donnerstag";"Freitag";"Samstag"
```

abmon *Keyword*

This keyword is used to define the abbreviated month names, corresponding to the `date %b` field descriptor. The operand must consist of twelve strings, separated by semicolons. The first string must be the abbreviated name of the first month of the year (January), the second string must be the abbreviated name of the second month, and so on. For example:

```
abmon      "Jan";"Feb";"Mar";"Apr";"May";"Jun";\
           "Jul";:Aug";"Sep";"Oct";"Nov";"Dec"
```

mon *Keyword*

This keyword is used to define the full month names, corresponding to the `date %B` field descriptor. The operand must consist of twelve strings, separated by semicolons. The first string must be the full name of the first month of the year (January), the second the full name of the second month, and so on. For example:

```
mon        "Januar";"Februar";"März";"April";\
           "Mai";"Juni";"Juli";"August";\
           "September";"Oktober";"November";"Dezember"
```

`d_t_fmt` *Keyword*

This keyword is used to define the appropriate date and time representation, corresponding to the `date %c` field descriptor. The operand must consist of a string and may contain any combination of characters and `date` field descriptors. In addition, the string may contain the `date %n` and `%t` field descriptors for newline and tab characters, respectively. For example:

```
d_t_fmt     "%a %b %d %H:%M:%S %Y"
```

`d_fmt` *Keyword*

This keyword is used to define the appropriate date representation, corresponding to the `date %x` field descriptor. The operand must consist of a string and may contain any combination of characters and date field descriptors. For example:

```
d_fmt       "%m/%d/%y"
```

`t_fmt` *Keyword*

This keyword is used to define the appropriate time representation, corresponding to the `date %X` field descriptor (see *date*(1)). The operand must consist of a string and may contain any combination of characters and date field descriptors. For example:

```
t_fmt       "%H:%M:%S"
```

am_pm *Keyword*

This keyword is used to define the appropriate representation of the ante meridiem and post meridiem strings, corresponding to the `date %p` field descriptor. The operand must consist of two strings, separated by a semicolon. The first string must represent the ante meridiem designation; the last string, the post meridiem designation. For example:

```
am_pm        "AM";"PM"
```

t_fmt_ampm *Keyword*

This keyword is used to define the appropriate time representation in the 12-hour clock format with `am_pm`, corresponding to the `date %r` field descriptor. The operand must consist of a string and may contain any combination of characters and date field descriptors. If this keyword is not defined, the default (`%I:%M:%S %p`) is used. For example:

```
t_fmt_ampm "%I.%M.%S %p"
```

A Sample File

```
LC_TIME
#
#
abday         "Son";"Mon";"Die"; \
              "Mit";"Don";"Fre";"Sam"
day           "Sonntag";"Montag";"Dienstag"; \
              "Mittwoch";"Donnerstag";"Freitag";"Samstag"
abmon         "Jan";"Feb";"März";"Apr"; \
              "Mai";"Juni";"Juli";"Aug"; \
              "Sept";"Okt";"Nov";"Dez"
mon           "Januar";"Februar";"März";"April"; \
              "Mai";"Juni";"Juli";"August"; \
              "September";"Oktober";"November";"Dezember"
d_t_fmt       "%I.%M.%S %p %m/%d/%y"
d_fmt         "%m/%d/%y"
t_fmt         "%I.%M.%S %p"
am_pm         "VM";"NM"
t_fmt_ampm    "%I.%M.%S %p"
#
END LC_TIME
```

How a Program Uses This Information

If a program needs to access the values in the current `locale`, it can do so via the library subroutine `nl_langinfo`, as well as by using the definition via the `strftime` library subroutine (refer to *ctime*(3P)).

Specifying Character Classification Information

The character classification category determines classification of characters as letters, digits, and so on, as well as some other information about the codeset and character set used. The default character classification only recognizes the 26 ASCII letters as such, which means that any program processing non-English text that depends on the classification will behave incorrectly. For example, take `vi`, which prints nonprintable characters using an octal notation. For `vi` to correctly display non-ASCII characters, you must change the character classification. Another example is a program that does uppercase to lowercase conversion; the standard table handles only ASCII.

Defining Character Classification

These definitions are created by placing a specification in the LC_CTYPE file in a locale directory. This specification is output by the chrtbl utility (refer to *chrtbl*(1M)). The created table should also be copied to the /lib/chrclass directory, to accommodate programs that still use the CHRCLASS variable, a System V feature.

When to Use the Character Classification locale Category

The created and installed definitions are not activated until the user specifies that they should be used. To do this, the user must set the LC_ALL, LC_CTYPE, or LANG environment variable to the directory in which the files are stored. This must be done before a program is executed. Note that the program must be set up to check and set the international environment (via the setlocale function). In the INTERACTIVE UNIX System, the standard utilities that depend on character classification, such as grep, ls, ed, and sort, have been modified to use the international environment.

However, the vi program has not been modified to use the international environment; it uses the information in the /lib/chrclass directory and the value of the environment variable CHRCLASS.

Creating a Character Classification Category Definition

Character classification definitions are created using the chrtbl utility. The source language for the character classification category in the INTERACTIVE UNIX Operating System allows the user to define the name of the data file created by chrtbl, the assignment of characters to character classifications, and the relationship between uppercase and lowercase letters. The character classifications recognized by chrtbl are:

chrclass
: Name of the data file to be created by chrtbl.

isupper
: Character codes to be classified as uppercase letters.

islower
: Character codes to be classified as lowercase letters.

`isdigit`
: Character codes to be classified as numeric.

`isspace`
: Character codes to be classified as spacing (delimiter) characters.

`ispunct`
: Character codes to be classified as punctuation characters.

`iscntrl`
: Character codes to be classified as control characters.

`isblank`
: Character code for the space character.

`isxdigit`
: Character codes to be classified as hexadecimal digits.

`ul`
: Relationship between uppercase and lowercase characters.

Any lines with a # sign in the first column are treated as comments and are ignored. Blank lines are also ignored.

A character can be represented as a hexadecimal or octal constant (for example, the letter `a` can be represented as 0x61 in hexadecimal or 0141 in octal). Hexadecimal and octal constants may be separated by one or more space or tab characters.

The dash character (–) can be used to indicate a range of consecutive numbers. Zero or more space characters may be used for separating the dash character from the numbers. The backslash character (\) is used for line continuation. Only a carriage return is permitted after the backslash character.

The relationship between uppercase and lowercase letters, `ul`, is expressed as ordered pairs of octal or hexadecimal constants: <*uppercase_character lowercase_character*>. These two constants may be separated by one or more space characters. Zero or more space characters may be used for separating the angle brackets (< >) from the numbers.

An Example of a Character Classification Definition

The following is an example of an input file:

```
chrclass       LC_CTYPE
isupper        0x41 - 0x5a
islower        0x61 - 0x7a
isdigit        0x30 - 0x39
isspace        0x20 0x9 - 0xd
ispunct        0x21 - 0x2f 0x3a - 0x40 \
               0x5b - 0x60 0x7b - 0x7e
iscntrl        0x0 - 0x1f  0x7f
isblank        0x20
isxdigit       0x30 - 0x39 0x61 - 0x66 \
               0x41 - 0x46
ul             <0x41 0x61> <0x42 0x62> <0x43 0x63> \
               <0x44 0x64> <0x45 0x65> <0x46 0x66> \
               <0x47 0x67> <0x48 0x68> <0x49 0x69> \
               <0x4a 0x6a> <0x4b 0x6b> <0x4c 0x6c> \
               <0x4d 0x6d> <0x4e 0x6e> <0x4f 0x6f> \
               <0x50 0x70> <0x51 0x71> <0x52 0x72> \
               <0x53 0x73> <0x54 0x74> <0x55 0x75> \
               <0x56 0x76> <0x57 0x77> <0x58 0x78> \
               <0x59 0x79> <0x5a 0x7a>
```

How a Program Uses This Information

Programs access this information by using the character classification and conversion library interfaces (refer to *ctype*(3C)). As `vi` does not use the information via the `locale`, it is useful to copy the table to the `/lib/chrclass` directory and give it the same name as the `locale`.

Use in Regular Expressions and Shell Pattern Matching

The information in the character classification definition can be directly used in regular expressions, via the character class syntax inside a bracket expression. The syntax is:

```
[:class-name:]
```

where *class-name* is the name of one of the following:

`alpha`	A letter
`upper`	An uppercase letter
`lower`	A lowercase letter
`digit`	A decimal digit
`xdigit`	A hexadecimal digit
`alnum`	An alphanumeric (letter or digit)
`space`	A character that produces white space in displayed text
`punct`	A punctuation character
`print`	A printing character
`graph`	A character with a visible representation
`cntrl`	A control character

For example, the following command will find all file names in the current directory that begin with an uppercase letter:

```
ls "[[:upper:]]*"
```

These specifications are primarily intended to replace the current use of expressions like `[A-Z]`, which are not portable (Z is not the last letter in all alphabets).

Configuring Network Drivers 13

The *Network Drivers* diskettes supplied with the INTERACTIVE UNIX System contain device drivers for various networking cards. With this software and a compatible networking card, you can set up Token Ring and Ethernet local area networks.

Installing the Network Drivers

1. To install a network driver, start `sysadm`. Select `Install a package` from the `Software` menu.
2. Answer the question regarding media type, and insert the *Network Drivers I* diskette.

3. Your screen will look similar to this:

```
 Disk        File        Machine     Software    User        Help        Quit
Install a software package
                 Install an INTERACTIVE-type Software Package

                         Select package(s) to install:

                Beep when diskette is finished:   <yes>

   <yes> Link Layer Driver (llc) for Host Based TCP/IP                4.1
   <no > 3COM Etherlink (3C501) Board                                 4.1
   <no > 3Com Etherlink II (3C503) Board                              4.1
   <no > 3COM 523 Ethernet Driver                                     4.1
   <no > Racal InterLan LLI ES-3210 STREAMS Driver                    4.1
   <no > NE 2000 Novell Ethernet Board                                4.1
   <no > Racal Interlan NI5210 and NI9210                             4.1
   <no > Racal Interlan NI6510                                        4.1

              OK                CANCEL              HELP
```

If the driver you are looking for isn't listed, choose `CANCEL`, and repeat steps 2 and 3 with the *Network Drivers II* diskette.

4. Use the up and down arrow keys to move to the driver for the network card that you are planning to use. Toggle the `no` to `yes`. Select `OK`.

Note – Since most drivers require that the Link Layer Driver for host-based TCP/IP (`llc`) be installed first, `llc` is set to install by default.

Note – The 3Com® 523 driver works only with a Micro Channel architecture bus.

You can install more drivers than you currently plan to use without affecting your system (except for the storage space required). By default, the system will install only the Link Layer Driver; however, you can install more than one driver by setting each desired driver's `Install` setting to `yes`. Only those drivers to be installed should be preceded by a `yes` on your screen.

Your screen will look similar to this:

```
Disk        File        Machine                 User        Help        Quit

                            Install Package Files
/etc/drivers/wd2/Driver.o
/etc/drivers/wd1
/etc/drivers/wd1/description
/etc/drivers/wd1/Master
/etc/drivers/wd1/Node
/etc/drivers/wd1/System
/etc/drivers/wd1/Driver.o
/etc/drivers/wd0
/etc/drivers/wd0/Node
/etc/drivers/wd0/Master
/etc/drivers/wd0/Space.c
/etc/drivers/wd0/description
/etc/drivers/wd0/System
/etc/drivers/wd0/Driver.o
106 blocks
When you are finished installing network drivers, you will
need to configure the one you wish to use. Execute the command
'sysadm netdrvrmgmt' to do this.
```

Similar information will be repeated for each driver you choose to install.

5. Press Escape when you have finished installing the drivers.

Configuring the Network Drivers

After you have installed the drivers that you want on your fixed disk, you must configure the ones you intend to use on your system.

Modifying I/O Address and IRQ Values

Each driver that controls hardware in the INTERACTIVE UNIX System has an I/O address range and an IRQ vector associated with it. In general, these values must be unique for each piece of hardware. The default values of the

I/O address range for each device driver supplied with the INTERACTIVE UNIX System have been set to an I/O address space that should not overlap with the I/O address space of any other supplied device driver. All drivers must be configured to use unique I/O address ranges. Additionally, each driver must use a unique IRQ line. There are only 16 available IRQ lines and many of them are used by the system, as shown in Table 13-1.

Table 13-1 IRQ Lines

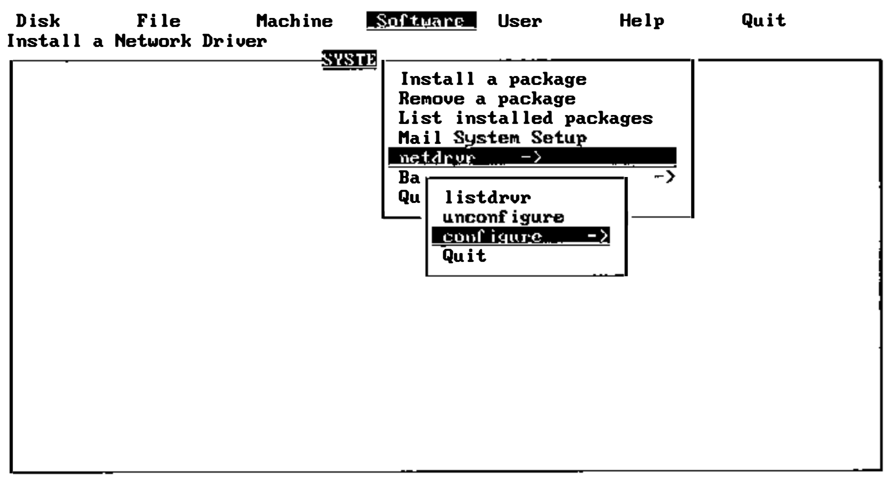
Interrupt	Usage
0	Timer
1	Keyboard
2	Used by system
3	COM2
4	COM1
5	LPT2
6	Diskette (floppy) controller
7	LPT1
8	Clock
9	Used by system
10	Free
11	Free
12	Free
13	Math coprocessor
14	Standard hard (fixed) disk controller
15	Free

There are a few free interrupts from 10 to 15. However, only 16-bit cards can utilize them. If an interrupt is used by a piece of hardware which is not present in the machine, that interrupt can be used by another peripheral. For example, if the machine has only one parallel port on IRQ 7, IRQ 5 can then be used by another card. IRQ 2 is a special interrupt line. It can be used by the operating system if the driver is set up to use IRQ 9. This technique will not work on all machines; some motherboards use IRQ 2 internally. Also, some older graphic video boards use IRQ 2. For information on modifying driver IRQ values and I/O addresses, refer to *sdevice*(7).

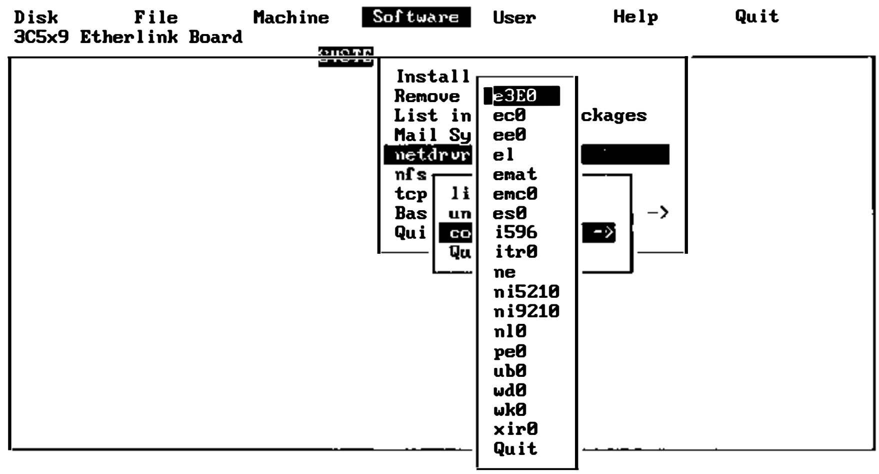
Configuring One Driver

Note – Experienced users can bypass all menus and begin driver configuration immediately by typing `sysadm configure`.

1. Log in as `sysadm` or use the `sysadm` command to access the Main menu. Select `Software`, then select `netdrvr`. The system displays the following menu:



2. To configure a network driver, select `configure`. The system then displays the driver setup options. Note that only the drivers you have previously chosen to install will appear on your screen. For example:



Note – The full name of the network driver currently highlighted will be displayed in the upper left portion of the screen.

---

3. Use the arrow keys, then press Enter to include a board in your system. Note that the information you must supply will vary depending on the driver selected. The following example is specific to the `wd0` driver for the Western Digital EtherCard PLUS board. To include this board in your system, select `wd0`, then press Enter. Your screen will look similar to this:

```
                Network Driver Software Installation

                    WD Hardware Configuration

The wd0 driver is currently configured with the following WD hardware
settings:

        Interrupt Request Line (IRQ)............ 3
        Start I/O Address on I/O bus............ 0x280
        End I/O Address on I/O bus.............. 0x29f
        Start Controller Memory Address......... 0xd0000
        End Controller Memory Address........... 0xd3fff

Do you want to use the current IRQ value [Yes]:
```

The system displays the default settings for this particular board. The IRQ value for this device is `3`, and the I/O address range is `0x280` through `0x29f`. This means that this I/O address range on the I/O bus is dedicated to this device. No other device should be communicating with the CPU on addresses falling in this range.

4. If you want to use the current IRQ value, type `y`. If you have set the IRQ jumper value on the board to be a number other than `3` and you want to change this value in the system files to match your board's setting, type `n`. You will want to do this if you already have a device driver configured into the kernel that occupies that particular value (IRQ `3`). If you type `n`, the system then prompts you to enter the new IRQ value:

```
Please enter the IRQ value for the board [ 3 ]:
```

5. Type in the new IRQ value, for example, `4`. The system will then ask:

```
Do you want to use the current I/O Port address range [Yes]?
```

6. Press Enter to accept the default if you want to use the current I/O port address range; type `n` if you want to change the value of the I/O port address range to match the value set on your board. You must be familiar with the specifications of this device to provide the beginning and ending values. Refer to the documentation that accompanied your board to determine the appropriate values. The system prompts you to enter the beginning I/O address (in hex) for this device. Your screen will look similar to this:

```
Please enter Start I/O address for the board in hex [ 280 ]:
```

7. Type in the new starting I/O address, for example, `400`. The system will then ask you for the new ending I/O address:

```
Please enter End I/O address for the board in hex [ 29f ]:
```

8. Type in the new ending I/O address in hex, for example, `40f`. The system will then check for interrupt vector conflicts that may exist between this device and other device drivers that are currently configured into the kernel. This check detects any conflicts early in the reconfiguration process, allowing you to go back and change any conflicting values.

   Note that if you are configuring the `wd0` driver *only*, you will be asked to respond to the following additional questions. (If you are configuring a different driver, different questions may be displayed.) Your screen will look similar to this:

```
Do you want to use the current Controller Shared Memory
address range [Yes]?
```

Type n. Your screen will look similar to this:

```
Please enter the Start Shared Memory address for the board
[d0000]:
```

Type in the starting shared memory address, or press Enter to accept the default. Your screen will look similar to this:

```
Please enter the End Shared Memory address for the board
[d3fff]
```

Type in the ending shared memory address, or press Enter to accept the default.

9. If there is no conflict, you can skip to step 11. If there is a conflict, your screen will look similar to this:

```
Checking for interrupt vector conflicts - Please wait...

There is a conflict with the device "asy" on interrupt
request line 4.  Please resolve this conflict by changing the
interrupt request line of either the conflicting device
or the WD board.

Change the IRQ value of "wd0" in the /etc/conf/sdevice.d/wd0
file or make modifications to the configuration file of each
conflicting device resident in the /etc/conf/sdevice.d
directory named after the corresponding device.  If you are
changing the interrupt vector and request line of the WD
board, then you will need to do the following:

   1   Abort this installation.
   2   Change the IRQ jumper on the WD board to the
       desired IRQ value.
   3   Restart this installation script by executing
       sysadm and giving the correct value for the interrupt
       value when asked by the script.

This conflict must be resolved before the new kernel can
be linked.  Please refer to the Network Driver
software and WD hardware installation manual for more
detailed instructions on changing the interrupt vector,
I/O address, etc.


Now interrupt this script (press CRTL \) to abort
the installation...
```

10. If this message appears, press Control-\ (or the interrupt key sequence on your system) to interrupt the configuration process. Make any necessary fixes as indicated, then return to step 1.

11. If there is no conflict between the IRQ value of this device and that of other devices currently listed in the file `/etc/conf/cf.d/sdevice`, your screen will look similar to this:

```
New /etc/conf/sdevice.d/wd0 file is:
wd0     Y    32   5   3   3    280    29f    d0000    d3fff

Do you wish to build a new kernel at this time? [y, n, q]
```

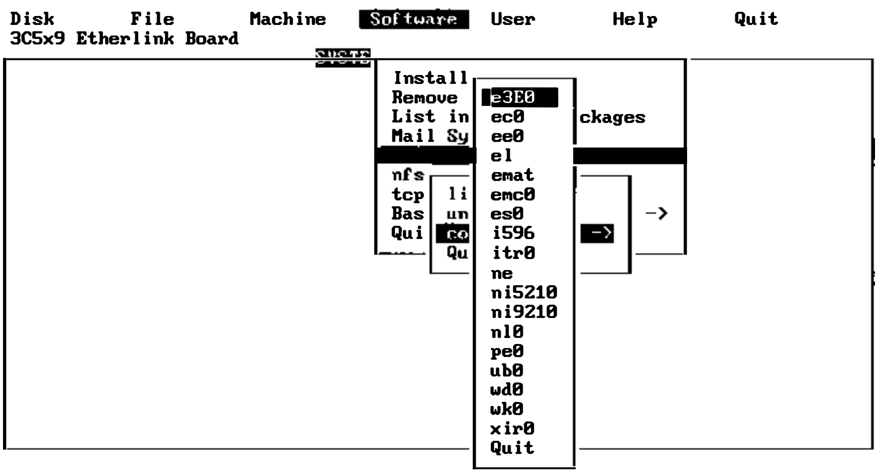
The system displays the device-dependent configuration file that you have just modified and asks if you are ready to build a new kernel.

12. If you want to configure other drivers before building a new kernel, type `n`; the system will then prompt you to return to the `Network Driver Software Installation` screen. If you are ready to build a new kernel, type `y`. The system will automatically call the `kconfig` program so that you can build and install a new kernel. See Chapter 5, "Using kconfig to Tailor Your System Kernel," for information on how to build and install a new kernel.

Configuring Additional Drivers

You can configure drivers for multiple boards from the same or different manufacturers on a system.

1. Access the `netdrvr` menu under `Software` and select `configure`. Your screen will look similar to this:



2. Select the driver option that corresponds to the second board you want to configure, for example, `wd0`, the Western Digital EtherCard PLUS. If you have already configured a driver for one Western Digital EtherCard PLUS board, the system asks:

```
Do you want to configure another Western Digital Board [Y/N]?
```

3. Type `y`. The system then displays:

```
Enter Multiple Western Digital Board #[wd1,wd2..etc]
```

4. If this is the second Western Digital board you are configuring, type `wd1`; if it is the third, type `wd2`, and so on. If you type `wd1`, your screen will look similar to this:

```
           Western Digital Hardware Configuration

   The wd1 driver is currently configured with the following WD
   hardware settings.
   NOTE: All hardware settings are in Hex value:

        Interrupt Request Line (IRQ)...............3
        Start I/O Address on I/O bus...............0x380
        End I/O Address on I/O bus ................0x39f
        Start Controller Shared Memory Address.....0xd6000
        End Controller Shared Memory Address.......0xd9fff
```

These are the default device-dependent values for this board as they appear in the file `/etc/conf/sdevice.d/wd1`.

5. Configuration of the second driver proceeds exactly as described for a single driver except your screen will look similar to this:

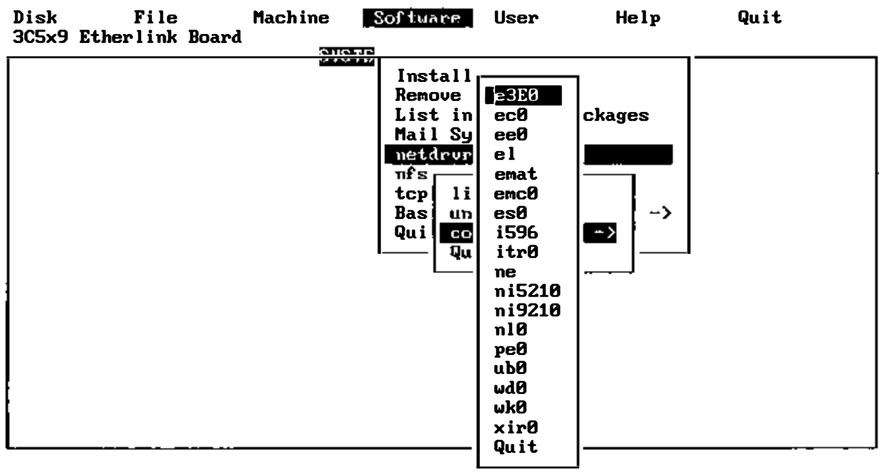
```
New /etc/conf/sdevice.d/wd1 file is:
wd1     Y   32   5   3  3   380    39f    d6000   d9fff

Do you wish to build a new kernel at this time? [y, n, q]
```

Changing the Parameters of Previously Configured Boards

If you want to change the parameters of a board for which you have already configured a driver, follow this procedure.

1. Access the `netdrvr` menu under `Software` and select `configure`. The system displays:



2. Select the driver option that corresponds to the board you want to change, for example, `wd0`, the Western Digital EtherCard PLUS. The system asks:

```
Do you want to configure another Western Digital Board [Y/N]?
```

3. If you have only configured one board, type `n`. The parameters of `wd0` are then displayed.

   If you want to change the parameters for the second Western Digital EtherCard PLUS board you have configured, answer `y` to this question. Your screen will look similar to this:

```
Enter Multiple Western Digital Board #[wd1,wd2..etc]
```

4. Type `wd1` if you want to change the parameters of the second Western Digital Board you configured; type `wd2` if you want to change the parameters of the third one, and so on. The parameters of the board you chose are then displayed, and you can change them just as during the initial configuration.

Configuring INTERACTIVE TCP/IP 14

Configuring the TCP/IP Interface

You must install and configure the Network Drivers subset (see Chapter 13, "Configuring Network Drivers") with the drivers you want to use and you must install INTERACTIVE TCP/IP *before* you can configure TCP/IP to use the interface. Once the software is loaded, a new kernel must be generated to support INTERACTIVE TCP/IP modules and the network board(s) of your choice.

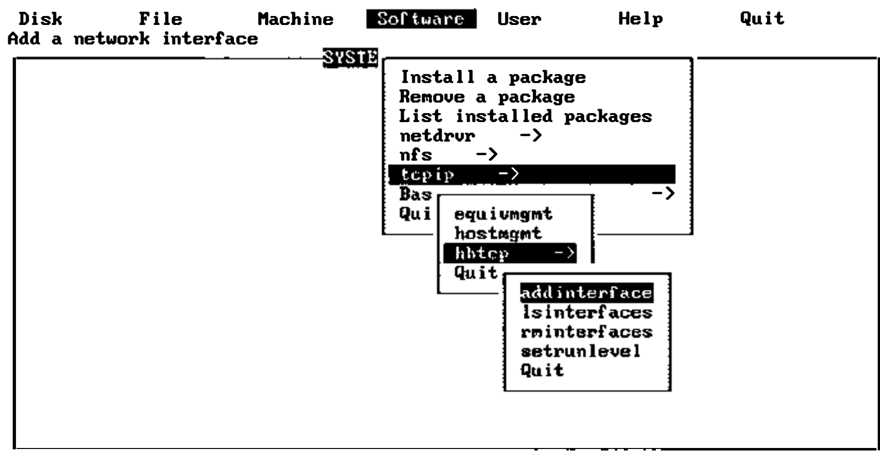
1. Log in as `sysadm` and select the `Software` menu option from the System Administration menu. (If you are not at the System Administration menu display, use the `sysadm` command or log in as `sysadm` to access this menu, then select `Software`.) Your screen will look similar to this:

```
 Disk      File      Machine    Software   User      Help      Quit
Install a software package
                                SYSTE
                                      Install a package
                                      Remove a package
                                      List installed packages
                                      netdrvr   ->
                                      nfs    ->
                                      tcpip    ->
                                      Basic Networking      ->
                                      Quit
```

2. Select the `tcpip` option from the `Software` menu. Your screen will look similar to this:

```
 Disk        File        Machine     Software   User        Help        Quit
Manage TCP/IP hosts.equiv entries (list, add, delete)
                                SYSTE
                                     Install a package
                                     Remove a package
                                     List installed packages
                                     netdrvr    ->
                                     nfs    ->
                                     tcpip    ->
                                     Bas                            ->
                                     Qui  equivmgmt
                                          hostmgmt
                                          hbtcp    ->
                                          Quit
```

3. Select the `hbtcp` option. The screen displays options for managing your network interface. Your screen will look similar to this:



You can use these options to add, list, or remove a network interface from your system. Note that both the appropriate network driver and INTERACTIVE TCP/IP must be installed on your system before INTERACTIVE TCP/IP can be used. Refer to Chapter 13, "Configuring Network Drivers," for more information about how to configure a specific driver into the kernel.

Adding a Network Interface

In order to use a driver that is installed on your system, you must add it as a new interface. The `addinterface` menu provides you with five options. You may add a driver from those installed with the Network Drivers subset, add a third-party driver, add a slip interface, configure TCP/IP for local operation only, or view a list of the network interfaces currently installed and configured on your system.

1. Select option `1` to see the `addinterface` menu. The system will display a list of options for adding a network interface. Your screen will look similar to this:

```
                     ADD INTERFACE COMMAND MENU
1 add a driver from the network driver subset as a new interface
2 add a third party Ethernet driver as a new interface
3 add a slip interface
4 configure TCP/IP for local operation only (no interface card)
5 list the installed and configured drivers
"q" to quit
Enter the operation you want to perform:
```

Adding an Installed Driver

1. To add a driver that is already installed on your system, select option `1` from the `addinterface` menu. The system will display a list of each driver that is configured into the kernel and can be used as a network interface. In this example, `ec0` is the only driver installed. Your screen will look similar to this:

```
Choose one of these configured drivers
        ec0, wd0, ne0
Please enter the name
```

2. Type the name of the driver you want to add. The system next prompts you for the host name. In this example, `myhost` is used. Your screen will look similar to this:

```
Please enter the hostname you wish to be associated
with this interface. This hostname must be in the
/etc/hosts file. Names can be added to this file with
the hostmgmt sysadm menus.
Enter the hostname for this interface :  myhost
```

3. The system prompts you for the netmask. Netmasks are most often needed if you are using more than one network interface or if your LAN has more than one segment (connected by a gateway). If you are configuring only one

interface and you have a small local area network, you will probably not need to change the netmask. Type n to use the default netmask. Your screen will look similar to this:

```
You will be configured to use the default netmask.
Change the netmask? [y, n, ?] n
```

If you want to change the netmask, see "Configuring a Network" later in this chapter or consult your network administrator.

4. The system prompts you for the broadcast address. Usually, you will not need to change the default broadcast address. Type n to use the default broadcast address.

```
You will be configured to use the default broadcast address.
Do you want to change the broadcast address? [y, n, ?] n
```

The addinterface screen will again be displayed.

Adding a Third-Party Driver

If you have a third-party driver that supports the INTERACTIVE TCP/IP interface, you can attempt to add it with this option. Note that this does not apply to network drivers installed on previous releases of INTERACTIVE TCP/IP. These previously installed drivers must be reinstalled from the Network Drivers extension before they can be used. Refer to Chapter 13, "Configuring Network Drivers," for information about reinstallation of these drivers.

1. To add a third-party driver, select option 2 from the addinterface menu.
2. The system will prompt you for the name of the driver you want to add. Type the name of the interface. In this example, ne0 is used. Your screen will look similar to this:

```
Enter the name of the driver you wish to use: ne0
```

3. If the driver you have selected is not installed, the system will remind you of this and ask you whether you want to (a) continue with the interface configuration and install the driver later or (b) choose another driver. Note that a driver must be installed through the Network Drivers extension, but it can be installed either before or after it has been associated with a host name.

4. If the driver you have selected is installed, you will be prompted for the host name you want associated with it. In this example, `myhost-ne0` is used. Your screen will look similar to this:

```
Please enter the hostname you wish to be associated
with this interface. This hostname must be in the
/etc/hosts file. Names can be added to this file
with the hostmgmt sysadm menus.
Enter the hostname for this interface : myhost-ne0
```

5. The system prompts you for the netmask. Your screen will then look similar to this:

```
You will be configured to use the default netmask.
Change the netmask? [y, n, ?]
```

6. Type `n` or press Enter to accept the default netmask.

7. The system prompts you for the broadcast address. Your screen will look similar to this:

```
You will be configured to use the default broadcast address.
Do you want to change the broadcast address? [y, n, ?]
```

   Type `n` to use the default broadcast address. The system will return you to the `addinterface` screen.

8. To list the currently configured network interfaces, first press Enter to access the `hbtcpmgmt` menu, then select option `2`. Your screen will look similar to this:

```
       Currently installed network interfaces:

Driver name      Hostname                 IP Address
    ec0         myhost                   128.212.16.59
    ne0         myhost-ne0               128.212.16.60

Press the RETURN key to see the hbtcpmgmt menu [?, ^, q]:
```

   The screen displays the interfaces you have installed.

Adding a SLIP Interface

A Serial Line Internet Protocol (SLIP) interface is a point-to-point connection, for example, between a tty and a modem. A SLIP interface permits users to use any tty driver as a network interface. Performance is dependent upon the speed of the serial line used to connect hosts. A limited dial-up utility that allows users to establish network links with modems over normal phone lines is also supported. You can run the SLIP interface alone or in addition to an Ethernet interface on your system. If you are using SLIP and an Ethernet interface, then you should first install the Ethernet interface driver from the Network Drivers subset.

SLIP performance is dependent on the speed of the serial line used to connect hosts. Although it is functional at speeds of 2400 baud, some users may find SLIP performance unacceptable at speeds of less than 9600 baud.

1. To add a SLIP interface, select option `3` from the `addinterface` screen. Your screen will look similar to this:

```
SLIP will be configured to use the "sl0" device
```

2. The system prompts you for a host name. Type in the host name you want to use. Your screen will look similar to this:

```
Please enter the hostname you wish to be associated
with this interface. This hostname must be in the
/etc/hosts file. Names can be added to this file
with the hostmgmt sysadm menus.
Enter the hostname for this interface : myhost-sl
```

In this example, the host name `myhost-sl` is used.

3. The system prompts you for a target host name. Type in the target host name you want to use. Your screen will look similar to this:

```
Please enter the target hostname you wish to be associated
with this interface. SLIP is a point to point link, so
INTERACTIVE TCP/IP needs to know the hostname of the system
that will be on the other side of the SLIP link.
This host name must also be in the /etc/hosts database.
Enter the target hostname for the slip interface : target-sl
```

In this example, the target host name `target-sl` has been used.

4. The system will now prompt you to use the default netmask and broadcast address. Type `n` at each prompt to use the system defaults.

5. The system now displays a screen similar to this:

```
               SLIP Hardware Configuration

The sl0 driver is currently configured to use
the following TTY Line settings:
        TTY Line...................... tty00
        Baud Rate..................... 9600

Do you want to use the current TTY Line [ Yes ]? [y, n]
```

Type y or press Enter to use the current line. If you do not wish to use the current line, type n. The system will then prompt you for the line you want to use. Your screen will look similar to this:

```
Do you want to use the current baud rate [ Yes ]? [y, n]
```

6. Type y or press Enter to use the current baud rate. Type n if you do not want to use the current baud rate. The system will then prompt you to supply a baud rate. Your screen will then look similar to this:

```
Is this SLIP connection going to be directly connected
to another system running SLIP (without a modem) [ No ] ?
[y, n]
```

7. If your SLIP connection will be with a modem, press Enter or type n. Type y if it will be a direct connection.

8. If your connection is with a modem, you will need to add an account for dialup SLIP access. Your screen will look similar to this:

```
Would you like to add an account for dialup Slip access
[ No ] [y, n]
```

Type y to add an account.

9. The system next prompts you for information about the new account. You must provide a login ID. You may either provide user and group ID numbers or select the system defaults by pressing Enter at the appropriate prompts. Your screen will look similar to this:

```
Enter SLIP login ID [?, q]: target1

Enter user ID number (default 836) [?, q]:

Enter group ID number or group name
(default 1) [?, q]:

This is the information for the new login:
        User's name:    Dialup SLIP account
        login ID:       target1
        user ID:        836
        group ID:       1       (other)
Do you want to install, edit, or skip this entry
[i, e, s, q]?

        SLIP configuration completed
```

10. If you want to list the currently configured network interfaces, return to the `hbtcpmgmt` menu and select option 2. Your screen will look similar to this:

```
Currently installed network interfaces:

Driver name     Hostname          IP Address
    pc0         myhost            128.212.16.59
    ne0         myhost-ne0        128.212.16.60
    sl          myhost-sl         128.212.16.61

Press the RETURN key to see the hbtcpmgmt menu [?, ^, q]:
```

Configuring TCP/IP for Local Operation

To use TCP/IP without a network, select option 4 from the `addinterface` menu and follow the menu instructions that appear on your screen.

Removing a Network Interface

You can remove a network interface through the `hbtcpmgmt` menu.

1. To remove a network interface, return to the `hbtcpmgmt` menu and select option `3`.
2. The system will prompt you for the name of the interface to be removed. In this example, the `sl` interface will be removed. Your screen will look similar to this:

```
Currently installed network interfaces:

Driver name     Hostname          IP Address
    pc0         myhost            128.212.16.59
    ne0         myhost-ne0        128.212.16.60
    sl          myhost-sl         128.212.16.61

Enter the name of the interface you wish to remove:
```

   After entering the name of the interface you want to remove, you will see the message:

```
The /etc/netd.cf file has been updated.

Press the RETURN key to see the hbtcpmgmt menu [?, ^, q]:
```

Listing Network Interfaces

To view the list of currently installed interfaces, use option `2` of the `hbtcpmgmt` menu. Your screen will look similar to this:

```
Currently installed network interfaces:

Driver name     Hostname          IP Address
    pc0         myhost            128.212.16.59
    ne0         myhost-ne0        128.212.16.60

Press the RETURN key to see the hbtcpmgmt menu [?, ^, q]:
```

Configuring a Network

A host machine is configured into the network through special configuration files. To link a host to the rest of the network, the host name and the internet address of the host must be added to the configuration files `/etc/hosts` and `/etc/hosts.equiv`. For more information about the internet address, refer to "Internet Address (Software "Logical" Address)" below. The following default configuration files are provided with the software:

- `/etc/hosts`
- `/etc/hosts.equiv`
- `/etc/networks`
- `/etc/protocols`
- `/etc/services`

Refer to "/etc/hosts," "/etc/hosts.equiv," "/etc/networks," "/etc/protocols," and "/etc/services" later in this section for more information about these files.

Host Naming Conventions

Each host in the network must be identified according to TCP/IP naming conventions. These conventions include two ways of identifying a network host: host name and internet address. The host name and the internet address *must* be set in the `/etc/hosts` file and may be set in the `/etc/hosts.equiv` file, using the `sysadm` menu.

Host Name

The host name (also called the node name) is initially set using the `setup` command during the installation of the INTERACTIVE UNIX Operating System. It can also be set with the `hostname` command or with the `syssetup` menu of the `sysadm` utility. A host name may be chosen arbitrarily. It is used primarily as a way for users to refer to a particular machine. For example, `ism780`, `market`, and `engineering` are all possible host names. To display a host name, use the `uname -n` command:

```
$ uname -n
market
$
```

Internet Address (Software "Logical" Address)

The internet address is a 32-bit (4-byte) number that includes a network number and a local address and uniquely identifies a host machine. Each internet address consists of two, three, or four parts. Each part is separated by a dot, for example, 128.212.32.1. The network number identifies your network to other networks and may consist of up to three parts of the internet address. In the above example, "128.212" is the network number; all hosts on a local network must have the same network number. The local address consists of the rest of the internet address and includes a node number, which identifies the host within the network and possibly a subnetwork number as described below. The node number is an arbitrary number assigned by the system administrator (the "32.1" in the above example).

Note – If your network will not be communicating with outside networks or already has an assigned network number, it is *not* necessary to obtain a network number from NIC (see below); the network number and local address may be selected arbitrarily. The network number should be the same for all machines in the network; the node number should be unique for each machine.

If your network is to communicate with other large networks, such as ARPANET, you must obtain a unique network number from the following organization:

Network Solutions
Attn: InterNIC Registration Services
505 Huntmar Park Drive
Herndon, VA 22070

1-703-742-4777

HOSTMASTER@RS.INTERNIC.NET

The network will be classified as A, B, or C and a network number assigned.

Class A
: A class A internet address begins with a number in the range of 0-127 and uses only the first byte for the network number; the other three bytes are available for the local address, which is assigned by the

system administrator. Class A network numbers are used for very large networks, such as ARPANET. A host on a class A network might have any of the following internet addresses:

```
57.0.1.1
57.0.1
57.1
```

In each case, the network number (the first byte) identifies the network as a class A network. All subsequent numbers are interpreted as part of the local address. The last three bytes are interpreted as the node number and should uniquely identify the machine.

Class B

A class B internet address begins with a number in the range of 128.1 to 191.254 and uses the first two bytes for the network number; the other two bytes are available for the local address, which are assigned by the system administrator. Class B addresses are typically assigned to most large organizations. The following is an example of a class B internet address:

```
150.8.8.16
```

The first two bytes identify the network as a class B network. All subsequent numbers are interpreted as part of the local address.

Class C

A class C internet address begins with a number in the range of 192.1.1 to 223.255.254 and uses the first three bytes for the network number; the last byte is used for the local address, which is assigned by the system administrator. The following is an example of a class C internet address:

```
192.20.0.1
```

The first three bytes identify the network as a class C network. The last byte is interpreted as the local address.

A subnetwork is a local interpretation of the internet address that allows bits from the local address to be used to extend the network number some additional bits, providing a subnetwork identifier.

Subnetworking provides a way for a large network to communicate with smaller networks through a *gateway*, rather than individually. A gateway serves as a way of routing information between smaller and larger networks. On large networks, such as ARPANET, there are a limited number of gateway routing table entries available; therefore, subnetworking can be used to subdivide an internet address into network, subnetwork, and local address parts with the subnetwork identifier interpreted locally. Instead of communicating with each subnetwork individually, the larger network routes all information for hosts within the subnetwork to the gateway, which then redistributes the data according to the subnetwork identifier and the local address.

For example, a class B network might have the following internet address:

```
128.212.0.0
```

Subnetworks within that address might be:

```
128.212.16.1
128.212.32.2
128.212.46.3
128.212.10.4
```

The first two bytes are the network number, the third byte is the subnetwork identifier, and the fourth byte is the local address (node number).

For more information about internet addresses and subnetworks, refer to the following Request For Comment (RFC) documentation available from NIC:

- RFC997 *Internet Numbers*
- RFC950 *Internet Standard Subnetting Procedure*

Configuring the Network

In order to make a machine known to the rest of the network, the `/etc/hosts` and the `/etc/hosts.equiv` files must be modified to include an entry for that host. This is done using the `sysadm` facility.

This section describes the `/etc/hosts` and `/etc/hosts.equiv` files and explains how to use the `sysadm` utility to make changes to these files.

/etc/hosts

The `/etc/hosts` file lists all hosts in the network. An entry is required for each host (including the local machine). Entries must have the following format:

```
internet_address host_name aliases [comments]
```

internet address
: The internet address is the logical address of the host, which consists of the network number (including subnetwork number and sub-subnetwork number, if any) and the local address (including the node number). Refer to "Internet Address (Software "Logical" Address)" earlier in this section for information about how to obtain an internet address.

host name
: The host name is an arbitrarily selected name set by the system administrator. For example, the host name for a machine that is used by the marketing staff of a corporation might be `market`.

aliases
: Aliases are nicknames for the host, separated by blanks. They are used to reduce the amount of typing required when referring to the host.

comments
: Comments must begin with the # character. They are used to provide descriptive information about a host.

The following shows a sample /etc/hosts file:

```
127.1       local       localhost
193.16.14.1 market      mktg m # Marketing system
193.16.14.2 train       trng t # Training system
193.16.14.3 accounting  acct a # Accounting system running
                               # Solaris
193.16.14.4 engineering eng  e # Engineering workstation running
                               # Solaris
```

An /etc/hosts file must exist on each host in the network.

/etc/hosts.equiv

The /etc/hosts.equiv file establishes "equivalent" accounts among hosts on the network. It enables users other than root on other hosts to access the local host without passwords. If a remote machine is listed in the /etc/hosts.equiv file located on the user's local host, the user may use rlogin (refer to *rlogin*(1C)) to log in to the remote machine. The user will not have to provide a login ID if the user has been assigned the same login ID on both the local and remote hosts. The /etc/hosts.equiv file located on the local host engineering might look like this:

```
accounting
market
```

The entries in this file specify that users on hosts accounting and market who maintain a login ID on host engineering can use the rlogin command to easily access engineering.

Other commands that use the hosts.equiv file are rcp and rsh.

Caution – Using hosts.equiv and .rhosts files is considered to be a security problem; they should not be used on an open network.

Using `sysadm` *to Modify Configuration Files*

The `sysadm` utility can be used to modify the configuration files `/etc/hosts` and `/etc/hosts.equiv`.

1. Use the `sysadm` command to display the System Administration menu, then select the `Software` menu. Your screen should look similar to this:

```
 Disk      File      Machine    Software   User      Help      Quit
Install a software package
                               SYSTE  Install a package
                                      Remove a package
                                      List installed packages
                                      netdrvr    ->
                                      nfs    ->
                                      tcpip    ->
                                      Basic Networking       ->
                                      Quit
```

2. From the `Software` menu, select the `tcpip` submenu:

```
  Disk      File      Machine   Software   User      Help      Quit
Manage TCP/IP hosts.equiv entries (list, add, delete)
                                SYSTE
                                     Install a package
                                     Remove a package
                                     List installed packages
                                     netdrvr    ->
                                     nfs    ->
                                     tcpip     ->
                                     Bas                          ->
                                     Qui  equivmgmt
                                          hostmgmt
                                          hbtcp     ->
                                          Quit
```

3. The first option, `equivmgmt`, allows you to list, add, or delete entries from the `/etc/hosts.equiv` file:

```
This procedure is used to list, add, and delete entries in
the Extended Networking Utilities '/etc/hosts.equiv' file.

Type 'q' at any time to quit the current operation.
If a '?' appears as a choice, type '?' for help.

If a default appears in the question, type <ENTER> for the
default.


Enter the operation you want to perform:
        1 list
        2 add
        3 delete

(default list)[q]:

Press the ENTER key to see the tcpipmgmt menu [?, ^, q]:
```

Whenever a host is added to or deleted from the network, this file should be modified on each host in the network.

4. The second option, `hostmgmt`, allows you to list, add, or delete entries from the `/etc/hosts` file:

```
This procedure is used to list, add, and delete entries in
the Extended Networking Utilities '/etc/hosts' file.

Type 'q' at any time to quit the current operation.
If a '?' appears as a choice, type '?' for help.

If a default appears in the question, type <ENTER> for the
default.


Enter the operation you want to perform:
        1 list
        2 add
        3 delete

(default list)[q]:

Press the ENTER key to see the tcpipmgmt menu [?, ^, q]:
```

Whenever a host is added to or deleted from the network, this file should be modified on each host in the network.

Other Default Files

Three other default files are included in the INTERACTIVE TCP/IP software package: `/etc/networks`, `/etc/protocols`, and `/etc/services`. These files are described below.

/etc/networks

The /etc/networks file contains the network names and numbers of those networks your network communicates with. The file should contain one entry per network, including the local network. Entries have the following format:

network_name network_number [*network aliases*] [# *comments*]

The following is an example of an /etc/networks file:

```
#
# Internet networks
#
loop    127     loopback
iconet  128.212.32
```

This file is used by programs concerned with internetworking, such as route, ifconfig, gated, named, and netstat.

A list of network and host names may be obtained from the NIC for users on the Internet.

/etc/protocols

The `/etc/protocols` file contains a list of protocols used on the network. Each protocol is associated with a unique, identifying number. The most commonly used protocols are `TCP`, `IP`, and `UDP`. The default `/etc/protocols` file should look similar to this:

```
#
# Internet (IP) protocols
#
ip      0     IP      # internet protocol, pseudo protocol number
icmp    1     ICMP    # internet control message protocol
ggp     3     GGP     # gateway-gateway protocol
tcp     6     TCP     # transmission control protocol
egp     8     EGP     # exterior gateway protocol
pup     12    PUP     # PARC universal packet protocol
udp     17    UDP     # user datagram protocol
hmp     20    HMP     # host monitoring protocol
```

The `/etc/protocols` file is provided as part of the software package and should not be modified.

/etc/services

The /etc/services file contains a list of services available on the network. Each service is associated with a unique, identifying number, and each service invoked uses one of two protocols to do data transfer. The two protocols used are TCP and UDP. The /etc/services file contains entries similar to these:

```
# Network services, Internet style
#
echo           7/udp
ftp            21/tcp
telnet         23/tcp
name           42/tcp      nameserver
whois          43/tcp      nicname
hostnames      101/tcp     hostname         # usually from sri-nic
#
# Host-specific functions
#
tftp           69/udp
finger         79/tcp
link           87/tcp      ttylink
ingreslock     1524/tcp
#
# UNIX-specific services
#
exec           512/tcp
login          513/tcp
shell          514/tcp     cmd              # no passwords used
who            513/udp     whod
ntalk          518/udp
```

The /etc/services file is provided as part of the software package. Care should be taken when modifying this file.

To invoke certain utilities, such as echo, enter telnet followed by the host name and its corresponding port number, as listed above. In the above example, the port number for echo is 7. After typing telnet *hostname* 7, the user will be in the echo program and may enter text to be echoed.

Advanced Configuration Topics

The topics covered in this section address issues that arise in a fairly complex network environment involving multiple networks, dissimilar TCP/IP implementations, or subnetworking. A simple Local Area Network made up of similar host machines running the same version of INTERACTIVE TCP/IP will not need these features.

Setting the Subnet Mask

If the host is to be used in a network with subnets, the software must know which parts of the internet address contain the network and subnet numbers. This is done by setting the subnet mask. One of the last lines in the `/etc/netd.cf` file is the `ifconfig` line for the network interface that you are using (refer to *ifconfig*(1M)). The correct `ifconfig` line to change is the one whose first parameter to `ifconfig` is the abbreviation for the network driver that you are using (such as `wd0` for a Western Digital board).

The netmask parameter and the network mask should be added to this `ifconfig` line. For example, if you are using network number 128.212, a Class B network, you have two bytes of local address and could use the first byte for a subnet number. In that case, you would set a mask using dot notation containing three full bytes (network and subnet numbers) and add the two parameters `netmask 255.255.255.0` to the `ifconfig` line in `/etc/netd.cf`.

Setting a Broadcast Address

The broadcast address used by default in INTERACTIVE TCP/IP is the internet address of the host, with the node number consisting entirely of ones. If the subnet number 32 is added to our previous example of network number 128.212, the default broadcast address would be 128.212.32.255. If you must change this value to be compatible with older TCP/IP hosts on your network, you can set the broadcast address in the `ifconfig` line of the `/etc/netd.cf` file, as described in "Setting the Subnet Mask." For example, to use a broadcast address of all ones, add the parameters `broadcast 255.255.255.255` to the `ifconfig` line.

Switching Network Boards

If you must change the network board on your host, the software may need reconfiguration to access the new board. If you use the same kind of board with the same jumper settings, no changes should be required. However, other machines may not recognize your new board until their `arp` table entry for your old board times out (refer to *arp*(1M) for instructions on how to delete the old `arp` table entries). If you use the same kind of board with different jumper settings, you should reconfigure the board through the Network Drivers subset.

If you replace your board with a different type of board, there are several steps that are required:

1. Reconfigure the board through the Network Drivers subset.
2. Use the `rminterface` option of the `hbtcpmgmt` menu to remove the interface for the old board from your system.
3. Use the `addinterface` option of the `hbtcpmgmt` menu to configure your new board. Refer to "Configuring the TCP/IP Interface" for more information about adding and removing interfaces.

Using and Running a Nameserver

If the host is to be run as part of a network that changes configuration often, it may be easier to use a nameserver on one machine instead of changing the `/etc/hosts` file on each machine for each configuration change. The INTERACTIVE TCP/IP software is configured to call a nameserver if you change the file `/etc/resolv.conf` (refer to *resolver*(5)). To use a remote nameserver, replace the `nonameserver` line with the `nameserver` option followed by the internet address of the host running the nameserver.

To use a local `nameserver`, configure `/etc/named`, automatically start `named` in `/etc/rc3.d/S05hbtcp`, and delete the `/etc/resolv.conf` file. Refer to *named*(1M).

Activating TCP/IP and Using Network Applications

The system is normally configured to run at level 2, which is specified by an entry in the file `/etc/inittab`. The TCP/IP software, however, is configured to run at level 3, which activates networking programs. After the INTERACTIVE TCP/IP software is installed, the entry in the `/etc/inittab` file is updated to specify level 3:

```
is:3:initdefault:
```

If you installed the software properly and initiated a system shutdown and reboot, the TCP/IP software should be running properly. However, if you made any changes to the `/etc/inittab` file or are experiencing problems with the TCP/IP software, make sure that the entry in the `/etc/inittab` file specifies level 3, not level 2.

`/usr/ucb`

The standard Berkeley utilities are installed in the directory `/usr/ucb`. In order to use the utilities, your search path has to include the `/usr/ucb` directory or you must type the full path name explicitly. The standard system files are set up to search `/usr/ucb`, but users with existing `.profile`, `.login`, or `.cshrc` files may need to edit them. Because the `/usr/ucb` path name has been compiled into some of the utilities, moving them is not recommended. However, copies of the utilities may be put elsewhere.

Initiating SLIP Over Modem and Serial Lines

Dialing Out

To make a connection via `sldialup` (refer to *sldialup*(1M)):

1. As user `root`, type: `sldialup` *port baud*, where *port* is the ttyname parameter and *baud* is the speed of the modem connected to the terminal line. For example, to use a 2400 baud modem connected to `/dev/tty00`, type:

```
# sldialup tty00 2400
```

2. You are now talking directly to the modem. Give the commands to dial the host. For example, if you are using a Hayes-type modem, and the phone number is 555-1212, type:

```
ATDT 5551212
```

3. When connected, press Enter a few times to get a `login:` prompt. You may have to press Control-b (to send a break signal) to change the baud rate.

4. Log in with the account set up on the target system. The account on the target system should have `sllogin` as its login shell. (Refer to *sllogin*(1M) for further information.)

5. When logged in, press Escape to return to the # prompt on the local system. You should now be connected.

6. If you are using one system to act as a gateway between a SLIP and a non-SLIP network, then you need to start up the gateway routing daemon. As user `root`, type:

```
# /etc/gated
```

7. It takes about 45 seconds to set up the routing tables for the network on the target system. You can see when they are there by typing:

```
$ netstat -r
```

Refer to *netstat*(1) for further information.

At this point, the network should be connected between the two systems.

Direct Lines

Direct lines are connected using `slattach` (refer to *slattach*(1M)). If you have configured SLIP, then `slattach` should be started automatically when the network is started. However, if for some reason it needs to be started manually, follow these steps:

1. Type: `slattach` *port baud*, where *port* is the ttyname parameter and *baud* is the optional baud rate parameter used to set the speed of the connection. For example, to use a baud rate 2400 on `/dev/tty00`, type:

```
$ slattach /dev/tty00 2400
```

2. If you are using one system to act as a gateway between a SLIP and a non-SLIP network, then you need to start up the gateway routing daemon. As user `root`, type:

```
# /etc/gated
```

3. It takes about 45 seconds to set up the routing tables for the network on the target system. You can see when they are there by typing:

```
$ netstat -r
```

Refer to *netstat*(1) for further information.

At this point, the network should be connected between the two systems.

INTERACTIVE NFS Management 15

After you have installed the INTERACTIVE NFS extension on your fixed disk, use the `sysadm nfsmgmt` command to initialize and maintain the INTERACTIVE NFS environment on your system.

You can access the system administration bar menu by logging in as `sysadm` or by typing the `sysadm` command. As an alternative, you can type `sysadm nfsmgmt` to access the NFS Management menu directly. Your screen will look similar to this:

```
Export a network file system
 ┌──────────────┐─────────SYSTEM ADMINISTRATION──────────────────────┐
 │ exportnfs    │                                                     │
 │ mountnfs     │                                                     │
 │ setupnfs     │                                                     │
 │ startstopnfs │                                                     │
 │ umountnfs    │                                                     │
 │ unexportnfs  │                                                     │
 │ Quit         │                                                     │
 └──────────────┘                                                     │
 │                                                                    │
 └────────────────────────────────────────────────────────────────────┘
```

Use the up and down arrow keys to move to the option you want, and press Enter to select it.

Each option and its use is described in detail in this chapter.

Setting Up INTERACTIVE NFS (`setupnfs`)

The `setupnfs` option is used once only, at the time of INTERACTIVE NFS installation. The `setupnfs` option enables the INTERACTIVE NFS servers. (A *server* is a daemon process that performs actions requested by the client machines.)

To set up INTERACTIVE NFS, type `sysadm nfsmgmt` and select the `setupnfs` option. Your screen will look similar to this:

```
Enabling NFS Server
Enabling NFS Mount Server
Enabling Automatic Startup of NFS
```

Starting INTERACTIVE NFS (`startstopnfs`)

The `startstopnfs` option is used to start INTERACTIVE NFS sessions. The system must be at run level 3 for these commands to work.

1. Access the NFS Management menu and select the `startstopnfs` option. Your screen will look similar to this:

```
Enter the item number of the operation you wish to use.
     1   Start NFS.
     2   Stop NFS.
     3   Start NFS and make entries to automatically start
         up NFS.
     4   Stop NFS and remove entries to automatically start
         up NFS.
     q   To exit.
Enter your selection:
```

2. There are two options for starting INTERACTIVE NFS:
   - Select option 1 if you want to start a single session of INTERACTIVE NFS. Your screen will look similar to this:

```
Starting NFS
NFS Startup...
ONC daemons: portmap rexd pcnfsd rwalld
Lock manager: statd lckclnt(4) lockd
NFS daemons: nfsclnt(4) biod(4)
NFS Startup complete.
NFS initialized

Press the RETURN key to see the nfsmgmt menu [?, ^, q]:
```

- If you want NFS to start automatically every time you bring the system up to multi-user/networking mode, select option 3 rather than option 1.

  If you select option 3, your screen will look similar to this:

```
Starting NFS
NFS Startup...
ONC daemons: portmap rexd pcnfsd rwalld
Lock manager: statd lckclnt(4) lockd
NFS daemons: nfsclnt(4) biod(4)
NFS Startup complete.
Checking for files
Linking appropriate files
NFS initialized

Press the RETURN key to see the nfsmgmt menu [?, ^, q]:
```

  The files necessary to run the INTERACTIVE NFS system are now in place.

Stopping INTERACTIVE NFS (`startstopnfs`)

At some time you may need to stop INTERACTIVE NFS, for example, if you are having network problems. The `startstopnfs` option is used to stop INTERACTIVE NFS sessions.

1. Access the NFS Management menu and select the `startstopnfs` option. Your screen will look similar to this:

```
Enter the item number of the operation you wish to use.
     1  Start NFS.
     2  Stop NFS.
     3  Start NFS and make entries to automatically start
        up NFS.
     4  Stop NFS and remove entries to automatically start
        up NFS.
     q  To exit.
Enter your selection:
```

2. There are two options for stopping INTERACTIVE NFS:
   - Select option 2 to simply stop the current session of INTERACTIVE NFS. Your screen will look similar to this:

```
Stopping NFS
NFS Shutdown: [NFS Shutdown Complete]
Press the RETURN key to see the nfsmgmt menu [?, ^, q]:
```

   - If you have previously configured the INTERACTIVE NFS extension to start automatically every time you bring the system up to multi-user/networking mode and you want to change this, select option 4 rather than option 2. INTERACTIVE NFS will no longer start every time you bring your system up to multi-user/networking mode; you will have to start it manually using the `startstopnfs` option. This operation may take up to 1 minute to complete. Your screen will look similar to this:

```
Stopping NFS
NFS Shutdown: [NFS shutdown complete]
Removing entries for automatically starting NFS

Press the RETURN key to see the nfsmgmt menu [?,q]:
```

Exporting a File System and Adding Allowed Hosts (`exportnfs`)

Exporting a file system notifies the other machines in the network of the availability of a file system on your local machine. Once a file system is exported, the system administrators of other machines in the network can mount that file system under the directories of their choice on their machines. Users on remote machines can then access files that exist in your local exported file system just as if they were physically located on the remote machines.

To export a file system, you provide the mount directory name of the file system you want to export and any restrictions you want to place on which *hosts* (machines) in the network may access the file system.

Start INTERACTIVE NFS (if it is not already running). Access the NFS Management menu and select the `exportnfs` option. Your screen will look similar to this:

```
Enter the item number of the export operation you wish to
execute. Options are:
       1 Export a file system.

       2 Add a host to the list of allowed hosts for a
         file system.

Select export operation (q):
```

The two export options are detailed below.

Option 1

1. To export a file system for the first time, select option `1`. The system then asks:

```
Available file systems:
    /home
    /usr/local
    /v
    /usr

Enter the mount directory of the file system you want
to export [?, q]:
```

2. Enter the full mount directory name of the file system you plan to export, `/usr` for example. The system then asks:

```
Should /usr be exported read-only? [y, n, q]
```

3. Type `y` to prevent all remote users from altering `/usr` files and directories; type `n` if changes will be allowed. The system then asks:

```
Continue adding Host Names for /usr? [y, n, q]
```

4. Type `y` to add the name of a host that will be allowed to access `/usr`. Your screen will look similar to this:

```
Enter a Host name for /usr:
```

5. Type in the network node name of the machine (for example, `dorrit`) you want to give access to `/usr`. Your screen will then look similar to this:

```
Should dorrit be given root access? [y, n, q]
```

6. Type `y` to allow the `root` user access to all files on this file system, regardless of the files' permissions. Type `n` to deny this access.

Caution – Selecting `y` could compromise your system's security. Select `y` *only* if you are sure you can trust the superuser(s) on the remote system to which you are exporting this file system.

Your screen will then look similar to this:

```
Should dorrit be given read-only access? [y, n, q]
```

7. Type `y` to give `read-only` access to users on the remote host; type `n` to allow the remote users to modify files, if their permissions allow it. Note that `read-only` access only affects this particular client system. Your screen will then look similar to this:

```
Continue adding Host Names for /usr? [y, n, q]
```

8. Type `y` to repeat these steps for another machine; `n` to stop. The system will then ask for a comment or description of the file system:

```
Enter a brief comment about /usr:
```

9. Type in a short description, such as `Shared usr file system`. The system will then ask:

```
Okay to export /usr? [y, n, q]
```

10. Type `y` if you have entered all the necessary information correctly. The file system is then exported and your screen will look similar to this:

```
Press the RETURN key to see the nfsmgmt menu [?,q]:
```

Option 2

1. To add a host to the list of allowed ones, start INTERACTIVE NFS (if it is not already running). Access the NFS Management menu and select the `exportnfs` option. Your screen will look similar to this:

```
Enter the item number of the export operation you wish to
execute.  Options are:
       1  Export a file system.

       2  Add a host to the list of allowed hosts for a
          file system.

Select export operation (q):
```

2. To add host machines for a file system you have already exported, select option `2`. The system then asks:

```
Available file systems:

    /usr

Enter the mount directory of the file system you want
to add hosts for [?, q]:
```

3. Type in the name of the mount directory for `/usr`. The system then asks:

```
Current hosts already available for /usr:

    root allowed:  dorrit
    read/write:    dorrit muffy
    read-only:     rangoon

Continue adding Host Names for /usr? [y, n, q]
```

4. To add a new host, type `y`. Your screen will look similar to this:

```
Enter a Host name for /usr:
```

5. Type in the network node name of the machine (for example, `dorrit`) that you want to give access to `/usr`. Your screen will then look similar to this:

```
Should dorrit be given root access? [y, n, q]
```

6. Type `y` to allow the `root` user access to all files on this file system, regardless of the files' permissions. Type `n` to deny this access.

Caution – Selecting `y` could compromise your system's security. Select `y` *only* if you are sure you can trust the superuser(s) on the remote system to which you are exporting this file system.

Your screen will then look similar to this:

```
Should dorrit be given read-only access? [y, n, q]
```

7. Type `y` to give `read-only` access to users on the remote host; type `n` to allow the remote users to modify files, if their permissions allow it. Note that `read-only` access only affects this particular client system. Your screen will then look similar to this:

```
Continue adding Host Names for /usr? [y, n, q]
```

8. Type y to repeat these steps for another machine; n to stop. The system will then ask:

```
Okay to export /usr? [y, n, q]
```

9. Type y if you have entered all the necessary information correctly. The file system is then exported and your screen will look similar to this:

```
Press the RETURN key to see the nfsmgmt menu [?,q]:
```

Mounting a Remote File System or Changing Its Entry (mountnfs)

Mounting a remote file system under a directory on your machine lets users transparently access files and directories on that machine. All that is required to mount a remote file system is a valid directory under which it can be placed on the local machine.

Before beginning the mount procedure, identify or create a directory where the remote file system will be mounted.

Access the NFS Management menu and select the mountnfs option. Your screen will look similar to this:

```
Enter the item number of the mount operation you wish to
execute.  Options are:
      1  Mount a file system.

      2  Make an entry for automatically mounting a file
         system when NFS is started.

      3  Change an entry for automatically mounting a file
         system when NFS is started.

Select mount operation (q):
```

The three mount options are detailed below.

Option 1

1. Select option `1` if you want the file system to be mounted during the current INTERACTIVE NFS session. Your screen will look similar to this:

```
Enter the Host you want to mount from:
```

2. Enter the name of the remote host system (for example, `dorrit`) that contains the file system that you want to mount on your local system. The screen will then look similar to this:

```
Available file systems:

    /v
    /y
    /etc
    /usr

Enter the file system you want to mount [q]:
```

3. Type `/usr` to select that file system for mounting. The system will then ask where it should be mounted:

```
Enter the directory to mount /usr under:
```

4. Enter the full path name of a valid directory under which `/usr` is to be mounted, for example, `/remote`. The system will then ask about restrictions:

```
Mount /usr as Read-Only? [y, n, q]
```

5. Type `y` if you do not want users on your local machine to have `write` permission in the remote file system. Type `n` if you want them to be able to modify the files. The screen will then look similar to this:

```
Mount /usr soft ? [y, n, q]
```

6. Type `y` if you want the system to return an error when the server does not respond. Type `n` if you want the system to continue to retry indefinitely. The screen will then look similar to this:

```
Enter the mount buffer size: [1) 1K, 2) 4K, 3) 8K]
```

7. Select an appropriate `read/write` buffer size. If there is no specific restriction on your LAN board, you should always select option `3` (8K) to get the best performance. The screen will then look similar to this:

```
OK to mount /usr under /remote? [y, n, q]
```

8. Type `y`. The system will then display the message:

```
/usr is now mounted under /remote.

Press the RETURN key to see the nfsmgmt menu [?, q]:
```

Option 2

1. Select option `2` to mount the file system automatically every time INTERACTIVE NFS is started in the future. Your screen will look similar to this:

```
Enter the Host you want to mount from:
```

2. Enter the name of the remote host system (for example, `dorrit`) that contains the file system that you want to mount on your local system. The screen will then look similar to this:

```
Available file systems:

    /v
    /y
    /etc
    /usr

Enter the file system you want to mount automatically [q]:
```

3. Type `/usr` to select that file system for mounting. The system will then ask about restrictions:

```
Automatically mount /usr read-only? [y, n, q]
```

4. Type `y` if you do not want users on your local machine to have `write` permission in the remote file system. Type `n` if you want them to be able to modify the files. The screen will then look similar to this:

```
Automatically mount /usr soft ? [y, n, q]
```

5. Type `y` if you want the system to return an error when the server does not respond. Type `n` if you want the system to continue to retry indefinitely. The screen will then look similar to this:

```
Automatically mount /usr in the background? [y, n, q]
```

6. Type `y` if you want system startup to continue immediately while the mounting operation runs in the background rather than waiting for the mount to complete before startup proceeds. If you type `y`, and the server on

which this file system is located is dead or hung, it will not delay the boot process on your machine. Type `n` if you do not want this protection. The screen will then look similar to this:

```
Enter the mount buffer size : [1) 1K, 2) 4K, 3) 8K]
```

7. Select an appropriate `read/write` buffer size. If there is no specific restriction on your LAN board, you should always select option 3 (8K) to get the best performance. The system will then ask where it should be mounted:

```
Enter the directory to mount /usr under:
```

Enter the full path name of a valid directory under which `/usr` is to be mounted, for example, `/remote`. The system then asks:

```
Make entry for mounting /usr under /remote automatically
when NFS is started? [y, n, q]
```

8. Type `y`. The system then displays the message:

```
Entry made.

Mount dorrit:/usr under /remote now? [y, n, q]
```

9. Type `y` if you want to mount `/usr` immediately. Your screen will look similar to this:

```
/usr is now mounted under /remote.

Press the RETURN key to see the nfsmgmt menu [?, q]:
```

Option 3

1. Select option 3 to change entries for resources that are to be mounted automatically when INTERACTIVE NFS is started. You can change the file system name, the directory under which it is mounted, and, if `write` permission is available on the file system, its permissions. If the resource is already mounted because an INTERACTIVE NFS session is currently running, the change will not take effect until INTERACTIVE NFS is stopped and started again. Your screen will look similar to this:

```
Current entries:
dorrit:/usr /remote -r NFS,rsize=4096,wsize=4096,soft,bg

Available File Systems:
dorrit:/usr
Enter the resource(s) you want to change [?, q]:
```

2. Type the name of the file system, for example, `dorrit:/usr`. The screen will then look similar to this:

```
Select:
       1      To change host/filesystem name
       2      To change directory name
       3      To turn on/off read-only
       q      To quit
Enter selection:
```

3. Type 2 to change the directory name. The system will then ask for a replacement name:

```
Enter replacement name for /remote
```

4. Type in the name of your new directory, for example, `/newremote`. The screen will then look similar to this:

```
Okay to change directory name /remote to
/newremote? [y, n, q]
```

5. Type y to change the directory. Your screen will then look similar to this:

```
/remote changed to /newremote

Press the RETURN key to see the nfsmgmt menu [?,q]:
```

Unexporting a File System (unexportnfs)

Unexporting a file system makes that file system on your local machine unavailable to remote users.

Note – If you unexport a file system during an INTERACTIVE NFS session and that file system has already been mounted on another machine, the other machine will see error messages such as `Cannot statfs /usr: file descriptor in bad state.`

Before unexporting a file system, it is a good idea to type the command `showmount -a` to list the machines that have your exported file systems currently mounted. You may want to ask the system administrators of those machines to unmount your file system before you unexport it.

To unexport a file system, access the NFS Management menu and select the `unexportnfs` option. Your screen will look similar to this:

```
Enter the item number of the export operation you wish to
execute.  Options are:

        1  Unexport a file system.

        2  Delete a host on the list of allowed hosts for
           a file system.

Select operation (q):
```

The two options for using `unexportnfs` are detailed below.

Option 1

1. If you want to unexport a file system, select option `1`. Your screen will look similar to this:

```
Available file systems:
/usr
Enter the mount directory(s) of the file systems you want
to unexport [?, q]:
```

2. Enter the name of the file system, for example, `/usr`. If you selected option `1`, the system then asks:

```
Okay to unexport /usr? [y, n, q]
```

3. Type `y` to unexport `/usr`.

Option 2

1. To delete a host on the list of allowed hosts, select option 2. Your screen will look similar to this:

```
Available file systems:
    /usr

Enter the mount directory(s) of the file systems
you want to delete hosts from [?, q]:
```

2. Type in the name of the mount directories you want. Your screen will look similar to this:

```
Current hosts available for deletion for /usr:

    root allowed:  dorrit
    read/write:    dorrit muffy kanga
    read-only:     rangoon

Enter Host name to delete for /usr:
```

3. Type in the remote host name. The system then asks:

```
Continue deleting host names for /usr? [y, n, q]
```

4. Type y if you want to delete additional hosts that are currently able to mount this file system. Type n if you do not want to delete any additional hosts. The system will then ask:

```
Okay to update /usr? [y, n, q]
```

5. Type y if you have entered all the necessary information correctly. The file system will then be unexported and your screen will look similar to this:

```
Press the RETURN key to see the nfsmgmt menu [?, q]:
```

Unmounting a Remote File System (`umountnfs`)

Unmounting a remote file system makes it unavailable to users on your local machine.

Note – When you unmount a remote file system, local users must not be using that file system. If a user is currently in that file system, the attempt to unmount it will fail.

To unmount a remote file system, access the NFS Management menu and select the `umountnfs` option. Your screen will look similar to this:

```
Enter the item number of the operation you wish to execute.
Options are:
       1  Unmount a file system.

       2  Remove an entry for automatically mounting a
          file system for when NFS is started.

Select operation (q):
```

The two options for unmounting a remote file system are detailed below.

Option 1

1. Select option `1` if you want to unmount a file system for the current INTERACTIVE NFS session only. If the file system is mounted automatically each time INTERACTIVE NFS is started, choosing option `1` will not affect its availability during future INTERACTIVE NFS sessions. Your screen will look similar to this:

```
Mounted file systems:
/remote

Enter the file system(s) you want to unmount [?, q]:
```

2. Type in the mount directory name of the file system you want to unmount, for example, `/remote`. The system then asks:

```
Okay to unmount /remote [y, n, q]
```

3. Type `y`. The machine will then display:

```
/remote has been unmounted.

Press the RETURN key to see the nfsmgmt menu [?, q]:
```

Option 2

1. To remove the entry in the file /etc/fstab that automatically mounts the selected file system, select option 2. Your screen will look similar to this:

```
Resources mounted automatically:
dorrit:/usr /remote -r NFS,rsize=4096,wsize=4096,soft,bg

Enter Resources to remove:
```

2. Enter the remote file system name, in this case, dorrit:/usr. The screen then prompts for your confirmation:

```
Okay to remove permanent entry for dorrit:/usr? [y, n, q]
```

3. Type y. Your screen will look similar to this:

```
Entry for dorrit:/usr has been removed.

Press the RETURN key to see the nfsmgmt menu [?, q]:
```

Configuring INTERACTIVE X11 16

If you have installed the INTERACTIVE X11 optional extension, follow the procedures in this chapter to add and remove X11 users and servers, configure X11 drivers, and modify configuration information.

Per-User X11 Configuration

The `addxuser` and `delxuser` functions of `sysadm` allow you to automatically configure a user's environment for INTERACTIVE X11.

Adding X11 Users

To enable a user to use X:

1. Use the `sysadm` command or log in as `sysadm` and access the Main menu.
2. Select the `x` submenu of the `Software` menu. The menu will show the following options:

```
configmgmt
hostmgmt
servermgmt
xdrivermgmt
addxuser
delxuser
```

3. Select the `addxuser` option to enable a user to use the X Window System. Your screen will look similar to this:

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

4. Type `y` to set up a new X user. Your screen will look similar to this:

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

5. Type the login name of the user you wish to add. Your screen will look similar to this:

```
Enter the window manager you want to use:
 1 mwm
 2 twm
 3 uwm

[default=1]
```

6. Enter the number that corresponds to the type of window manager the new user will use, or press Enter to accept the default, the Motif® Window Manager `mwm`.

   The display below shows a sample entry for a user with the login name `rws` who is using the Motif Window Manager. The name `rws` will be replaced with the login name of the user you want to add. Your screen will look similar to this:

```
Installing /usr/rws/.mwmrc
Installing /usr/rws/.xdefaults
Modifying /usr/rws/.profile

User rws can now use the X Window System

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

7. Type `y` to set up additional users, or type `q` to quit. If you type `q`, your screen will look similar to this:

```
Press <ENTER> to continue.....
```

8. Press Enter to return to the X Window System Management menu.

Deleting X11 Users

To disable a user's ability to run X, select the `delxuser` option:

1. Your screen will look similar to this:

```
Do you wish to disable an X user's ability to run X?
[y, n, ?, q]
```

2. Type `y` to disable an X user's ability to run X. Your screen will look similar to this:

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

3. Now type the login name of the user who will no longer be running X. The display below shows a sample entry for the login name `rws`. The name `rws` will be replaced with the login name of the user you want to delete. Your screen will look similar to this:

```
rws's /usr/rws/.Xdefaults and .uwmrc files will now be removed.
Proceed? [y, n, ?, q]
```

4. The system asks you to confirm that you want to disable this user's ability to run X. If you type `y`, your screen will look similar to this:

```
User rws can no longer use the X Window System

Do you wish to disable an X user's ability to run X?
[y, n, ?, q]
```

5. Type y to disable another user's ability to run X, or type q to quit. If you type q, your screen will look similar to this:

```
Press <ENTER> to continue.....
```

6. Press Enter to return to the X Window System Management menu.

X Host Management

If you are running INTERACTIVE TCP/IP, X Host Management allows you to manage the list of X hosts that are allowed to connect across the net to your display server. Refer to *xhost*(1) for additional information.

To access the X Host Management menu, type `sysadm hostmgmt` or select the `hostmgmt` option from the `x` submenu under `Software`.

Your screen will look similar to this:

```
This procedure is used to list, add, and delete entries in
the X Window System '/etc/X0.hosts' file.

Type 'q' at any time to quit the current operation.
If a '?' appears as a choice, type '?' for help.

If a default appears in the question, press <ENTER> for the
default.

Enter the operation you want to perform:

        1 list
        2 add
        3 delete

[default: q]:
```

Listing Remote Hosts

To list the remote hosts that are currently allowed access to a display server:

1. Type `1` to see a listing of host entries for a display. Your screen will look similar to this:

```
Enter the number of the display [0]:
```

2. Type `0` (the default display server) or the number of the display for which you want a list. The display below shows a sample entry for display number `0`. A host named `expo.lcs.mit.edu` is listed. The display number and host name shown will be replaced with the number and list of host names for the display you choose to list. Your screen will look similar to this:

```
This is the current list of hosts allowed access to the X
Window System on display 0:
expo.lcs.mit.edu

Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

3. Type `q` to quit or `1` to list another display. If you type `q`, your screen will look similar to this:

```
Press <ENTER> to continue.....
```

4. Press Enter to return to the X Window System Management menu.

Adding Remote Hosts

To enable a remote host to have network access to your display server:

1. First, access the `hostmgmt` option from the `x` submenu under the `Software` menu. Your screen will look similar to this:

```
Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

2. Type `2` to add a host entry. Your screen will look similar to this:

```
This procedure is used to create entries in the
X Window System '/etc/X0.hosts' file.
This file contains information which X uses to determine
which remote hosts are allowed to access the display server.

Enter the number of the display [0]:
```

3. Type `0` (the default display server), or type the number of the display for which you wish to add a host entry. Your screen will look similar to this:

```
Enter the name of the host you want to add:
```

4. Type the name of the host you want to have access to this display server. The display below shows a sample entry for a host named `expo.lcs.mit.edu`. The name `expo.lcs.mit.edu` will be replaced with the name of the host you want to add. Your screen will look similar to this:

```
Here is the entry for machine 'expo.lcs.mit.edu.'

SYSTEM NAME:   expo.lcs.mit.edu

Should this be entered into the '/etc/X0.hosts' file?
[y, n, q]
```

5. The system asks you to confirm that you want to add this entry. If you type `y` to add the entry, your screen will look similar to this:

```
* host 'expo.lcs.mit.edu' has been added to '/etc/X0.hosts' *

Add another entry to the /etc/X0.hosts file? [y, n, q]
```

6. Type `y` to add another entry, or type `q` to quit adding entries for this display. If you type `q`, your screen will look similar to this:

```
Enter the operation you want to perform:

        1 list
        2 add
        3 delete

[default: q]:
```

7. Type `q` to quit, or type `2` to add a host to another display. If you type `q`, your screen will look similar to this:

```
Press <ENTER> to continue.....
```

8. Press Enter to return to the X Window System Management menu.

Deleting Remote Hosts

To remove a remote host's entry in the file `/etc/X0.hosts`, so that the host will no longer have network access to your server:

1. First, access the `hostmgmt` option of `sysadm`. Your screen will look similar to this:

```
Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

2. Type `3` to delete a host entry. Your screen will look similar to this:

```
This procedure is used to remove entries in the
X Window System '/etc/X0.hosts' file.
This file contains information which X uses to determine which
remote hosts are allowed to access the display server.

Enter the number of the display [0]:
```

3. Type `0` (the default display server) or the number of the display for which you want to delete a host entry. The display below shows a sample entry for a host named `expo.lcs.mit.edu`. The name `expo.lcs.mit.edu` will be replaced with the name of the host you want to delete. Your screen will look similar to this:

```
This is the current list of host entries known to the X Window
System:

expo.lcs.mit.edu

Which host entry do you want to delete? [q]
```

4. Type the name of the host entry you wish to delete. Your screen will look similar to this:

```
Here is the file entry for machine 'expo.lcs.mit.edu'.

SYSTEM NAME: expo.lcs.mit.edu
Do you want to delete the host entry named 'expo.lcs.mit.edu'?
[y, n, ?, q]
```

5. The system asks you to confirm that you want to delete this entry. If you type `y` to delete the entry, your screen will look similar to this:

```
* entry 'expo.lcs.mit.edu' has been deleted from
'/etc/X0.hosts' *
Any other deletions? [y, n, q]
```

6. Type `y` to delete another entry, or type `q` to quit deleting entries for this display. If you type `q`, your screen will look similar to this:

```
Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

7. Type `q` to quit or `3` to delete a host from another display. If you type `q`, your screen will look similar to this:

```
Press <ENTER> to continue.....
```

8. Press Enter to return to the X Window System Management menu.

Display Configuration

The Display Configuration menu of `sysadm` allows you to interactively create, revise, list, and delete X11 display configuration information stored in the system configuration file. You should refer to *xconfig*(1), *Xconfig*(5), and server-specific manual entries for additional configuration information. (The server-specific manual entries are those Section 1 entries in which the entry names begin with an uppercase X.)

To list, add, or delete X display configuration information, use the `sysadm` command or log in as `sysadm` and access the Main menu. Select the `configmgmt` option from the `x` submenu under `Software`.

Your screen will look similar to this:

```
This procedure is used to list, add, and delete entries in the
X Window System Utilities '/usr/lib/X11/Xconfig' file.

Type 'q' at any time to quit the current operation.
If a '?' appears as a choice, type '?' for help.

If a default appears in the question, press <ENTER> for the
default.

Enter the operation you want to perform:

        1 list
        2 add
        3 delete

[default: q]:
```

Listing a Display Configuration

To list a display configuration entry:

1. First, access the `configmgmt` menu option of `sysadm`. Your screen will look similar to this:

```
Enter the operation you want to perform:

        1 list
        2 add
        3 delete

[default: q]:
```

2. Type `1` to list the display configuration. Your screen will look similar to this:

```
Enter display number to list ('l' for list, 'q' to quit) [0]:
```

3. Type the number of a display you want to list. Your screen will look similar to this:

```
Values for display "0":
Resource Type   Info             Display Device
display  VGA    "640x480 16 11x8" 0 /dev/vt00
keyboard AT     101              0   /dev/vt00:/dev/vt%02d
mouse    LOGI-S "1200 3"         0   /dev/tty00

Enter display number to list ('l' for list, 'q' to quit) [q]:
```

4. To list another entry, type a display number. If you do not wish to list another entry, type q to quit. If you type q, your screen will look similar to this:

```
Enter the operation you want to perform:

        1 list
        2 add
        3 delete

[default: q]:
```

5. Type q to quit. Your screen will look similar to this:

```
Press <ENTER> to continue.....
```

6. Press Enter to return to the X Window System Management menu.

Adding or Modifying a Display Configuration

To add or modify a display configuration entry:

1. First, access the `configmgmt` option of `sysadm`. Your screen will look similar to this:

```
Enter the operation you want to perform:

        1 list
        2 add
        3 delete

[default: q]:
```

2. Type 2 to add a display configuration. Your screen will look similar to this:

```
Enter display number to add ('l' for list, 'q' to quit) [0]:
```

3. Type the number of the display you want to add. For example, type `1`. Your screen will look similar to this:

```
Display "1" doesn't exist, create it? [y]:
```

4. If there is no configuration information for this screen in the system configuration file, the system will ask you to confirm that you want to create it. If the configuration information already exists, you may modify any of the resources for the screen. If you type `y`, your screen will look similar to this:

```
Create display "1":
Select resource to add:
        1       display
        2       keyboard
        3       mouse
        4       tablet
(select 'l' to list, 'q' to quit)

Enter selection [display]:
```

5. The information in this section presents an alternate configuration using the same steps that you used when you initially configured your X11 on your system (see the *INTERACTIVE UNIX System Installation Guide*). The configuration shown, adding an IBM 8514/A display adapter, could be used, for example, on a virtual terminal using VT flip. Note that only one IBM 8514/A adapter can be run at a time. For additional information on virtual terminals, refer to the *INTERACTIVE UNIX System User's Guide*.

   When a menu has a default selection, that choice is shown in brackets in the line at the bottom of the screen that says `Enter selection [`*selection*`]`. If you want to select the default resource (`display`), press Enter. If you want to select a resource other than the `display` resource, you may type either the option number (`2`, `3`, or `4`) or the selection name (`keyboard`, `mouse`, or `tablet`). The X configuration procedures automatically cycle between the various resource options.

   When configuring your resources, note that your display and the information requested will vary according to the display type you select.

Note – You cannot include both a mouse and a tablet in the same configuration.

If you installed the *Display Servers I* diskette and you select `display`, your screen will look similar to this:

```
Select display type:
 1  CVC     Cornerstone Displays (and compatibles)
 2  8514    IBM 8514/A Display Adapter and compatibles
 3  HRC     Hercules monochrome display
 4  LVP     Sigma Designs LaserView
 5  V256    VGA and compatibles (256 Colors)
 6  VGA     VGA and compatibles
 7  EGA     EGA and compatibles
 8  VIKING  Moniterm Viking
(select 'q' to quit)

Enter selection [VGA]:
```

6. Type the option number or the name that corresponds to the type of display you are using. For example, to select the IBM 8514/A display adapter, type `2` or type `8514`. If you select the `8514` display type, your screen will look similar to this:

```
Select display adapter:
 1   IBM
 2   Matrox
 3   ADEX

Enter selection [IBM]:
```

7. Type the option number or the name that corresponds to the display adapter you are using. For example, to select the `IBM` adapter, type `1` or `IBM`, or press Enter to accept the default. If you select the `IBM` display adapter, your screen will look similar to this:

```
Select screen resolution:
 1   640x480
 2   1024x768

Enter selection [1024x768]:
```

8. To select the default, `1024x768`, type `2` or press Enter. Your screen will look similar to this:

```
Select number of colors:
  1   16 colors
  2   256 colors

Enter selection [256]:
```

9. To select 16 colors, type `1`. To select the default, `256` colors, type `2` or press Enter. Your screen will look similar to this:

```
Enter screen size (in inches) in the form WIDTHxHEIGHT [11x8]:
```

10. Type the screen size you will be using, or press Enter to select the default.

    Your screen will look similar to this:

```
Select resource to add:
       1      display
       2      keyboard
       3      mouse
       4      tablet
(select 'l' to list, 'q' to quit)

Enter selection [keyboard]:
```

11. The display resource has now been configured. The next resource, keyboard, has been automatically selected as the default selection. If you made a mistake in your display selection and wish to edit the display resource again, type 1 or display; otherwise press Enter to select the default option (keyboard). If you select the keyboard option, your screen will look similar to this:

```
Select keyboard type:
 1   AT    Standard PC/AT keyboard
 2   WY60  Wyse 60 terminal in scancode mode
(select 'q' to quit)

Enter selection [AT]:
```

12. Type 1 or press Enter to select the AT keyboard type. Your screen will look similar to this:

```
Enter keyboard type:
       1      101 keys (with separate arrow keys)
       2      84 keys  (escape key above number pad)

Enter selection [101]:
```

13. Type the option that corresponds to the number of keys on your AT keyboard. You can press Enter to accept the default (101). If you do not know whether you have an 84- or 101-key keyboard, you can use the following scheme to determine which type you have:
    - If your keyboard has arrow keys that are separate from the ones on the numeric keypad, then you have a 101-key keyboard.
    - If the arrow keys on your keyboard are located on the numeric keypad only, then you have an 84-key keyboard.

14. After you have made your selection, your screen will look similar to this:

```
Enter keyboard device name (from the manual entry)
[/dev/vt00:/dev/vt%02d]:
```

15. Press Enter to select the default keyboard device name unless you are using a SunRiver display.

If you are using a SunRiver® display, the keyboard device name for the first workstation is `/dev/st00:/dev/st0%d`. A second SunRiver workstation would have `/dev/st10:/dev/st1%d` specified as the device name, etc. Refer to *Xvga*(1) for additional information. For information on the SunRiver keyboard display driver and SunRiver serial ports, refer to *skd*(7) and *sasy*(7).

After selecting the keyboard device name, your screen will look similar to this:

```
Select resource to add:
        1       display
        2       keyboard
        3       mouse
        4       tablet
(select 'l' to list, 'q' to quit)

Enter selection [mouse]:
```

16. The keyboard resource has now been configured. The next resource, `mouse`, has been automatically selected as the default selection. If you made a mistake in your keyboard selection and wish to edit the keyboard resource again, type `2` or `keyboard`; otherwise press Enter to select the default option (`mouse`). (If you will be using a tablet instead of a mouse, type `tablet` and skip to step 21 to add the tablet resource to your configuration.) If you select the `mouse` option, your screen will look similar to this:

```
Select mouse type:
        1       LOGI-S      Logitech Serial Mouse
        2       LOGI-B      Logitech Bus Mouse
        3       MS-S        Microsoft Serial Mouse
        4       MS-B        Microsoft Bus Mouse
        5       MSC-S       Mouse Systems Serial Mouse
        6       MSC-B       Mouse Systems Bus Mouse
        7       OMNI        MSC OmniMouse
        8       PS/2        IBM PS/2 on-board mouse
        9       COMPAQ      Compaq on-board mouse
(select 'q' to quit)

Enter selection [LOGI-S]:
```

17. Type the option number or the name of the mouse you plan to use, or press Enter to select the default (`LOGI-S`). If you select the LOGITECH Serial Mouse, your screen will look similar to this:

```
Select Logitech Mouse baud rate:
        1       1200 baud
        2       2400 baud
        3       4800 baud
        4       9600 baud

Enter selection [1200]:
```

18. Type the option number or the baud rate you plan to use, or press Enter to select the default (`1200`). Your screen will look similar to this:

```
Select number of buttons on mouse:
        1       1 button
        2       2 buttons
        3       3 buttons

Enter selection [3]:
```

19. Type the option number that corresponds to the number of buttons on your mouse, or press Enter to select the default (`3`). Your screen will look similar to this:

```
Enter mouse device name [/dev/tty00]:
```

20. Type the mouse device name, or press Enter to select the default (`/dev/tty00`) (serial port 0, that is, `COM1`).

    To select serial port 1, that is, `COM2`, enter `/dev/tty01` as the mouse device name. (If you are using the SunRiver mouse devices, the names are `/dev/ser00` for the first workstation, `/dev/ser10` for the second workstation, etc. For information on SunRiver serial ports, refer to *sasy*(7).)

Your screen will look similar to this:

```
Select resource to add:
       1      display
       2      keyboard
       3      mouse
       4      tablet
(select 'l' to list, 'q' to quit)

Enter selection [tablet]:
```

21. The mouse resource has now been configured. The next resource, `tablet`, has been automatically selected as the default selection. If you made a mistake in your mouse selection and wish to edit the mouse resource again, type `3` or `mouse`; otherwise press Enter to select the default option (`tablet`). If you select the `tablet` option, your screen will look similar to this:

```
Select tablet type:
    1    SUMMA    Summagraphics SummaSketch Plus tablet
(select 'q' to quit)

Enter selection [SUMMA]:
```

22. Type `1` or press Enter to select the Summagraphics® SummaSketch® Plus tablet or a compatible. You will then see a series of displays for configuring the tablet you have chosen. First, you will be asked to select the tablet model and its orientation. Your screen will look similar to this:

```
Select tablet model and orientation:
  1  MM961 (6x9" tablet) oriented vertically
  2  MM961 (6x9" tablet) oriented horizontally
  3  MM1201 (12x12" tablet) [in any position]

Enter selection [3]:
```

23. Type the number that corresponds to your selection, or press Enter to accept the default.

You will then be asked to select the type of pointer you will be using. Your screen will look similar to this:

```
Select type of pointer installed:
  1  4-button cursor (puck)
  2  2-button stylus (pen)

Enter selection [1]:
```

24. Type the number that corresponds to your selection, or press Enter to accept the default. You will then be asked to select the input mode. Your screen will look similar to this:

```
Select input mode:
  1  absolute mode (conventional tablet)
  2  relative mode (like a mouse)

Enter selection [1]:
```

25. Type the number that corresponds to your selection, or press Enter to accept the default. You will then be asked to select the lines of resolution. Your screen will look similar to this:

```
Select lines of resolution:
  1  1000 lines per inch
  2  500  lpi
  3  400  lpi
  4  200  lpi
  5  100  lpi
  6  40 lines per mm (1016 lpi)
  7  20 lpmm (508 lpi)
  8  10 lpmm (254 lpi)

Enter selection [1000]:
```

26. Type the number that corresponds to your selection, or press Enter to accept the default. You will then be asked to select the report rate divisor. Your screen will look similar to this:

```
Select report rate divisor:
  1  maximum throughput (fast systems only)
  2  throughput / 2
  3  throughput / 8
  4  throughput / 32

Enter selection [3]:
```

27. Type the number that corresponds to your selection, or press Enter to accept the default. You will then be asked to enter the tablet device name. Your screen will look similar to this:

```
Enter tablet device name [/dev/tty00]:
```

28. You have now configured all of the resources for the display. Press Enter to see a listing of all of the options you have selected. Here is a sample listing that assumes you have configured a mouse instead of a tablet; your screen will look similar to this:

```
Display "1" is now:
Resource Type   Info            Display Device
display  8514   "IBM 43 1024x768 256 11X8" 1 /dev/vt00
keyboard AT     101             1      /dev/vt00:/dev/vt%02d
mouse    LOGI-S "1200 3"        1     /dev/tty00

Save these changes? [y]:
```

29. Press Enter to save the configuration you have selected. Your screen will look similar to this:

```
Enter display number ('l' for list, 'q' to quit) [q]:
```

30. To add another entry, type a display number; or type `q` to quit. If you type `q`, your screen will look similar to this:

```
Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

31. Type `q` to quit. Your screen will look similar to this:

```
Press <ENTER> to continue.....
```

32. Press Enter to return to the X Window System Management menu.

Deleting a Display Configuration

To delete a display configuration entry:

1. First, access the `configmgmt` option of `sysadm`. Your screen will look similar to this:

```
Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

2. Type `3` to delete a display configuration. Your screen will look similar to this:

```
Enter display number to delete ('l' for list, 'q' to quit) [0]:
```

3. Type the number of the display you want to delete. If you type 1, your screen will look similar to this:

```
Values for display "1":
Resource Type    Info              Display Device
display  8514    "IBM 43 1024x768 256 11X8" 1 /dev/vt00
keyboard AT      101               1     /dev/vt00:/dev/vt%02d
mouse    LOGI-S "1200 3"           1    /dev/tty00

Delete display "1"? [y]:
```

4. The system asks you to confirm that you want to delete this entry. Press Enter or type y to delete the entry; type n if you do not want to delete this entry. If you type y, your screen will look similar to this:

```
Display "1" has been deleted.
Enter display number to delete ('l' for list, 'q' to quit) [q]:
```

5. To delete another entry, type a display number; or type q to quit. If you type q, your screen will look similar to this:

```
Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

6. Type q to quit. Your screen will look similar to this:

```
Press <ENTER> to continue.....
```

7. Press Enter to return to the X Window System Management menu.

Server Management

To list, add, or delete an X server, use the sysadm command or log in as sysadm and access the Main menu. Select Software, then x, then the servermgmt option.

Your screen will look similar to this:

```
This procedure is used to list, add, and delete X11 servers
for use.

Type 'q' at any time to quit the current operation.
If a '?' appears as a choice, type '?' for help.

If a default appears in the question, press <ENTER> for the
default.

Enter the operation you want to perform:

        1 list
        2 add
        3 delete

[default: q]:
```

Listing Servers

To see a listing of the servers you have created on your system:

1. First, access the servermgmt option of sysadm. Your screen will look similar to this:

```
Enter the operation you want to perform:

        1 list
        2 add
        3 delete

[default: q]:
```

2. Type 1 to see a list of servers you have created. Your screen will look similar to this:

```
This procedure is used to list X11 servers.

The following is a list of X11 servers:

Name     built Description
----     ----- -----------
Xcvc           Server for Cornerstone displays
Xgp            Server for IBM 8514/A and compatible displays
Xhrc           Server for Hercules Monochrome Graphics
                 displays
Xlvp           Server for Sigma Designs LaserView Plus
                 displays
Xv256          Server for 256 color VGA and compatible
                 displays
Xvga     yes   Server for VGA/EGA and compatible displays
Xviking        Server for Moniterm Viking 21/91 Display

Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

3. Type q to quit. Your screen will look similar to this:

```
Press <ENTER> to continue.....
```

4. Press Enter to return to the X Window System Management menu.

Adding Servers

To add a new server to your system:

1. First, access the `servermgmt` option of `sysadm`. Your screen will look similar to this:

```
Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

2. Type `2` to add a new server to your system. Your screen will look similar to this:

```
This procedure is used to create new X11 servers. A new X11
server must be created for each type of display.

The following is a list of the possible X11 servers to build:

Name     built Description
----     ----- -----------
Xcvc           Server for Cornerstone displays
Xgp            Server for IBM 8514/A and compatible displays
Xhrc           Server for Hercules Monochrome Graphics
                 displays
Xlvp           Server for Sigma Designs LaserView Plus
                 displays
Xv256          Server for 256 color VGA and compatible
                 displays
Xvga     yes   Server for VGA/EGA and compatible displays
Xviking        Server for Moniterm Viking 21/91 Display

Enter the name of the server you want to create or 'q' to quit
[q]:
```

3. Type the name of the server you want to create. The display below shows a sample entry for a server named `Xgp`. Replace `Xgp` with the name of the server you want to create. Your screen will look similar to this:

```
Should server 'Xgp' be created? [y, n, q]
```

4. The system asks you to confirm that you want to create this server. If you type `y`, your screen will look like this:

```
 **** server 'Xgp' has been created ****

Create another X11 server? [y, n, q]
```

5. Type `y` to create another server, or type `q` to quit. If you type `q`, your screen will look similar to this:

```
Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

6. Type `q` to quit. Your screen will look similar to this:

```
Press <ENTER> to continue.....
```

7. Press Enter to return to the X Window System Management menu.

Deleting Servers

To delete a server from your system:

1. First, access the `servermgmt` option of `sysadm`. Your screen will look similar to this:

```
Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

2. Type `3` to delete a server that was previously created from your system. Your screen will look similar to this:

```
This procedure is used to delete X11 servers.


The following is a list of the possible X11 servers to delete:

Xgp     Server for IBM 8514/A and compatible displays
Xvga    Server for VGA/EGA and compatible displays

Enter the name of the server you want to delete:
```

3. Type the name of the server you want to delete. The display below shows a sample entry for a server named `Xgp`. Replace `Xgp` with the name of the server you want to delete. Your screen will look similar to this:

```
Should server "Xgp" be deleted? [y, n, q]
```

4. The system asks you to confirm that you want to delete this server. If you type y, your screen will look like this:

```
 **** server 'Xgp' has been deleted ****

Delete another X11 server? [y, n, q]
```

5. Type y to delete another server, or type q to quit. If you type q, your screen will look similar to this:

```
Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

6. Type q to quit. Your screen will look similar to this:

```
Press <ENTER> to continue.....
```

7. Press Enter to return to the X Window System Management menu.

Configuring X11 Drivers

Some high-resolution displays require UNIX System kernel drivers in order to operate in high-resolution modes. `xdrivermgmt` is the `sysadm` interface used to configure these drivers into the kernel. Configuration is limited to turning the driver on or off in the kernel. In order to change specification settings, such as I/O address, DMA, and shared memory address, you must edit the appropriate file in `/etc/conf/sdevice.d` and edit the `/etc/conf/cf.d/mdevice` file. Consult your hardware manual and the appropriate INTERACTIVE X11 manual entries for the correct values for these settings. When a driver is configured, checking will be done automatically to ensure that conflicts in the above parameters do not occur. If an error does occur, you will be notified of the error but you will still be permitted to configure the driver. Before the kernel can be built, however, these conflicts must be resolved.

It is easier to list, add, and delete driver entries using `sysadm`, but the `kconfig` program can also be used. For information on `kconfig`, refer to Chapter 5, "Using kconfig to Tailor Your System Kernel."

Using `sysadm` *to Configure X11 Drivers*

Use the `sysadm` command to list, add, or delete X display server configuration information.

Adding X11 Drivers to the Configuration Using `sysadm`

1. On the system console, use the `sysadm` command or log in as `sysadm` and access the Main menu. Select the `xdrivermgmt` menu option from the `x` submenu under `Software`.

   Your screen will look similar to this:

```
This procedure is used to list, add, and delete entries in the
kernel configuration.

Type 'q' at any time to quit the current operation.
If a '?' appears as a choice, type '?' for help.

If a default appears in the question, press <ENTER> for the
default.

Enter the operation you want to perform:

        1 list
        2 add
        3 delete

[default: q]:
```

2. Type `2` to add an X11 kernel driver. Your screen will look similar to this:

```
 1      Cornerstone Memory Mapping Driver
 2      Logitech Bus Mouse Driver
 3      Microsoft Bus Mouse Driver
 4      Built-in Mouse Driver

Which kernel driver would you like to add? [q]
```

Note – If you are performing a new installation (*not* overwriting the previous X11 release) and you are running a type of mouse other than serial, you will need to install the appropriate mouse driver from the Additional Drivers subset.

---

3. Type the number of the kernel driver you would like to add. Type `1` to add the `Cornerstone Memory Mapping Driver`. Your screen will look similar to this:

```
Checking for interrupt conflicts
Checking for DMA channel conflicts
Checking for shared memory address conflicts

The kernel must be rebuilt in order for the drivers that
have been configured to take effect. Would you like to build
a kernel at this time?  [y, n]
```

4. Type `y` to build a kernel at this time. Type `n` if you plan to add additional kernel drivers or remove kernel drivers from the configuration. If you type `y`, your screen will look similar to this:

```
The following X device drivers are configured in the kernel:
       Cornerstone Memory Mapping Driver
```

5. The `Cornerstone` driver has been added to the kernel configuration.

Removing X11 Drivers From the Configuration Using `sysadm`

1. Use the `sysadm` command or log in as `sysadm` and access the Main menu. Select `Software`, then `x`, then the `xdrivermgmt` menu option. Your screen will look similar to this:

```
This procedure is used to list, add, and delete entries in the
kernel configuration.

Type 'q' at any time to quit the current operation.
If a '?' appears as a choice, type '?' for help.

If a default appears in the question, press <ENTER> for the
default.

Enter the operation you want to perform:

       1 list
       2 add
       3 delete

[default: q]:
```

2. Type `3` to remove an X11 kernel driver. Your screen will look similar to this:

```
 1      Cornerstone Memory Mapping Driver

Which kernel driver would you like to remove?
```

3. Type the number of the kernel driver you would like to remove from the configuration. For example, to remove the `Cornerstone Memory Mapping Driver`, you would type `1`. Your screen will look similar to this:

```
The kernel must be rebuilt in order for the drivers that
have been configured to take effect. Would you like to build
a kernel at this time? [y, n]
```

4. Type `y` to build a kernel at this time. Type `n` if you plan to add additional kernel drivers or remove kernel drivers from the configuration. If you type `y`, your screen will look similar to this:

```
The following X device drivers have been removed from kernel
     Cornerstone Memory Mapping Driver

Enter the operation you want to perform:

        1 list
        2 add
        3 delete

[default: q]:
```

5. The `Cornerstone Memory Mapping Driver` has been removed from the kernel configuration.

Setting Up Your Printer

A

A System V lp printer is added to the system with the `sysadm lp Printer Management` option of the `Machine` menu. Printers can be either serial or parallel. A serial printer must be plugged into a serial port, such as `tty00` or `tty01` (`COM1` or `COM2` under DOS); a parallel printer must be plugged into a parallel port, such as `lp0` or `lp1` (`LPT1` or `LPT2` under DOS).

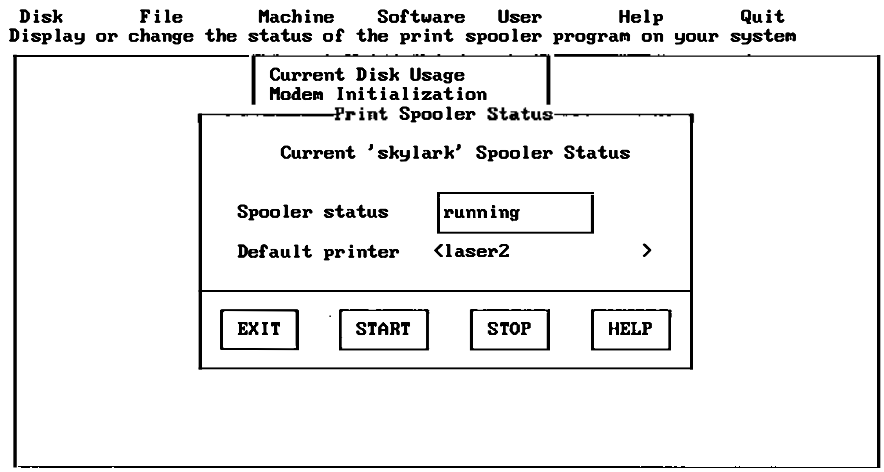
The system supports a set of default printer types. However, the printers shown in this chapter are only examples. Your system may be configured to support additional or different printers. Before you attempt to install a new printer, refer to the documentation provided by your vendor or to your hardware manufacturer's instructions.

You can also refer to Chapter 10, "The Berkeley Line Printer Spooler (lpr) System," for more information about the line printer driver.

Starting and Stopping the Print Spooler

The print *spooler* is the daemon that provides system-wide control of all jobs queued to all printers on your system. The print spooler should only be turned off when the entire printer system is to be shut down. In this case, you should turn off the spooler and flush the printer queues before shutting your system down. Initially, the print spooler is off; it should be turned on before adding a printer to your system. To stop or start the print spooler, follow this procedure:

1. Select the `Spooler Management` option of the `lp Printer Management` menu. Your screen will look similar to this:



2. The `Print Spooler Status` form contains these fields:

   `Spooler status:`

   This field displays the current status of the print spooler, `running` or `not running`. If the spooler is running, jobs will be queued to the various printers on your system. If it is not running, the system will not accept or queue jobs to any printer on the system.

   To start the print spooler, select the `START` button at the bottom of the form. To stop the print spooler, select the `STOP` button. Your screen will be updated to reflect the current state of the spooler.

   `Default printer:`

   Access the options in this field to change the default printer for your system, that is, the one used by the print spooler when no printer is specified in a command. For example, the command `lp` *filename* would cause the print spooler to send the job to the default printer, in this case `laser`, since no printer is specified in the command.

3. Select the `EXIT` button to return to the `lp Printer Management` menu.

Adding a New Printer

To add a new printer to your system, follow this procedure:

1. Plug the printer into the appropriate serial or parallel port on your system. It may be located on your main board or on an adapter board. Refer to the instructions provided by your hardware manufacturer to determine the correct connection.

2. From the `Machine` menu select `lp Printer Management`, then select `Add a Printer`. Your screen will look similar to this:

```
 Disk      File      Machine    Software   User      Help      Quit
Add a printer to your system
                          Configure a New Printer
              LP Print Service - Printer Parameters

   Printer name:                     Class name:      <none          >

   Alert type:    <mail     >        Fault handling:  <wait        >
   Model type:
   <standard (standard printer interface program)                  >

   Type of hardware interface:        <parallel>

   Device name:   </dev/null >       Force Banner?    <yes>

  DESCRIPTION:

                 OK              CANCEL              HELP
```

3. The `Configure a New Printer` form contains these fields:

   `Printer name:`
   : Type the name of the printer. For example, type `laser1`. You cannot skip this field.

   `Class name:`
   : A class name can be used to group printers in any way you prefer. You can use a class name instead of a printer name when you send a job to a printer. For example, you might specify `lasers` as the class name for all laser printers on your system. You can then specify the

class name instead of a specific printer name when you send your job. The print spooler will attempt to print the job on the first printer in the `lasers` class. If this printer is busy, the job will be sent to each printer in the class, in turn, until one is found that is available to print the job.

`Alert type:`
This field allows you to specify how the system should notify you when a problem is detected with the printer. The system default is `mail`. This means that a mail message will be sent when a problem is detected. If you want to specify a different method of notification, pop up the list of alternative options, which are `write`, `no alert`, and `custom`. The `write` option will send a message directly to your screen when a printer problem is detected. The `no alert` option will prevent the sending of any alert messages. The `custom` option is provided so that you can specify to whom alert messages should be sent. By default, messages are sent to `root`. If you want to specify other users instead of or in addition to this user, use the `custom` option.

`Fault handling:`
This field allows you to specify what the printer should do once a problem has been taken care of. The default is `wait`, which means that the printer will wait for further instructions once a problem has been cleared. To specify a different type of fault handling, access the options `continue` and `restart`. The `continue` option directs the printer to continue printing the current job from the point where it was stopped. This option is only useful if your interface can detect the current position of the job. The `restart` option directs the printer to start printing the job from the beginning once the problem is cleared.

`Model type:`
In this field, provide the name of the interface program. The default standard program should work with most printers. However, if you want to specify another interface program, pop up the list in this field and select the appropriate interface for your system. The last option on the list is `other`, which allows you to specify an interface program that is not already listed.

`Type of hardware interface:`

In this field, specify the type of hardware interface. The default is `parallel`, but you can toggle the spacebar in this field if you want to specify `serial`. Depending on which option you select, you will need to indicate some specifics such as the *baud rate* for your type of hardware interface.

`Device name:`

In this field, specify the appropriate device name for your interface. The system default is `/dev/null`. To specify a different device name, access the list in this field and select an appropriate name. If the device name you want is not listed, select the `other` option and type the name in the box provided.

The list of device names that appears when you access the menu in this field is constructed according to whether you select `serial` or `parallel` in the `Type of hardware interface:` field. If you select `serial`, the list will look similar to this:

```
/dev/tty00
/dev/tty01
other
```

`/dev/tty00` corresponds to the first serial port, and `/dev/tty01` corresponds to the second serial port. If you select `parallel`, the list will look similar to this:

```
/dev/null
/dev/lp
other
```

`/dev/null` is the system default, and `/dev/lp` is the first parallel port on the system.

Note that in order to make the entries on these lists available for use, the appropriate hardware must be present.

`Force Banner?`

This field allows you to specify on a system-wide basis that a banner page should be printed with each job. The default setting in this field is `yes`. A `yes` in this field, overrides an individual user's command and prints a banner page for every job. If you select `no`, an individual user can specify on the command line that a banner page should be printed for a particular job.

`DESCRIPTION:`
You may use this field to add a line of text describing the printer, since printer names are sometimes cryptic.

4. Now that you have provided the system with the information necessary to add the printer, you may install the entry by selecting the `OK` button at the bottom of the form. If you want to delete the entry, select `CANCEL`. You will be returned to the `lp Printer Management` menu.

5. Test your new printer by sending a file to it using the `lp` command.

Instructing a Printer to Accept or Reject Print Jobs

At times you will want to instruct a printer to stop accepting jobs, for example, while you perform routine maintenance. At other times, you may want a printer that you have stopped to begin accepting jobs again. The `Change Printer Status` option of the `lp Printer Management` menu provides a way to implement either of these changes. This option also allows you to take a printer off-line temporarily while continuing to queue jobs for it.

1. From the `lp Printer Management` menu, select the `Change Printer Status` option. Your screen will look similar to this:

```
 Disk       File       Machine    Software   User       Help       Quit
Display or change the status of printers on your system

          Display/Change System Printer Status

            LP Print Service - Current Printer Status

   Printer name:   <              >       Class: [          ]

                                  if not, reason:
   Accepting jobs: <   >          [                         ]

                                  if not, reason:
   Print enabled:  <   >          [                         ]

             [ OK ]         [ CANCEL ]         [ HELP ]
```

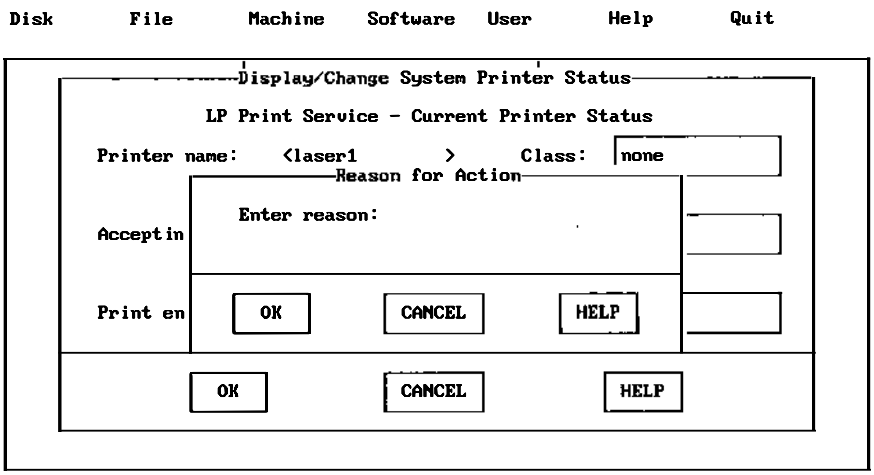
The `Display/Change System Printer Status` form contains these fields:

`Printer name:`
: Press the spacebar to pop up the list of available printers, then select the appropriate printer.

`Class:`
: The system automatically supplies the name of the class the printer belongs to. If the printer is not part of a class, the word `none` will appear in this box.

`Accepting jobs:`
: This field allows you to make a printer (or a class of printers) temporarily unavailable by turning off the queue to the printer(s). Access the list of choices in this field and select `no` or `yes`. If you select `no`, jobs will no longer be queued to the printer. In this case, your screen will display a form in which you can provide an explanation of why the printer is not accepting jobs. This

explanation will be displayed on the screen when a user tries to send a job to this printer. The form displayed when you select `no` will look similar to this:



Type your explanation, for example, `Routine Maintenance`, then select the `OK` button at the bottom of the form. If you select `yes`, jobs will once again be queued to the printer.

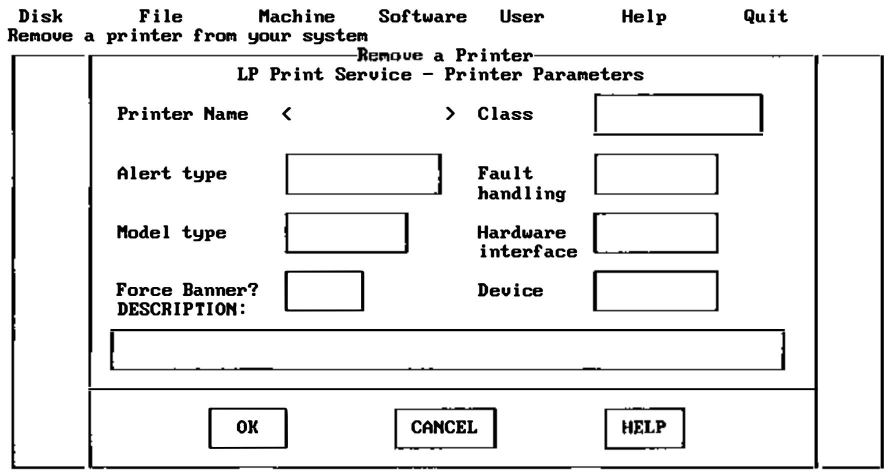
`Print enabled:`

This field allows you to take a printer (or a class of printers) off-line temporarily while leaving the queue on. Select `no` or `yes` from the list in this field. If you select `no`, the printer will be disabled but jobs will continue to be queued to it. In this case, your screen will display a form in which you can provide an explanation of why the printer has been disabled. The explanation will be displayed on the screen when a user tries to send a job to this printer. If you select `yes`, the printer will be put back on-line and will begin to print all jobs waiting in its queue.

Removing a Printer From the System

If you want to remove a printer from your system, follow this procedure:

1. Select the `Delete a Printer` option of the `lp Printer Management` menu. Your screen will look similar to this:



2. Access the list in the `Printer Name` field and select the printer you want to remove from the system, for example, `laser1`. The system displays the printer's configuration, which was set with the `Add a Printer` option. Your screen will look similar to this:

```
 Disk      File      Machine    Software   User      Help      Quit
Remove a printer from your system
                       Remove a Printer
              LP Print Service - Printer Parameters

  Printer Name   <laser1        >  Class       [lasers     ]

  Alert type     [mail      ]      Fault       [wait   ]
                                   handling

  Model type     [standard]        Hardware    [parallel]
                                   interface

  Force Banner?  [yes  ]           Device      [/dev/lp ]
  DESCRIPTION:
  [Sales department laser printer                          ]

            [ OK ]          [ CANCEL ]          [ HELP ]
```

3. To remove the printer, select the `OK` button at the bottom of the form.

Troubleshooting B

This chapter describes some of the most frequently asked questions that arise during the installation and configuration of the INTERACTIVE UNIX Operating System. These may help you to troubleshoot problems with your system.

- **What does the error message** `Interrupt conflict between wt and gendev` **mean when building a kernel?**

  Generally this means that there is a conflict between two (or more) drivers in their interrupt level. The most common cause for this is that you have not used the `kconfig` utility to previously configure the High Performance Device Driver. The default HPDD is delivered with four interrupts configured (5, 11, 14, 15). If you configure the HPDD, you will generally free up the interrupts needed by other devices.

  Refer to the driver conflict information found in Chapter 6, "Hardware Compatibility and Configuration," for the appropriate drivers for your devices.

- **What does the error message** `getty respawning too rapidly` **mean?**

  There are two common causes for this:

  - **1a)** A serial terminal may be configured with the incorrect baud rate, or it may be improperly wired. Refer to the documentation that accompanied your hardware to verify the correct baud rate and proper wiring.

1b) The second asy port may not be configured properly into the kernel. Refer to Chapter 6, "Hardware Compatibility and Configuration," to verify the proper configuration of the second asy device, and use the kconfig program to verify that the second asy device is configured. To do this, select Drivers under Configure on the bar menu. Select the asy driver and access the Driver Configuration Data form. Select asy2 from the list of asy drivers to review its configuration. If the Driver Status button has an asterisk in the On field, then it is configured into the current kernel.

- **2)** A modem may be configured incorrectly, or you may be using a device name that does not support modem control. The modem should be in "quiet" mode, that is, its echo and verbose modes should be off. The modem should use a device that uses modem control, such as /dev/ttyd0 or /dev/ttyd3, rather than /dev/tty00 or /dev/tty03, which do not.

- **My printer only gives me blank pages.**

  This is most probably caused by the printer not interpreting the NL character that the INTERACTIVE UNIX System provides for new lines. The DOS environment uses a combination of carriage return and line-feed (CR/LF).

  - If you use the System V lp print system, this problem can be solved by using the INTERACTIVE UNIX System utod program, which translates the UNIX System NL characters to CR/LF and passes command sequences without translation. Replace the word cat in the file in /usr/spool/lp/model/* (your printer mode) with utod.
  - If you use the BSD lpr print system, edit /etc/printcap and add OPOST ONLCR to the tt= string.

- **My printer does not print at all.**

  First, check the physical connections to the printer. Assuming the parallel port is used, determine the proper INTERACTIVE UNIX System device for the printer by typing:

```
$ cat /etc/passwd > /dev/lp1
```

If this is the correct device, a copy of `/etc/passwd` will print on your printer. Although `/dev/lp1` is correct on most machines, you may have to try `/dev/lp0` or `/dev/lp2` if `/dev/lp1` does not work. When you have determined the proper device, reinstall your printer using the `sysadm lpmgmt` menu.

- **My parallel printer prints very slowly, like one line every several seconds.**

  The kernel may be configured for the wrong interrupt (IRQ) value. By default, the INTERACTIVE UNIX System uses IRQ 7 for the parallel port. However, a number of LPT ports use IRQ 5 instead. Use `kconfig` to change the interrupt request line for the `lp` driver to 5, then build a new kernel, install it, and reboot the system.

- **How do I change my `ULIMIT`?**

  The preferred method is to change the `ULIMIT` within the kernel. Access the `kconfig` program and select `Kernel` on the `Parameters` menu under `Configure` on the bar menu. When asked for the value, change the default value that is displayed to the value desired (remembering that 1024 = approximately 1 MB). After you have rebuilt and installed the kernel and rebooted the system, the new `ULIMIT` will be in effect.

  You can also change the `ULIMIT` by changing `/etc/default/login`, but this is discouraged because it doesn't affect all processes on your system. To use this method, edit the file `/etc/default/login` and change the `ULIMIT` entry to the desired number. Remember that the entry is in 512-byte blocks, so an entry of 4096 indicates that the largest file size will be about 2 MB.

- **How do I verify that my mouse is working?**

  The easiest way to do this is to type:

```
$ cat -v < /dev/tty00
```

for a serial mouse using `/dev/tty00`, or to type:

```
$ cat -v < /dev/mouse
```

for a bus mouse using /dev/mouse. The bus mouse simply returns the prompt. If you have a serial mouse, move the mouse and see if something appears on the screen. If the prompt does not return (bus mouse) or nothing appears (serial mouse), check your configuration against the appropriate mouse driver configuration in Chapter 6, "Hardware Compatibility and Configuration."

- **I keep building kernels but no change occurs.**

  Make sure that there is no kernel (a file named unix) in the /etc/conf/cf.d directory. Kernels present in this directory are automatically linked to /unix at shutdown time. If a unix file exists in this directory, remove or rename it.

- **How do I mount the DOS partition of my fixed disk or a DOS diskette on my INTERACTIVE UNIX System?**

  Use the following command for a partition:

```
# mount -f DOS /dev/dsk/c0t0p? /dir
```

  Replace the *?* with the number of the fdisk partition on which DOS resides.

  Use the following command for a diskette. Refer to the *INTERACTIVE UNIX System User's Guide* for diskette naming conventions.

```
# mount -f DOS /dev/dsk/f0q15dt /mnt
```

  Note that this device name mounts a 1.2 MB 5.25-inch diskette.

  You can also have the system automatically mount DOS partitions at system startup by adding the following line to /etc/fstab:

```
/dir /dev/dsk/c0t0p?
```

  Automatic mounting of diskettes is discouraged, as the system will display error messages if the diskette is not in the diskette drive. Refer to the *INTERACTIVE UNIX System User's Guide* for more information about accessing DOS files on the INTERACTIVE UNIX System.

- **I modified the** `/etc/inittab` **file and built a new kernel, but my changes did not remain.**

  You must change both of the files `/etc/conf/node.d/`*filename* and `/etc/conf/cf.d/init.base` (or `/etc/conf/init.d/`*driver*, such as `asy`) to make permanent changes in the `inittab` file.

- **I can't install my system. All I get is the** `Booting the INTERACTIVE UNIX Operating System` **message and then my disk light goes out.**

  This may be caused by hardware that is not being recognized correctly. Simplify the hardware environment by removing unnecessary adapters, and verify that the INTERACTIVE UNIX Operating System is supported on your hardware. This problem may also be caused by bad media. If your *Boot/Install* diskette cannot be used on another system, contact your distributor for a replacement.

- **I've built a kernel that won't boot, and I can't find** `OLD.unix` **to boot an old kernel. Now what can I do?**

  This problem can be solved by mounting the fixed disk partition on the boot disk and copying the kernel found on the *Boot/Install* diskette to the INTERACTIVE UNIX System partition on the fixed disk. See "Recovery of the INTERACTIVE UNIX System" in Chapter 4, "File System Maintenance," for details.

Kernel Error Messages

C

The INTERACTIVE UNIX System kernel displays an error message on the system console when it encounters an internal error condition. If you are using virtual terminals, you will not see an error message unless you are switched to the console virtual terminal.

Kernel error messages are divided into three severity classes: PANIC, WARNING, and NOTICE. The severity class of a message is displayed at the beginning of the message.

Panic Messages

The kernel displays a PANIC error message when it encounters a problem that is so severe that it cannot continue to function and must stop. This is usually caused by a hardware problem, an incorrectly configured device, or a logic error in a device driver that has been configured into the kernel. It is only occasionally caused by a logic error in the kernel itself.

Common PANIC Error Messages

The following are the most common PANIC error messages:

```
Kernel mode trap. Type 0xD (general protection exception)

Kernel mode trap. Type 0xE (page fault)

User mode trap. Type 0xD

User mode trap. Type 0xE
```

PANIC error messages can have many causes. If your system displays a PANIC message after having run without problems for some time, it is likely that there has simply been a random hardware error that will not happen again soon. Turn off the power, wait 30 seconds, and turn the power back on. The system should check the disk file systems, correct any problems it finds, and resume normal operation. If the system resumes operation without problems and does not produce the original error message again, then your system is probably fine. On the other hand, if the previous error message or some other PANIC error message appears again, your system has a problem that must be corrected.

If you get the message `Kernel mode trap. Type 0xD (general protection exception)`, look in `/usr/include/sys/trap.h` for a description of trap-type values.

If you have reconfigured the kernel recently, you probably have a configuration or driver problem. If you have not reconfigured the kernel recently, you probably have a hardware problem. If you have added a device driver not supplied by SunSoft, there may be an error in the driver software logic.

Hardware Parity Errors

The following messages indicate a hardware parity error:

```
Parity error address unknown

Parity error at address 0x

FATAL: Parity error on an add-on card
```

The most common causes are:

- RAM memory failure has occurred. This may be a random error that will not occur again for a long time, or it may have been caused by a permanent memory chip failure. Turn the power off and on to reboot the system, and if the problem recurs, have the hardware repaired.
- A device configuration conflict has occurred. This will occur if, for example, two pieces of hardware are attempting to share the same memory address.
- A conflict in DMA addressing has occurred. This can be seen when some display cards conflict with the diskette controller. The symptom is that a `cpio` out to the diskette fails with a parity error panic message.

Warning Messages

The kernel displays a WARNING error message when it encounters a problem that is not serious enough to require stopping the system suddenly. The system administrator should correct the cause of the problem. The following are the most common WARNING messages:

`iget - inode table overflow`
: This indicates that the number of entries in the inode table is inadequate. Use `kconfig` to increase the `NINODE` tunable parameter and then build and install a new kernel.

`Swap space running out. Needed` *x* `pages`
: This indicates that the maximum total process storage space has been exceeded. You can either:

  - Add RAM memory. This will reduce the need for swapping and improve system performance.

- Add swap space to the system. This generally requires reinstalling the system with more swap space allocated. This rather drastic step can be avoided if there is unallocated disk space. Use the command `swap -a` to add another swap device.

`Cannot load floating point emulator`
: This indicates that there was trouble loading the floating point emulator. The file `/etc/emulator` may be corrupted or missing.

Notice Messages

NOTICE messages provide information about the status of the system. In some cases this information can be used avoid problems. Some commonly encountered NOTICE messages include:

`File table overflow`
: The number of entries in the file table is inadequate. Use `kconfig` to increase the `NFILE` tunable parameter and build and install a new kernel.

`Unexpected NMI in system mode!`
`Unexpected NMI in user mode!`
: This message appears when a Non-Maskable Interrupt (NMI) occurs. This message may be an indication that the hardware is failing.

`getcpages - waiting for` *x* `contiguous pages`
: If *x* is a very large number, this usually indicates an application software error. If it is a small number (less than 100,000), the kernel is not able to allocate enough contiguous pages of memory and is waiting until it is able to get them. This occurs when the system does not have enough RAM to run the processes requested. This is most common while running INTERACTIVE X11 on a system with only 4 MB of memory, although depending on the number of clients, it may happen on any system configuration.

Advanced Internationalization Features

D

This chapter defines and outlines the collation sequence, numeric and monetary formatting, and yes/no response category capabilities for C programmers who want to develop internationalized applications. You must have the optional International Supplement subset installed to use these features. See Chapter 12, "Setting Up an International Environment," for more general information and for how to set up an international environment on an INTERACTIVE UNIX System.

Preparing and Installing a Collation Sequence

A collation sequence specifies how characters and collating elements should be sorted, that is, the order between characters and collating elements. Collation sequences are created using the `colldef` processor (refer to *colldef*(1P) for more information). This section describes how to set up a source collation sequence definition and use it to create a collation sequence. Once the source definition is created and tested, you can use it to create "object" collation sequences, which are stored in a file named `LC_COLLATE` in a `locale` directory.

When to Use a Collation Sequence

A created and installed collation sequence definition is not activated until the user specifies that it should be used. To do this, set the `LC_ALL`, `LC_COLLATE`, or `LANG` environment variable to the directory in which the files are stored.

This must be done before a program using the stored definitions is executed. Note that the program must be set up to check and set the international environment (via the `setlocale` function).

User-defined collation is supported through the `colldef` utility and the library functions `strxfrm` and `strcoll` (refer to *strxfrm*(3P) and *strcoll*(3P) for more information). These functions are used to compare strings based on the defined collation order and rules. Traditional programs that need to do sorting use `strcmp`, which does byte-to-byte comparison. In the INTERACTIVE UNIX Operating System, the standard utilities that depend on collation, such as `sort` and `ls`, have been modified to use the international environment (refer to *string*(3P) for more information).

Defining Collation

Collation, according to a dictionary, is the "act of putting things in their proper order." Collation rules define how the data are put in the proper order, or *sorted*. Traditionally, the collating order in the UNIX System has been ASCII order, that is, the order in which the characters appear in the ASCII codeset. This is also the natural collating order for the English language.

For most languages in the world, however, this is not enough. Most European languages contain more letters than the 26 in the English language, with the additional letters typically collating between the letters in the ASCII set. For example, an `á` sorts between `a` and `b`. The European user expects sorted lists (for instance, the output from the `ls` command) to appear in the collation order of his or her language.

The INTERACTIVE UNIX Operating System provides users with the ability to define their own collation order. This capability is a superset of the X/Open™ requirement for an internationalized system, and it is expected to satisfy the requirements for dictionary ordering for most European languages and non-European alphabetic languages.

Capabilities

The following capabilities are provided:

Multicharacter collating elements
: The term *collating element* is used to describe the basic entities that are compared in collation. All characters in the character set are

automatically collating elements. In addition, the user can define multicharacter collating elements (sequences of two or more characters to be collated as a single entity). For example, the Spanish `ch` collates as an entity between `c` and `d`.

User-defined ordering of collating elements

The user has complete control over the order in which characters (and multicharacter collating elements) are sorted.

Multiple weights and equivalence classes

For many languages, the basic ordering is sufficient, but others require more complex rules. For example, in German, the `ö` and the `o` collate as the same character, but if two words are equal except for the `o` and the `ö`, then the word with `o` comes first. In French, all accented letters collate equally with the base character; if the words are equal, there is a defined "secondary ordering" among these characters. All characters (or collating elements) that initially collate equally are said to belong to an *equivalence class*. Such characters typically have more than one "weight." The first (primary) weight is that of the equivalence class; the second weight is determined by their relative order. The INTERACTIVE UNIX System supports up to `{COLL_WEIGHTS_MAX}` (defined in `/usr/include/sys/limits.h`) different weights for each character or collating element.

One-to-many mapping

A single character is mapped into a string of collating elements. An example of this is the German `ß`, which collates as `ss`.

Many-to-many substitution

A string is substituted for another string of one or more characters. The string that is substituted can be an empty string. In other words, the character or characters are ignored for collation purposes.

Ordering by weights

To determine their relative order, two strings are first compared based on the primary weight. If they are equal, and more than one weight has been assigned, then the strings are compared again and again until the strings either compare unequally or the weights are exhausted. Comparisons may proceed either from the beginning of the strings toward the end, or from the end toward the beginning.

Creating a Collation Sequence Definition

The source language for collation definitions in the INTERACTIVE UNIX System is the language specified by the IEEE POSIX 1003.2 standard for the LC_COLLATE locale category.

A collation sequence definition describes the relative order among collating elements (characters and multicharacter collating elements) in the locale. This order is expressed in terms of collation values or weights by assigning each element one or more collation values. The collation sequence definition is used by regular expressions, pattern matching, and sorting.

A collation source definition consists of a collation header, a collation body, and a collation trailer. The collation header is the word LC_COLLATE. The collation trailer is the string END LC_COLLATE.

The collation body consists of one or more lines of text, each of which contains an identifier, optionally followed by one or more operands. Identifiers are either keywords or collating elements. Identifiers are separated from the operands by one or more blank characters (space or tab).

Operands are characters, collating elements, or strings of characters. When a keyword is followed by more than one operand, the operands must be separated by semicolons; blanks are allowed before and/or after a semicolon.

A line modifying the comment character (the default is #) can be inserted before the header. The format is:

```
comment_char new-comment-character
```

starting in the first column. Empty lines and lines containing the *new-comment-character* in the first position are ignored.

A line modifying the escape character (the default is a backslash, \) can also be inserted before the header. The format is:

```
escape_char escape-character
```

starting in the first column. A line can be continued by placing an escape character as the last character on the line. Comment lines cannot be continued on a subsequent line using an escaped newline character.

Individual characters, characters in strings, or collating elements can be represented in operands in any of the following formats:

Symbolic notation

A character is specified via a symbolic character name, enclosed within angle brackets (< >). A symbolic name, including the angle brackets, must either be a symbol defined via a `collating-symbol` or `collating-element` keyword or must exactly match a symbolic name defined in the `charmap` file specified via the `colldef -f` option. It is not an error to specify a collating element via a `charmap` symbol that does not exist in the current `charmap` file (refer to *charmap*(5P)). The processor assumes that the definition is a "generic" one, intended for use with many codesets. Such a generic definition may contain characters not present in all codesets. Therefore, the `colldef` processor assumes that the character should simply be ignored and issues a warning message to that effect. Note that any escape character or right angle bracket in a symbolic name must be preceded by the escape character.

Using symbolic names rather than any other notation makes it possible to use the same source definition with several codesets. For example:

```
<c>;<a\>>;<c-cedilla>   "<M><a><y>"
```

Character notation

A character is specified by the character itself. The quote, comma, semicolon, angle brackets, and escape character (`" , ; < >` and *escape-character*) must be escaped (preceded by the escape character) if they are found outside strings enclosed by double quotes; only the double quote must be escaped inside quoted strings. For example:

```
c;ç;â   "May"
```

Octal notation

An octal constant must be specified as the escape character, followed by two or three octal digits. For example:

```
\143;\347      "\115\141\171"
```

Hexadecimal notation

A hexadecimal constant must be specified as the escape character, followed by an x, followed by one or two hexadecimal digits. For example:

```
\x63;\xe7      "\x4d\x61\x79"
```

Decimal notation

A decimal constant must be specified as the escape character, followed by a d, followed by one, two, or three decimal digits. For example:

```
\d99;\d231     "\d77\d97\d121"
```

`charmap` *Files*

The `colldef` program (as well as the `iconv` utility) can use the information stored in a `charmap` file. (Refer to *iconv*(1P) for more information.) These files are used to document the supported codesets. Each character in the coded character set is described with a symbolic name and the character encoding. The following is an excerpt from the `charmap` file describing IBM codepage 437. Refer to *charmap*(5P) for more information.

```
<C-cedilla>    \d128   LATIN CAPITAL LETTER C WITH CEDILLA
<u-diaeresis>  \d129   LATIN SMALL LETTER U WITH DIAERESIS
<e-acute>      \d130   LATIN SMALL LETTER A WITH ACUTE
<a-circumflex> \d131   LATIN SMALL LETTER A WITH CIRCUMFLEX
<a-diaeresis>  \d132   LATIN SMALL LETTER A WITH DIAERESIS
<a-grave>      \d133   LATIN SMALL LETTER A WITH GRAVE
<a-ring>       \d134   LATIN SMALL LETTER A WITH RING ABOVE
<c-cedilla>    \d135   LATIN SMALL LETTER C WITH CEDILLA
<e-circumflex> \d136   LATIN SMALL LETTER E WITH CIRCUMFLEX
<e-diaeresis>  \d137   LATIN SMALL LETTER E WITH DIAERESIS
<e-grave>      \d138   LATIN SMALL LETTER E WITH GRAVE
<i-diaeresis>  \d139   LATIN SMALL LETTER I WITH DIAERESIS
<i-circumflex> \d140   LATIN SMALL LETTER I WITH CIRCUMFLEX
<i-grave>      \d141   LATIN SMALL LETTER I WITH GRAVE
<A-diaeresis>  \d142   LATIN CAPITAL LETTER A WITH DIAERESIS
<A-ring>       \d143   LATIN CAPITAL LETTER A WITH RING ABOVE
<E-acute>      \d144   LATIN CAPITAL LETTER E WITH ACUTE
<ae>           \d145   LATIN SMALL LETTER AE
<AE>           \d146   LATIN CAPITAL LETTER AE
<o-circumflex> \d147   LATIN SMALL LETTER O WITH CIRCUMFLEX
```

Source File Organization

The source file contains the following keywords, described in detail in the following sections:

`LC_COLLATE`
: The header.

`collating-element`
: A `collating-element` keyword is used to specify multicharacter collating elements. This keyword is optional.

`collating-symbol`
: A `collating-symbol` keyword is used to specify collation symbols for use in collation order statements. This keyword is optional.

`substitute`
: Zero or more `substitute` keywords define mapping between strings. This keyword is optional.

`order_start`
: This keyword is followed by one or more collation order statements, assigning character collation values and collation weights to collating elements.

`order_end`
: This keyword terminates the collation order lines.

`END LC_COLLATE`
: The trailer.

`collating-element` *Keyword*

Every character in the character set is also a collating element. If the language (or application) for which this collation sequence definition is intended also recognizes multicharacter collating elements (such as the Spanish `ch`), these must be specified via a `collating-element` keyword. The syntax is:

```
collating-element symbol from string
```

The *symbol* operand must be a string of one or more characters, enclosed between angle brackets (< >), which cannot duplicate any symbolic name in the current `charmap` file or any other symbolic name defined in this collation definition. The *string* operand is a string of two or more characters to be collated as an entity. For example:

```
collating-element <ch> from <c><h>
collating-element <ss> from ss
```

`collating-symbol` *Keyword*

In addition to characters and multicharacter collating elements, you can also define special symbols for use in collation sequence statements, that is, between the `order_start` and the `order_end` keywords. Such a symbol does not have any character associated with it, as the `charmap` symbols do.

However, placing such a symbol in the collating sequence assigns to it a relative order that can be used in other collating element specifications. The syntax is:

```
collating-symbol symbol
```

The *symbol* is a string of one or more characters, surrounded by angle brackets, which must not duplicate any symbolic name in the current `charmap` file or any other symbolic name defined in this collation definition. For example:

```
collating-symbol <UPPER_CASE>
collating-symbol <LOWER_CASE>
collating-symbol <NO_ACCENT>
collating-symbol <GRAVE>
collating-symbol <ACUTE>
```

`substitute` *Keyword*

The `substitute` keyword is used to define a substring substitution in a string to be collated. The syntax is:

```
substitute "regexp" with "repl"
```

The first operand is treated as a simple regular expression. The replacement operand consists of zero or more characters and regular expression backreferences (for example, `\1` through `\9`).

When strings are collated based on a collation definition containing substitute statements, any substitutions are performed before strings are compared. For instance, if you have a substitute statement:

```
substitute "Mc" with "Mac"
```

and you compare the two strings `McArthur` and `MacArthur`, the substitute is first applied to both strings. As a result, the first string is replaced by `MacArthur` and the two strings compare as equals.

Ranges in the regular expression are interpreted according to the current character collation sequence, and character classes are interpreted according to the character classification specified via the LC_CTYPE environment variable at collation time. If more than one substitute statement is present in the collation definition, the substitute statements are applied in the order in which they occur in the source definition.

Both operands must be enclosed within double-quotes (" ") or a null replacement is indicated by two adjacent double-quotes. For example:

```
substitute "Mc" with ""
```

order_start *Keyword*

The order_start keyword precedes collation order entries and also defines the number of weights for this collation sequence definition and other collation rules.

The syntax of the order_start keyword is:

```
order_start     sort-rules;sort-rules; ...
```

The operands to the order_start keyword are optional. If present, the operands define rules to be applied when strings are compared. The number of operands defines how many weights each element is assigned; if no operands are present, one *forward* operand is assumed. If present, the first operand defines rules to be applied when comparing strings using the first (primary) weight; the second, when comparing strings using the second weight; and so on. Operands are separated by semicolons (;). Each operand consists of one or more collation directives, separated by commas (,). If the number of operands exceeds the {COLL_WEIGHTS_MAX} limit, the utility ignores the operands in excess of the limit and issues a warning message. The following directives are supported:

forward
: Specifies that comparison operations for the weight level proceed from the beginning of the string to the end of the string.

`backward`
: Specifies that comparison operations for the weight level proceed from the end of the string to the beginning of the string.

`position`
: Specifies that comparison operations for the weight level will consider the relative position of non-`IGNORE`d elements in the string such that, if strings compare as equals, the element with the shortest distance from the starting point of the string is collated first.

The directives `forward` and `backward` are mutually exclusive. For example:

```
order_start    forward;backward;forward
```

The absence of operands for this keyword is taken as a directive to perform comparisons on a character basis rather than on a string basis.

Collation Order

The `order_start` keyword is followed by `collating-element` entries. The syntax for the `collating-element` entries is:

```
collating-element weight;weight; ...
```

Each `collating-element` consists of either a character (in any of the forms defined above), a `collating-element` symbol, a `collating-symbol` symbol, an ellipsis (`...`), or the special symbol `UNDEFINED`. The order in which `collating-elements` are specified determines the character collation sequence, such that each `collating-element` compares less than the elements following it. The `NULL` character compares lower than any other character.

A `collating-element` symbol is used to specify multicharacter collating elements and indicates that the character sequence specified via the `collating-element` symbol is to be collated as a unit and in the relative order specified by its place. A `collating-symbol` symbol is used to define a position in the relative order for use in weights.

The ellipsis symbol (...) specifies that a sequence of characters collates according to their encoded character values. That is, all characters with a coded character set value higher than the value of the character in the preceding line and lower than the coded character set value for the character in the following line are placed in the character collation order between the previous and the following character. This is done in ascending order according to their coded character set values. An initial ellipsis is interpreted as if the line preceding it specified the NULL character; a trailing ellipsis is interpreted as though the line following it specified the highest coded character set value in the current coded character set. An ellipsis is treated as invalid if the lines preceding or following it do not specify characters in the current coded character set. Note that the use of the ellipsis symbol ties the definition to a specific coded character set and may preclude the definition from being portable. The colldef utility issues a warning to this effect if an ellipsis is detected. The explicit specification elsewhere of a character automatically included via an ellipsis symbol is treated as an error.

All characters not defined in the order sequence (either explicitly or via an ellipsis) are placed in the collation order via the special symbol UNDEFINED. All such characters are placed into the existing order at the point of the UNDEFINED symbol, and ordered according to their coded character set values. If no UNDEFINED symbol is specified, and the current coded character set contains characters not specified in this clause, colldef issues a warning message and places such characters at the end of the character collation order.

The optional operands for each collating-element are used to define the primary, secondary, or subsequent weights for the collating-element. The first operand specifies the relative primary weight, the second the relative secondary weight, and so on. Two or more collating-elements can be assigned the same weight. They are said to belong to the same *equivalence class*. In string collation, each pair of strings is first compared based on primary weight. If equal, collating-elements belonging to primary equivalence classes are compared again based on their secondary weights. If still equal, secondary equivalence class elements are compared again based on tertiary weights, up to the limit {COLL_WEIGHTS_MAX}.

Weights must be expressed as characters (in any of the forms specified above), collating-symbols, collating-elements, an ellipsis, or the special symbol IGNORE. A single character, a collating-symbol symbol, or a `collating-element` symbol represents the relative order in the character collating sequence of the character or symbol, rather than its absolute value. Multiple characters or symbols indicate one-to-many mapping.

The special symbol `IGNORE` means that this character is to be ignored at the defined weight level for collation purposes.

For example, if the dash (-) is `IGNORE`d, then the two strings

```
co-ordinate
```

and

```
coordinate
```

collate as equals. In regular expressions, such characters are never ignored. Ranges are based on the order in which elements are listed in the definition (basic character ordering sequence), and all characters are explicitly or implicitly listed.

All characters specified via an ellipsis are assigned unique weights and are ordered according to their coded character set values. Characters specified via an explicit or implicit `UNDEFINED` special symbol are by default assigned the same primary weight (that is, they belong to the same equivalence class). An ellipsis symbol as a weight is interpreted to mean that each character in the sequence must have unique weights, equal to the relative order of the character in the character collation sequence. Secondary and subsequent weights have unique values. The use of the ellipsis as a weight is treated as an error if the `collating-element` is neither an ellipsis nor the special symbol `UNDEFINED`.

An empty weight implies that the `collating-element` will be assigned a weight equal to the current position in the order. In other words, the `collating-element` "collates as itself."

`order_end` *Keyword*

The `order_end` keyword terminates the ordering statements.

An Example (See also notes on next page)

```
LC_COLLATE
#
collating-element <ch> from <c><h>              # See Note 1
collating-element <ss> from ss
#
collating-symbol <UPPER_CASE>
collating-symbol <LOWER_CASE>
collating-symbol <NO_ACCENT>
collating-symbol <GRAVE>
collating-symbol <ACUTE>
#
substitute "Mc" with "Mac"
#
order_start       forward;backward;forward
#
<UPPER_CASE>                                    # See Note 2
<LOWER_CASE>
<NO_ACCENT>
<GRAVE>
<ACUTE>
<space>
\...              IGNORE;IGNORE;IGNORE          # See Note 3
<A>               <A>;<UPPER_CASE>;<NO_ACCENT>  # See Note 4
<a>               <A>;<LOWER_CASE>;<NO_ACCENT>
<a-acute>         <A>;<LOWER_CASE>;<ACUTE>
<a-grave>         <A>;<LOWER_CASE>:<GRAVE>
<B>
<b>
<C>               <C>;<C>;<C>                   # See Note 5
<C-cedilla>       <C>;<C>;<C-cedilla>
<c>               <C>;<c>;<c>
<c-cedilla>       <C>;<c>;<c-cedilla>
<ch>              <ch>;<ch>;<ch>                # See Note 6
<S>               <S>;<S>;<S>
<s>               <S>;<s>;<s>
<ss>              <S><S>;<s><s>;<s><s>
<sharp-s>         <S><S>;<s><s>;<s><s>          # See Note 7
UNDEFINED         IGNORE;IGNORE;IGNORE          # See Note 8
order_end
END LC_COLLATE
```

Notes

1. The character sequences `ch` and `ss` are defined as collating elements.
2. The `collating-symbols` <UPPER_CASE>, <LOWER_CASE>, <NO_ACCENT>, <GRAVE>, and <ACUTE> are placed first in the ordering sequence, followed by the `space` symbol.
3. Characters with code values between `space` and `A` are placed in the basic ordering sequence after the space, but are ignored for collation purposes.
4. The accented and unaccented `A`'s have the same primary weight, that is, they belong to an equivalence class. The secondary weight is based on case, but ignores accents. The third weight considers accents. This definition uses the collating symbols and their relative order (uppercase before lowercase, no accents before accents).

   The definition can be viewed as a directive to transform strings by weight before comparing them. For example, when comparing the strings `abbà` and `Abba`, the two strings are first compared using the primary weight. This equates to comparing `ABBA` with `ABBA`, that is, they compare as equals.

   On secondary weighting, they compare as follows:

```
<LOWER_CASE><LOWER_CASE><LOWER_CASE><LOWER_CASE>
```

   against:

```
<UPPER_CASE><LOWER_CASE><LOWER_CASE><LOWER_CASE>
```

   The first collates after the second.

5. The accented and unaccented `C`'s also belong to an equivalence class. Secondary ordering and tertiary ordering are defined using the characters themselves. The uppercase letters collate before the lowercase ones and the accented letters after the unaccented ones.

   The two strings `Ça` and `Ca` first compare as `CA` versus `CA`. Based on secondary weights, they still compare as equals: (C<LOWER_CASE> versus C<LOWER_CASE>). On tertiary weight comparison, the two strings compare as Ç<LOWER_CASE> versus C<LOWER_CASE>, that is, the second compares lower.

6. The string ch compares as a single element. The string Bach consists of three collating elements and collates after the string Back.

7. The character ß (eszet or "sharp s") is a German character that collates as two "esses" (ss). This means that the two strings Strasse and Straße should collate as equals.

8. All characters not explicitly defined (or implicitly included via an ellipsis) are placed last in the collation sequence, in order according to their coded values. They are ignored for collation purposes.

Use in Regular Expressions and Shell Pattern Matching

The collation sequence determines how bracket expressions in regular expressions are interpreted:

- All characters are valid in a bracket expression. Multicharacter collating elements (such as <ch> in the example above) are also recognized.
- Multicharacter collating elements must be entered using a special "bracket-dot" syntax, for example, [.ch.], to distinguish the multicharacter element from the sequence "ch".
- All characters belonging to an equivalence class can be referenced using the special "bracket-equal" syntax; [=a=] is shorthand for A, a, à, á in the example above.
- Range expressions are interpreted according to the basic character collation order, that is, the order in which the characters are listed in the definition. In the previous example, all characters not explicitly specified collate last via the UNDEFINED statement. This means that, using the previous example, [a-s] only specifies the characters in the list between a and s:

```
a à á B b C Ç c ç "ch" S s
```

  Likewise, a range such as [r-t] will not contain s.

- To be able to find both "Strasse" and "Straße" in text with one expression, it is necessary to make ss into a collating element. Then, the following regular expression will find both strings: "Stra[[.ss.][.ß.]]e".

Specifying Numeric and Monetary Information

Numeric and monetary formatting determines how numeric and monetary items appear. This section explains how it can be used and how the files that contain the information should be set up.

Reasons for Defining Numeric and Monetary Formatting

The default conventions for decimal delimiter and other numeric formatting rules are seldom appropriate in an international environment. For example, the default decimal delimiter is a period, but in most European countries the comma is used instead. By defining numeric and monetary formatting with the correct values, programs display fractions using the appropriate decimal delimiter.

Defining Numeric and Monetary Formatting

These definitions are created by placing a specification in the appropriate file (either `LC_NUMERIC` or `LC_MONETARY`) in a `locale` directory.

When to Use the Numeric and Monetary `locale` Category

The created and installed definitions are not activated until the user specifies that they should be used. The user must set the `LC_NUMERIC` environment variable to the directory in which that file is stored and the `LC_MONETARY` environment variable to the directory in which that file is stored. Alternately, the user can set the `LC_ALL` or `LANG` environment variable to the directory to specify both. This must be done before a program using the stored definitions is executed. Note that the program must be set up to check and set the international environment (via the `setlocale` function). In the INTERACTIVE UNIX System, the standard utilities that depend on numeric editing, such as `awk`, have been modified to use the international environment.

Numeric Editing

Numeric editing controls the appearance of (nonmonetary) numbers, as well as the input format. The following three aspects of numeric editing are controlled via the `LC_NUMERIC locale` category:

- The character used as a decimal delimiter
- The character used to separate groups of digits (thousands separator)
- The size of such groups

It should be noted that while the standard INTERACTIVE UNIX System library subroutines `printf`, `scanf`, and `strtod` (refer to *printf*(3P), *scanf*(3P), and *strtod*(3C) for more information) are sensitive to the decimal delimiter, they do not support grouping of digits. Consequently, while user-developed functions can (and should) take into account grouping and thousands separators, the standard functions do not.

Creating a Numeric Category Definition

The source language for the numeric category in the INTERACTIVE UNIX System is the language defined by the POSIX.2 group for the `LC_NUMERIC locale` category.

A numeric editing source definition consists of a header, a numeric editing body, and a trailer. The header is the word `LC_NUMERIC`. The trailer is the string `END LC_NUMERIC`.

The numeric editing body consists of one or more lines of text. Each line contains a keyword followed by one or more operands. Keywords are separated from the operands by one or more blank characters (space or tab).

Operands are characters, strings of characters, or digits. When a keyword is followed by more than one operand, the operands must be separated by semicolons (`;`). Blank characters are allowed before and/or after a semicolon. Strings must be surrounded by quotes. Individual characters may be surrounded by quotes, but it is not required. Blank lines or lines containing a # sign in the first column are ignored.

The following keywords are recognized:

`LC_NUMERIC`
: The header.

`decimal_point`
: Defines the decimal delimiter character.

`thousands_sep`
: Defines the thousands separator character.

`grouping`
Defines the grouping of digits.

`END LC_NUMERIC`
The trailer.

`decimal_point` *Keyword*

This keyword specifies the character to use as the decimal delimiter in the editing of floating-point numbers (both on input and output). The format is:

```
decimal_point character
```

where *character* is the character chosen as the decimal delimiter.

`thousands_sep` *Keyword*

This keyword specifies the character to be used as the thousands separator. The format is:

```
thousands_sep character
```

where *character* is the character chosen to separate groups of digits to the left of the decimal delimiter in formatted nonmonetary quantities. Note that none of the standard INTERACTIVE UNIX System subroutines or commands recognizes a thousands separator.

`grouping` *Keyword*

The `grouping` keyword defines the size of each group of digits in formatted nonmonetary quantities. The format is:

```
grouping digit [ ; digit ] ...
```

where the operands are integers separated by semicolons. Each integer specifies the number of digits in a group, with the initial integer defining the size of the group immediately preceding the decimal delimiter and the following integers defining the preceding groups. Grouping is performed only for groups with a defined size unless the last integer is zero, in which case the size of the last group is used repeatedly for the remainder of the digits.

As an example of the interpretation of the `grouping` keyword, assume that the value to be formatted is 123456789 and the `thousands_sep` is `" "`. The following are the results with the various groupings shown:

Grouping	**Formatted Value**
3	123456 798
3;0	123 456 789
3;2	1234 56 789
3;2;0	12 34 56 789

An Example of a Numeric Category Definition

```
LC_NUMERIC
#
decimal_point       ","
#
thousands_sep       "."
#
grouping            3;0
#
END LC_NUMERIC
```

How a Program Uses This Information

If a program needs to access the values in the current `locale`, it can do so via the library interfaces `localeconv` and `nl_langinfo`. Refer to *localeconv*(3P) and *nl_langinfo*(3P) for more information.

Monetary Editing

Monetary editing controls the appearance of monetary numbers. Note that no standard INTERACTIVE UNIX System library routines or commands take into account monetary editing. The following aspects of monetary editing are controlled via the `LC_MONETARY` `locale` category:

- The character used as a monetary decimal delimiter
- The number of fractional digits
- The character used to separate groups of digits (thousands separator)
- The size of such groups
- The content (and placement) of strings used to denote the currency
- Positive and negative signs and their placement

Creating a Monetary Category Definition

The source language for the monetary category in the INTERACTIVE UNIX Operating System is the language defined by the POSIX.2 group for the `LC_MONETARY locale` category.

A monetary editing source definition consists of a header (the word `LC_MONETARY`), a monetary editing body, and a trailer (the string `END LC_MONETARY`).

The monetary editing body consists of one or more lines of text. Each line contains a keyword followed by one or more operands. Keywords are separated from the operands by one or more blank characters (space or tab).

Operands are characters, strings of characters, or digits. When a keyword is followed by more than one operand, the operands must be separated by semicolons (;). Blank characters are allowed before and/or after a semicolon. Strings must be surrounded by quotes. Individual characters may be surrounded by quotes, but it is not required. Blank lines or lines containing a # sign in the first column are ignored.

The following keywords are recognized:

`LC_MONETARY`
: The header.

`int_curr_symbol`
: Defines the ISO standard four-character (three letters and a space) code for currency, for example, `"USD "` for U.S. dollar.

`currency_symbol`
: Defines the character to be used as the currency symbol, for example `"$"`.

`mon_decimal_point`
: Defines the decimal delimiter for monetary quantities.

`mon_thousands_sep`
Defines the thousands separator for monetary quantities.

`mon_grouping`
Defines the grouping of digits.

`positive_sign`
Defines the positive sign.

`negative_sign`
Defines the negative sign.

`int_frac_digits`
Defines the number of fractional digits displayed when formatting using the `int_curr_symbol`.

`frac_digits`
Defines the number of fractional digits displayed when formatting using the `currency_symbol`.

`p_cs_precedes`
Defines whether the `currency_symbol` succeeds or precedes a positive quantity.

`p_sep_by_space`
Defines whether a space separates the `currency_symbol` from a positive quantity.

`n_cs_precedes`
Defines whether the `currency_symbol` succeeds or precedes a negative quantity.

`n_sep_by_space`
Defines whether a space separates the `currency_symbol` from a negative quantity.

`p_sign_posn`
Defines the placement of the sign and a positive quantity.

`n_sign_posn`
Defines the placement of the sign and a negative quantity.

`END LC_MONETARY`
The trailer.

int_curr_symbol *Keyword*

This keyword is used to define the international currency symbol. The operand must be a four-character string, with the first three characters containing the alphabetic international currency symbol in accordance with those specified in ISO 4217 (*Codes for the representation of currencies and funds*). The fourth character must be the character used to separate the international currency symbol from the monetary quantity, normally a space. For example:

```
int_curr_symbol    "FMK "
```

currency_symbol *Keyword*

This keyword defines the string to be used as the local currency symbol. For example:

```
currency_symbol    $
```

mon_decimal_point *Keyword*

The operand is the character to be used as the decimal delimiter to format monetary quantities. For example:

```
mon_decimal_point  "$"
```

is the Portuguese monetary decimal delimiter.

mon_thousands_sep *Keyword*

This operand is the string to be used as the separator for groups of digits to the left of the decimal delimiter in formatted monetary quantities. For example:

```
mon_thousands_sep  " "
```

`mon_grouping` *Keyword*

This keyword is used to define the size of each group of digits in formatted monetary quantities. The operand is a sequence of integers separated by semicolons. Each integer specifies the number of digits in each group, with the initial integer defining the size of the group immediately preceding the decimal delimiter and the following integers defining the preceding groups. Grouping is performed only for groups with a defined size, unless the last integer is zero, in which case the size of the last group is repeatedly used for the remainder of the digits. For example:

```
mon_grouping        3;0
```

`positive_sign`/`negative_sign` *Keywords*

Each operand is a string used to indicate positive or negative values. For example:

```
positive_sign       ""

negative_sign       "C"
```

`int_frac_digits` *Keyword*

This keyword is an integer that represents the number of fractional digits (those to the right of the decimal delimiter) to be displayed in a formatted monetary quantity using `int_curr_symbol`. For example:

```
int_frac_digits   2
```

frac_digits *Keyword*

This keyword is an integer that represents the number of fractional digits (those to the right of the decimal delimiter) to be displayed in a formatted monetary quantity using `currency_symbol`. For example:

```
frac_digits    2
```

p_cs_precedes/n_cs_precedes *Keywords*

Each keyword is an integer that is set to `1` if the `currency_symbol` precedes the value for a positive or negative formatted monetary quantity, respectively, and set to `0` if the symbol succeeds the value. For example:

```
p_cs_precedes  1
```

p_sep_by_space/n_sep_by_space *Keywords*

Each keyword is an integer that is set to `1` if a space separates the `currency_symbol` from the value for a positive or negative formatted monetary quantity, respectively. They are set to `0` if no space separates the symbol from the value.

p_sign_posn/n_sign_posn *Keywords*

Each keyword is an integer that is set to a value indicating the positioning of the `positive_sign` or `negative_sign` for a positive or negative formatted monetary quantity, respectively. The following integer values are recognized:

- `0` Parentheses enclose the quantity and the `currency_symbol`
- `1` The sign string precedes the quantity and the `currency_symbol`
- `2` The sign string succeeds the quantity and the `currency_symbol`
- `3` The sign string immediately precedes the `currency_symbol`
- `4` The sign string immediately succeeds the `currency_symbol`

An Example of a Monetary Category Definition

```
LC_MONETARY
#
int_curr_symbol       "CHF "
currency_symbol       "SFrs."
mon_decimal_point     "."
mon_thousands_sep     ","
mon_grouping          3;0
positive_sign         ""
negative_sign         "C"
int_frac_digits       2
frac_digits           2
p_cs_precedes         0
p_sep_by_space        0
n_cs_precedes         1
n_sep_by_space        0
p_sign_posn           1
n_sign_posn           2
#
END LC_MONETARY
```

With the above definition, a monetary quantity should be edited as follows:

```
Positive      SFrs.1,234.56
Negative      SFrs.1,234.56C
```

How a Program Uses This Information

If a program needs to access the values in the current `locale`, it can do so via the library interfaces `localeconv` and `nl_langinfo`. Refer to *localeconv*(3P) and *nl_langinfo*(3P) for more information.

Specifying Yes/No Response Information

The "yes/no" response category determines the correct string to be used as affirmative (yes) and negative (no) responses to program queries.

Reasons for Defining Yes/No Responses

The standard UNIX System utilities that require this kind of interaction (such as `rm`) normally expect either a `y` or an `n`. In countries that do not normally use the English language, this is not the obvious response. In France, for instance, the obvious affirmative response would be `o` (for oui); in Spain, it would be `s` (for si).

Defining Yes/No Responses

These definitions are created by placing a specification in the `LC_MESSAGES` file in a `locale` directory.

When to Use the Yes/No Response `locale` Category

The created and installed definitions are not activated until the user specifies that they should be used. To do this, the user must set the `LC_ALL`, `LC_MESSAGES`, or `LANG` environment variable to the directory in which the files are stored. This must be done before a program using the stored definitions is executed. Note that the program must be set up to check and set the international environment (via the `setlocale` function). In the INTERACTIVE UNIX System, the standard utilities that depend on a yes/no response, such as `ln` and `rm`, have been modified to use the international environment. Note that while the internationalized yes/no response is required by XPG3 for certain commands, the `LC_MESSAGES` category is not part of the `locale` as defined by XPG3.

Creating a Yes/No Response Category Definition

The source language for the yes/no response category in the INTERACTIVE UNIX Operating System is the language defined by the POSIX.2 group for the `LC_MESSAGES` category.

A yes/no response source definition consists of a header, a response body, and a trailer. The header is the word LC_MESSAGES. The trailer is the string END LC_MESSAGES.

The response body consists of one or more lines of text. Each line contains a keyword, followed by one or more operands. Keywords are separated from the operands by one or more blank characters (space or tab).

Operands are characters, strings of characters, or digits. When a keyword is followed by more than one operand, the operands must be separated by semicolons (;). Blank characters are allowed before and/or after a semicolon. Strings must be surrounded by quotes. Individual characters may be surrounded by quotes, but it is not required. Blank lines or lines containing a # sign in the first column are ignored.

The following keywords are recognized:

`LC_MESSAGES`
: The header.

`yesexpr`
: Defines the affirmative (yes) response.

`noexpr`
: Defines the negative (no) response.

`END LC_MESSAGES`
: The trailer.

`yesexpr` *Keyword*

This keyword specifies the character or string to use as the affirmative (yes) response. The format is:

```
yesexpr regular-expression
```

where *regular-expression* is a regular expression which, when used to match affirmative responses, will report a match.

noexpr *Keyword*

This keyword specifies the character or string to use as the negative (no) response. The format is:

```
noexpr regular-expression
```

where *regular-expression* is a regular expression which, when used to match negative responses, will report a match.

An Example of a Response Category Definition

```
LC_MESSAGES
#
yesexpr             "^[Yy].*"
#
noexpr              "^[Nn]on"
#
END LC_MESSAGES
```

How a Program Uses This Information

If a program needs to access the values in the current `locale`, it can do so via the `nl_langinfo` library interface. Refer to *nl_langinfo*(3P) for more information.

Index

D

E

F

L

M

O

P

R

S

T

U

V

W

X

Y

SunSoft
A Sun Microsystems, Inc. Business

2550 Garcia Avenue
Mountain View, CA 94043 U.S.A.

Part No.: 801-7872-10
Revision A of June 1994