

CLIPPER System Administrator's Guide

DSA029910

January 1990

SSS0061, SSS0062, SSS0075, SSAD075

DSA02991S (Spine), DSA02991T (Bleed Tabs), DIXGDMRS4 (Binder)

INTERGRAPH

Warranties and Liabilities

All warranties given by Intergraph Corporation about equipment or software are set forth in your purchase contract.

The information and the software discussed in this document are subject to change without notice and should not be considered commitments by Intergraph Corporation. Intergraph Corporation assumes no responsibility for any errors that may appear in this document.

The software discussed in this document is furnished under a license and may be used or copied only in accordance with the terms of this license.

No responsibility is assumed by Intergraph for the use or reliability of software on equipment that is not supplied by Intergraph or its affiliated companies.

Trademarks

Intergraph is a registered trademark of Intergraph Corporation. InterPro, InterAct, InterView, and InterServe are registered trademarks of Intergraph Corporation. CLIPPER and CLIX are trademarks of Intergraph Corporation.

Other brands and product names are trademarks of their respective owners.

Classifications

This equipment is designed to comply with the requirements in Part 15 of the FCC rules for a class A computing device.

CLIPPER System Administrator's Guide

DSA029910

January 1990

SSS0061, SSS0062, SSS0075, SSAD075

DSA02991S (Spine), DSA02991T (Bleed Tabs), DIXGDMRS1 (Binder)

Warranties and Liabilities

All warranties given by Intergraph Corporation about equipment or software are set forth in your purchase contract.

The information and the software discussed in this document are subject to change without notice and should not be considered commitments by Intergraph Corporation. Intergraph Corporation assumes no responsibility for any errors that may appear in this document.

The software discussed in this document is furnished under a license and may be used or copied only in accordance with the terms of this license.

No responsibility is assumed by Intergraph for the use or reliability of software on equipment that is not supplied by Intergraph or its affiliated companies.

Trademarks

Intergraph is a registered trademark of Intergraph Corporation. InterPro, InterAct, InterView, and InterServe are registered trademarks of Intergraph Corporation. CLIPPER and CLIX are trademarks of Intergraph Corporation.

Other brands and product names are trademarks of their respective owners.

Classifications

This equipment is designed to comply with the requirements in Part 15 of the FCC rules for a class A computing device.

Copyright

© 1990 INTERGRAPH CORPORATION

INCLUDING THIS DOCUMENTATION, AND ALL SOFTWARE AND ITS FILE FORMATS AND AUDIO-VISUAL DISPLAYS DESCRIBED HEREIN; ALL RIGHTS RESERVED; CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF INTERGRAPH AND/OR OTHER THIRD PARTIES WHICH IS PROTECTED BY COPYRIGHT, TRADE SECRET AND TRADEMARK LAW AND MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE WITHOUT PRIOR WRITTEN AUTHORIZATION.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subdivision (c)(1)(ii) of the rights in technical data and computer software clause at 52.227-7013.

Unpublished—rights reserved under the Copyright Laws of the United States.

Intergraph Corporation
One Madison Industrial Park
Huntsville, AL 35807-4201

Preface

This guide provides system managers with instructions for using and managing Intergraph workstations and servers.

The information in this guide is built on the information presented in the *CLIPPER User's Guide* (DSA0300). For information on topics such as the screen display, input devices, basic CLIX operating system concepts, ASCII text editors, the floppy disk drive, and Intergraph Toolbox utilities, refer to the *CLIPPER User's Guide*.

This document includes information on system management tasks such as setting up a new system, creating user accounts, modifying the login environment, customizing the user interface (including information on creating windows using *vterm*), creating and maintaining file systems, setting up a line printer, monitoring processes, backing up and restoring files, rebuilding the hard disk, repartitioning the hard disk, and using the Utility pages.

Document Conventions

The following conventions and symbols appear in this guide:

1. Command names and user actions appear in **bold type**.
2. The term *workstation* refers to an InterPro, InterAct, or InterView workstation. The term *server* refers to an InterServe processor.
3. *Select* means to place the screen cursor on the desired option and tap <D> (middle cursor button).
4. **Warnings** are used to emphasize potential physical danger. **Cautions** are used to emphasize critical information or potential errors. **Notes** are used to emphasize important information.
5. *Press a button* means to press a button and hold it down.
6. *Tap a button* means to press and immediately release a button.
7. *Key in* means to enter input at the keyboard and press <Return>.
8. Various symbols appear in the text. They represent the following keys or cursor buttons:

<Ctrl> Control key

<Ctrl-D> Unless otherwise noted in the text, you must hold down the <Ctrl> key while pressing the second key.

<Return> Return key

<Esc> Escape key

Additional References

The UNIX System V documentation is useful reference material. The following Release 5.3 documents can be purchased individually or in sets from Intergraph:

<i>Title</i>	<i>Release V.3</i>
<i>UNIX System V User's Guide—CLIPPER</i>	DSYS08010
<i>UNIX System V Programming Guide—CLIPPER</i>	DSYS08410
<i>UNIX System V Administrator's Guide—CLIPPER</i>	DSYS08210
<i>UNIX System V User's Reference Manual—CLIPPER</i>	DSYS08110
<i>AT&T 5.3.1 User's Reference Addendum</i>	DSYS19410
<i>UNIX System V Administrator's Reference Manual—CLIPPER</i>	DSYS08310
<i>AT&T 5.3.1 Administrator's Reference Addendum</i>	DSYS19710
<i>UNIX System V Programmer's Reference Manual—CLIPPER</i>	DSYS08510
<i>AT&T 5.3.1 Programmer's Reference Addendum</i>	DSYS19510
<i>CLIX System Administrator's Reference Manual</i>	DSYS18310
<i>Addendum to CLIX System Administrator's Reference Manual</i>	DSYS18311
<i>Addendum to CLIX System Administrator's Reference Manual</i>	DSYS18312
<i>CLIX Programmer's & User's Reference Manual</i>	DSYS18410
<i>Addendum to CLIX Programmer's & User's Reference Manual</i>	DSYS18411
<i>Addendum to CLIX Programmer's & User's Reference Manual</i>	DSYS18412
<i>CLIX System Guide</i>	DSYS18510
<i>Addendum to CLIX System Guide</i>	DSYS18511
<i>Addendum to CLIX System Guide</i>	DSYS18512
<i>CLIX Permuted Index (master index to CLIX and AT&T reference manuals)</i>	DSA027410

Ordering and Support Information

To order any of these documents:

- Within the United States contact your customer engineer or sales account representative.
- For international locations, contact the Intergraph subsidiary or distributor where you purchased your workstation.

If you have trouble with the workstation/server or the procedures described in this guide, contact Intergraph Customer Support at 1-800-633-7248. International customers should contact the Intergraph subsidiary or distributor where the workstation/server was purchased.



Contents

Chapter 1: Setting Up a New System

1.1	Setting Up and Adjusting the Hardware	1-3
1.2	Evaluating the Default Partitions	1-4
1.2.1	Default Partitions	1-4
1.2.2	Determining Partition Sizes	1-5
1.3	Configuring the Workstation Using the Utility Pages	1-6
1.4	Setting Up User Accounts	1-8
1.5	Assigning an Internet Address	1-9
1.6	Delivering Software	1-10
1.7	Determining a Backup Schedule	1-11
1.8	Using Intergraph System Documentation	1-12

Chapter 2: System Startup and Initialization

2.1	The /etc/init Process.....	2-3
2.2	The /etc/getty Process	2-5
2.3	Modifying the inittab and gettydefs Files for a 32-Channel RS-232 Interface Assembly.....	2-6
2.3.1	Modifying the /etc/inittab File.....	2-6
2.3.2	Modifying the /etc/gettydefs.....	2-8
2.4	The /etc/rc2 Process	2-9
2.4.1	The /etc/init.d Directory	2-10
2.4.2	Initializing Additional Processes from Multiuser Mode	2-10

Chapter 3: Creating and Maintaining User Accounts

3.1	System and Administrative Accounts	3-3
3.1.1	The root Account	3-3
3.1.2	The sysadm Account	3-3
3.1.3	The setup Account	3-4
3.1.4	The sys Account	3-4
3.1.5	The nuucp Account	3-4
3.2	Adding User Accounts	3-5
3.2.1	Using addusers to Add User Accounts	3-6
3.2.2	Using sysadm to Add User Accounts	3-8
3.3	Removing User Accounts	3-10
3.3.1	Using rmusers to Remove User Accounts	3-10
3.3.2	Using sysadm to Remove User Accounts	3-11
3.4	Assigning, Changing, and Removing Passwords	3-13
3.4.1	Restrictions on Passwords	3-13
3.4.2	Creating a Password	3-13
3.4.3	Forcing a Password Change	3-14
3.4.4	Removing a Password	3-15
3.4.4.1	Removing a Password from User Accounts	3-15
3.4.4.2	Removing the root Password	3-15
3.4.4.3	Removing the Utility Pages Password	3-18

3.5	Using Protection Codes to Manage Files	3-19
3.5.1	File Protection Codes	3-19
3.5.1.1	Directory Files	3-20
3.5.1.2	Ordinary Files	3-21
3.5.2	Modifying File Permissions	3-21
3.5.3	Modifying File Ownership	3-22
3.5.4	Changing Group Access	3-23
3.5.4.1	Changing Your Active Group	3-23
3.5.4.2	Changing the Group of a File	3-24

Chapter 4: Modifying the Login Environment

4.1	The <code>getty</code> and <code>login</code> Processes	4-3
4.1.1	The <code>/etc/passwd</code> File	4-4
4.1.2	Customizing the <code>/etc/passwd</code> File	4-5
4.2	The <code>/etc/profile</code> File	4-6
4.3	The <code>.profile</code> file	4-8
4.4	The <code>.env</code> File	4-10
4.4.1	Command Recall (The <code>.history</code> File)	4-11
4.4.1.1	Command Recall Using the <code>vi</code> Editor	4-12
4.4.1.2	Command Recall Using the <code>emacs</code> Editor	4-12
4.4.1.3	Temporary Editors for Command Recall	4-12
4.5	The <code>.exrc</code> File	4-13
4.6	Creating a Reminder Service	4-14
4.7	Using <code>cron</code>	4-15
4.7.1	The <code>crontab</code> Command	4-16
4.7.2	Executing One Command at a Specified Time	4-17

Chapter 5: Window and Screen Configuration

5.1	Setting the Workstation User	5-3
5.2	Creating VT220 Windows	5-4
5.2.1	Using <code>vterm</code>	5-4
5.2.2	Creating Local Windows Through the Workstation Menu	5-8
5.2.3	Creating a Terminal Window	5-8
5.3	Using the General Setup Menu to Define VT220 Characteristics	5-10
5.4	Using the Display Setup Menu to Define Display Characteristics	5-12
5.5	Using the Keyboard Setup Menu to Define Keyboard Characteristics	5-13
5.6	Using the Tab Setup Menu to Define Tabs	5-14
5.7	Using the Printer Setup Menu to Define a Window Attached to a Dedicated Printer	5-15
5.8	Using the Macro Setup Menu to Assign Macros to Keys	5-16

5.9	Creating Color Windows	5-18
5.9.1	The Intergraph-Developed regis Color Command	5-18
5.9.2	The ReGIS Color Command	5-19
5.9.3	The ANSI Color Escape Sequences	5-22
5.10	Using the Catalog Function to Save Customized Windows	5-24
5.11	Altering the Workstation Pulldown Menu	5-26
5.11.1	Adding Processes to the Workstation Pulldown Menu	5-26
5.11.2	Modifying the Workstation Pulldown Menu	5-28
5.11.3	Deleting Processes from the Workstation Pulldown Menu	5-28
5.12	Changing Screen Configuration	5-30
5.12.1	Changing the Number of Virtual Screens (Single-Monitor Workstations Only)	5-30
5.12.2	Changing the Number of Message Strips (Dual-Monitor Workstations Only)	5-31
5.12.3	Changing the Message Strip(s) Location	5-31
5.13	Changing the Background Pattern	5-32
5.14	Changing the Screen Color	5-33
5.15	Invoking Processes Automatically	5-34
5.15.1	The def_proc File	5-34
5.15.2	The .def_proc File	5-35
5.15.3	The sys_proc File	5-36

Chapter 6: Setting Up a Line Printer and Printing

6.1	Configuring a Line Printer	6-3
6.2	Using the Line Printer	6-4
6.2.1	Submitting a Print Request	6-4
6.2.2	Monitoring a Print Request	6-4
6.2.3	Canceling a Print Request	6-5

Chapter 7: Creating and Maintaining File Systems

7.1	Creating a File System	7-3
7.1.1	Creating a Device File	7-3
7.1.2	Creating a File System on a Partition	7-9
7.1.2.1	Creating a Standard File System	7-10
7.1.2.2	Creating a Fast File System	7-11
7.1.3	Mounting a Partition	7-12
7.2	Monitoring Disk Space	7-13
7.2.1	Displaying the Amount of Free Disk Space	7-13
7.2.2	Displaying the Amount of Disk Space Used	7-14
7.2.3	Freeing Disk Space	7-14
7.3	Checking the Integrity of a File System.....	7-17

Chapter 8: Monitoring Processes with CRM

8.1	Installing CRM	8-3
8.2	Invoking the Curses-Based FMLI Interface for CRM	8-4
8.2.1	Graphics Monitors.....	8-4
8.2.2	Alphanumeric Terminals.....	8-4
8.3	Using the FMLI Interface	8-5
8.4	System Monitors	8-7
8.4.1	Running System Monitors from the Command Line	8-7
8.4.2	Running Graphics-Based System Monitors	8-9
8.4.3	Using the System Monitors from the Curses-Based FMLI Menus	8-10
8.4.3.1	Change Defaults Form	8-10
8.4.3.2	Monitor Parameters (monparam)	8-11
8.4.3.3	Top Fault Monitor (topfault)	8-12
8.4.3.4	Top Memory Monitor (topmem)	8-14
8.4.3.5	Top CPU Monitor (topcpu)	8-15
8.4.3.6	Top I/O Monitor (topio)	8-15
8.4.3.7	Top System Monitor (topsys)	8-16
8.4.3.8	Show Open Files (showfiles)	8-17
8.4.3.9	Show Memory Usage (showmemory)	8-17
8.5	Process Monitors	8-19
8.5.1	Running Process Monitors from the Command Line	8-19
8.5.2	Using the Process Monitors Interactively from Curses-Based FMLI Menus	8-24
8.5.2.1	Select Process to Monitor	8-24
8.5.2.2	Select Profiler Options	8-25
8.5.2.3	Enable Page Faults	8-26
8.5.2.4	Enable System Calls	8-26
8.5.2.5	Run Profiler	8-27
8.5.2.6	Run Memory Monitor	8-28
8.5.2.7	Run Process Monitor	8-29

Chapter 9: Backing Up and Restoring Files

9.1	Determining a Backup Schedule	9-3
9.2	Devising a Backup Plan According to System Configuration	9-4
9.2.1	Full Network Configuration	9-4
9.2.2	Standalone Configuration	9-5
9.3	Backup Commands and Utilities	9-6
9.3.1	The dd Command	9-10
9.3.2	The tar Command	9-13
9.3.3	The cpio/scpio Commands	9-18
9.3.4	The backup and restore Commands	9-26
9.3.5	The ansitape Utility	9-36
9.3.6	The vmsbackup Utility	9-42
9.3.7	The rtc Utility	9-44
9.3.8	The compress Utility	9-47

9.4	Additional Commands for Floppy Disks	9-51
9.4.1	The format Command	9-52
9.4.2	The mkfs Command	9-54
9.4.3	The cpflop Command	9-56
9.4.4	The to_flop Command	9-56
9.4.5	The fr_flop Command	9-56

Chapter 10: Using the Utility Pages

10.1	Workstation Utility Menus	10-3
10.1.1	Main Utility Page	10-4
10.1.2	Peripheral Configuration Page	10-7
10.1.3	Operating System Parameters Page	10-11
10.1.4	Disk Maintenance Utility Page	10-13
10.1.5	Hard Disk Flaw Data Utility Page	10-16
10.1.6	Verify Utility Page	10-19
10.1.7	Format Utility Page	10-21
10.1.8	Disk Partitioning Utility Page	10-23
10.1.9	Device Image Copy Utility	10-26
10.1.10	Rebuild Utility Page.....	10-30
10.2	Server Utility Menus	10-32
10.2.1	Startup Configuration Utility Menu	10-34
10.2.2	Main Utility Menu	10-35
10.2.3	Date and Time Menu	10-36
10.2.4	Peripheral Configuration Menu	10-39
10.2.5	Operating System Parameters Menu	10-42
10.2.6	Disk Maintenance Utilities Menu	10-43
10.2.7	Disk Partitioning Menu.....	10-46
10.2.8	Device Image Copy Utility Menu	10-48
10.2.9	Rebuild Utility Menu.....	10-54

Chapter 11: Rebuilding and Repartitioning the Hard Disk

11.1	Backing Up Files	11-3
11.2	Creating Current Rebuild Floppy Disks	11-4
11.3	Booting from the Rebuild Boot Floppy Disk	11-7
11.4	Entering the Utility Pages	11-8
11.4.1	Entering the Utility Pages on a Workstation	11-8
11.4.2	Entering the Utility Pages on a Server	11-9
11.5	Verifying Hard Disk Flaws	11-10
11.5.1	Verifying a Workstation Hard Disk	11-11
11.5.2	Verifying a Server Hard Disk	11-12
11.6	Formatting the Hard Disk	11-13
11.6.1	Formatting Workstation Hard Disks	11-13
11.6.2	Formatting Server Hard Disks	11-14
11.7	Selecting the Rebuild Utility Page Icon	11-16

11.8	Repartitioning the Hard Disk	11-17
11.8.1	Using the Current Partition Table	11-18
11.8.2	Repartitioning Using Default Partitions	11-19
11.8.3	Repartitioning Using Custom Partitions	11-20
11.8.3.1	Partitioning Overview	11-20
11.8.3.2	Creating Nonstandard Partitions	11-27
11.9	Loading the Rebuild Root Media	11-31
11.10	Entering the Rebuild Environment	11-32
11.11	Restoring the File Systems	11-33
11.12	Loading Nucleus Software with newprod	11-37
11.12.1	Preparing the Hard Disk to Load Software from CDROM	11-37
11.12.2	Loading Software	11-38
11.13	Preparing Nonstandard Partitions for Use	11-40
11.14	Loading Application Software with the newprod Utility	11-41
11.15	Restoring Files from Backups	11-42
Appendix A: Keyboard Character Codes.....		A-1
Appendix B: Read/Write Compatibility of Floppy Disks with Intergraph Workstations and IBM PCs.....		B-1
Appendix C: System V/PC-DOS Utilities.....		C-1
Glossary.....		gl-1
Index.....		in-1



Chapter 1: Setting Up a New System

This chapter familiarizes the system manager with the responsibilities involved in setting up and maintaining an Intergraph workstation or server. Tasks that should be performed when a system is first set up are listed and briefly explained, and the reader is referred to the appropriate documents for more thorough documentation.

This chapter contains the following sections:

1.1	Setting Up and Adjusting the Hardware	1-3
1.2	Evaluating the Default Partitions	1-4
1.2.1	Default Partitions	1-4
1.2.2	Determining Partition Sizes	1-5
1.3	Configuring the Workstation Using the Utility Pages	1-6
1.4	Setting Up User Accounts	1-8
1.5	Assigning an Internet Address	1-9
1.6	Delivering Software	1-10
1.7	Determining a Backup Schedule	1-11
1.8	Using Intergraph System Documentation	1-12

1-2 *Setting Up a New System*

1.1 Setting Up and Adjusting the Hardware

When you receive an Intergraph workstation or server, your first concern will be setting up the system. Your Intergraph Field Representative will unpack and set up all Intergraph systems when they are delivered. Thus, you will not need to be concerned with this task.

However, once the systems have been set up, you can adjust the monitors, keyboards, and digitizing tables to meet individual (*ergonomic*) preferences. The *CLIPPER Hardware Setup and Maintenance Guide* (which is shipped with each Intergraph workstation or server) describes hardware adjustments that can be made to Intergraph workstations and also points out hardware main components and ports.

Please refer to the *CLIPPER Hardware Setup and Maintenance Guide* for information on setting up and maintaining the hardware.

1.2 Evaluating the Default Partitions

Your next concern in setting up a new system should be to evaluate hard disk partitions and, if necessary, repartition the hard disk. This section lists the default partitions that will already exist on a new system. It also provides background information to help you determine the partition sizes best suited for your system. For step-by-step instructions for repartitioning the hard disk, refer to Chapter 11, "Rebuilding and Repartitioning the Hard Disk."

1.2.1 Default Partitions

Intergraph ships all new systems with the following standard (or default) hard disk partitions:

	80 MB	156 MB	355 MB	584 MB	670 MB
boot:	3988	7988	7988	7988	7988
root:	25000	25000	25000	25000	25000
swap:	27360	71000	71000	71000	71000
PC-DOS:	5000	none	none	none	none
usr:	100000	200600	590298	1037988	1204900
usr2:	none	none	none	none	none

Note:

The total number of blocks in the partition table and the disk capacity differ. They differ because a one-block header is associated with each partition. The header blocks are included in the disk capacity count and not in the partition table count.

1.2.2 Determining Partition Sizes

The following partition requirements must be met for all Intergraph internal hard disks. You may change the hard disk partitions, but you must meet these requirements:

- The root partition must have at least 25,000 blocks.
- The swap partition must have at least 15,000 blocks.
- The usr partition must exist.

Additional partition requirements will differ from machine to machine, depending on the application software being used.

When you are determining partition sizes, you may want to consider the following:

- Will you be using InterPlot software? If so, you may need a **stash** partition. Refer to the *InterPlot User's Guide* for information on determining whether you need a **stash** partition and, if so, determining its size.
- Will you be using any application (such as EMS or DP/Publisher) that requires a certain amount of swap space? To check for application requirements, use the **review** option in the **newprod** utility. This option allows you to review information about a software product (such as fixes, features, open problems, comments, and installation help) before you load the product. To use this option, invoke **newprod**, move the cursor to the product for which you need to check partition requirements, and press **r**. You will need to review each application product that you will load on the system.
- Do you want to create a **usr2** or **usr3** partition? Often it is helpful to divide the hard disk into separate partitions (usually **usr2** and **usr3**) for user functions. However, the default partitions do not include **usr2** and **usr3** and no software products require them. Thus, you may divide your **usr** partition into separate **usr** partitions if you wish.

1.3 Configuring the Workstation Using the Utility Pages

Another task that you must perform when you first set up a system is to set certain system parameters such as system time, date, and nodename. These system parameters can be set using a series of menus called the *Utility Pages*. This section lists the parameters that need to be configured when a system is being configured initially. For documentation on all of the Utility Page parameters, refer to Chapter 10, "Using the Utility Pages."

To access the Utility Pages *on a workstation*, reboot the system and, at the introductory screen, move the cursor within five seconds. Then, select the Utility icon. To access the Utility Pages *on a server*, reboot the system and, at the introductory screen, press any key within five seconds. Then, key in UT at the menu.

You should set the following Utility Page parameters when you first receive an Intergraph system:

- **Workstation Password.** From the Main Utility Page, you can assign a password that restricts entrance to the Utility Page environment. It is important to restrict access to the Utility Pages because these menus allow you to format and repartition the hard disk. Only the system administrator should be allowed to access the Utility Pages.
- **System Date.** From the Main Utility Page, you can set the system date using the Month, Day, and Year, fields. When you first receive the system, you should set the system date.
- **System Time.** From the Main Utility Page, you can set the system time using the Timezone, Daylight Savings, and Current Time fields. When you first receive the system, you should set the system time.
- **Screen Saver.** From the Peripheral Configuration Utility Page, you can activate/deactivate the Screen Saver feature. Screen Saver causes the screen to dim automatically after a certain period of inactivity. You will probably wish to activate this feature to help preserve the screen.
- **Floating Menu Present.** From the Peripheral Configuration Utility Page, you must specify whether a floating menu tablet will be used.
- **Keyboard.** From the Peripheral Configuration Utility Page, you may specify to activate/deactivate the Key Click and Membrane Click (for the function keys) keyboard features and the volume level (none, soft, or loud) for the Bell Tone.

- **Serial Port Parameters.** From the Peripheral Configuration Utility Page, you can configure the system for peripheral devices such as an alphanumeric terminal or a modem. Refer to the *CLIPPER Hardware Setup and Maintenance Guide* for more information on configuring a system for peripheral devices.
- **Automatic Dump.** From the Operating System Parameters Utility Page, you can set the automatic dump switch on the system so that information pertinent to the cause of the crash will be provided.
- **Automatic Reboot.** From the Operating System Parameters Utility Page, you can determine whether the system will automatically reboot after a system crash.
- **Node Name.** From the Operating System Parameters Utility Page, you can specify the nodename for the system. When you first receive the system, you should assign a nodename, or a unique logical name, representing the unit's hard-coded network address. The nodename for an Intergraph workstation or server can be a maximum of eight alphanumeric characters. The nodename cannot contain any uppercase characters. Before you assign a nodename, check the Clearinghouse to ensure that the nodename is unique. For information on using the Clearinghouse, please refer to the *Intergraph Network Core User's Guide*.

1.4 Setting Up User Accounts

Before you can log in to a system, you must have a *user account*. Intergraph systems are delivered with certain system accounts that you can use to log in to the system initially. Once you have logged in using a system account, you can create your user account(s).

Please read this section to become familiar with the system accounts that exist on Intergraph systems. Then refer to Chapter 3, "Creating and Maintaining User Accounts," for step-by-step instructions for creating user accounts.

The following are system and administrative accounts that you can use to log in to a new Intergraph workstation or server:

- *sys* (log in to a local window)
- *root* (log in to the console window)
- *setup* (log in to the console window)
- *sysadm* (log in to the console window)

Note:

The *setup* and *sysadm* accounts are special accounts used to perform system administrative tasks such as setting up a new system and adding users.

1.5 Assigning an Internet Address

Another initial task is to assign an Internet address to your system. This task is necessary only if you will be using the TCP/IP, NFS/RFS, or NQS products. You will not be able to load these software products until you have assigned a valid Internet address to the node.

For information on assigning an Internet address, see the *CLIX TCP/IP User's Guide* (DSA025720) or Appendix A in the *NQS User's Guide* (DSA025410).

1.6 Delivering Software

The next task is to load software on your system(s). Intergraph workstations and servers are delivered with a set of *nucleus* products. These products are the minimum products required to run the System V operating system. Although these products will already reside on a new system, you must load any additional application software products that you purchased. Please refer to the *CLIPPER Software Delivery Guide* (DSA026420) for step-by-step instructions on delivering software products.

1.7 Determining a Backup Schedule

A very important initial task for new systems is to determine and implement a backup schedule. Please refer to Chapter 9, "Backing Up and Restoring Files," for guidelines for determining the backup schedule and for choosing the backup utility to use.

1.8 Using Intergraph System Documentation

Documentation for Intergraph workstation nucleus software products include the following:

CLIPPER User's Guide (DSA030010)
CLIPPER System Administrator's Guide (DSA029910)
CLIPPER Hardware Setup and Maintenance Guide (DSA025121)
CLIPPER Software Delivery Guide (DSA026320)
Intergraph Network Core User's Guide (DSA030110)
InterPlot User's Guide (DPA025320)
IGDS Plotting Guide (DPA029610)
Network Queuing System User's Guide (DSA025411)
CLIPPER Release News (DSA026920)

The following chart specifies the intended audience and purpose for each of these manuals.

Document	Purpose
<i>CLIPPER User's Guide</i> (DSA030010)	To provide new users with a basic understanding of the workstation interface, including the CLIX operating system.
<i>CLIPPER System Administrator's Guide</i> (DSA029910)	To provide system administrators with an understanding of the tasks involved in managing an Intergraph system.
<i>CLIPPER Hardware Setup and Maintenance Guide</i> (DSA025131)	To provide workstation users with instructions for adjusting hardware components such as monitors and digitizing tables to meet individual ergonomic requirements. In addition, this document contains information on maintaining hardware components.
<i>CLIPPER Software Delivery Guide</i> (DSA026320)	To provide system administrators with instructions for setting up a delivery source and delivering software to Intergraph systems.
<i>Intergraph Network Core User's Guide</i> (DSA030110)	To provide workstation/server users with instructions for using the network utilities available on Intergraph systems.

Document

Purpose

InterPlot User's Guide
(DPA025320)

To provide workstation/server users with instructions for using the utilities available on Intergraph systems to print jobs.

IGDS Plotting User's Guide
(DPA029610)

To provide IGDS plotting users with instructions for using the VPLOT, IPLOT, and VAXpath interfaces.

Network Queuing System User's Guide
(DSA025410)

To provide workstation/server users with instructions for using the NQS product to print.

CLIPPER Release News
(DSA026920)

To inform workstation/server users of the software changes being implemented in the current software release.



Chapter 2: System Startup and Initialization

This chapter describes the processes invoked when a workstation or server is booting into the System V operating system.

This chapter contains the following sections:

2.1	The /etc/init Process	2-3
2.2	The /etc/getty Process	2-5
2.3	The /etc/rc2 Process	2-6
2.3.1	The /etc/init.d Directory	2-7
2.3.2	Initializing Additional Processes from Multiuser Mode	2-7

2.1 The /etc/init Process

The first process executed in System V version 3.1 is */etc/init*. The **init** process controls System V initialization.

The **init** process invokes, or *spawns*, other processes. **init** is invoked with one of seven arguments called *run levels*. The current run level determines the processes that **init** spawns. A run level is the current mode of system operation. The following chart lists the system run levels (arguments for **init**) and describes their associated mode.

Run Level	Operation
0	shutdown
1	single-user
2	multiuser, NFS support
3	multiuser, RFS support
4	not used
5	not used
6	reboot

For example, when the system is rebooting, it is operating at run level 6. When the workstation is running a MicroStation application and using NFS, it is functioning at run level 2.

The **who -r** command displays the current run level, as shown in the following example:

```
$ who -r
. run-level 2  May 29 14:44  2  0  S
$
```

The **init** process reads the records in the */etc/inittab* file to determine the default run level and processes to invoke.

The `/etc/inittab` file associates processes with run levels. Study the following sample `inittab` file:

```
nt::sysinit:/etc/incl.start >/dev/console 2:&1
fs::bootwait:/etc/bcheckrc </dev/console >/dev/console 2:&1
mt::bootwait:/etc/brc </dev/console >/dev/console 2:&1
sf:23:bootwait:/etc/sfsck -y </dev/console >/dev/console 2:&1
is:2:initdefault:
p3s1234:powerfail:/etc/shutdown -y -i0 -g0 >/dev/console 2:&1
s0:056:wait:/etc/rc0 >/dev/console 2:&1 </dev/console
sl:1:wait:/etc/shutdown -y -iS -g0 >/dev/console 2:&1 </dev/console
n1:1:wait:/etc.incl.start >/dev/console 2:&1
of:0:wait:/etc/uadmin 2 0 >/dev/console 2:&1 </dev/console
RB:6:wait:echo "\nThe system is being rebooted." >/dev/console 2:&1
rb:6:wait:/etc/uadmin 2 1 >/dev/console 2:&1 </dev/console
co:234:respawn:/etc/getty console console
t0:234:respawn:/etc/getty tty00 9600 none
t1:234:off:/etc/getty tty01 9600 none
t2:234:off:/etc/getty tty02 9600 none
s2:23:wait:/etc/rc2 >/dev/console 2:&1 </dev/console
s3:3:wait:/etc/rc3 >/dev/console 2:&1 </dev/console
```

Each line, called a *record*, invokes a process. For example, the following *record* invokes the `/etc/rc2` process:

```
s2:23:wait:/etc/rc2 >/dev/console 2:&1 </dev/console
```

The following describes each field in this record:

- `s2` is the unique identifier
- `23` is the run level(s) at which the process executes
- `wait` is the action status that determines when to execute the process
- `/etc/rc2 >/dev/console 2:&1 </dev/console` is the process that `init` executes

The second field indicates when (at which run level) the process in the last field is invoked. A process is executed at all run levels (all the time) if the second field is blank.

In the sample `/etc/inittab` file, `init` will execute the processes in the first three records because the second field is blank. The `init` process then continues to read the `/etc/inittab` file and looks for the `initdefault` record (the fourth record in the sample file). The `initdefault` record specifies the default run level, which is 2 in this case. The `init` process then executes all other `/etc/inittab` records associated with run level 2 (those that have a 2 in the second field). Some of the processes invoked by `init` are described in the following sections.

For more information on `init` and the processes invoked in the `/etc/inittab` file, refer to the *CLIX System Administrator's Reference Manual* and the *AT&T System V System Administrator's Reference Manual*.

2.2 The /etc/getty Process

The */etc/getty* process opens a device and spawns a login process to that device. For example, *getty* sets up a terminal line between workstation windows and the operating system or between CTTs and the operating system.

The *getty* process reads the */etc/gettydefs* file, which contains information on setting the terminal type, modes, speed, and line discipline. The *gettydefs* file also contains the default login prompt for the device.

The *gettydefs* file is commonly modified so that the login prompt that appears on the device is changed. For example, if your configuration includes several servers with alphanumeric terminals connected to them, you can modify the login prompts so that they uniquely identify the server. For example, you can modify a console login prompt to appear as "IS4000 console>>".

2.3 Modifying the *inittab* and *gettydefs* Files for a 32-Channel RS-232 Interface Assembly

If your workstation/server is equipped with a 32-channel RS-232 interface assembly, you may need to modify the */etc/inittab* file before attaching external devices that require a *getty* process to be running. (Any device that requires a login prompt requires a *getty* process to be running.) To determine if your workstation/server is equipped with a 32-channel RS-232 interface assembly, use the *showconfig* command to list the boards in the system. If the 32-channel serial interface board (MPCB821) is listed, the workstation/server is equipped with the 32-channel interface assembly.

If the device you are attaching is not supported by the current *gettydefs* settings, the */etc/gettydefs* file may also need to be modified. The 9600 baud rate (and its associated settings) is the default *gettydefs* setting and is suitable for most terminal devices. Refer to the information delivered with the external device being connected to determine the correct settings.

2.3.1 Modifying the */etc/inittab* File

In this subsection the terms *respawn* and *off* refer to how the port interprets the device attached to it. For example, a device (such as an alphanumeric terminal) that requires a login prompt requires the *respawn* delimiter. A device (such as a printer or plotter) that does not require a login prompt should use the *off* delimiter. A modem should be set to *respawn* when used for dialing in only and *off* when used for dialing out only.

Note:

For more information on the *inittab* and *gettydefs* files, refer to the *inittab(4)* and *gettydefs(4)* entries in the *AT&T UNIX System V Programmer's Reference Manual*. For more information on the *getty* process, see the *AT&T UNIX System V Administrator's Reference Manual*.

The */etc/inittab* file must include an entry beginning with "r" followed by the number of the port (0-31) you are connecting a device to. Proceed through the following steps to modify the */etc/inittab* file:

1. Access the superuser account by keying in *su -* at the system prompt.

Note:

Be sure to include the "-" after the "su" to access the login environment of root.

2. Change to the */etc* directory.
3. Edit the *inittab* file and create an entry corresponding to the port the device is being connected to. Be sure the device parameters entered (for example, 9600 baud rate in the following examples) match those of the device being connected. The format for the *inittab* entry is as follows:

id:state:action:process

Notes:

For more information on the fields for the *inittab* entries, refer to the *inittab(4)* and *gettydefs(4)* entries in the *AT&T UNIX System V Programmer's Reference Manual* and the entry for the *getty* entry in the *AT&T UNIX System V System Administrator's Reference Manual*.

In the following examples, "9600" is a label that refers to the associated baud rate entry in the *gettydefs* file.

- For a device requiring the respawn delimiter, an example of the syntax for port 0 is as follows:

r0:234:respawn:/etc/getty ttr00 9600 none

An example of the syntax for port 1 is as follows:

r1:234:respawn:/etc/getty ttr01 9600 none

- For a device requiring the off delimiter, an example of the syntax for port 0 is as follows:

r0:234:off:/etc/getty ttr00 9600 none

An example of the syntax for port 1 is as follows:

r1:234:off:/etc/getty ttr01 9600 none

4. Continue adding the appropriate line for each port being connected.
5. Initialize the new *inittab* settings by keying in *init q* at the superuser prompt.

2.3.2 Modifying the */etc/gettydefs* File

If you are connecting a device whose parameter setup is not supported in the */etc/gettydefs* file, you need to modify the file to support the new device. The new entry should match the terminal settings (such as baud rate, local echo, stop bits, or data bits) of the device being connected. Refer to the information delivered with the device to determine its correct settings.

Note:

Refer to the *termio(7S)* entry in the *CLIX System Administrator's Reference Manual* for more information on terminal settings.

Proceed through the following steps to modify the */etc/gettydefs* file:

1. Access the superuser account by keying in *su -* at the system prompt.

Note:

Be sure to include the "-" after the "su" to access the root login environment.

2. Change to the */etc* directory.
3. Edit the *gettydefs* file and create an entry whose parameters correspond to those of the device being connected.

Note:

When editing the file, verify that the entry contains no carriage returns before the end of the line.

2.4 The /etc/rc2 Process

The */etc/rc2* file is a shell script executed (spawned) on run level 2 (multiuser environment). The */etc/rc2* file starts the processes for run level 2. The processes invoked by *rc2* reside in files in an associated directory, */etc/rc2.d*.

The */etc/rc2.d* directory contains startup processes for run level 2. Each startup file starts with an S and is followed by a two-digit priority number. The following is a sample listing of the *rc2.d* directory:

```
# ls /etc/rc2.d
S01MOUNTFSYS  S04errod          S70uucp          S85inc
S01SWAPADD    S05RMTMPFILES    S75cron          S90nqs
S02DODINI     S20stssetup      S80mmgr
```

These startup files are shell scripts. These shell scripts invoke standard processes (such as setting up and mounting the */usr* file system(s), cleaning up and removing the */tmp* and */usr/tmp* directories, and starting cron) that must be started for run level 2. In addition, they initialize products (such as the Screen Manager and Intergraph Network Core (INC)) specific to the Intergraph multiuser environment.

Any other processes that need to be initialized when the system boots into multiuser mode (run level 2) must be linked from the */etc/init.d* directory to the *rc2.d* directory.

Note:

"Initializing Additional Processes from Multiuser Mode" in 2.4.2 describes how to link and initialize processes such as *lp*, *lpr*, *perf*, and accounting so they start automatically in multiuser mode.

2.4.1 The /etc/init.d Directory

All startup and kill processes invoked at system initialization or shut down are stored in one common directory, */etc/init.d*. The files in *rc2.d* and *rc3.d* are linked to */etc/init.d* so they are executed at the specified run level. The *init.d* directory acts as a common holding place for startup and kill shell scripts.

Some startup and kill files, such as accounting, *lp*, *lpr*, and *perf* may not be linked to the *rc<n>.d* directory, and, therefore, will not be started automatically. The following section describes how to add processes to *rc2.d* so they are automatically started at bootup.

2.4.2 Initializing Additional Programs from Multiuser Mode

The startup files for accounting, lp (line printer spooling utilities), lpr (Berkley printer spooling utilities), and perf (performance measurement utilities) are stored in the */etc/init.d* directory but are not linked to a multiuser directory (*/etc/rc2.d* or */etc/rc3.d*) by default.

Follow these steps to link accounting, lp, lpr, or perf to a multiuser directory so they are invoked automatically when the system boots:

1. Access the superuser account and change to the */etc/init.d* directory as follows:

```
# cd /etc/init.d
```

2. Key in the program filename followed by *init*, as shown in the following example:

```
# /lp init
# /acct init
# /lpr init
# /perf init
```

The programs are now linked to the *rc2.d* directory and will be automatically invoked when you enter run level 2. They are automatically assigned a new filename and priority number.

Note:

For information on lpr refer to the *CLIX Programmer's and User's Reference Manual*. For information on the lp command refer to the *AT&T UNIX System V User's Reference Manual*. For information on accounting refer to the *AT&T UNIX System V Programmer's Reference Manual*. For information on the perf program refer to the *sa(1)* and *sa(1M)* commands in the *AT&T UNIX System V System Administrator's Reference Manual*.



Chapter 3: Creating and Managing User Accounts

Before you can add user accounts to a new system, you must be able to log in to the system. Several system and administrative accounts are available for this purpose. This chapter begins by listing and describing these accounts. Then, procedures for adding and removing user accounts and administering passwords are provided. In addition, this chapter discusses file protection codes and how they apply to user accounts.

This chapter contains the following sections:

3.1	System and Administrative Accounts	3-3
3.1.1	The root Account	3-3
3.1.2	The sysadm Account	3-3
3.1.3	The setup Account	3-4
3.1.4	The sys Account	3-4
3.1.5	The nuucp Account	3-4
3.2	Adding User Accounts	3-5
3.2.1	Using addusers to Add User Accounts	3-6
3.2.2	Using sysadm to Add User Accounts	3-8
3.3	Removing User Accounts	3-10
3.3.1	Using rmusers to Remove User Accounts	3-10
3.3.2	Using sysadm to Remove User Accounts	3-11
3.4	Assigning, Changing, and Removing Passwords	3-13
3.4.1	Restrictions on Passwords	3-13
3.4.2	Creating a Password	3-13
3.4.3	Forcing a Password Change	3-14
3.4.4	Removing a Password	3-15
3.4.4.1	Removing a Password from User Accounts	3-15
3.4.4.2	Removing the root Password	3-15
3.4.4.3	Removing the Utility Pages Password	3-18
3.5	Using Protection Codes to Manage Files	3-19
3.5.1	File Protection Codes	3-19
3.5.1.1	Directory Files	3-20
3.5.1.2	Ordinary Files	3-21
3.5.2	Modifying File Permissions	3-21
3.5.3	Modifying File Ownership	3-22
3.5.4	Changing Group Access	3-23
3.5.4.1	Changing Your Active Group	3-23
3.5.4.2	Changing the Group of a File	3-24

3-2 Creating and Managing User Accounts

3.1 System and Administrative Accounts

The following are system and administrative login accounts. The system administrator will need to log in using one of these accounts to set up user accounts on a new system.

- root*
- sysadm*
- setup*
- sys*

Each of these accounts is described in the following sections.

3.1.1 The root Account

The *root* account is a special account that only the system administrator should access. This account (also known as the *superuser* account) has special privileges, including access to all system files and resources. Thus, this account should be passworded at all times.

You can gain superuser privileges by logging in as *root* in the console window or by issuing the *su* (set user) command at the system (\$) prompt. The default prompt is # and the home directory is the *root* (/) directory when you are working in superuser mode.

Note:

Any user who knows the *root* password can key in *su* at the system prompt and gain superuser privileges. To maintain system security, keep the *root* password confidential. Only users who should have system management privileges should know the *root* password.

3.1.2 The sysadm Account

The *sysadm* account is an administrative account that invokes the *sysadm* utility. You should restrict access to this account by assigning a password to it because this utility provides a mechanism for adding and removing users from the system.

You must log in to this account through the console window. The *sysadm* account uses the restricted shell (*rsh*). When you log in, the *sysadm* utility is invoked automatically, and when you quit this utility, you will automatically be logged-out. Thus, when you log in using this account, you can access only the *sysadm* utility.

The **sysadm** utility is a menu-driven utility that allows you to perform file management, machine management, package management, system setup, and user management tasks. For more information on this utility, refer to the AT&T System V manual page for **sysadm(1M)**.

3.1.3 The **setup** Account

The **setup** account is an administrative account that invokes the **setup** utility. You should restrict access to this account by assigning a password to it because this utility establishes the current user as the *owner* of the machine.

You must log in to this account through the console window. The **setup** account uses the restricted shell (**rsh**). When you log in, the **setup** utility is invoked automatically, and when you quit this utility, you will automatically be logged out. Thus, when you log in using this account, you can access only the **setup** utility.

The **setup** utility is a menu-driven utility that is actually a part of the **sysadm** utility. The **setup** utility allows you to set the system time and date, system owner, and system and administrative passwords. This utility is intended to be used for initial system setup tasks. For more information on this utility, refer to the AT&T System V manual page for **setup(1)**.

3.1.4 The **sys** Account

The **sys** account can be used to log in to the console window or a local window. This account does not assume superuser privileges. The home directory for **sys** is **/usr/src**.

3.1.5 The **nuucp** Account

The **nuucp** account is not used as a login account. **nuucp** allows a node to receive electronic mail. Thus, do not assign a password to **nuucp** unless you wish to restrict the amount of mail you receive to only those people who know your **nuucp** password.

3.2 Adding User Accounts

One primary duty for the system manager is to set up and maintain user accounts.

The following steps are required to add user accounts to the system. You can perform these steps manually, or you use the `addusers` or `sysadm` utility to automate the process.

1. Create a new user record for the account in the `/etc/passwd` file.
2. Create a default user directory (`/usr/<logln>`) for the new user.
3. Change ownership of the new user directory.

Note:

When you add user accounts, the username must consist of eight or fewer alphanumeric characters, beginning with an alphabetic character.

The `addusers` utility has the following advantages:

- `addusers` copies skeleton files (`env`, `exrc`, and `.profile`) to the new user directory, and `sysadm` does not.
- `addusers` allows you to specify the login shell, whereas `sysadm` does not.
- If you specify the new username as an argument to the `addusers` command (as in `addusers john`), the user account will be added and you will not be prompted for information. If you normally accept all installation defaults, this feature allows you to add user accounts easily and quickly.

The `sysadm` utility has the following features:

- `sysadm` allows you to edit or skip (cancel without saving) a user account before it is saved, whereas `addusers` does not allow input errors.
- `sysadm` allows you to assign a password to the new user account, whereas `addusers` does not.
- `sysadm` allows you to assign a user ID number to the new account, whereas `addusers` does not.

This subsection describes how to use the `addusers` and `sysadm` utilities to add user accounts.

3.2.1 Using `addusers` to Add User Accounts

The `addusers` utility automates the process of adding user accounts to the system. Follow these steps to use this utility:

1. Access the superuser account by keying in `su` at the system prompt.
2. To initiate the `addusers` utility, key in the following:

```
# addusers
```

Note:

As an alternative, if you specify the new username as an argument to the `addusers` command, the user account will be added and you will not be prompted for information. If you normally accept all installation defaults, this feature allows you to add user accounts easily and quickly.

```
# addusers john
Username "john" added
```

3. The utility prompts for the new login name(s). The login name can contain a maximum of eight characters.

You can add one or many user accounts. When `addusers` adds multiple user accounts, it assigns the same group name, user information, login directory (such as `/usr` or `/usr2`), and login shell for all new users. If you need to set up the user accounts differently for the new users, add each account separately.

To add one user account, key in the new username as follows:

```
New login name(s) → scott
```

To add multiple user accounts, key in the new usernames separated by spaces as follows:

```
New login name(s) → donna deana paul
```

4. The `addusers` utility prompts you for the following information:

- **Group name? [users]**

The group name determines the group identification number (GID) in the `/etc/passwd` file. The default group name is `users`, which corresponds to the GID 500.

- **User info? [new user]**

The user information that you supply will be placed in the miscellaneous information field (the fifth field) in the */etc/passwd* file. This information is used for identification. For example, if the system manager is checking the */etc/passwd* file, information such as employee department, position, and phone number might be useful. User information defaults to "new user."

- **Login directory? [/usr]**

Login directory defaults to */usr/<login>*, where *<login>* represents the new login name. However, the system manager may place a user in */usr2*, */usr3*, or any other file system.

- **Login shell? [/bin/ksh]**

The login shell is the UNIX shell that the user will log in under. The default is the Korn shell (*/bin/ksh*), but the system manager may choose to have users log in under the Bourne shell (*/bin/sh*) or the C shell (*/bin/csh*). The Bourne shell is the standard System V shell; the Korn shell is very similar to the Bourne shell but has additional features such as command recall; the C shell has a syntax similar to the C programming language.

After you have responded to all prompts, the following message appears for each user added:

```
Username "scott" added
```

When accounts are created with the *addusers* utility, the new user will automatically be prompted for a password when he or she logs in. For more information on creating passwords, refer to 3.4.2.

3.2.2 Using `sysadm` to Add User Accounts

The `sysadm` utility also automates the process of adding user accounts to the system. Follow these steps to use this utility:

1. Access the superuser account by keying in `su` at the system prompt.
2. To initiate the `sysadm` utility, key in the following:

```
# sysadm
```

3. The following menu appears:

SYSTEM ADMINISTRATION

1 filegmt	file management menu
2 machinmgmt	machine management menu
3 packagemgmt	package management menu
4 syssetup	system setup menu
5 usermgmt	user management menu

To add user account(s), key in 5.

4. The following menu appears:

USER MANAGEMENT

1 addgroup	add a group to the system
2 adduser	add a user to the system
3 delgroup	delete a group from the system
4 deluser	delete a user from the system
5 lsgroup	list groups in the system
6 lsuser	list users in the system
7 modadduser	modify defaults used by adduser
8 modgroup	menu of commands to modify group attributes
9 moduser	menu of commands to modify a user's login

To add a user, key in 2.

5. The `sysadm` utility prompts for the following information. If you wish to quit the utility (without adding the user), key in `q` at any of the following prompts. If you wish to see additional information on any prompt, key in `?` at that prompt.

- Enter user's full name [`?`, `q`]:

This information is not the login name. The information will be placed in the `/etc/passwd` file's miscellaneous information field. For example, you might enter `John P. Smith` in this field.

- **Enter user's login ID [?, q]:**

The login ID is the username. The login ID can be any combination of alphanumeric characters, beginning with an alphabetic character. A minimum of one and a maximum of eight characters are allowed.

- **Enter user ID number (default #) [?, q]:**

The user ID number (UID) will determine file ownership. Each user has a different UID. You can derive the default UID (in parentheses) by adding one to the highest UID on the system. If you specify the UID, ensure that the number you choose is unique to that user account.

- **Enter group ID number or group ID name (default 1) [?, q]:**

The group ID number (GID) or group name places users in groups. Users may access files of other users in the same group. The `users` group has a GID of 500.

- **Enter user's login (home) directory name (default `/usr/username`) [?, q]:**

The default login directory for the new user is `/usr/<username>`, where `<username>` represents the new user account. However, the system manager may place a user in `/usr2`, `/usr3`, or any other file system.

- **Do you want to install, edit, or skip this entry [i, e, s, q]?**

If you wish to add the user account, key in `i` to install the new account.

- **Do you want to give the user a password? [y, n]**

Key in `y` to assign a password. At the new password prompt, key in the password for the new user account. This password must have a minimum of six and a maximum of eight characters. At least two characters must be alphabetic and at least one must be numeric (or a special character).

The following section describes how to manually assign a password to user accounts

- **Do you want to add another login? [y, n, q]**

Key in `y` to add another user account. Key in `n` to return to the User Management menu. Key in `q` to quit `sysadm`.

3.3 Removing User Accounts

The following steps are required to remove user accounts:

1. Remove the user's directory.
2. Remove the user's record from the */etc/passwd* file.

You can perform these steps manually or you can use the *rmusers* or *sysadm* utilities, which automate the process:

- The *rmusers* utility asks whether you wish to remove the user's directory. You can remove the user's login ID without removing the corresponding directory.
- The *sysadm* utility removes the user's login ID and directory.

Once a user account has been removed, you can no longer log in under the username and password for that account.

The following sections provide procedures for removing users with the *rmusers* and the *sysadm* utilities.

3.3.1 Using *rmusers* to Remove User Accounts

The *rmusers* utility automates the process of removing user accounts from the system. Follow these steps to use this utility:

1. Access the superuser account by keying in *su* at the system prompt.
2. To initiate the *rmusers* utility, key in the following:

```
# rmusers
```

3. This utility displays the following prompt:

```
Do you wish to delete the user's directories? (y,n)
```

When you remove the user account, you will no longer be able to log in under that username. However, you can keep the user directory (containing any files that the user saved). To remove the user directory and files, key in *y*. To save the user directory and files, key in *n*.

4. The utility prompts for the login name(s) to remove as follows. You can remove one or multiple user accounts.

Remove login name(s) ->

Key in the name (or names separated by spaces) of the user accounts to remove.

5. The utility removes each user account specified and exits back to the # prompt.

3.3.2 Using sysadm to Remove User Accounts

The `sysadm` utility also automates the process of removing user accounts from the system. Follow these steps to use this utility:

1. Access the superuser account by keying in `su` at the system prompt.
2. To initiate the `sysadm` utility, key in the following:

```
# sysadm
```

3. The following menu appears:

SYSTEM ADMINISTRATION

1 filemgmt	file management menu
2 machinmgmt	machine management menu
3 packagemgmt	package management menu
4 syssetup	system setup menu
5 usermgmt	user management menu

To remove user account(s), key in 5.

4. The following menu appears:

USER MANAGEMENT

1 addgroup	add a group to the system
2 adduser	add a user to the system
3 delgroup	delete a group from the system
4 deluser	delete a user from the system
5 lsgroup	list groups in the system
6 lsuser	list users in the system
7 modadduser	modify defaults used by adduser
8 modgroup	menu of commands to modify group attributes
9 moduser	menu of commands to modify a user's login

To remove a user, key in 4.

5. The following message displays:

This function COMPLETELY REMOVES THE USER, their mail file, home directory, and all files below their home directory from the machine.

Once this is done, there is no way guaranteed to get them all back.

BE SURE THIS IS WHAT YOU WANT TO DO!

Enter the login ID you wish to remove [q]:

Enter the username of the user to remove or **q** to quit.

6. The following additional prompt appears:

Do you want to remove login ID 'username'? [y, n, ?, q]

Key in **y** to remove the login ID specified. Key in **n** to return to the User Management menu. Key in **?** for help responding to the prompt. Key in **q** to quit the utility.

7. If you keyed in **y** to remove the user account, **sysadm** removes the specified account and displays the following prompt:

Enter login ID you wish to remove [q]:

8. To remove additional user accounts, key in the next login ID that you will remove. To quit the utility, key in **q**.

3.4 Assigning, Changing, and Removing Passwords

This section explains how to create, change, and remove a password. The `passwd` command can be used to add a new password to a login name or change an existing password. A user can change only the password that corresponds to his or her login name, whereas a superuser can change any password. Restrictions on the length and characters in a password are defined in 3.4.1.

When you assign a password, the password is not echoed on the screen. The system encrypts the password and adds it to the user account record in the `/etc/passwd` file. Each time you log in, the system compares the login name and password to the information in the `/etc/passwd` file. The login process and the `/etc/passwd` file are detailed in Chapter 4, "Modifying the Login Environment."

Note:

The passwords in the `/etc/passwd` file are encrypted. Thus, you cannot add a password simply by editing the `/etc/passwd` file.

3.4.1 Restrictions on Passwords

If the superuser creates or changes the password, the only restriction is that passwords can consist of a maximum of eight characters. Any additional characters will be truncated.

The following restrictions apply for passwords created or changed by the user:

- The password must consist of at least six characters.
- At least two characters must be alphabetic and at least one must be a numeric or special character.
- The password must differ from the login name and from the old password by at least three characters.
- Passwords can have a maximum of eight characters. Any additional characters will be truncated.

3.4.2 Creating a Password

The `passwd` command creates and changes a password. As the system administrator, you can either assign passwords to users or let them assign their own. The password will not display as it is keyed in at the prompt.

The following example creates a new password:

```
login: kelley
$ passwd
Changing password for kelley
New password:
Re-enter new password:
```

The following example changes an existing password:

```
login: kelley
password:
$ passwd
Changing password for kelley
Old password:
New password:
Re-enter new password:
```

3.4.3 Forcing a Password Change

To maintain system security, you should force users to periodically change their passwords. As system administrator, you can force a password change by editing the */etc/passwd* file, removing the second field (the encrypted password) for each user account, and placing a „ (comma dot) in the second field. Users will be prompted to enter a new password when they log in. The */etc/passwd* file is described in detail in Chapter 4, “Modifying the Login Environment.”

Follow these steps to force a password change:

1. Access the superuser account by keying in `su` at the system prompt.
2. Edit the */etc/passwd* file with the text editor of your choice. Delete the contents of the password field; however, be careful that you do not delete the colons. Then, key in „ (comma dot) in the password field as shown in the following example:

```
kelley:,:72:500:new user:/usr/kelley:/bin/ksh
```

3. Exit and save the file.

The next time user `kelley` logs in, she will be prompted to key in a new password.

3.4.4 Removing a Password

To remove a password, you must remove the contents of the password field (the second field) from the */etc/passwd* file. Refer to Chapter 4, "Modifying the Login Environment," for a description of the */etc/passwd* file.

- If you do not remember the root password and you wish to remove it, follow the procedures in 3.4.4.2, "Removing the root Password."
- If you do not remember the root password and the Utility Pages are passworded, follow the procedures in 3.4.4.3, "Removing the Utility Pages Password."

3.4.4.1 Removing a Password from User Accounts

Follow these steps to remove a password. In this example, the password for kelley is removed.

1. Access the superuser account by keying in `su` at the system prompt.
2. Edit the */etc/passwd* file with the text editor of your choice. Delete the contents of the password field for your user account. Be careful that you do not delete the colons as shown in the following example:

```
kelley::72:500:new user:/usr/kelley:/bin/ksh
```

3. Exit and save the file.

User kelley can now log in without being prompted for a password.

3.4.4.2 Removing the root Password

Use these guidelines to remove the root password:

- Follow the previous procedures (3.4.4.1) if you know the root password.
- If you do not know the root password and the Utility Pages are passworded, follow the procedures in 3.4.4.3.
- If you do not know the root password, you will be unable to access the superuser account (as described in the previous section); thus, you will be unable to edit the */etc/passwd* file. Follow these steps:

1. Reboot the workstation/server.
2. Access the Utility pages by selecting the Utility screen button on the Introductory screen (for workstations) or by keying in **UT** at the Introductory screen (for servers).
3. From the Main Utility page, access the Operating System Parameters page by selecting the Operating System Parameters screen button (for workstations) or by keying in **OS** (for servers).
4. On the Operating System Utility page, change the value for the Boot Single User parameter to **yes** by toggling the Boot Single User box to Yes (for workstations) or by keying in **SU Y** (for servers).
5. Load System V by selecting the System V screen button (for workstations) or by keying in **UN** (for servers).
6. The workstation/server boots to single-user mode, which places you in superuser mode. Issue the **passwd** command to change the password for root. You will be prompted to key in a new password, as shown in the following example:

```
# passwd
Changing password for root
New password: sysmgr32
Re-enter new password: sysmgr32
#
```

Note:

When you key in the new password, the text will not display on the screen.

7. You can use the **cat** command to view the contents of the */etc/passwd* file.

```
$ cat /etc/passwd
root:sFghYWagtTrM:0:3:0000-Admin(0000):/bin/ksh
daemon:1:12:0000-Admin(0000):/bin:
bin:2:2:0000-Admin(0000):/bin:
sys:3:3:0000-Admin(0000):/usr/src:
adm:4:4:0000-Admin(0000):/usr/adm:
uucp:5:1:0000-uucp(0000):/usr/lib/uucp:
sysadm:0:0:administration login:/usr/admin:/bin/ksh
sync:20:1:0000-Admin(0000):/bin/sync
lisa:Cw32S1rHIvg1Q:502:500:new user:/usr/lisa:/bin/ksh
kelley:3Y9xSI83rA:72:500:new user:/usr/kelley:/bin/ksh
```

The password field for root contains the encrypted password, **sFghYWagtTrM**.

8. Boot the system into multiuser mode by keying in one of the following command lines:

init 2

OR

/etc/reboot

The system automatically boots and the root password is removed.

3.4.4.3 Removing the Utility Pages Password

The following procedure describes how to remove the Utility Pages password if you do not know what it is.

As described in the previous section, you can remove the root password by booting the workstation/server into single-user mode from the Utility Pages. However, if the Utility Pages require a password that you do not know, you will first need to remove the password from the Utility Pages.

To remove the Utility Page password and boot the system to single-user mode, you must boot from the rebuild floppy as follows:

1. Insert the Rebuild Boot (#1) floppy in the floppy disk drive.
2. Turn on the workstation/server or reboot. The system will boot from the rebuild floppy instead of from the hard disk.
3. When the blue Introductory Screen appears, select the Utility option to enter the Utility Pages.
4. From the Main Utility Page, select the Workstation/Server Password field and press <Return> twice. This action deletes the current Utility Pages password.
5. Select the Save button to save the change (the deleted password).
6. Select the System V button to boot into System V.
7. Remove the Rebuild Floppy from the floppy disk drive.

The Utility Pages are no longer passworded. If you need to remove the root password, continue with the steps in subsection 3.4.4.2.

3.5 Using Protection Codes to Manage Files

As system administrator, you are responsible for managing all user accounts on the system. The best way to allow or restrict access to a user is through the use of file protections.

File protection codes control how other users access and manipulate files in System V. This section describes file protection codes and how you modify them.

Not all users can access all files on the workstation. Your ability to access a file depends on the following factors:

- Your relationship to the file. Your relationship to the file determines whether you will belong to the “user,” “group,” or “other” category.
 - If you are the file’s user, or owner (you created or copied the file), your category is “user.”
 - If you are in the same group as the file’s owner, your category is “group.”
 - If you are not the file’s owner or are not in the same group as the file’s owner, your category is “other.”
- The file permissions assigned to your category.

3.5.1 File Protection Codes

Each file (directory or ordinary) has a set of associated protection codes, or *permissions*. The file protection code contains three separate fields:

- User
- Group
- Other

```
rwX rwX rwX
|   |   |
|   |   other
|   group
user
```

A file’s protection codes display when you list a file with the `ls -l` command. For example, the long listing for the `/usr/lp32/inc` directory displays as follows:

```
drwxrwxrwx 3 root    root    816 May 12 12:54  inc
```

The first field of the long listing contains the file type and protection codes. In the previous example, `d` (directory) is the file type. Four file types exist: directory (`d`), character special file (`c`), block special file (`b`), and ordinary (`-`).

The file protections for the INC product directory (`/usr/ip32/inc`) are `rwXrwXrwX`. This character string indicates read (`r`), write (`w`), and execute (`x`) permissions for each user category (user, group, and other).

```
rwX rwX rwX
|   |   |
|   |   other
|   group
user
```

The read, write, and execute (`rwX`) permissions determine how each user type can access and manipulate a file. Each user type can have one of the following combinations:

- All privileges (`rwX`)
- A combination of privileges (`rw-`)
- No privileges (`---`).

File manipulations differ for directory and ordinary files.

3.5.1.1 Directory Files

When a directory file is created, the system automatically assigns the following default permissions to the directory:

```
rwXrwXr-x
```

These permissions give the user and group read (`r`), write (`w`), and execute (`x`) access to that particular directory file. Other users have only read (`r`) and execute (`x`) access to the directory file.

The following describes how each type of permission determines acceptable manipulations for a directory file.

- Read access (`r`) allows you to read a directory's contents.
- Write access (`w`) allows you to add or delete a directory's files.
- Execute access (`x`) allows you to access a directory. If you do not have execute access for a directory, you will not be able to use the `cd` command to change to that directory, and you will not be able to access that directory at all.

3.5.1.2 Ordinary Files

When an ordinary file is created, the system automatically assigns the following default permissions to it:

```
rw-rw-r--
```

These permissions give the user and group read (r) and write (w) access to the file. Other users can only read and copy (r) the file. They cannot modify the file contents or delete the file. When adding a user account, the `addusers` utility places a `umask` value in the user's `.profile` file; this value determines the default permissions for directories and ordinary files. For more information on `umask`, refer to the System V online manuals.

The following describes how each type of permission determines an ordinary file's acceptable manipulations. To access any ordinary file in a directory, you must have execute (x) permission for that directory.

- Read (r) allows you to display a file's contents and copy the file to your directory.
- Write (w) allows you to modify or write to the file.
- Execute (x) allows you to execute a file such as a shell script or a program.

3.5.2 Modifying File Permissions

As file owner or superuser, you can change the permissions of that file. The `chmod` (change mode) command, used to add or deny privileges for users, must have the following format:

```
$ chmod <user_type><operator><permissions> filename
```

The following abbreviations are used with the `chmod` command to specify the user type, the operator, and the permissions.

User Type:

```
u user
g group
o other
a all
```

Operator:

```
+ add
- deny
= change all
```

Permissions:

r read
w write
x execute

The following example of using the `chmod` command grants all users execute access to the `/usr/joe/graphics` file:

```
$ ls -l /usr/joe/graphics
-rw-rw-r-- 1 joe users 94 Jan 09 05:22 graphics

$ chmod a+x /usr/joe/graphics

$ ls -l /usr/joe/graphics
-rwxrwxr-x 1 joe users 94 Jan 09 05:22 graphics
```

The following example denies other users write access to the `/usr/frank/report` file:

```
$ ls -l /usr/frank/report
-rwxrwxrwx 1 frank users 103 Jul 18 01:39 report

$ chmod o-w /usr/frank/report

$ ls -l /usr/frank/report
-rwxrwxr-x 1 frank users 103 Jul 18 01:39 report
```

The following example grants "group" users read, write, and execute access to the `/usr/kathy/edits` file:

```
$ ls -l /usr/kathy/edits
-rwx----- 1 kathy users 36 May 09 10:54 edits

$ chmod g-rwx /usr/kathy/edits

$ ls -l /usr/kathy/edits
-rwxrwx--- 1 kathy users 36 May 09 10:54 edits
```

3.5.3 Modifying File Ownership

You are classified as a file owner when you create or copy a file. Only the file owner or the superuser can change the owner of a file or the permissions assigned to a file.

The `chown` command is used to change the owner of a file. To change a file's owner, you must be either the current owner or superuser. The `chown` command must have the following format:

```
$ chown <new_owner> <filename>
```

The following example shows how joe transfers ownership of a file called *project* to sam.

```
$ ls -l /usr/joe/project
-rw-rw-r- 1 joe users 10 Apr 01 10:44 project

$ chown sam /usr/joe/project

$ ls -l /usr/joe/project
-rw-rw-r- 1 sam users 10 Apr 01 10:44 project
```

3.5.4 Changing Group Access

You are in the group category when you belong to the file owner's group. The `/etc/group` file determines which user group, or groups, you are associated with. Users in a group can access one another's files (based on the second set of read, write, and execute permissions) and still maintain file security from others outside the group. If you belong to the file owner's group, the second group of read, write, and execute (`rwX`) permissions determine how you can manipulate the file.

The `addusers` utility, when used to set up your account on the workstation/server, automatically places you in the users group. This default group has a Group Identification (GID) number of 500.

3.5.4.1 Changing Your Active Group

You can edit the `/etc/group` file so that you belong to several groups. However, you can only actively belong to one group at a time. The `newgrp` command changes your active group. You must belong to the new group as defined in the `/etc/group` file. The syntax for the `newgrp` command is as follows:

```
$ newgrp <group_name>
```

For example, if your active group is *programmers*, you would key in the following to change your active group to *writers*.

```
$ newgrp writers
```

You are now permitted group access to the files of users in the group, *writers*.

Note:

The `newgrp` command does not change your default group (the group you are placed in at login). The change is effective only until you log out. You must edit the group field in the `/etc/passwd` file to change your default group.

3.5.4.2 Changing the Group of a File

The file owner or superuser can use the `chgrp` command to change the group access for a file. The syntax for the `chgrp` command is as follows:

```
$ chgrp <new_group> <filename>
```

The following example illustrates how the group access of a file called *text* was changed from *grp1* to *grp2*.

```
$ ls -l /usr/brad/text
-rw-rw-r-- 1 brad grp1 37 May 16 01:13 text
```

```
$ chgrp grp2 /usr/brad/text
```

```
$ ls -l /usr/brad/text
-rw-rw-r-- 1 brad grp2 37 May 16 01:13 text
```

grp2 now has read and write permissions for the file *text*.



Chapter 4: Modifying the Login Environment

This section describes the processes invoked and files read when you log in to the system. Ways to customize the login environment by altering the */etc/passwd*, */etc/profile*, *env*, *exrc*, and *.profile* files are included. In addition, procedures for creating a reminder service and for using the **cron** (commands run on notice) commands are included.

Note:

The */etc/passwd*, */etc/profile*, and *cron* files affect the login environment for all workstation/server users, while *env*, *exrc*, and *.profile* (*skeleton*, or hidden files) affect only the user's environment where the files reside. Skeleton files all begin with a dot (*.*), which causes them not to display when the standard **ls** command is used. To list the skeleton files, key in **ls -a** from your user directory.

This chapter contains the following sections:

4.1	The <i>getty</i> and login Processes	4-3
4.1.1	The <i>/etc/passwd</i> File	4-4
4.1.2	Customizing the <i>/etc/passwd</i> File	4-5
4.2	The <i>/etc/profile</i> File	4-6
4.3	The <i>.profile</i> file	4-8
4.4	The <i>.env</i> File	4-10
4.4.1	Command Recall (The <i>.history</i> File)	4-11
4.4.1.1	Command Recall Using the <i>vi</i> Editor	4-12
4.4.1.2	Command Recall Using the <i>emacs</i> Editor	4-12
4.4.1.3	Temporary Editors for Command Recall	4-12
4.5	The <i>.exrc</i> File	4-13
4.6	Creating a Reminder Service	4-14
4.7	Using cron	4-15
4.7.1	The crontab Command	4-16
4.7.2	Executing One Command at a Specified Time	4-17

4.1 The `getty` and `login` Processes

The `getty` process waits for input at the login prompt. When you key in your login name, `getty` invokes the `login` process.

The `login` process checks the `/etc/passwd` file to determine whether the username you keyed in matches a record in this file. Each line of the `/etc/passwd` file, called a *record*, represents a user account. All user accounts are included in this file.

If the login name you keyed in matches a record in the `/etc/passwd` file and if your login has an associated password, the `login` process displays the `passwd` prompt and turns off echoing to the screen. Since `echo` is turned off, your password does not display when you key it in.

Again, `login` compares your password to the record in the `/etc/passwd` file. If they match, `login` places you in your home directory, as specified in the sixth field of the `/etc/passwd` file. The `login` process also invokes the shell as specified in the last field of the `/etc/passwd` record. The following section describes the fields of the `/etc/passwd` file.

4.1.1 The `/etc/passwd` File

To display the contents of the `/etc/passwd` file, key in the following command:

```
# cat /etc/passwd
```

The contents of the `/etc/passwd` file varies depending on the system users. Sample output follows:

```
root:sFghYWagtTrM:0:3:0000-Admin(0000):/bin/ksh
daemon:1:12:0000-Admin(0000):/bin:
bin:2:2:0000-Admin(0000):/bin:
sys:3:3:0000-Admin(0000):/usr/src:
adm:4:4:0000-Admin(0000):/usr/adm:
uucp:5:1:0000-uucp(0000):/usr/lib/uucp:
sysadm:0:0:administration login:/usr/admin:/bin/ksh
sync:20:1:0000-Admin(0000):/bin/sync
lisa:Cw32S1rHIvg1Q:502:500:new user:/usr/lisa:/bin/ksh
kelley:3Y9xSI83rA:72:500:new user:/usr/kelley:/bin/ksh
```

The fields of the `/etc/passwd` file are as follows. The fields in this file are separated by colons (:), as shown in the following example.

```
<login_name>:<encrypted_password>:<UID>:<GID>:<misc_info>:<home_directory>:<default_shell>
```

Each field is described using `kelley` as an example.

`kelley:3Y9xSI83rA:72:500:new user:/usr/kelley:/bin/ksh`

- `kelley` is the login name that the user `kelley` must key in at the login prompt. The system administrator supplies the login name when adding the account. The login name may contain a maximum of eight characters.
- `3Y9xSI83rA` is the (encrypted) password for the user account. The system administrator can specify a password when adding the account, or the password can be added later with the `passwd` command. If users forget their passwords, the superuser can edit the `/etc/passwd` file and delete the encrypted passwords.
 - The password may contain a maximum of eight characters. If the user adds the password, the password must have at least six characters with at least one numeric or special character (such as `?` or `4`) and at least two alphabetic characters.
 - If the superuser adds the password, the password may have less than six characters and does not require any numeric or special characters.
- `72` is the User Identification code (UID). The UID is provided by the `addusers` or `sysadm` utility. Derive the UID by adding one to the highest UID number in the `/etc/passwd` file. The UID may range from 0 to 65,535. (0 is reserved for the root account and should **never** be assigned to other users.) Because the system uses the UID to determine file ownership, the UID should be unique for each user account.
- `500` is the Group Identification code (GID). The GID determines which group in the `/etc/group` file a user is associated with. The GID may range from 1 to 65,535. GIDs for all user accounts created with `addusers` default to 500; accounts created with `sysadm` default to 1. Users with the same GID can access one another's files based on the second set of read, write, and execute permissions.
- `new user` is miscellaneous information that you can change by editing the `/etc/passwd` file. This field can contain any relevant miscellaneous information, such as an employee's department, phone, or location.
- `/usr/kelley` is the default directory (also known as the *home* directory) where users are automatically placed when they log in to the system. The `addusers` or `sysadm` utility creates this directory.

- `/bin/ksh` is the operating system shell where users are placed when they log in. The default shell for user accounts is the Korn Shell (specified as `/bin/ksh`). The C-Shell is specified as `/bin/csh`.

To place users in the Bourne Shell when they log in, leave the last field blank, as shown in the following example:

```
root:12k1*)@#:0:3:0000-Admin(0000):/:
```

4.1.2 Customizing the `/etc/passwd` File

By editing the `/etc/passwd` file, you can alter user accounts. For example, you can remove a user's password, change a user's home directory, or change the user's group identification number. Do not delete the colons.

4.2 The `/etc/profile` File

After the login process checks the `/etc/passwd` file and you can access the system, the shell executes the `/etc/profile` file. The `/etc/profile` file creates a standard operating environment and defines shell variables for all workstation/server users.

To customize a particular user's environment, modify the `.profile` file in the user's *home* directory, as described in 4.3. Note that the user's `.profile` file will override the variables set in the `/etc/profile` file.

Note:

For a list of shell variables, refer to the manual page for the appropriate shell (such as `ksh`) in the *CLIX Programmer's and User's Reference Manual*.

Lines in the `/etc/profile` file are described below:

- `./etc/TIMEZONE` executes the `/etc/TIMEZONE` shell script. This shell script defines the time relative to your timezone. You must manually set the `TZ` variable in the `/etc/TIMEZONE` file to correspond to your timezone. The `TZ` variable value defaults to Central Standard Timezone (CST) during daylight savings time (CDT). The setting for the `TZ` variable defaults to "`TZ=CST6CDT`." For additional timezone values, refer to the chart in 8.2.3, "Date and Time Menus."
- `cat /etc/motd` displays the contents of the `/etc/motd` (message-of-the-day) file on the screen. The `/etc/motd` file is empty by default. You can edit this file and use it to display messages to all users when they log in.
- `if mail -e echo "you have mail"` displays the "you have mail" message when mail has been sent to your username. You can modify this command string to display another message or add `<Ctrl-G>` to make the workstation beep when the message displays.
- `stty intr "^C" erase "^?" echoe kill "^U"` sets the erase character to the Delete key, the line kill character to `<Ctrl-U>`, and the interrupt character to `<Ctrl-C>`.
- `ulimit 65536` sets the maximum file size to 65,536 blocks.

- **PATH=**./bin:/usr/bin:/etc defines the path of directories (separated by colons) searched when you execute a command. If you keep commonly used commands in a file such as */usr/jlm/mycommands*, you can add this line to the PATH variable.
- **TERM=VT100** sets the terminal type to VT100.
- **EDITOR=vi** sets the default editor to vi. To change the default editor (to an editor such as *emacs*), redefine this variable for **EDITOR=emacs**. Note that any changes to this file affect all system users.
- **Export PATH TERM EDITOR** exports shell variable assignments so they are known to all processes in any level or subshell.

For additional information on the */etc/profile* file, refer to the System V manual page for **profile(4)**.

4.3 The `.profile` file

After the login process checks the `/etc/passwd` file and the shell executes the `/etc/profile` file, the shell checks your *home* (`/usr/<login>`) directory and executes the `.profile` file if it exists. The `.profile` file affects only your personal login environment. You can customize this file by adding shell variable assignments.

This file is similar to the `/etc/profile` file, except that `/etc/profile` affects the login environment for all users on a particular workstation/server. The `/usr/<login>/profile` assignments override those in the `/etc/profile` file.

To view the contents of your `.profile` file, key in the following from your user directory:

```
$ cat .profile
```

The default `.profile` file (created with `addusers`) appears as follows:

```
#ident "@(*)stdprofile 1.2 9/1/88 Copyright (c) 1986,1987 by Intergraph Corp."

# This is the default standard profile provided to a user.
# They are expected to edit it to meet their own needs.

# This file has been modified by Intergraph to define a standard
# customer environment.

umask 2
PATH=/usr/bin:/bin:
ENV="$HOME/.env"
EDITOR=vi
TERM=vt100
export PATH ENV EDITOR TERM
```

This file defines the following:

- `umask 2` sets the default permission mode so that files are created with the following permissions:

```
directory files:  drwxrwxr-x
ordinary files:   -rw-rw-r-
```

- `PATH=./usr/bin:/bin` defines the path of directories searched when you execute a command. By default, the system searches your current directory, the `/usr/bin` directory, and the `/bin` directory. You can add other commonly accessed directories (such as `/etc`) to this variable assignment.

- **ENV="\$HOME/.env"** defines your environment file as */usr/<login>/env* when you use the Korn shell.
- **TERM=VT100** defines your current terminal type as a VT100.
- **EDITOR=vi** defines your default editor as **vi**. If you are using the Korn shell and change your default editor (to an editor such as **emacs**), you will also want to change the **FCEDIT** and **VISUAL** variables in the *env* file. This ensures that command recall will use the specified default editor.
- **Export ENV, TERM, EDITOR** allows all processes in any level or subshell to recognize new shell variable assignments.

4.4 The `.env` File

If you are using the Korn shell, the shell executes the `.env` file in your *home* directory if the file exists. The `.env` file affects only your personal login environment. You can customize this file by adding shell variable assignments.

The `.env` (environment) file is used to define the history file as `/usr/<login>/history`, define several command aliases, and set up the variables necessary for command recall. The `.env` file is unique to the Korn shell and therefore is executed only when you invoke the Korn shell. Since the `addusers` and `sysadm` utilities automatically establish the Korn shell as the default login shell, the `.env` file is automatically executed when you log in after the `.profile` file has been executed.

The Korn (k-shell environment file) allows you to define simple command aliases, include functions used by the Korn shell, and recall recent commands using your default text editor. You can view this file by using the `pg` command, which prints the first screen-length page of the `.env` file. Key in this command from your user directory as follows:

```
$ pg .env
```

The default `.env` file appears as follows:

```
alias -x h='fc -l'  
alias -x ls='ls -C'  
alias -x more='pg -n'  
alias -x lo='exit'  
  
HISTFILE=$HOME/.history  
FCEDIT=vi  
VISUAL=vi  
export HISTFILE FCEDIT VISUAL
```

This file defines the following:

- `alias -x h='fc -l'` prints a range of commands from the *history* file when you key in `h [first] [last]`, where `first` and `last` can be either a number or a string.
- `alias -x ls='ls -C'` causes output from the `ls` command to display in columns as it would for the `ls -C` command.
- `alias -x more='pg -n'` allows you to use `more` instead of the `pg -n` command.

- `alias -x lo='exit'` allows you to log out by keying in `lo` instead of `exit`.
- `HISTFILE="$HOME/.history"` defines your history file as `/usr/<login>/history`.
- `FCEDIT=vi` defines your editor for the `fc` command. If you prefer an editor other than `vi` for command recall, redefine this variable (for example, `FCEDIT=emacs`).
- `VISUAL=vi` defines your editor for command recall. If you prefer another editor for command recall, redefine this variable (for example, `VISUAL=emacs`).
- `export HISTFILE FCEDIT VISUAL` allows all processes in any level or subshell to recognize new shell variable assignments.

4.4.1 Command Recall (The `.history` File)

Command recall enables you to recall and edit your previous commands (from the command line) using a series of key strokes. This section describes how to use the `vi` or `emacs` editor for command recall when you are working in the Korn shell. It also includes instructions for using the `set` command to set your editor temporarily to one other than your default. To redefine the command recall editor permanently, change the `VISUAL` variable in the `.env` file, as explained in the previous section.

The `history` file, created by the Korn shell, stores your most recent commands. This file is defined by and stored in the `.env` file, which is also specific to the Korn shell. You can use the default text editor (or a temporary editor) to recall your most recent commands from the `history` file, execute them again, or edit them on the command line.

You cannot use the `pg` command to display the `history` file.

4.4.1.1 Command Recall Using the vi Editor

By default, your workstation/server is set up for command recall with the vi editor. Follow these steps to use command recall with vi:

1. To recall your last command, press <Esc> and key in **k**. The “k” recalls your last command (one line up in the *history* file) just as the **k** command in vi moves the cursor up one line in a file.
2. To edit the command line, use vi commands to position the cursor and edit the text. You are in command mode until you enter **i** for insert mode or **a** to append. To issue the edited command-line, press <Return>.

4.4.1.2 Command Recall Using the emacs Editor

You do not need the emacs software to use emacs commands for command recall. If emacs is your default editor or you want to use emacs as your default editor for command recall, redefine the VISUAL variable in the *.env* file so that it is now emacs.

Once the VISUAL variable is set to emacs, you can use <Ctrl-P> to recall your last commands. Other emacs commands can be used to edit and move around in the line.

4.4.1.3 Temporary Editors for Command Recall

You can temporarily assign an editor to use for command recall. This option is useful when you are working on other workstations/servers that may not be set up for the editor you prefer.

To temporarily set command recall for emacs, key in the following:

```
$ set -o emacs
```

To temporarily set command recall for vi/vedit, key in the following:

```
$ set -o vi
```

4.5 The .exrc File

The `.exrc` file is an initialization script that contains abbreviations and keyboard maps that simplify the text-editing environment. This file affects only your personal login environment. `.exrc` is executed only when the `vi`, `vedit`, or `ex` editor is invoked. You can customize `.exrc` by assigning (mapping) additional editor commands to keys or by adding `vi/ex` parameters.

Key in the following to view the contents of the `.exrc` file:

```
$ cat .exrc
```

The default `.exrc` file appears as follows:

```
map g G
ab IP32 InterPro 32
map #1 :sh
map #2 :set number
map #3 /
map #4 :dd
map p
map M
```

This file defines the following:

- `map g G` ensures that a lowercase `g` and an uppercase `G` are interpreted the same. This map is used for the `goto` command.
- `ab IP32 InterPro 32` changes all text occurrences of “IP32” to “InterPro 32.”
- `map #1 :sh` allows PF1 to return you to the shell. From the shell, `<Ctrl D>` returns you to your `vi` file.
- `map #2 :set number` allows PF2 to add line numbers to your `vi` file.
- `map #3 /` allows PF3 to search for a specified string.
- `map #4 :dd` allows PF4 to delete an entire line.
- `map p` allows the `Enter` key on the keypad to invoke a `<Return>`.
- `map M` allows the `0` key on the keypad to invoke a `<Return>`.

You can map additional editor commands to keys. You can also add `vi/ex` parameters (such as `set tabstop=4` to change tab settings to four spaces). For a complete listing of `vi` parameters, edit a file using `vi` and (in command mode) key in the following:

```
:set all
```

4.6 Creating a Reminder Service

Another way to customize the login environment for an individual user is to create a reminder service.

Using the `calendar` command, you can create a reminder service that prints important dates and events on the screen when you log in. The `calendar` command knows the value of *today* and *tomorrow* and searches your calendar file for any lines that include *today* or *tomorrow's* date. On Fridays, the command interprets **Monday** as *tomorrow*.

To create a reminder service, follow the steps below:

1. Create a file called `calendar` in your user directory.
2. Enter important dates (month/day combinations) and events in the file as follows:

```
$ vi calendar
8/24   *** Department Meeting, 10:00 ***
Aug 30  Back up all files
```

3. Enter the `calendar` command in your `/usr/login/.profile` file.

```
$ cat >> /usr/login/.profile
calendar
<Ctrl-D>
$
```

Notes:

- The `calendar` command recognizes standard month/day combinations (Jul 24; 7/24) but does not recognize day/month combinations (24 Jul).
- When the `calendar` command is inserted in your `.profile` file, login time may increase slightly as the system searches your calendar file.

4.7 Using cron

Using the **cron** (commands run on notice) commands, you can create a file that will execute commands at specific times. This capability is especially useful for running commands at regular intervals as a reminder service or for initiating a process (such as a backup script) to run after hours.

Note:

If you use **cron** to run processes after hours, you may log out, but do not shut the system down. If you shut the system down, the processes will not run.

To set up a **cron** file, follow these steps:

1. Add yourself to the `/usr/lib/cron/cron.allow` file as a valid user as follows:

```
# cat >> /usr/lib/cron/cron.allow
username
<Ctrl-D>
#
```

2. Create a **cron** file in your user directory using the following format:

```
minutes hours days months weeks command > device
```

The following list defines valid values for each component in the previous format:

```
minutes = 0-59
hours = 0-23
days = 1-31
months = 1-12
days of the week = 0 (Sunday) - 6 (Saturday)
```

The following is an example showing how to create a **cron** file.

```
# vi cronfile
30 9 * * 1 echo "10:00 Meeting!" > /dev/tty00
```

This example instructs the system to display the message "10:00 Meeting!" at 9:30 a.m. on every Monday of every month. The message will display in the `tty00` window.

Notes:

- An asterisk in a field is interpreted as all values. To enter multiple values for one field, separate the numbers by commas.
- Ensure that you redirect the output of a **cron** command to an appropriate file or device.

3. Copy your cron file to the `/usr/spool/cron/crontabs` directory using the `crontab` command as follows:

```
# crontab /usr/login/cronfile
warning: commands will be executed using /bin/sh
#
```

Notes:

- If you change the contents of the cron file you created in your home directory, you must recopy the file to `/usr/spool/cron/crontabs` so the system will recognize the changes.
- See the following subsection for additional information on the `crontab` command.

4.7.1 The `crontab` Command

You can use the `crontab` command to add your cron command file to the appropriate directory (`/usr/spool/cron/crontabs`) for execution. The following table shows options that can be used with the `crontab` command:

Command	Function
<code>crontab <filename></code>	Adds your <i>cron</i> command file to the <code>/usr/spool/cron/crontabs</code> directory. In the <i>crontabs</i> directory, a file called <i>login</i> (where <i>login</i> is your login name) contains your cron commands.
<code>crontab -l</code>	Lists your cron commands as they are entered in the <code>/usr/spool/cron/crontabs</code> directory.
<code>crontab -r</code>	Removes your cron commands from <code>/usr/spool/cron/crontabs</code> .

Notes:

- You must be listed as a valid user in the `/usr/lib/cron/cron.allow` file before you can execute the `crontabs` command.
- Cron should be reserved for commands executed regularly. For commands that are executed only once, use the `at` command. The `at` command is discussed in the following section.

Refer to the System V manuals for more information on `cron` and `crontab`.

4.7.2 Executing One Command at a Specified Time

The `at` command is used to execute one command at a specific time, whereas the `cron` command allows you to execute a command at regular intervals. This command is useful for creating one-time reminders or for initiating a process to run at a specific time.

To use the `at` command, follow these steps:

1. Add yourself to `/usr/lib/cron/at.allow` file as a valid user.

```
$ cat >> /usr/lib/cron/at.allow
username
<Ctrl-D>
$
```

2. Key in the time, date, and command to be executed using the following format:

```
at <time> <date> <command> <Ctrl-D>
```

Valid forms of `time` include the following:

```
05:15
5:15pm
5 pm
```

Valid forms of `date` include the following:

```
month day (Jun 14)
week_day (Friday)
today
tomorrow
```

The following is an example of how to use the `at` command:

```
# at 4:15pm Jun 6
echo "4:30 meeting." > /dev/tty00
<Ctrl-D>
warning: commands will be executed using /bin/sh
job 518476140.a at Fri Jun 6 16:15:00 1989
#
```

Refer to the System V manuals for more information on the `at` command.



Chapter 5: Window and Screen Configuration

This chapter contains the following sections:

5.1	Setting the Workstation User	5-3
5.2	Creating VT220 Windows	5-4
5.2.1	Using vterm	5-4
5.2.2	Creating Local Windows Through the Workstation Menu	5-8
5.2.3	Creating a Terminal Window	5-8
5.3	Using the General Setup Menu to Define VT220 Characteristics	5-10
5.4	Using the Display Setup Menu to Define Display Characteristics	5-12
5.5	Using the Keyboard Setup Menu to Define Keyboard Characteristics	5-13
5.6	Using the Tab Setup Menu to Define Tabs	5-14
5.7	Using the Printer Setup Menu to Define a Window Attached to a Dedicated Printer	5-15
5.8	Using the Macro Setup Menu to Assign Macros to Keys	5-16
5.9	Creating Color Windows	5-18
5.9.1	The Intergraph-Developed regis Color Command	5-18
5.9.2	The ReGIS Color Command	5-19
5.9.3	The ANSI Color Escape Sequences	5-22
5.10	Using the Catalog Function to Save Customized Windows	5-24
5.11	Altering the Workstation Pulldown Menu	5-26
5.11.1	Adding Processes to the Workstation Pulldown Menu	5-26
5.11.2	Modifying the Workstation Pulldown Menu	5-28
5.11.3	Deleting Processes from the Workstation Pulldown Menu	5-28
5.12	Changing Screen Configuration	5-30
5.12.1	Changing the Number of Virtual Screens (Single-Monitor Workstations Only)	5-30
5.12.2	Changing the Number of Message Strips (Dual-Monitor Workstations Only)	5-31
5.12.3	Changing the Message Strip(s) Location	5-31
5.13	Changing the Background Pattern	5-32
5.14	Changing the Screen Color	5-33
5.15	Invoking Processes Automatically	5-34
5.15.1	The def_proc File	5-34
5.15.2	The .def_proc File	5-35
5.15.3	The sys_proc File	5-36

This chapter describes how to configure the window and screen to create an environment most suitable to your purposes. It starts with a discussion of the *Workstation User* concept. You should understand this concept before you customize your system. Then, it discusses ways to customize your system using VT220 Emulation and Screen Manager features.

Note:

Most procedures described here depend on graphics display and therefore apply only to workstations.

The following *vterm* features can be customized:

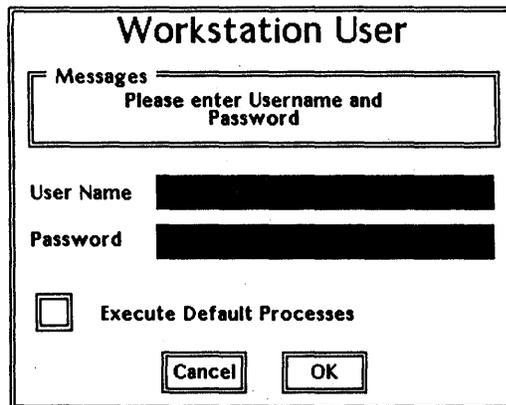
- General VT220 characteristics
- VT220 display characteristics
- Keyboard characteristics
- Tab definitions
- Printer setup
- Keyboard macro definitions
- Window colors
- Save feature for customized VT220 windows (catalog function)

The following Screen Manager features can be customized:

- Workstation Pulldown menu
- Number of virtual screens
- Background pattern
- Screen color
- Default processes

5.1 Setting the Workstation User

After the workstation boots, a login menu appears. This menu prompts for the username and password for any valid user account on the system. No windows can be created until you log in successfully from this menu.



The image shows a dialog box titled "Workstation User". Inside the dialog, there is a "Messages" section with a border containing the text "Please enter Username and Password". Below this, there are two input fields: "User Name" and "Password", both of which are currently filled with blacked-out text. At the bottom left, there is a checkbox labeled "Execute Default Processes" which is currently unchecked. At the bottom right, there are two buttons: "Cancel" and "OK".

This login menu establishes a workstation user. All processes will run under the protections/privileges established by the workstation user. An entry in the Workstation Pulldown menu (Wkst. User) can be used to check for or change the workstation user. When you are finished using the workstation, use the "log out" feature on the Workstation Pulldown menu.

If you check the Execute Default Processes box on the login menu, the *def_proc* file in your home directory will run. For more information on *def_proc*, refer to 5.15, "Invoking System Processes Automatically."

Note:

If you do not want the Workstation User menu to appear each time you boot the system, you can create a file called */etc/wkstuser*. This file should contain the username that you wish to establish as the workstation user.

For more information on the workstation user, refer to the */usr/ip32/smgr/smgr.doc* file.

5.2 Creating VT220 Windows

Intergraph's VT220 Terminal Emulation software allows local and network windows on Intergraph workstations to emulate a DEC VT220 terminal. The VT220 software uses a process called `vterm` to create VT220 windows.

You can create VT220 windows through the command line or through the Workstation Pulldown menu. This section describes the `vterm` options available for creating windows from the command line, local windows through the Workstation Pulldown menu, and terminal windows through the Workstation Pulldown menu.

5.2.1 Using `vterm`

The `vterm` command allows you to create VT220 windows. The following options can be used with this command:

- `-l [tty#]`

The `-l` option creates a local VT220 window. If the optional `tty` (pseudo-terminal) number is specified, `vterm` will connect to the device specified by the pseudo-terminal number. If no pseudo-terminal number is specified, `vterm` will automatically assign the next number. A pseudo-terminal number can range from 0 to 18.

- `-a <aux#>`

The `-a` option creates a virtual terminal connected to a serial device through the specified auxiliary (RS-232) port. For example, for a modem connected to auxiliary port 2, use the `vterm -a 2` command to create a window connected directly to the modem.

- `-n <nodename>`

The `-n` option creates a network window to the `nodename` specified. The `nodename` can be the network address (for example, 08-00-36-XX-XX-00) or the logical `nodename` assigned to the node. For example, the command `vterm -n sys44` creates a window to the `sys44` node.

- **-x [s,r,e] <command>**

The **-x** option creates a window and executes the specified command. For example, the `vterm -x df` command creates a window, logs in, executes the `df` command once, logs out, and deletes the window. If the command string contains spaces, enclose the command string in quotation marks (for example, `vterm -x "pg mbox"`).

The **-x** option has three optional modifiers (**s**, **r**, and **e**) that prevent the window from being deleted after the command has completed.

The **s** (stay) modifier causes the window to remain on the screen after the command has finished executing. For example, the `vterm -xs df` command creates a window, logs in, and executes the `df` command once and stops. The window will stay on the screen until you delete it.

The **r** (respawn) modifier causes the command to be executed continuously. For example, the `vterm -xr df` command creates a window, logs in, and executes the `df` command continuously. You must delete the window to stop the process.

The **e** (error) modifier causes the window to stay on the screen if an error occurs while the command executes. If no error occurs, the window is deleted automatically. For example, the `vterm -xe df` command creates a window, logs in, executes the `df` command, and exits (if the `df` command returned no error).

- **-f <catalog file>**

The **-f** option creates a window using predefined VT220 and window characteristics. To use this option, you must supply the filename of the predefined (or *cataloged*) window. (Refer to 5.10, "Using the Catalog Function to Save Customized Windows," for more information.) For example, the `vterm -l -f red_window` command creates a local window previously saved as *red_window*.

- **-p <aux#>**

The **-p** option creates a window attached to a dedicated printer connected to the specified auxiliary port. For example, the command `vterm -l -p0` creates a local window attached to a printer on port 0. From a VT220 created with the **-p** option, you can use the Print Screen function key to print the window's text to the dedicated printer. You can also use the auxiliary printer as a logging device. In addition, an application running in the window can control this printer.

- **-console**

The **-console** option creates a console window. You can have only one console window. The console window must be created as a local window. The default console window is created in the */usr/lp32/smgr/sys__proc* file.

- **-c**

The **-c** option creates a window that can become a color window. For example, the **vterm -l -c** command creates a local window for which you can change the text and background colors. Refer to 5.9, "Creating Color Windows."

- **-small**

The **-small** option creates a scaled-down (604 X 407) window. A window created with this option displays the same 24 rows and 80 columns in a smaller character font than standard windows do. For example, the **vterm -l -small** command creates a small-sized local window.

- **-passive**

The **-passive** option creates window that does not become active until you explicitly activate it. By default, a newly created window becomes the active process. The **vterm -l -passive** command creates a local window that is inactive when created.

- **-show**

The **-show** option uncollapses the window when it receives output from the host. The default console window uses this option to cause the console window to uncollapse when system messages are received. If you wish to keep the console window from popping to the top when system messages have been received, remove the **-show** option from the string that creates the console window in the */usr/lp32/smgr/smgrutil* file.

- **-origin x y**

The **-origin** option places the window's upper left-hand corner at the specified coordinates. The **vterm -l -origin 100 300** command creates a local window at screen coordinates 100 300. By default, a window is created at coordinates 41 34.

- **-size x y**

The **-size** option defines the size (in pixels) of the window. The command **vterm -l -size 900 400** command creates a 900- X 400-pixel local window. By default, a standard window is 1004 X 526 pixels.

- **-collapse x y**

The **-collapse** option creates a collapsed window and places it in the specified screen coordinates. The **vterm -l -collapse 900 765** command creates a collapsed local window at 900 765.

- **-log <logfile>**

The **-log** option specifies a file for storing the window's activity. The **vterm -l -log /usr/tmp/john.log** command creates a local window with the */usr/tmp/john.log* log file.

- **-u <username>.<password>**

The **-u** option, used with the **-x** option, causes the command to be executed under the specified user. For example, the **vterm -xs ksh -u john.doe** command creates a local window, logs in to the window with the **john.doe** username/password combination, and places you in the Korn shell.

- **-T <title>**

The **-T** option creates a window with a specified title. The **vterm -l -T "Hello there"** command creates a local window entitled "Hello there." By default, a window's title is the name of the host or device to which the window is connected, followed by "- VT220." Note that the title of the console window cannot be overridden.

- **-s <symfile> <index>**

The **-s** option specifies the collapse icon (instead of the standard Environ V collapse icon) for a window. For example, if you created a symbol file (using Symbol Editor), you could specify for a collapsed window to use the symbol. To use this option, you must specify the symbol file and the symbol index. The **vterm -l -s /usr/john/symbol_file 3** command creates a local window that will, when collapsed, display as symbol 3 from the file */usr/john/symbol_file*.

- **-t <logical name>**

The **-t** option creates a window associated with a specific device's logical name. If this option is not specified, the host will assign the first available device. This option is valid only with the **-n** (network) option.

5.2.2 Creating Local Windows Through the Workstation Menu

To create a local window, you can use the **vterm** commands described previously or you can use the Workstation Pulldown menu. To create a local window from the menu, select the Local option from the Workstation Pulldown menu.

From a local window (local VT220 virtual terminal), you can communicate with the UNIX System V operating system.

5.2.3 Creating a Terminal Window

To create a terminal window, you can use the **vterm** command described previously or you can use the Workstation Pulldown menu. To use the menu, select the Terminal option from the Workstation Pulldown menu.

When you choose the Terminal option from the Workstation Pulldown menu, the Create VT220 menu appears. This menu contains the following options for creating virtual terminal windows:

- **Local Host.** This option creates a local window and allows you to specify its pseudo terminal number. If you do not specify a number, **vterm** assigns the next available pseudo terminal.
- **RS-232 Port Host.** This option associates the creation of virtual terminal windows with an auxiliary port.
- **Network Host.** This option creates a virtual terminal window to a host on the network. You must provide the nodename or node address in the Host Parameters box.
- **Command Host.** This option creates a window for executing a specific command. You must specify the command on the Create VT220 menu. In addition, you can check one of three checkboxes to specify how the command will be executed.

- **Respawn.** This option creates the window and executes the command continuously until you delete the window. The Stay option creates the window and executes the command once. The window remains on the screen until you delete it. The Error option creates the window, executes the command, and immediately deletes the window (if no error occurred).
- **Printer.** The printer option assigns a local printer to a particular auxiliary port.
- **Catalog File.** This option creates a VT220 window that has been saved in a file. For information on saving customized windows, see 5.10, "Using the Catalog Function to Save Customized Windows."
- **Origin.** This option specifies the window's placement on the screen. You must specify the x and y coordinates of the upper left-hand corner of the window. The screen coordinates are measured in pixels.
- **Size.** This option defines the size of the window. You must specify the x and y coordinates that define the size. The screen coordinates are measured in pixels.
- **Collapse.** This option creates a collapsed window and places it at the specified coordinates.
- **Color.** This option must be selected if you wish to create color windows (using the ReGIS commands or ANSI escape sequences). For more information, see 5.9, "Creating Color Windows."
- **Small.** This option creates a window the size of a console window, which displays text in a smaller font than a standard window does.
- **Output Uncollapse.** This option forces a collapsed window to uncollapse when system messages or other output is sent to the window.

5.3 Using the General Setup Menu to Define VT220 Characteristics

The General Setup menu allows you to set various terminal characteristics. To access this menu, select the Terminal icon on the window icon strip and select the General option from the menu that appears. The General Setup menu contains the following options:

- **Emulation.** Use this option to specify VT100 or VT220 terminal emulation. When VT220 is set, the virtual terminal emulates a VT220 terminal; when VT100 is set, the virtual terminal emulates a VT100 terminal.
- **Data Size.** Use this option to set the mode of the virtual terminal for 7-bit or 8-bit data, depending on the requirements of the application running in the virtual terminal window.
- **Erase Extent.** This setting directly affects the erase screen escape sequences (ESC[2J, ESC[0J, and ESC[1J) described in Appendix A, "Keyboard Character Codes." The scroll area creation (using ESC[x1;x2r, where x1 and x2 are line numbers) and the cursor positioning will also affect the erase screen escape sequences.
- **Form Feed.** When Line Feed is set, formfeeds will function as line feeds. When Home & Clear is set, a formfeed will cause that line of text to jump to the top of the window and successive lines of text to display beneath it.
- **Online.** When this option is selected, input is sent to the host. Otherwise, input is sent to the terminal window.
- **Local Echo.** When this option is not set, `vterm` displays input on the terminal window as the host prescribes. When this option is set, `vterm` displays input on the terminal window both as the host prescribes and as you key it in. Hence, when Local Echo is set, data you key in is almost always duplicated on the terminal window.
- **New Line.** When this option is set, pressing <Return> produces a carriage return and a line feed. Otherwise, pressing <Return> produces a single carriage return.
- **User-Defined Keys Locked.** Use this option to prevent user-defined keys from being assigned.

- **Log File.** This option enables you to capture data created during a session on the workstation. For example, you can save all output that scrolls out of the window in a file called */usr/tmp/log*. (The name of the log file can be changed.) By default, output is not saved. To use this option, select the checkbox beside the Log File option and key in a filename to indicate where you want the output stored. Select OK to dismiss the menu and activate the Log File option. After you create the file, recall the General Setup menu and select the Log File checkbox again to terminate the log session.
- **Buffered Screens.** Use the Buffered Screens data-entry field to set the number of screens of data that you want saved in a buffer. These saved screens can be viewed with the scroll bar at the side of the window. The default number of buffered screens is 10.

5.4 Using the Display Setup Menu to Define Display Characteristics

The Display Setup menu allows you to set display characteristics of your virtual terminal. To access this menu, select the Terminal icon from the window icon strip and select the Display option from the menu that appears. The Display Setup menu contains the following options:

- **Cursor.** To display the cursor, select the Visible option. To display a blinking cursor, select the Blinking option. For an invisible cursor, do not select either option.
- **Type.** This option specifies the type of cursor (block or underline). You must set your cursor to appear as either a block or as an underline.
- **Columns.** Use this option to set the number of columns for your terminal window. The 80-column setting allows 80 characters to display in a window. The 132-column setting uses a smaller text font to display 132 characters in a window.
- **Autowrap.** When this option is set, text that exceeds the designated column width (80 or 132) displays on the next line. When Autowrap is not set, the last character in a maximum width line is overwritten by successive characters.
- **Display Controls.** When this option is set, `vterm` displays the codes sent to the window without interpreting them. When it is not set, `vterm` interprets control codes sent by the host without displaying the codes in the terminal window.
- **Jump Scroll.** This option is not yet implemented.
- **Light Background.** You can set your terminal window to display light text on a dark background or dark text on a light background. If color attributes are set, this option can reverse them so that the background color becomes the foreground color and the foreground color becomes the background color.
- **Erase Screen On Column Change.** As window column widths are changed from 80 to 132 characters and back again, you can use this option to choose whether data on the screen remains or is erased.

5.5 Using the Keyboard Setup Menu to Define Keyboard Characteristics

The Keyboard Setup menu allows you to customize your keyboard by defining certain keyboard characteristics. To access the Keyboard Setup menu, select the Terminal icon on the window icon strip and select the Keyboard option from the Setup menu that appears. From the Keyboard Setup menu, you can define the following options:

- **Warning Bell.** Some applications acknowledge operator errors by sounding the keyboard bell. When the Warning Bell option is checked, the keyboard beeps at operator errors; when this option is not set, the keyboard does not beep.
- **Auto Repeat.** If you press a key and hold it down, the character will display repeatedly when the Auto Repeat option is set. If this option is not set, the character will display only once.
- **Answerback.** The message in the Answerback entry box will display when the <Ctrl-E> character sequence occurs. The default message is "Hello World." You can alter this message.
- **Cursor Keys.** When the Normal Cursor Keys option is set, character codes for the up, down, left, and right arrow keys are sent to the host. When the Application Cursor Keys option is set, a different set of character codes is sent to the host.
- **Keypad.** When the Numeric Keypad option is set, the comma, period, minus sign, and numbers of the numeric keypad are sent to the host and recognized as the value on the face of the key. When the Application Keypad option is set, a different set of character codes is sent to the host.

5.6 Using the Tab Setup Menu to Define Tabs

The Tab Setup menu sets tab stops. To access this menu, select the Terminal icon from the window icon strip and select the Tab option from the menu that appears. Use the mouse or 12-button cursor to toggle tab stops on and off on the Tab Setup menu. The Tab Setup menu contains the following options:

- **Tab Motion.** When the Direct option is set, the cursor tabs to column locations without affecting text along the way. When the Spaces option is set, text is erased when the tab passes over it.
- **Clear Tabs.** Select this option to remove all tab stops.
- **Set 8-Column Tabs.** Select this option to set tab stops at every eighth column location.

5.7 Using the Printer Setup Menu to Define a Window Attached to a Dedicated Printer

Before you can use this menu, your virtual terminal must be created with the `-p` (Printer) option. You can select the Printer option from the Create VT220 menu (from the Terminal option on the Workstation Pulldown menu) or by keying in the `vt220` command string (for example, `vterm -l -p`). In addition, you can add this command string to the Workstation Pulldown menu or alter the Local option (used to create a local window) by adding the `-p` option to the existing command string. Refer to 5.11, "Altering the Workstation Pulldown Menu" for more information.

To access the Printer Setup menu, select the Terminal icon from the window icon strip and select the Printer option from the menu that appears.

The Printer Setup menu contains the following options:

- **Printer Mode.** When the Normal option is set, the Print Screen key can be used to print the current contents of the window. When the Auto option is set, a line is printed when the cursor is moved off from the current line by a received line feed, formfeed, tab, or autowrap. The Print Screen key may also be used when Auto is set. When the Controller option is set, received characters do not display. All received data except for XON, XOFF, and NUL characters and `ESC[5i` and `ESC[4i` escape sequences is transmitted directly to the printer.
- **Terminator.** When the Form Feed option is set, a formfeed character is transmitted after a Print Screen operation. If the None option is set, a formfeed is not transmitted.

In addition, a Printer Attached field displays at the top of the Printer setup menu. If a printer is attached to the workstation, this field contains a check. If not, the field is empty.

5.8 Using the Macro Setup Menu to Assign Macros to Keys

The macro facility allows you to assign character strings, such as command lines or ASCII text, to the typewriter and membrane keys on the workstation keyboard. Once a macro is assigned to a key, you can call up the assigned character string on the present line by simply pressing that key.

Follow these steps to assign macros to keys:

1. Select the Terminal icon from the window icon strip.
2. Select the Macro option from the Setup menu that appears.
3. Use the cursor to select the key you will assign a macro to. The key highlights. You can assign a macro to any key except for those that are disabled (and display in a lighter text font). These keys will not highlight when you select them.
4. Key in the key's new value in the Macro String entry box, keeping in mind these points:
 - Each character you key in (including <Return>) will be included as part of the macro.
 - Use the arrow button to the right of the Macro String entry box to delete characters in the macro.
 - Use the mouse to select characters in the macro string. Characters keyed in will replace the currently selected characters.
 - To see the value assigned to a key, select the key on the Macro menu. The macro will display in the Macro String entry box.
 - To clear all macros, select the Clear All button.
5. To assign another macro select another key and repeat the process.
6. Select the OK button to exit the menu, saving macro assignments. Select the Cancel button to exit the menu without saving macro assignments.

For example, you could assign a commonly used text phrase to a key. To assign the phrase "CDROM Software Delivery" to function key **a1**, you would select function key **a1** from the Macro Setup menu, key in "CDROM Software Delivery" in the Macro String entry box, and select the OK menu button. Then, any time you want to insert the phrase "CDROM Software Delivery" in a text file, you will press the **a1** function key, rather than key in the phrase.

You could also assign a commonly used command string to a key. To assign the command `/usr/ip32/deltools/remove` (to invoke the `remove` shell script) to function key **a2**, you would select function key **a2** from the Macro Setup menu, key in `/usr/ip32/deltools/remove`, press <Return> to cause the command to be entered automatically, and select the OK menu button. Then, any time you wish to invoke the `remove` shell script to remove products, you will press the **a2** function key rather than keying in the command and pressing return.

5.9 Creating Color Windows

To change the color of a window, three methods are available: Intergraph **regis** commands, DEC ReGIS commands, and ANSI escape sequences. The Intergraph **regis** and DEC ReGIS commands change the color of text, bolded text, and background of the virtual terminal. The ANSI color escape sequences change the color of text and text blocks (the rectangular region surrounding each character).

Note:

Color attributes of a virtual terminal created with the Intergraph **regis** or DEC ReGIS commands can be saved with the catalog feature. Color attributes of a virtual terminal created with the ANSI escape sequences cannot be saved.

Before you can change the color of a window, you must have created the window using the color (-c) option. To create a window using the color option, you can create the window using the **vterm -l -c** command or you can select the Color option from the Create VT220 menu. To access the VT220 menu, invoke the Workstation Pulldown menu and select the Terminal option.

5.9.1 The Intergraph-Developed **regis** Color Command

The Intergraph-developed **regis** command is much less complex than the standard ReGIS command because **regis** accepts logical color names.

Follow these steps to create a color window using the **regis** command:

1. Create the window using the following command:

```
$ /usr/ip32/vt200/vterm -l -c
```

2. Log in to the window.
3. Key in the following to display all available colors:

```
$ regis
```

4. You can use the **regis** command to change the color of text, background, or bold text. The syntax for the **regis** command is as follows:

```
$ regis -t <text_color> -b <background_color> -h <bold_color>
```

For example, to change text color to red, background color to navy blue, and bold text color to yellow, issue the following command:

```
$ regis -t red -b navyblue -h yellow
```

5.9.2 The ReGIS Color Command

Unlike the Intergraph-developed `regis` command, the standard ReGIS command does not accept logical color names. If you are familiar with the standard ReGIS command, use the following as a quick reference. Extended procedures for using the ReGIS command follow the quick reference.

Sample Command:	ESC P2ps(m0(b))
Enter ReGIS	ESC P2p
Exit ReGIS	ESC
Text Color	3
Bold Text Color	2
Background Color	0
Clear Mistakes	;

To use the standard ReGIS command, which allows you to define colors according to their individual color component values, follow these steps:

1. To successfully use the ReGIS command, you must block communication with the host so that only the virtual terminal receives the commands. Thus, you must set the virtual terminal offline. To do this, invoke the Setup menu by selecting the Terminal icon from the window icon strip. In the General Setup menu, set Online to off. Select OK to dismiss the menu.
2. Press the <Esc> key and then key in P2p to initiate the ReGIS command.

Note:

The P2p does not appear on the screen.

3. Key in s(m0(b)) to initiate the color segment of the ReGIS command.

s(m initiates the color segment.

0 (zero) indicates a color change to the background. Alternatives include 3 for text and 2 for bolded text.

(b) is the color choice (blue). Single-letter alternatives include the following:

r for red
g for green
d for black (dark)
c for cyan
y for yellow
m for magenta
w for white

These color alternatives require a single letter. Other alternatives require values for hue, lightness, and saturation. Refer to the chart on the next page for a complete list of alternatives. When you use any of these alternatives, specify the color choice in the following format:

(h260l65s60)

The **h** specifies hue, **l** specifies lightness, and **s** specifies saturation. The previous example specifies the color aquamarine.

) ends the color segment.

4. Press the <Esc> key and key in a backslash (\) to exit the ReGIS command.
5. Set the terminal back online.

Notes:

- If you make an error while issuing a ReGIS command, press the semicolon (;) key to restart an escape sequence. Resume by reissuing the color segment of the escape sequence. For example, key in s(m0(b)) to set the background to blue.
- After you set the background color, you can immediately set the text color by keying in the ReGIS command again using the 3 option for text, instead of 0 for background color.

The chart on the following page provides the hue, lightness, and saturation values for all colors available with the ReGIS commands.

Color	Hue	Lightness	Saturation
Aquamarine	260	65	60
Aquamarine (med)	280	50	60
Black (dark)	0	0	0
Blue	0	50	100
Blue, cadet	300	50	25
Blue, cornflower	0	35	25
Blue, dark slate	40	35	60
Blue, light	300	80	25
Blue, light steel	0	65	25
Blue, medium	0	50	60
Blue, midnight	0	25	25
Blue, navy	0	35	60
Blue, sky	320	50	60
Blue, slate	330	50	100
Blue, steel	320	35	60
Coral	150	50	100
Cyan	300	50	100
Firebrick	120	35	60
Gold	150	50	60
Goldenrod	180	65	60
Goldenrod, medium	180	80	60
Green	240	50	100
Green, dark	240	25	25
Green, dark olive	180	25	25
Green, forest	240	35	60
Green, lime	240	50	60
Green, med forest	200	35	60
Green, med sea	240	35	25
Green, med spring	210	50	100
Green, pale	280	65	25
Green, sea	280	35	60
Green, spring	270	50	100
Green, yellow	200	50	60
Grey, dark slate	300	25	25
Grey, dim	0	33	0
Grey, light	0	66	0
Khaki	180	50	25
Magenta	60	50	100
Maroon	80	35	60
Orange	120	50	60
Orchid	60	65	60
Orchid, dark	40	50	60
Orchid, med	20	65	60
Pink	120	65	25
Plum	60	80	60
Red	120	50	100
Red, indian	120	25	25
Red, med violet	100	65	60
Red, orange	90	50	100
Red, violet	80	50	60
Salmon	120	35	25
Sienna	160	35	60
Tan	140	65	60
Thistle	60	80	25
Turquoise	300	80	60
Turquoise, dark	340	65	60
Turquoise, med	300	65	60
Violet	60	25	25
Violet, blue	60	50	25
Wheat	180	80	25
White	0	99	0
Yellow	180	50	100
Yellow, green	220	65	60

5.9.3 The ANSI Color Escape Sequences

The ANSI color escape sequences are used to change the color of text and text blocks (the rectangular region surrounding each character). As mentioned, character attributes created using the ANSI escape sequences are not saved with the catalog feature. Hence, the ANSI sequences satisfy short-term needs.

Unlike the ReGIS commands, the ANSI escape sequences provide a limited number of color alternatives. The ANSI sequences offer the same eight colors for both text and text blocks as the following chart illustrates. Each color has a corresponding code number.

Text		Text Block	
Black	30	Black	40
Red	31	Red	41
Green	32	Green	42
Yellow	33	Yellow	43
Blue	34	Blue	44
Magenta	35	Magenta	45
Cyan	36	Cyan	46
White	37	White	47

If you are familiar with the ANSI escape sequences, use the following as a quick reference. Extended procedures follow the quick reference.

Sample Sequence:	ESC[33;44m
Enter sequence	<ESC>[
Terminate sequence	m
Separate parameters	;
Text colors	30 through 37
Text block colors	40 through 47
Default colors	<ESC>[m

Follow these steps to change the color of a window using ANSI escape sequences:

1. To successfully use the ANSI escape sequences, you must block communication with the host so that only the virtual terminal receives the commands. To do this, invoke the Setup menu by selecting the Terminal icon from the window icon strip. In the General Setup menu, set Online to off. Select OK to dismiss the menu.
2. Press the <Esc> key and then key in a square bracket ([) to indicate that the following information is control data.
3. Key in 31;43m to set the text to red (31) and the text block to yellow (43).

4. Set the virtual terminal back online.

The following include additional uses for ANSI escape sequences:

- The escape sequence example can be used to change only the text color. For example, use this command-line to set the text to red (31):

ESC[31m

- The escape sequences can also be used for changing only the text block color. For example, use this command-line to set the text block color to yellow (43):

ESC[43m

- A third use for the escape sequences is for setting the text color to the window's foreground color and the text block to the window's background color. This variation is represented here:

ESC[m

5.10 Using the Catalog Function to Save Customized Windows

After you have used the various Setup menus and ReGIS color combinations, you will probably have created a virtual terminal that you would like to use regularly. The Catalog function allows you to save the characteristics of that virtual terminal. You can then add the cataloged file to the Workstation Pulldown menu to access it regularly.

Follow these steps to save a customized virtual terminal and add it to the Workstation Pulldown menu:

1. Create the virtual terminal that you will save as a cataloged file.
2. Select the Catalog icon (to the right of the Terminal icon) on the window icon strip.
3. The Catalog Terminal Environment menu displays. Key in the name that you will assign to this terminal in the File Name entry box. The name must have 14 characters or less and cannot include blank spaces.
4. Select the Save option to save the window as a file. `vterm` stores your terminal as a file in the `/usr/lp32/vt200/catalog` directory. Now you can invoke the cataloged window from the command line or you can add it to the Workstation Pulldown menu.

Key in the following command to invoke the window from the command line:

```
$ vterm -l -f <window_name>
```

Follow these steps to add the catalog file to the Workstation Pulldown menu:

1. Access the Configure Pulldown menu. (To do this, select the workstation icon in the upper left-hand corner of the screen. Then, select the Configure option. Select the Menu option to access the Configure Pulldown menu. If you have a root password, you must key it in before you can access the menu.)
2. Set the User/System toggle switch to specify the menu you wish to alter. (The user pulldown menu is valid only for your user account, while the system pulldown menu is valid for all users.)
3. To add a process to the Workstation Pulldown menu, select the New button and key in the process name that will appear on the menu.

4. Key in the command string that creates the virtual terminal. The `-f` option is reserved for creating cataloged virtual terminals. The `-f` immediately followed by the catalog name (in this example, "window_name") creates your custom virtual terminal.

```
vterm -l -f <window_name>
```

Use the same procedure for adding a network virtual terminal to the workstation menu, replacing `-l` with `-n` and your node address.

5. Select the OK button to save all changes to the Workstation Pulldown menu.
6. Select the Workstation Pulldown menu to see the new option.

5.11 Altering the Workstation Pulldown Menu

When certain software products are delivered to the workstation, the command strings that invoke those products are automatically added to the Workstation Pulldown menu, allowing you to invoke processes from this menu. You can customize this menu by adding, modifying, or deleting processes on it. For example, you may add processes that you use regularly. These procedures may include connecting to a specific host, creating a cataloged window, or invoking a product such as Actem or MicroStation, which are not automatically placed on the menu.

Note:

When you alter the Workstation Pulldown menu, you can change the menu for all users or just for your account. The contents of the */usr/lp32/smgr/smgr_pd* determine the entries on the system menu, while the *smgr_pd* file in your login directory determines the menu entries (in addition to the system menu entries) that are available when you are the workstation user. A toggle box on the Configure Pulldown menu allows you to specify whether you are changing the system menu or the workstation user menu.

This section provides separate instructions for adding, modifying, or deleting processes on the Workstation Pulldown menu.

5.11.1 Adding Processes to the Workstation Pulldown Menu

Follow these steps to add processes to the Workstation Pulldown menu:

1. Select the workstation icon in the upper left-hand corner of the screen.
2. From the Workstation Pulldown menu, select the Configure option.
3. Select the Menu option to access the Configure Pulldown menu.
4. Set the User/System toggle box. Your additions to the menu will be reflected in the *smgr_pd* file in your home directory if the switch is set to User and in the */usr/lp32/smgr/smgr_pd* file if the switch is set to System.
5. To add a process to the Workstation Pulldown menu, select the New button.
6. Key in the process name that will appear in the menu.

7. Key in the command string that invokes the process. The Screen Manager software, which controls the menu, does not check command string entries for validity. If you select a menu item and no process is invoked, either the command was not found or the command could not be executed properly. The following examples of command strings are commonly added to the menu:

- To add a specific host that you connect to frequently, key in the following command string, where “node__address” is the host’s nodename or network address:

```
vterm -n node__address
```

- To add a cataloged window to the menu, key in the following command string, where “window__name” is the name of the cataloged window:

```
vterm -l -fwindow__name
```

- To add the process that invokes the crm product to the menu, key in the following command string:

```
/usr/ip32/crm/crm.sh
```

- The following table lists the command strings associated with standard workstation products and utilities. You can use this list to restore an option that may have been deleted.

Here are some common processes and their corresponding command strings:

Process	Command String
terminal	/usr/ip32/vt200/creat__term
local	vterm -l
shutdown	/usr/ip32/smgr/smgrutil -confirm -shutdown
reboot	/usr/ip32/smgr/smgrutil -confirm -reboot
console	/usr/ip32/smgr/smgrutil -console -passive -origin 578 446 -collapse 1119 765 -small -log /usr/tmp/console.log -show

8. Select the OK button to save all changes.
9. Select the Workstation Pulldown menu to see the new option.

5.11.2 Modifying the Workstation Pulldown Menu

Follow these steps to modify the command string or the menu name associated with each process on the Workstation Pulldown menu:

1. Select the workstation icon in the upper left-hand corner of the screen.
2. From the Workstation Pulldown menu, select the Configure option.
3. Select the Menu option to access the Configure Pulldown menu.
4. Set the User/System toggle box. Your modifications to the menu will be reflected in the *smgr_pd* file in your home directory if the switch is set to User and in the */usr/lp32/smgr/smgr_pd* file if the switch is set to System.
5. Select the process you will modify from the list.
6. Key in the new name that will appear in the menu. There is no maximum character length for menu entries.
7. Key in the new command string. The Screen Manager software, which controls the menu, does not check command string entries for validity. If you select a menu item and no process is invoked, either the command was not found or the command could not be executed properly. Refer to 5.11.1 for process command string examples.
8. Select the OK button to save all changes.
9. Select the Workstation Pulldown menu to see the new option.

5.11.3 Deleting Processes from the Workstation Pulldown Menu

Follow these steps to delete processes from the Workstation Pulldown menu:

1. Select the workstation icon in the upper left-hand corner of the screen.
2. From the Workstation Pulldown menu, select the Configure option.

3. Select the Menu option to access the Configure Pulldown menu.
4. Set the User/System toggle box. Your deletions from the menu will be reflected in the `.smgr_pd` file in your home directory if the switch is set to User and in the `/usr/lp32/smgr/smgr_pd` file if the switch is set to System.
5. To delete a process on the Workstation Pulldown menu, select the process name to delete on the list.
6. Select the Delete button.
7. Select the OK button to save all changes.
8. Select the Workstation Pulldown menu to see the new option.

5.12 Changing Screen Configuration

You can change the number of virtual screens (single-monitor workstations only) or the number of message strips (dual-monitor workstations only) and the location of the message strip (single- or dual-monitor workstations) any time. This section describes each process.

5.12.1 Changing the Number of Virtual Screens (Single-Monitor Workstations Only)

Intergraph single-monitor workstations can be configured for one or two virtual screens. You can run multiple processes on each virtual screen.

By default, single-monitor workstations are configured for two virtual screens. To toggle between virtual screens, select the virtual screen icon to the right of the workstation icon.

Some processes require one virtual screen while others require two. You can reconfigure the number of virtual screens for single-monitor workstations any time (as long as all windows have been deleted).

All 240-series (and above) workstations have 512 colors available to both virtual screens. All 220-series (and below) workstations configured for one virtual screen have 2^5 (32) colors available for that virtual screen. All 220-series (and below) workstations configured for two virtual screens have 2^4 (16) colors available for the first virtual screen and 2^1 (2) colors available for the second virtual screen.

Follow these steps to reconfigure the number of virtual screens for single-monitor workstations:

1. Log out and delete all windows, including the console window.
2. Select the workstation icon.
3. Select the Configure option on the Workstation Pulldown menu.
4. Select the Screen(s) option to access the Configure Virtual Screens menu.
5. Select either the single-screen or dual-screen configuration.
6. Select the OK button to reconfigure the virtual screens.

5.12.2 Changing the Number of Message Strips (Dual-Monitor Workstations Only)

For dual-monitor workstations, you can reconfigure the number of message strips.

By default, dual-monitor workstations are configured for two message strips (one on each screen). You can configure the workstation for two message strips, one message strip on the left screen, or one message strip on the right screen.

Follow these steps to reconfigure the number of virtual screens for dual-monitor workstations:

1. Select the workstation icon.
2. Select the Configure option on the Workstation Pulldown menu.
3. Select the Screen(s) option to access the Configure Virtual Screens menu.
4. Select the appropriate box to configure the workstation for two message strips, one message strip on the left screen, or one message strip on the right screen.
5. Select the OK button to reconfigure the virtual screens.

5.12.3 Changing the Message Strip(s) Location

By default, the message strip appears at the top of each screen. To move the message strip to the bottom of the screen, tap the process ID field, located in the top, right-hand corner of the screen. Tap the process ID again to move the message strip back to its original position at the top of the screen.

This process works the same for single- and dual-monitor workstations.

5.13 Changing the Background Pattern

The Background option on the Configure menu enables you to choose a background pattern for the Screen Manager environment. The Screen Manager software honors the background and foreground colors defined in */usr/lp32/smgr_vlt*.

Follow these steps to change the background pattern:

1. Select the workstation icon.
2. Select the Configure option from the Workstation Pulldown menu.
3. Select the Background option from the Configure menu.
4. Ten background patterns display; nine are predefined. The pattern in the lower right-hand corner is user-configurable.

To create a new pattern for the user-configurable choice, edit the */usr/lp32/smgr/bgpattern* file. This file is a collection of background patterns. The first pattern displays in the Background menu. The patterns in this file consist of ASCII characters. A zero specifies that a screen pixel is turned off (pixel off), while any other character specifies that a screen pixel is turned on (pixel on). If you edit this file, you must reboot before the Screen Manager software recognizes the new background pattern.

Select one predefined background pattern or the user-configurable pattern for your workstation.

5. Select the OK button to save the changes. If you choose a predefined pattern, the background pattern changes immediately. If you choose the user-configurable pattern, edit the */usr/lp32/smgr/bgpattern* file to define a new pattern. Then, reboot to see the new background pattern.

5.14 Changing the Screen Color

To change the screen's default background and foreground colors, edit the `/usr/lp32/smgr/smgr_vlt` file. The default file delivered specifies black as the background color and white as the foreground color. You can also change the color of individual windows. For instructions, follow steps in 5.9, "Creating Color Windows."

Each line in the `/usr/lp32/smgr/smgr_vlt` file consists of an integer from 0 to 15, 0 defining dark and 15 defining bright. The first three lines define the background color and the last three define the foreground. The format of this file is as follows:

```
background red value
background green value
background blue value
foreground red value
foreground green value
foreground blue value
```

For example, the default file consists of the following lines:

```
0
0
0
15
15
15
```

The first three lines are all zero, setting the background color to black. The last three are all 15, setting the foreground color to white.

The following combination sets define the foreground color to white and the background color to dark blue, royal blue, red, green, and grey:

Dark Blue	Royal Blue	Red	Green	Grey
0	0	15	0	9
0	0	0	15	9
8	15	0	0	9
15	15	15	15	15
15	15	15	15	15
15	15	15	15	15

Note:

After you edit the file, you must either reboot the workstation or reconfigure the number of virtual screens before the screen's color will change.

5.15 Invoking Processes Automatically

The `/usr/ip32/smgr/def_proc`, `/usr/<login>/def_proc`, and `/usr/ip32/smgr/sys_proc` files allow you to specify certain processes to be automatically invoked by the Screen Manager software. This section describes each of these files.

5.15.1 The `def_proc` File

The `/usr/ip32/smgr/def_proc` shell script specifies the user and system processes that will be invoked automatically. This script is invoked after the workstation user has been established.

Notes:

- All programs invoked by this script should be followed by an ampersand (&) so they will be invoked as background processes.
- The command string to invoke a local VT220 window is the first command in the file and should be followed by a *sleep* of at least 5 seconds. A sleep is simply a pause that slows the activation of the other processes. This ensures that VT220 (the ttx00 window) is the active process before other processes execute.
- Lines that begin with a # are comments. They do not invoke any processes. (This is a standard System V shell script convention.)

The following is an example of a `/usr/ip32/smgr/def_proc` file.

```
# The def_proc is a shell script executed by Screen Manager
#
# Console is invoked in sys_proc.
#
/usr/ip32/vt200/vterm -1 &
sleep 5
```

5.15.2 The `.def_proc` File

The `def_proc` shell script specifies the user processes that will be invoked automatically. This script is run at Screen Manager startup if the Execute Default Processes box was checked on the Workstation Login menu or if the workstation user was determined by the `/etc/wkstuser` file.

Notes:

- Use the ampersand (&) symbol to specify for a process to be invoked as a background process.
- The # symbol denotes a comment. (This is a standard System V shell script convention.)

The `def_proc` file will not automatically exist in your home directory. Instead, you must create it. The following is an example of a `def_proc` file:

```
# This sample .def_proc file will cause several processes to be
# invoked at Screen Manager initialization. Note that this
# file must exist in a user directory. The .def_proc file for
# a particular user is run only when the user has been
# established as the current Workstation User.
#
# The following line invokes the Toolbox clock utility.
/usr/ip32/toolbox/iclock/iclock.sh
#
# The following line invokes the CLIX Resource Monitor (CRM).
/usr/ip32/crm/crm.sh passive
```

5.15.3 The `sys_proc` File

The `/usr/tp32/smgr/sys_proc` file is a shell script that runs default system processes as the superuser. By default, only one process is invoked with this script (the console window).

Notes:

- Ideally, `sys_proc` should contain only the line to invoke the console so that output is available before any user has logged in.
- Programs invoked by this script should be followed by an ampersand (&) so they will be invoked as background processes.
- The # symbol denotes a comment. (This is a standard System V shell script convention.)

The following is the default `/usr/tp32/smgr/sys_proc` file:

```
# All processes added are run as ROOT
#
/usr/tp32/smgr/smgrutil -console -passive -origin 578 446 -collapse \
1119 765 -small -log /usr/tmp/console.log -show &
```

Note:

You will notice that the Console window pops up when it receives a system message. If you wish to change the console window so that it does not pop up, edit the `sys_proc` file and remove the `-show` option.



Chapter 6: Setting Up and Using a Line Printer

This chapter provides instructions for connecting a dedicated line printer to a workstation and using the `lp` command to print to it.

This chapter contains the following sections:

6.1	Configuring a Line Printer	6-3
6.2	Using the Line Printer	6-4
6.2.1	Submitting a Print Request	6-4
6.2.2	Monitoring a Print Request	6-4
6.2.3	Canceling a Print Request	6-5

6-2 *Setting Up and Using a Line Printer*

6.1 Configuring a Line Printer

A dedicated line printer is a useful tool for workstation users. Follow these steps to configure a line printer:

1. Connect the line printer to the RS-232 port on the workstation. A special serial cable must be used to connect the line printer to the workstation. The pin connections for the cable are as follows:

Workstation Pin	Line Printer Pin
2	3
3	2
5	20
7	7

2. Configure the RS-232 port for the line printer on the Peripheral Configuration Utility Page. To do this, reboot the system and enter the Utility Pages. Access the Peripheral Configuration Utility Page. On the Serial Parameters section of this Utility Page, set the Port Type to RS-232 and then set the parameters appropriately for the line printer you are using.
3. You must alter the *lp* file to reflect site-specific factors such as the RS-232 port number and the printer model type. To do this, use an ASCII text editor to edit the */etc/init.d/lp* file. This file (by default) contains the following line:

```
/usr/lib/lpadmin -plp0 -v/dev/tty00 -mepson
```

Substitute the proper port and model type for your line printer in this line. The following are valid printer model types. You can list these model types by keying in the following command:

```
$ ls /usr/spool/lp/model
1640      epson      hp          ph.daps    prx
dumb      f450      model.mk   pprx
```

If the name of the printer model you are using is not listed, specify *dumb* as the model type.

4. Edit the */usr/spool/lp/model/<model_type>* file for the printer model you are using. In this file, check the printer characteristics to ensure that they match your printer. (Note that baud rate is specified by *stty*.)

5. Initialize the lp software so that the processes required to operate the line printer will be started each time the system is booted. To initialize software, the *root* user must execute the */etc/init.d/lp* file as follows with an argument of *init*:

```
# cd /etc/init.d  
# ./lp init
```

The line printer is now ready to be used. Proceed to the next section for instructions for printing to this printer.

6.2 Using the Line Printer

After you have properly configured the line printer (as described in the previous section), you can print to it. To use the line printer from the System V shell environment, you should be familiar with submitting, monitoring, and canceling a print request. These tasks are described in this section.

6.2.1 Submitting a Print Request

Use the `lp` command to submit a print request to the printer. In addition, the `pr` command can be used with `lp` to format the output. (`pr` labels each page every 66 lines.) To use these commands, you will pipe the output from the `pr` command into the `lp` command, as shown in the following example:

```
$ pr <filename> | lp
```

Both `lp` and `pr` have various command-line options. For more information on using the `lp` and `pr` commands, refer to the manual page in the System V Online Documentation product.

6.2.2 Monitoring a Print Request

Use the `lpstat` command to monitor the print request. To use this command, key in `lpstat` at the command line. The job currently printing and all jobs waiting to be printed will display on the screen, as shown in the following example:

```
$ lpstat
```

```
lp0-54      sb      12470    Feb 12 14:39 on lp0
lp0-55      sb      4950     Feb 12 14:39
lp0-56      tb      5197     Feb 12 14:39
lp0-57      lwc     11821    Feb 12 14:39
```

Various command-line options exist for `lpstat`. Refer to the System V Online Documentation for more information on using this command.

6.2.3 Canceling a Print Request

Use the cancel command to abort a job currently printing or jobs that are waiting to print.

Key in the following (where <printer_name> represents the name of the printer, such as lp0) to cancel a job currently printing:

```
$ cancel <printer_name>
```

Key in the following (where <job_ID> represents the job identification number, such as lp0-57) to cancel a job waiting to print:

```
$ cancel <job_ID>
```

For more information on the cancel command, refer to the manual page in the System V Online Documentation product.



Chapter 7: Creating and Maintaining File Systems

File systems are the structures that organize data logically and accessibly on the hard disk. They allow you to access *devices* such as hard disk partitions and peripherals. By default, internal hard disks for Intergraph systems have *root* and */usr* file systems created on them. The *root* (/) file system is used to organize operating system files and the */usr* file system is used to organize application software and user files. Other file systems may exist on additional user partitions on the disk (usually called *usr2*, *usr3*, and so on). File systems for these nonstandard partitions are not created by default; they must be created manually, as described in this chapter.

In addition to explaining how to create a file system and prepare it for use, this chapter provides information essential for maintaining file systems.

This chapter contains the following sections:

7.1	Creating a File System	7-3
7.1.1	Creating a Device File	7-3
7.1.2	Creating a File System on a Partition	7-9
7.1.2.1	Creating a Standard File System	7-10
7.1.2.2	Creating a Fast File System	7-11
7.1.3	Mounting a Partition	7-12
7.2	Monitoring Disk Space	7-13
7.2.1	Displaying the Amount of Free Disk Space	7-13
7.2.2	Displaying the Amount of Disk Space Used	7-14
7.2.3	Freeing Disk Space	7-14
7.3	Checking the Integrity of a File System	7-17

7-2 *Creating and Maintaining File Systems*

7.1 Creating a File System

A file system allows you to access a hard disk partition. File systems already exist for standard partitions such as `root` and `usr`, but not for nonstandard partitions such as `usr2` and `usr3`. Thus, to allow a nonstandard partition to be accessed, you must create a file system on it.

To create and be able to access a file system on a hard disk partition, you must complete the following steps:

1. Create a device file for the partition if one does not already exist. (Refer to 7.1.1.)
2. Create the file system (S51K or FFS). (Refer to 7.1.2.)
3. Mount the partition. (Refer to 7.1.3.)

7.1.1 Creating a Device File

The first step in preparing a nonstandard partition for use is creating the device file for the partition if it does not already exist. The device file must be created in the `/dev/dsk` directory. This section explains the procedure for checking to ensure that the device file exists and creating it if it does not exist.

The example in this section shows how to create a device file for the `s0u0p7.4 (usr2)` internal hard disk partition. The procedure for creating a device file for an external hard disk is no different than the procedure for an internal hard disk except that the device file reflects a different SCSI ID for the external hard disk. The SCSI ID for an internal hard disk is always 0; the SCSI ID for an external hard disk is always greater than 0 and is usually 1, 2, or 3.

Follow these steps to check for and create a device file:

1. Secure the following information about the partition:
 - Partition name (for example, `usr2`)
 - Size in blocks
 - Block major (b-maj) number
 - Block minor (b-min) number
 - SCSI ID number

You can find this information in the Partition Table on the Disk Partitioning Utility Page. In addition, the charts on the following pages provide the partition name, size, block major number, and block minor number if you already know the SCSI ID.

7-4 Creating and Maintaining File Systems

Table 7-1. Device File Information for Creating Partitions for SCSI ID 0, LUN 0

Name	Device	Device File Type **	maj #	min #
root	/dev/dsk/s0u0p7.0	b	64	112
	/dev/rdisk/s0u0p7.0	c	64	112
swap*	/dev/dsk/s0u0p7.1	b	64	113
	/dev/rdisk/s0u0p7.1	c	64	113
tmp	/dev/dsk/s0u0p7.2	b	64	114
	/dev/rdisk/s0u0p7.2	c	64	114
usr	/dev/dsk/s0u0p7.3	b	64	115
	/dev/rdisk/s0u0p7.3	c	64	115
usr2	/dev/dsk/s0u0p7.4	b	64	116
	/dev/rdisk/s0u0p7.4	c	64	116
usr3	/dev/dsk/s0u0p7.5	b	64	117
	/dev/rdisk/s0u0p7.5	c	64	117
usr4	/dev/dsk/s0u0p7.6	b	64	118
	/dev/rdisk/s0u0p7.6	c	64	118

* Not used for file systems; reserved for swap space.

** The device file can be block (b) or character (c) type.

Table 7-2. Device File Information for Creating Partitions for SCSI ID 1, LUN 0

Name	Device	Device File Type **	maj #	min #
root	/dev/dsk/s1u0p7.0	b	66	112
	/dev/rdisk/s1u0p7.0	c	66	112
swap*	/dev/dsk/s1u0p7.1	b	66	113
	/dev/rdisk/s1u0p7.1	c	66	113
tmp	/dev/dsk/s1u0p7.2	b	66	114
	/dev/rdisk/s1u0p7.2	c	66	114
usr	/dev/dsk/s1u0p7.3	b	66	115
	/dev/rdisk/s1u0p7.3	c	66	115
usr2	/dev/dsk/s1u0p7.4	b	66	116
	/dev/rdisk/s1u0p7.4	c	66	116
usr3	/dev/dsk/s1u0p7.5	b	66	117
	/dev/rdisk/s1u0p7.5	c	66	117
usr4	/dev/dsk/s1u0p7.6	b	66	118
	/dev/rdisk/s1u0p7.6	c	66	118

* Not used for file systems; reserved for swap space.

** The device file can be block (b) or character (c) type.

7-6 *Creating and Maintaining File Systems*

Table 7-3. Device File Information for Creating Partitions for SCSI ID 2, LUN 0

Name	Device	Device File Type **	maj #	min #
root	/dev/dsk/s2u0p7.0	b	68	112
	/dev/rdisk/s2u0p7.0	c	68	112
swap*	/dev/dsk/s2u0p7.1	b	68	113
	/dev/rdisk/s2u0p7.1	c	68	113
tmp	/dev/dsk/s2u0p7.2	b	68	114
	/dev/rdisk/s2u0p7.2	c	68	114
usr	/dev/dsk/s2u0p7.3	b	68	115
	/dev/rdisk/s2u0p7.3	c	68	115
usr2	/dev/dsk/s2u0p7.4	b	68	116
	/dev/rdisk/s2u0p7.4	c	68	116
usr3	/dev/dsk/s2u0p7.5	b	68	117
	/dev/rdisk/s2u0p7.5	c	68	117
usr4	/dev/dsk/s2u0p7.6	b	68	118
	/dev/rdisk/s2u0p7.6	c	68	118

* Not used for file systems; reserved for swap space.

** The device file can be block (b) or character (c) type.

Table 7-4. Device File Information for Creating Partitions for SCSI ID 3, LUN 0

Name	Device	Device File Type **	maj #	min #
root	/dev/dsk/s3u0p7.0	b	70	112
	/dev/rdisk/s3u0p7.0	c	70	112
swap*	/dev/dsk/s3u0p7.1	b	70	113
	/dev/rdisk/s3u0p7.1	c	70	113
tmp	/dev/dsk/s3u0p7.2	b	70	114
	/dev/rdisk/s3u0p7.2	c	70	114
usr	/dev/dsk/s3u0p7.3	b	70	115
	/dev/rdisk/s3u0p7.3	c	70	115
usr2	/dev/dsk/s3u0p7.4	b	70	116
	/dev/rdisk/s3u0p7.4	c	70	116
usr3	/dev/dsk/s3u0p7.5	b	70	117
	/dev/rdisk/s3u0p7.5	c	70	117
usr4	/dev/dsk/s3u0p7.6	b	70	118
	/dev/rdisk/s3u0p7.6	c	70	118

* Not used for file systems; reserved for swap space.

** The device file can be block (b) or character (c) type.

2. Boot to System V, log in, and access the superuser account as follows:

```
login: username
$ su
#
```

Note:

If you restored the */usr* file system(s), user accounts will no longer exist. Log in using the *sys* account. You will need to restore user accounts by recovering the */etc/passwd* and */etc/group* files from the previous backup or recreate user accounts. Refer to Chapter 3, "Creating and Maintaining User Accounts."

3. List the contents of the */dev/dsk* or */dev/rdisk* directory as follows to verify that a file with the same name as the new partition (s0u0p7.4 in this example) does not already exist. If it already exists, you will not need to create the device file, and you can proceed to 7.1.2 for instructions on creating a file system.

```
# ls /dev/dsk
```

OR

```
# ls /dev/rdisk
```

4. If the appropriate device file does not exist, create it by using the **mknod** (make node) command. This command creates a device file, or *special file*.

The syntax for this command is as follows, where "name" is the device name, "b" specifies block device file type (*/dev/dsk*) and "c" specifies character device file type (*/dev/rdisk*), "b-maj" is the block major number, and "b-min" is the block minor number:

```
# /etc/mknod /dev/dsk/name b b-maj b-min
```

OR

```
# /etc/mknod /dev/rdisk/name c b-maj b-min
```

For example, to create a System V block device file for partition s0u0p7.4 (for *usr2*), key in the **mknod** command and the partition information obtained in step 1 as follows:

```
# /etc/mknod /dev/dsk/s0u0p7.4 b 64 116
```

For more information about the **mknod** command and device numbering scheme, refer to the */dev/dsk/README* or */dev/rdisk/README* file.

Note:

If you created a device file for a **stash** partition, you must key in the following command to change the mode for this device so that all users can access the new partition:

```
# chmod 666 /dev/dsk/s0u0pf.0
```

After you execute this command, the permissions for this file will be as follows:

```
rw-rw-rw-
```

5. List the contents of the *root* (/) directory with the following command to ensure that a */usr2* directory (or name of another */usr* directory that you are creating) does not already exist.

```
# ls /
```

6. If this directory does not exist, create it by keying in the following. (This example creates the */usr2* directory).

```
# mkdir /usr2
```

7. Continue to 7.1.2 to create a file system on the new partition.

7.1.2 Creating a File System on a Partition

Once a partition has a device file, a file system must be created on the partition. Creating a new file system (or overwriting an old file system) overwrites any existing data on the disk device. Thus, back up any files on the existing file system because all data will be lost.

You can create one of two types of file systems: a standard file system (S51K) or a Fast File System (FFS). S51K file systems contain 1K-byte units, whereas Fast File Systems contain 8K-byte units. For this reason, Fast File Systems read large files more efficiently than standard S51K file systems do. Fast File Systems are most efficiently used for file systems that contain many large files such as design files. Thus, you can convert file systems such as */usr2* to Fast File Systems to store and access large files efficiently. By default, the *root* (/) file system is created as a standard file system and the */usr* file system is created as a Fast File System.

Section 7.1.2.1 describes how to create a standard file system, while 7.1.2.2 describes how to create a Fast File System.

7.1.2.1 Creating a Standard File System

Follow these steps to create a standard file system on a partition:

1. Use the `mkfs` (make file system) command to create a standard file system. The syntax for this command is as follows, where "name" is the partition name and "size" is the partition size (in blocks):

```
# /etc/mkfs /dev/dsk/name size
```

For example, to create a file system on the `s0u0p7.4 (usr2)` partition, key in the following:

```
# /etc/mkfs /dev/dsk/s0u0p7.4 100000
```

Messages similar to the following appear:

```
MKFS: /dev/rdisk/s0u0p7.4  
(DEL if wrong)
```

Then, messages similar to the following appear:

```
bytes per logical block = 1024  
total logical blocks = 21800  
total inodes = 5440  
gaps (physical blocks) = 1  
cylinder size (physical blocks) = 128
```

```
mkfs: Available blocks = <size of partition>
```

These messages confirm that the file system has been created.

2. After creating a file system on the desired partition, run the `/etc/labelit` program to label the partition and attach a logical base directory name to the file system. For instance, to run `labelit` on the `s0u0p7.4` device (`usr2`), key in the following, where the first `usr2` represents the file system's mounted name and the second represents the volume name (which is user-definable):

```
# /etc/labelit /dev/dsk/s0u0p7.4 usr2 usr2
```

3. Continue to 7.1.3, "Mounting a Partition."

7.1.2.2 Creating a Fast File System

Follow these steps to create a Fast File System on a partition:

1. To create a Fast File System use the `newfs` (new file system) command. The syntax for this command consists of the following, where "name" is the partition name and "disk" is the disk type:

```
# /etc/newfs /dev/dsk/name disk
```

Note:

Fast File Systems allocate space for the file system structure. You will not be able to access this space. By default, 10 percent of the FFS is allocated for the file system structure. You can alter this percentage by using the `-m` option on the command line when you create the FFS. The following example specifies that only 5 percent of the file system will be allocated:

```
# /etc/newfs -m5 /dev/dsk/name disk
```

Notice that the `newfs` command requires a disk type. The following chart displays the disk types that Intergraph supplies.

Disk Size	Disk Type
80 MB (Quantum)	FDSK150
80 MB (Priam)	FDSK131
156 MB (CDC)	FDSK155
355 MB (Maxtor)	FDSK226
584 MB (Fuji)	FDSK211
670 MB (Maxtor)	FDSK230

For example, key in the following to create a file system on the `s0u0p7.4` (`usr2`) partition for a 156-MB hard disk:

```
# /etc/newfs /dev/dsk/s0u0p7.4 FDSK155
```

Unlike the `mkfs` command, the `newfs` command does not require the file system size. Instead, it derives the file system size from the partition size.

2. After creating a file system on the desired partition, run the `/etc/labelit` program to label the partition and attach a logical base directory name to the file system. For instance, to run `labelit` on the `s0u0p7.4` device (`usr2`), key in the following, where the first `usr2` represents the file system's mounted name and the second represents the volume name (which is user-definable):

```
# /etc/labelit /dev/dsk/s0u0p7.4 usr2 usr2
```

3. Continue to 7.1.3, "Mounting a Partition."

7.1.3 Mounting a Partition

To access and use a partition, you must *mount* it on a base directory (an empty directory that serves as the top directory for the partition). Mounting a partition sets up a logical definition for the file system so that it can be recognized by the kernel and accessed by users. Partitions can be mounted on standard or Fast File Systems.

To mount a partition, you will use the `mount` command, located in the `/etc` directory. Follow these steps to mount a partition:

1. The syntax for the `mount` command is as follows, where `type` represents the file system type, `name` represents the partition name, and `directory` represents the directory where the file system will reside:

```
# /etc/mount -f type /dev/dsk/name directory
```

For example, assuming that a standard (S51K) file system now exists on `/dev/dsk/s0u0p7.4`, you could mount the `usr2` partition with the following command:

```
# /etc/mount -f S51K /dev/dsk/s0u0p7.4 /usr2
```

The following command mounts a partition with a Fast File System created on it:

```
# /etc/mount -f FFS /dev/dsk/s0u0p7.4 /usr2
```

2. To mount the partition and check the file system automatically each time you boot, add the partition to the `/etc/fstab` file. Otherwise, you will need to mount the partition manually every time you boot.

The following examples modify the appropriate operating system file so that `/dev/dsk/s0u0p7.4` mounts automatically on directory `/usr2` when the workstation boots.

- For a standard file system, add a line similar to the following to the `/etc/fstab` file:

```
/dev/dsk/s0u0p7.4 /usr2 S51K
```

- For a Fast File System, add a line similar to the following to the `/etc/fstab` file:

```
/dev/dsk/s0u0p7.4 /usr2 FFS
```

4. Now that the partition has been mounted, you will be able to access it through System V commands.

7.2 Monitoring Disk Space

To optimize system performance, you should remove unneeded files regularly. In addition, the hard disk may run out of space and you will need to free disk space. For example, when you install new products on the hard disk, you may run out of space.

This section provides instructions for determining the amount of space used and suggestions for freeing disk space.

7.2.1 Displaying the Amount of Free Disk Space

The System V `df` (disk space free) command displays the amount of unused disk space (in blocks) and i-nodes. You will want to use this command frequently to determine whether you have enough space to load or save files.

The `df` command syntax is as follows:

```
$ df [-t] [<filesystem>]
```

For example, key the following to display the amount of free disk space on all mounted file systems:

```
$ df

/      (/dev/dsk/s0u0p7.0):      8674 blocks      2015 i-nodes
/usr   (/dev/dsk/s0u0p7.3):      5758 blocks      12496 i-nodes
/usr3  (/dev/dsk/s1u0p7.0):      6914 blocks      14687 i-nodes
```

The `-t` option lists the total number of blocks and i-nodes allocated to the file system. It also displays the number of free blocks and i-nodes.

```
$ df -t

/      (/dev/dsk/s0u0p7.0):      8674 blocks      2015 i-nodes
      total: 25000 blocks  3120 i-nodes
/usr   (/dev/dsk/s0u0p7.3):      5758 blocks      12496 i-nodes
      total: 100052 blocks 12496 i-nodes
/usr3  (/dev/dsk/s1u0p7.0):      6914 blocks      14687 i-nodes
      total: 154602 blocks 19312 i-nodes
```

7.2.2 Displaying the Amount of Disk Space Used

The `du` (disk space used) command displays the size of a directory or file. This command is commonly used before a file is copied to another system.

The `du` command syntax is as follows:

```
$ du -[asr] [<filename>]
```

For example, you would key in the following to display the size of a directory:

```
$ du /usr/jim
6416 /usr/jim
```

- The `-a` option causes an output line to be generated for each file.

```
$ du -a /usr/jim
63 /usr/jim/house.dgn
7 /usr/jim/mbox
```

- The `-s` option causes only the total (for each specified file) to display.

```
$ du -s /usr/jim/house.dgn
63 /usr/jim/house.dgn
```

- The `-r` option causes messages concerning any files that cannot be read to display. If this option is not specified, no output will be displayed for these files.

7.2.3 Freeing Disk Space

This section provides suggestions for freeing disk space. You are not required to follow all or any of them. However, each step will help free disk space.

1. Product install scripts place products in `/usr/tmp` before they are moved to the appropriate product directory. This directory's contents are removed when you reboot. Thus, you can regain disk space by rebooting or removing the contents. To remove the contents of the `/usr/tmp` directory, key in the following command:

```
# rm -r /usr/tmp/*
```

2. *Core* files can be created through several possible occurrences ranging from machine checks to graphics aborts. Use the `find` command to locate all core files on the disk. *Core* files will be printed on the screen as they are located. An example of possible output follows:

```
# find / -name core -print
```

```
/usr/ip32/vt200/core  
/core
```

Delete the core files using the `rm` command as follows:

```
# rm /usr/ip32/vt200/core  
# rm /core
```

Do not delete any directories called *core*.

3. The file system checker, `fsck`, checks the file system's connectivity and integrity and places files in the *lost+found* directory. The `fsck` program runs automatically at boot-up when the system was not shut down properly. This program finds and stores files not linked orderly to the file system. Remove all files in the *lost+found* directory by keying in the following:

```
# rm -r /usr/lost+found/*
```

Note:

If this command will not remove all of the files in this directory, key in the following to remove (unlink) the */usr/lost+found* directory. Then, you will need to recreate this directory:

```
# unlink /usr/lost+found  
# mkdir /usr/lost+found
```

4. The following utilities are delivered with the `systemv` product. If you do not use any of these utilities (`assist`, `help`, and `graf`), you can free approximately 8,000 blocks by removing them. Remove these utilities by keying in the following commands:

```
# rm -r /usr/lib/assist  
# rm -r /usr/lib/help  
# rm -r /usr/bin/graf
```

5. To remove Intergraph products that you do not use, you can use the remove shell script. This shell script allows you to remove any software products that have a *remove.sh* file. To use this utility, you must be in superuser mode. Key in the following to invoke the remove utility:

```
# /usr/ip32/deltools/remove
```

First, the remove shell script prompts you for the file system to reclaim space in. Then, the remove shell script lists products that can be removed. At this menu, specify any products you wish to remove, and the remove utility will remove the products, including all related files from the hard disk.

Do not remove any of the following products. These products are collectively called *CLIPPER System Software*. The CLIPPER System Software is required to run the System V operating system. You may remove any product except these:

- Delivery Tools (*deltools*)
- System V 3.1 File System (*systemv*)
- System V 3.1 Boot Images (*unixboot*)
- CLIPPER Graphics Libraries (*environ_s*)
- Screen Manager (*screenmgr*)
- DEC VT220 Emulation (*vt220*)
- Workstation Network Software (*inc*)
- Geometry Pipeline Host Shared Library (*gpipe_s*)
- Workstation Graphic Resources (*resources*)
- Workstation/Server Diagnostics (*diag*)
- I/Forms Run-Time Package (*forms_s*)
- RIS Client Support Package (*risccu*)

6. If you have an external hard disk, you can load some software products to it. For example, if your external hard disk has a */usr2* file system created on it, you can load software products such as Intergraph Online News (*internews*) and System V Online Documentation (*sysvdoc*) to */usr2/ip32* rather than */usr/ip32*. To do this, remove the products from the */usr* file system with the remove utility described in the previous step. Then, invoke *newprod*. In *newprod*'s interactive mode, key in *f* and specify the */usr2* file system. Then, load the *internews* and *sysvdoc* products to */usr2*.

Note:

Products loaded to an alternate file system are symbolically linked to */usr/ip32*.

7.3 Checking the Integrity of a File System

The **fsck** (file system check) utility checks a file system for inconsistencies. When problems with the file system are discovered, the utility asks whether to attempt to repair them. Often, problems cannot be repaired without the loss of data. However, if you ignore problems discovered by **fsck**, they may spread to the rest of the file system.

If a system was not shut down properly, the **fsck** utility runs automatically the next time the system boots. You can also run this utility interactively through the command line. Only the superuser can execute this command. **Never run fsck on a mounted file system (with the exception of root).**

The syntax for this command is as follows, where the **-y** option allows you to specify a **yes** response for all questions, the **-n** option allows you to specify a **no** response for all questions, and **<file_system>** represents the file system to check (such as **/dev/dsk/s1u0p7.5**):

```
# fsck [-y] [-n] <file_system>
```

The following example checks the **/usr3** file system on an external hard disk and responds to all prompts with a **no** response:

```
# fsck -n /dev/dsk/s1u0p7.5
```

It is a good idea to respond to all prompts with **no** the first time you run **fsck** so that you are familiar with the problem areas and how many files will be affected. Then, you can run **fsck** again and respond to each prompt individually.

For more information on the **fsck** utility, refer to the manual page in the System V Online Documentation product.



Chapter 8: Monitoring Processes with CRM

This chapter contains the following sections:

8.1	Installing CRM	8-3
8.2	Invoking the Curses-Based FMLI Interface for CRM	8-4
8.2.1	Graphics Monitors	8-4
8.2.2	Alphanumeric Terminals	8-4
8.3	Using the FMLI Interface	8-5
8.4	System Monitors	8-7
8.4.1	Running System Monitors from the Command Line	8-7
8.4.2	Running Graphics-Based System Monitors	8-9
8.4.3	Using the System Monitors from the Curses-Based FMLI Menus	8-10
8.4.3.1	Change Defaults Form	8-10
8.4.3.2	Monitor Parameters (monparam)	8-11
8.4.3.3	Top Fault Monitor (topfault)	8-12
8.4.3.4	Top Memory Monitor (topmem)	8-14
8.4.3.5	Top CPU Monitor (topcpu)	8-15
8.4.3.6	Top I/O Monitor (topio)	8-15
8.4.3.7	Top System Monitor (topsys)	8-16
8.4.3.8	Show Open Files (showfiles)	8-17
8.4.3.9	Show Memory Usage (showmemory)	8-17
8.5	Process Monitors	8-19
8.5.1	Running Process Monitors from the Command Line	8-19
8.5.2	Using the Process Monitors Interactively from Curses-Based FMLI Menus	8-24
8.5.2.1	Select Process to Monitor	8-24
8.5.2.2	Select Profiler Options	8-25
8.5.2.3	Enable Page Faults	8-26
8.5.2.4	Enable System Calls	8-26
8.5.2.5	Run Profiler	8-27
8.5.2.6	Run Memory Monitor	8-28
8.5.2.7	Run Process Monitor	8-29

8-2 Monitoring Processes with CRM

System managers and end-users can monitor the system processes of their CLIPPER workstation or server with the CLIX Resource Monitor (CRM). Programmers may also use CRM during program development to analyze program behavior. CRM allows you to monitor either the system as a whole (through system monitors) or individual processes (through process monitors).

The system monitors collect information about various system functions such as I/O activity, and file, memory, and CPU use. The process monitors allow you to profile a process and show its paging, I/O, system call, and instruction execution.

CRM provides either FMLI (alphanumeric) displays based on the Curses facilities or graphics displays based on Environ V facilities. The following sections will discuss how to create each type of display. The chart below lists each type of monitor provided by CRM and illustrates which display method(s) can be used by each monitor.

CRM Monitors	Graphics-based	Curses-based
<i>System Monitors</i>		
Monitor Parameters (monparam)	—	X
Top System (topsys)	X	—
Top Memory (topmem)	X	X
Top Fault (topfault)	—	X
Top I/O (topio)	—	X
Top CPU (topcpu)	X	X
Show Open Files (showfiles)	—	—
Show Memory Usage (showmemory)	—	—
<i>Process Monitors</i>		
Profiler (watcher)	—	X
Memory (monregion)	X	X
Process (monproc)	X	X

CRM is a baseline supplemental product delivered free with the workstation/server standard system software. However, it might not be installed on your workstation/server. If necessary, refer to the *CLIPPER Software Delivery Guide* for instructions on loading baseline application products to a workstation or server.

8.1 Installing CRM

When loading the CRM product with `newprod`, you will receive the following prompts:

```
Do you want to have CRM added to your def__proc file (y/n)? [n]:
Do you want to have CRM added to your pulldown menu (y/n)? [n]:
```

If you choose to have CRM added to your `def__proc` file, a collapsed CRM window will be automatically created (in the form of an icon) every time you boot your workstation. This option does not apply to InterServe processors since windows are graphics-based.

If you choose to have CRM added to your workstation pull-down menu, the CRM option will be automatically added to your menu as an option. Again, this option does not apply to InterServe processors.

Once you have installed CRM on your workstation or server, you must be located in the `/usr/ip32/crm` directory (unless you installed the CRM product in another file system such as `/usr2`) on your workstation/server to execute the CRM interface menus or any of the individual monitors from the command line. You may want to change to this directory before you get started:

```
$ cd /usr/ip32/crm
```

Or, if you will be executing the CRM interface or the individual monitors regularly, you may want to link the executables for each process to `/usr/bin` so they can be executed from any directory in the system. If you loaded the CRM product in another file system (such as `/usr2` or `/usr3`), you must use the `-s` (symbolic) option with the `ln` command to link across file systems.

To link `crm` to `/usr/bin`, use the following command line:

```
$ ln /usr/ip32/crm/crm.sh /usr/bin
```

To link `crm` to `/usr/bin` if `crm` is loaded in an alternate file system (`/usr2` in this example), use the following command line:

```
$ ln -s /usr2/ip32/crm/crm.sh /usr/bin
```

To link any of the individual monitors to `/usr/bin`, use the following command line:

```
$ ln /usr/ip32/crm/<monitor_executable> /usr/bin
```

For example, to link the Top Fault (`topfault`) monitor to `/usr/bin` you would use the following command line:

```
$ ln /usr/ip32/crm/topfault /usr/bin
```

8.2 Invoking the Curses-Based FMLI Interface for CRM

Use one of the following procedures, depending on whether you are invoking CRM from a (local) graphics monitor or on an alphanumeric terminal (for an InterServe processor or a remote workstation).

8.2.1 Graphics Monitors

To invoke the CRM curses-based FMLI interface on a local (not remote) graphics monitor, use one of the following procedures:

- Key in the following from the default `/usr/lp32/crm` directory (or from any directory if you linked to `/usr/bin`):

```
$ crm.sh
```

OR

- Select CRM from the workstation pull-down menu *if* you opted to have it added to the menu when you installed CRM (or if you added it to the menu since then).

OR

- Select the CRM icon *if* you opted to have CRM added to the `def_proc` file. Processes listed in the `def_proc` file execute automatically when the system is booted.

If you have a password on your root account, you will receive a small menu in which you must enter your root (superuser) password when you execute by either the first or second method. The window will disappear as soon as the correct root password is entered.

8.2.2 Alphanumeric Terminals

To enter the CRM curses-based interface on an alphanumeric terminal (such as an InterServe processor or a remote workstation), key in the following:

```
$ crm.server
```

8.3 Using the FMLI Interface

A window with the Main CRM menu appears when you invoke CRM. The Main CRM menu provides the following selections:

- > Instructions
- > System Monitors
- > Process Monitors
- > Exit

This menu allows you to enter the System Monitor and Process Monitor menus, access online instructions for using CRM, or exit CRM. Use your arrow keys to scroll through these choices and press <Return> to execute.

You can select the > **Instructions** option to display information on using the CRM menus, forms, and labels (at the bottom of the CRM window). The information in the online instructions is summarized here:

- **Functions Keys and Labels.** Eight highlighted "labels" at the bottom of each CRM screen correspond to the top two rows on the keyboard's numeric keypad. The keypad keys perform the function shown on the corresponding label.

Ten possible labels can appear on CRM screens; however, not all labels display in every area of CRM. Labels that share the same function key never display as options at the same time. The ten possible labels, their corresponding keypad key, and their functions are as follows:

CANCEL (PF1) dismisses the present screen and returns to the previous menu.

PREVPAGE (PF2) displays the previous page of text.

NEXTPAGE (PF3) displays the next page of text.

PREV-FRM (PF4) moves to the previous form.

NEXT-FRM ("7" on keypad) moves to the next form.

HELP ("8" on keypad) displays a help screen that defines the purpose and available options of a menu or form.

CMD-MENU ("9" on keypad) displays a menu of commands that allow you to perform operations such as redisplaying the screen, temporarily escaping to the UNIX environment, and exiting the CRM program.

CONTENTS (“-” on keypad) displays a menu (table of contents) that allows you to obtain information about the following topics related to using menus and forms:

- Screen Layout
- Navigating within Menus
- Navigating within Forms
- Navigating within Help Frames
- Editing Form Key-in Fields
- The Command Menu (CMD-MENU)

CHOICES (PF2 on keypad) displays available values that you can enter in the field the cursor is on. This option is not presently implemented.

SAVE (PF3 on keypad) saves a form in which you have selected options for running a monitor. Even if you plan to accept the default values for all fields in a form, you *must* select the Save button before you will be able to execute the related monitor.

- **Menus** list operations (such as executing a monitor) that you can perform within CRM. You saw the first menu when the CRM window appeared. To select an item in a CRM menu, use the arrow keys on your keyboard to scroll through and highlight the item. Press <Return> to execute the highlighted choice. Other ways to move around in CRM menus are described online in “Navigating within Menus.” You can display this section by selecting CONTENTS (pressing the “-” on the keypad).

The Cancel, Help, and Cmd-menu functions are available to use with menus. The labels for these functions will appear at the bottom of the window while a menu displays.

- **Forms** provide key-in fields that allow you to enter information needed to run a monitor. When a form initially displays, a blinking cursor will appear in the first key-in field. Enter the required information and press <Return>. The cursor will move to the next field. After you have entered all required information, press the SAVE key (PF3) to execute. CANCEL dismisses the form without recording any changes.

The Cancel, Save, Help, and Cmd-menu functions are available to use with forms. The labels for these functions will appear at the bottom of the window while a form displays.

8.4 System Monitors

System monitors provide information about the system in areas such as I/O activity, and file, memory, and CPU use.

CRM provides the following system monitors:

- Monitor Parameters (monparam)
- Top Fault Monitor (topfault)
- Top Memory Monitor (topmem)
- Top CPU (topcpu)
- Top I/O (topio)
- Top System (topsys)
- Show Open Files (showfiles)
- Show Memory Usage (showmemory)

To exit a graphics-based system monitor, select the delete window icon for that monitor. To exit a curses-based system monitor, press Q, X, or <Ctrl-C>.

8.4.1 Running System Monitors from the Command Line

You can execute each of these monitors from the CRM menus or directly from the command line. To execute a system monitor from the command line, simply key in the executable name (listed in parentheses after each monitor listed above) at the prompt. While running any monitor, you can display a Help screen by keying in ?. A brief explanation of the display fields in the monitor will appear.

Command-Line Syntax

For Top System (topsys), Top Fault (topfault), and Top I/O (topio), use the following syntax:

```
$ <monitor_name> [-I interval] [-i input_file] [-o output_file]
```

For Top Memory (topmem) and Top CPU (topcpu), use the following syntax:

```
$ <monitor_name> [-I interval] [-i input_file] [-o output_file]
```

OR

```
$ <monitor_name> [-I interval] [-i input_file] [-w]
```

You can use the following options in the command line:

-I (Interval); Allows you to specify (in seconds) how frequently you want the monitor to gather and display information. Enter a positive number for this option. The interval defaults to two seconds. For example, you would use the following command to set the sample interval to five seconds:

```
$ topmem -I 5
```

-i (Input File); Allows you to enter the pathname of a file in which you previously recorded a monitoring session. The monitor will exit automatically when it finishes running the recorded session. To record a monitoring session in an output file, use the **-o** option described next.

For example, enter this command line to input previously recorded data from a file called **faultfile**:

```
$ topfault -i faultfile
```

To specify standard in (STDIN) as the input file, you would simply put a dash (-) after the **-i** option.

-o (Output File); Allows you to enter the pathname of a file in which you want to record a monitoring session. For example, you would key in the following to record a **topfault** monitoring session in a file called **faultfile**:

```
$ topfault -o faultfile
```

To specify standard out (STDOUT) as the output file you would place a dash (-) after the **-o** option.

The recording will begin when you enter a monitor and end when you exit that monitor (by pressing <Ctrl-C>.) If you execute the monitor as a background process, the only way to exit the monitor is to exit (kill) the process. The output file created can be read as an input file with the **-i** option.

This option causes raw data to be sent to a file that you cannot read. Only a program that can interpret the data can read the file. Presently, only CRM monitors can read and interpret the raw output data that is generated.

8.4.2 Running Graphics-Based System Monitors

Each system monitor except for Topsy will display in a curses-based format by default. Topsy displays only in a graphics-based format. Topmem and Topcpu will display in both formats.

To display Topmem and Topcpu in a graphics-based format, key in the executable name of the monitor with the `-w` (window) option. For example, to run the Topcpu monitor in graphics format, you would key in the following from the default `/usr/lp32/crm` directory:

```
$ topcpu -w
```

Presently, graphics-based system monitors can be invoked only through the command line.

Since the graphics-based monitors run in an Environ V window, you can manipulate the monitor windows with Environ V window icons. To access the icons, press any mouse button when your cursor is in the monitor window. The rest of this section describes the functions provided by the CRM icons.

Moving and Expanding the Monitor Window

When the monitors first appear on your screen, they are in small boxes with room to show only a few of the main processes being monitored. To display an expanded list of processes being monitored, you can stretch and broaden the window with the standard modify icon.

Obtaining Help

To receive additional information about a monitor, select the `?` icon from the window icon box. A Help window will appear.

Changing the Colors of your Bar Graphs

The colors on the monitor bar graphs are selected from color slots in the currently loaded VLT (Video Look-up Table). If you want to change the colors of your bar graphs, select the color palette icon from the window icon box. A Colors menu will appear.

The foreground color displays when the menu first appears. If you want to change the foreground color, select a new color from the palette. The change will appear immediately in the Top Sys window. To move to the next selection (background color), select the roll-through arrows icon (or the text next to it).

Continue scrolling through and changing colors until you have the color combination that you want. Exit and save the changes by selecting the delete icon in the Colors window. The colors you select are saved for this monitoring session only.

Saving the Monitor Location and Size

If you have changed the size and location of the monitor window on your screen, you may want the monitor to appear with the same values the next time you execute. The Save icon allows you to save the present size and position of the monitor display and the monitor icon (the collapsed window).

Changing to Bar Graph Only

To delete the list of color labels and display only the bar graph, select the white and grey block icon. Selecting this icon again will restore the header. (This option is available for Topmem and Topcpu only).

8.4.3 Using the System Monitors from the Curses-Based FMLI Menus

If you are not in the CRM interface window, key in `crm.sh` on the command line from the `/usr/lp32/crm` directory to create a CRM window. To enter the System Monitor menu, select **> System Monitors** from the main CRM menu. The main System Monitors menu appears with the following selections:

- > Change Defaults
- > Monitor Parameters
- > Top Fault Monitor
- > Top Memory Monitor
- > Top CPU Monitor
- > Top I/O Monitor
- > Top Sys Monitor
- > Show Open Files
- > Show Memory Usage

8.4.3.1 Change Defaults Form

The Change Defaults form allows you to change the default options of system monitors. For instance, you can reset the frequency with which the monitor polls the system for status information, record a monitoring session, or replay a previously recorded session. After filling out the Change Defaults form, press the Save key (PF3 on the keypad) to save your changes.

Note:

Use the Backspace key to delete characters in form fields.

The Change Defaults Form has the following entry fields:

Sample Interval:
Input File [Optional]:
Output File [Optional]:

Separate Windows [Y/N]: N
Learn Mode [Y/N]: N

An explanation of these fields follows:

Sample Interval allows you to define how frequently (in seconds) you want a monitor to gather information and update the monitor fields. Enter a positive number in this field. The default setting is two seconds.

Input File allows you to enter the pathname of a file in which you previously recorded a monitoring session. To record a monitoring session, go to the next field, Output File.

Output File allows you to enter the pathname of a file in which you want to record a monitoring session. The recording will begin when you enter a monitor and end when you exit that monitor. (A very large amount of data may be produced.)

Separate Windows [Y/N] allows you to execute monitors in windows separate from the CRM interface when it is set to "Y." This option enables you to run several monitors at once (in separate windows). The default setting is "N."

Learn Mode [Y/N] displays the command and options used to execute a monitor. This is a useful tool for learning the direct pathname and options (that you defined in your Change Default Option menu) used to invoke specific monitors. After the system displays the command, you will be prompted to press <Return> to execute the monitor. The default setting is "N."

8.4.3.2 Monitor Parameters (monparam)

Monitor Parameters (monparam) monitors the parameters of a running system. For example, Monitor Parameters can be used to show if you are running out of a resource. This monitor is a complement for the Configurable UNIX utility. (Configurable UNIX allows you to change system parameters.)

The following describes each Monitor Parameters field:

Sample Time specifies how frequently (in seconds) the monitor gathers and displays information on running processes. You can change this time interval by pressing the up arrow key (to increment) and the down arrow key (to decrement).

Name displays the name of the parameter being monitored.

Current displays the current value of the parameter being monitored.

Max displays the maximum value of the parameter since the system was booted last.

Configured displays the value specified for the parameter under Configurable UNIX.

Highlight specifies that the parameter line will be highlighted when the maximum value is 90 percent or more of the configured value.

8.4.3.3 Top Fault Monitor (topfault)

Top Fault (topfault) monitors the page faults being encountered by each process running on the system. Top Fault displays in a curses-based format only.

A brief explanation of the Top Fault fields follows. You can access a similar list online by keying in a ? while the monitor is running.

Sample time displays how frequently (in seconds) the monitor gathers and displays information. The default setting is two seconds. You can reset the value for sample time on the Change Defaults Options page, or you can change it interactively while running the monitor. Press the up or down arrow keys to increase or decrease the value.

Max displayed determines the maximum number of faulting processes that will display. You can also reset this value interactively with the right and left arrow keys. For example, if you are interested in seeing only the top three faulting processes, you could set this value to 3 instead of letting it default to 20.

vmfault displays virtual faults. The **vmfault** value is the sum of the four following values defined by UNIX. Remember that out of the four following types of faults, only swap and file faults go to the disk; demand and cache faults are satisfied in memory.

demand - Demand zero and demand fill pages

swap - Fault satisfied by swap

cache - Fault satisfied in the cache

file - Fault satisfied from a file

vmprfault displays protection faults. The **vmprfault** value is the sum of the two following values:

cop_wrt - copy-on-write. If two processes are sharing a copy-on-write page in memory, the page must be copied when one process needs to write to the page.

steal - If a page is marked copy-on-write but only one process is accessing it, it does not need to be copied. Instead, the protections are changed on the page so that one process can write to it.

vmfreedpgs displays the number of pages that are freed on the system during the last sample interval

vmunmodsw displays the number of unmodified pages in swap (as determined by *getpages*) during the sample time period.

vmunmodfl displays the number of unmodified pages in all files (as determined by *getpages*) during the sample time period.

vmswapin displays the number of pages swapped into memory during the sample time period.

vmswapout displays the number of pages swapped out of memory during the sample time period.

8.4.3.4 Top Memory Monitor (topmem)

Top Memory monitors the amounts of physical and virtual memory that system processes are using. The following fields appear in the Top Memory monitor. (A similar list is available online when you press ?):

freepages displays the average number of pages that were free (available) during the last sample interval.

proc_phys displays the sum of all weighted **_physical_size** values. The resulting sum indicates the total physical memory used by all processes.

freeswap displays the amount of space available on the swap device.

physical size displays the total amount of physical memory (valid pages) being used by the indicated process.

virtual size displays the size of the virtual address space being used by the indicated process. This value indicates the amount of swap space being allocated to processes.

weighted physical size displays the sum of valid pages used by a process. This sum is modified by the number of processes that share it. When several processes can share memory pages, fewer pages will need to be allocated for the later processes since they will share some of the pages that have already been allocated by earlier processes. This value indicates the amount of physical memory actually being used.

For example, if three **vterm** processes were running, the first process executed would be allocated the memory pages needed to run. The second and third **vterm** processes would not require as many memory pages because they could share some of the pages allocated to the original process. Therefore, the weighted physical size of each process will vary depending on the number of pages already allocated to another process that the processes can share.

8.4.3.5 Top CPU Monitor (topcpu)

Top CPU monitors the amount of CPU time being used at a particular moment in each of the following modes: user, kernel (system), wait I/O, swap I/O, phys I/O (physical I/O), and sxbrk (time spent allocating memory for a new job).

The following fields appear in the Top CPU monitor for the processes that were the top CPU users for the sample period. (A similar list is available online when you press ?):

% Used displays the amount of overall CPU time being used by a process.

% User/System used displays a type of bar graph composed of U's (user) and S's (system) to illustrate how much CPU time needed for a process is required by the user (the process itself) and how much is required by the system. Each U or S represents about 2 percent.

8.4.3.6 Top I/O Monitor (topio)

Top I/O monitors the I/O activity on the system and displays which processes are performing the activity.

The following fields appear in the Top CPU monitor. (A similar list is available online when you press ?):

b_read and **b_wrt** display the number of reads (**b_read**) and writes (**b_wrt**) to the block-oriented device (the disk). The "b" represents "block."

l_read and **l_wrt** display the number of data accesses made (by a program) to the system buffer cache.

cache displays the percent of I/O satisfied by the buffer cache (rather than the block-oriented device, or disk). This value is the difference between the total "b" reads and writes and the total "l" reads and writes.

phread and **ph_wrt** display the number of physical reads and writes to the raw disk.

sysrd and **syswrt** display the number of system calls made to the read and write routines in the system.

rdch and **wrtch** display the total number of bytes (characters) transferred by all read and write calls from a program, regardless of where the data came from (cache, disk, or memory).

device lists the SCSI devices involved in I/O on your system.

ops displays the number of I/O operations that occurred on the corresponding SCSI bus.

busy indicates the percentage of time that the SCSI device was busy with I/O operations (versus the amount of idle time).

bcnt displays a count of disk blocks that were transferred.

avque displays the average number of times that I/O had to wait because the SCSI device was busy servicing other I/O requests.

currque displays the current I/O queue depth (how many I/O requests waiting in the queue to be serviced).

ioch displays the number of characters transferred by the corresponding process.

8.4.3.7 Top System Monitor (topsys)

Top System monitors the activity of the entire system. Top System simultaneously displays activities that the other four system monitors (Top Memory, Top CPU, Top I/O, and Top Fault) show individually to give a complete overview of system activity. Top System is a graphics-based monitor in which the percentage of system resources being used by each process is represented by bars of contrasting colors.

Top System cannot presently display in a curses-based format. Although the alphanumeric console of InterServe processors will not display graphics, the Top System monitor can run on a local server and display on a remote workstation.

To run Top System on a graphics monitor from a remote server you can use a command with the following syntax. This syntax designates <node>.<user>.<pass> as the nodename (or address) and username.password of the graphics workstation where the monitor will display:

```
$ topsys -o - | rpipe <node>.<user>.<pass> topsys -i -
```

8.4.3.8 Show Open Files (showfiles)

This monitor lists all processes on the system, along with a list of open files for each process. Device names and inode numbers are provided for each open file.

Once you have an open file's inode number, you can use the `ncheck` command to generate the pathnames from the inode numbers. Refer to the System V Online manuals for more information on `ncheck(1M)`.

8.4.3.9 Show Memory Usage (showmemory)

This system monitor lists all processes on the system, including attached memory regions associated with each process.

The following information is given for each displayed region:

Region Type can be one of the following:

TEXT: Main executable code
DATA: Main data region
STACK: Process stack
SHMEM: Shared memory
DMM: Double mapped memory
LIBTXT: Shared library code
LIBDAT: Shared library data

Region Number is the CLIX internal identification number of the region. If the region number displays in more than one process, the region is being shared among those processes.

Virtual Size is the amount of virtual memory allocated to the region. The virtual size of the region is also allocated from the available swap space.

Physical Size is the amount of real memory currently being used by the region.

Shared is the number of processes currently attached to the region. If no number appears in this column, the region is being used by only one process.

Percent Memory is the amount of physical memory allocated to the region. Percent memory is weighted by the number of processes sharing the region. For example, if the total amount of physical memory is 10 MB and two processes are sharing a 1 MB region, the percent memory for each process would be five percent.

Cumulative is the cumulative percentage of physical memory used by the regions.

After the list of processes displays, a system summary displays. The summary provides the following information:

Total Physical Memory On System is the amount of real memory the system has, in megabytes.

Used By Processes is the final cumulative total of physical memory being used by the process regions.

Process Overhead is the amount of memory used by the system to keep page tables and user blocks.

Unattached Regions is the amount of physical memory being used by regions with no attached processes. Unattached regions occur when a program has the *sticky bit* set in its mode. For information on the sticky bit, refer to the System V Online manual page for `chmod(2)`.

Available Memory is the amount of physical memory available for processes to use.

Initial Clix Size is the amount of physical memory used by the CLIX operating system when the system boots.

Allocated By Clix is the amount of physical memory allocated when the CLIX operating system is running. For example, when a driver is loaded, its text and data section occupies a section of physical memory.

8.5 Process Monitors

The CRM process monitors provide the capability to profile a process and show its paging, I/O, system call, and instruction execution. CRM provides the following three process monitors:

- Profiler (watcher)
- Memory (monregion)
- Process (monproc)

All CRM process monitors display in curses-based format by default. However, you can also display the Memory and Process monitors in a graphics-based format by selecting an option from CRM menus (to be discussed shortly) or by using the `-w` option on the command line.

To exit each of the Process Monitors, use the commands shown in the following chart:

Monitor	Graphics-based	Curses-based
Profile (watcher)	—	<Ctrl-C>
Memory (monregion)	Delete icon	Q, X, <Ctrl-C>
Process (monproc)	Delete icon	O, X, <Ctrl-C>

8.5.1 Running Process Monitors from the Command Line

You can enter each process monitor through the CRM menus or directly from the command line. If you execute from the command line, you must supply these options for the command to run:

- All process monitors require an input option to specify the process you want to monitor as shown in this example:

```
$ monregion -n <program_name>
```

- The Profiler (watcher) also requires that you enable page faults (`-f`) and/or system calls (`-s`). For example, to view all faults and all system calls for a process you would use the following command line:

```
$ watcher -f ALL -s ALL -n <program_name>
```

Command Syntax

The command-line syntax for each process monitor is as follows:

Memory monitor (monregion):

```
$ monregion [output_option] <input_option>
```

Process monitor (monproc):

```
$ monproc [output_option] <input_option>
```

Profiler (watcher):

```
$ watcher <event_options> [output_options] <input_options>
```

Command-Line Options

The command-line options available for the process monitors are listed and described in this subsection. Remember, an input option *must* be used for each monitor, and the `-f` or `-s` option must also be used for the Profiler. You can also select these options through the CRM *forms* when using the curses-based FMLI interface.

To see a similar online list of options for a monitor, key in the executable name of the monitor and then a `-?`. For example, to see options for the Memory Monitor, key in the following from the default `/usr/lp32/crm` directory:

```
$ monregion -?
```

Output options for the process monitors are as follows:

`-o <raw_output_file>` Allows you to direct raw output data to a specified file. When the `-o` option is not specified, the data goes to the standard output device by default.

For example, key in the following to send the output of a monregion monitor to a file called `rawfile1`, where `program1` is the process being monitored:

```
$ monregion -n program1 -o rawfile1
```

`-o -` Allows you to send the raw output data of a monitor directly to a program that can interpret it. Presently, only CRM monitors can read and interpret the raw output data generated. This option was designed to pipe the output of one CRM monitor to another CRM monitor across a network.

For example, you would use the following command line to pipe the output of a local monproc monitor to another monproc monitor on node3, where program2 is the process being monitored:

```
$ monproc -n program2 -o - | rpipe
node3.joe.smith monproc -i - -w
```

The `-i -` option at the end of the example directs the input to standard in (STDIN). The `-w` option specifies that a graphics-based monitor will be created in a window. Therefore, this example would not work with the Profiler monitor since it cannot display in a graphics-based format.

For example, you would use the following command line to pipe the output of a Profiler monitor:

```
$ watcher -n program2 -o - | rpipe node3.joe.smith
vterm -x watcher -i -
```

-w

Allows you to execute this monitor in a graphics-based format (in a window). This option does not apply to alphanumeric terminals.

For example, you would use the following command line to execute monregion in a graphics-based format where program3 is the process being monitored:

```
$ monregion -n program3 -w
```

Input options for the process monitors are as follows:

-p <pid>

Allows you to specify the ID number of the process you want to monitor. You can key in `ps -e` at the system prompt to determine the PID of a process already running.

For example, you would use the following command line to monitor the memory use of process 151:

```
$ monregion -p 151
```

-i <raw_input_file> Allows you to display a previously recorded monitoring session that was saved in a file. The filename will be your raw input file.

For example, you would key in the following to replay a Profiler (watcher) monitoring session recorded in a file called *rawfile1*:

```
$ watcher -i rawfile1
```

-i - Allows you to pipe the output of one monitor to the input of another. Input is from STDIN standard input. See the example for the **-o -** option.

-n <process_name> Allows you to specify the name of the process you want to monitor. You can key in **ps -e** at the system prompt to determine the name of a process already running.

For example, you would key in the following to monitor the screen manager (smgr) process.

```
$ monproc -n smgr
```

If the process that you intend to monitor has not been executed yet, you will not see any response from the monitor until you start the process.

-e <command_to_exec> [command args...] Allows you to run, provide arguments for, and monitor a program.

For example, you would key in the following to monitor all system calls from the **ls** process when you are using the **-l** argument:

```
$ watcher -s ALL -e ls -l
```

All options on the command line following the **-e** are used to invoke the child process.

The **-I <interval>** option is also available for process monitors. This option is outlined as follows:

-I <interval> Allows you to define how frequently (in seconds) you want a monitor to gather information and update the monitor fields. (This option applies to the Process monitor (monproc) only.)

For example, to set the sample interval to 5 seconds:

```
$ monproc -I 5
```

In addition to supplying input options (and other desired options), you must also supply the following Event Options with the Profiler (watcher):

-f <fault_options> Enables system faults to be monitored. The ALL option (all capitals required) will monitor all system faults. Other fault options that can be defined are demand, swap, cache, file, cw, and steal.

For example, you would key in the following command line to monitor the demand and swap faults for program3:

```
$ watcher -f demand swap -n program3
```

-s <system_call_types> Enables system calls to be monitored. The ALL option (all capitals required) enables all system calls to be monitored. Other system calls are program-specific and therefore user-definable.

For example, you would key in the following command line to monitor all system calls made by program4:

```
$ monproc -s ALL -n program4
```

8.5.2 Using the Process Monitors Interactively from Curses-Based FMLI Menus

To enter the main process monitors menu, select **> Process Monitors** from the main CRM menu. The main process monitors menu appears with the following selections:

- > Select Process to Monitor
- > Select Profiler Options
- > Run Profiler
- > Run Memory Monitor
- > Run Process Monitor

8.5.2.1 Select Process to Monitor

Before you can monitor a process, you must specify the process you will monitor. You can key in **ps -e** at the system prompt to determine the name or PID (process ID) of processes running on the system.

Then, to select the process that you will monitor, choose the "Select Process to Monitor" option from the main process monitor menu. A Change Default options form appears with the following entry fields:

Name of program to monitor:
Or PID:
Or program to execute:
Or pre-recorded file:
Output File [Optional]:
Separate Windows: N
Sample Interval:
Learn Mode (Y/N): N

You must enter data in one of the first four fields to specify the process you want to run, but completing the rest of the form is optional. A brief description of each field follows:

Name of program to monitor allows you to enter the name of the process you want to monitor.

PID allows you to enter the PID of an active process that you want to monitor.

Program to Execute allows you to enter the pathname (and options) of a program that you want to execute and monitor simultaneously.

Pre-recorded File allows you to enter the filename (pathname) of a previously recorded monitoring session.

Output File [Optional] allows you to enter the filename (pathname) of a file where the monitoring session will be recorded.

Separate Windows allows you to execute a monitor in graphics-based format when it is set to "Y." This field applies only to the Memory and Process monitors; the Profiler does not run in graphics-based format. This option should be set to "N" when you attempt to run Profiler; otherwise, it will not execute.

Sample Interval allows you to define how frequently (in seconds) you want a monitor to gather information and update the monitor fields. Enter a positive number in this field. This field applies only to the Process monitor (monproc).

Learn mode (Y/N) displays the command and options used to execute a monitor. This is a useful tool for learning the direct pathname and options (defined in the Change Default Option menu) used to create specific monitors. After the system displays the command, it prompts you to press <Return> to execute the monitor.

Note:

If you forget to complete the Change Default form before you execute one of the monitors, the form will automatically display so you can fill it out before the monitor begins.

8.5.2.2 Select Profiler Options

The Profiler monitors the page faults and system calls of a specified process. Before you can run the Profiler monitor, you must enable page faults and system calls. Choose "Select Profiler Options" from the main process monitor menu. A menu with the following selections appears:

- > Enable Page Faults
- > Enable System Calls

Notes:

- Even if you want to accept the default values for each option listed on the Page Faults and System Calls forms, you must still enter each form and press the Save key (PF3 on the keypad) to accept the default settings.
- If you forget to complete the Profiler Options forms before you execute the Profiler, the forms will automatically display for you to fill out before the monitor begins.
- Remember to set the Separate Windows option on the Change Defaults form to "N"; otherwise, this monitor will not execute.

8.5.2.3 Enable Page Faults

Select the > **Enable Page Faults** option and a form with the following entry fields appears:

Demand Zero [Y/N] Y
Swap [Y/N] Y
Cache [Y/N] Y
File [Y/N] Y
Copy on Write [Y/N] Y
Steal Y

Starting Virtual Address [Hex] 0
Ending Virtual Address [Hex] FFFFFFFF

Maximum Samples [Dec] 100000

See 8.4.3.2, "Top Fault Monitor," for descriptions of the Demand Zero, Swap, Cache, File, Copy on Write, and Steal fields. The last two fields on the form are described here:

Starting Virtual Address [Hex] and Ending Virtual Address [Hex] allows you to monitor faults occurring only at certain addresses in the process. These fields allow you to define the section of the process in which faults will be watched.

Maximum Samples allows you to define the number of samples for the monitor to collect. This definition may prevent the monitor from running indefinitely.

8.5.2.4 Enable System Calls

Select the > **Enable System Calls** option and a form with the following fields appears:

All System Calls [Y/N] Y
I/O [Y/N] N
Summary Only [Y/N] N

A description of these fields follows:

All System Calls [Y/N] directs the Profiler to report all system calls. The default setting is "Y."

I/O [Y/N] directs the Profiler to ignore all system calls except for I/O calls. The default setting is "N."

Summary Only [Y/N] directs monitor to print only a summary of system calls when the monitoring interval is over instead of listing all system calls as they are encountered. The default setting is "N."

8.5.2.5 Run Profiler

To run the Profiler monitor, select > **Run Profiler** option from the menu. If you did not specify that you want to view *only* a summary of the system calls and faults, you may see a list of the system calls and faults as they are encountered by the specified process. If you chose to see only a summary, nothing will appear until the monitor finishes or you exit the monitor. (Press <Ctrl-C> twice.)

The following is a sample of system call and system fault fields that appear in Profiler:

```
System Fault: DEMAND
PC: 0x00004400
Virtual address: 0x0040157d

OPEN system call
PC: 0xff804e62
arg0:      647773 7361702F 6374652F /etc/passwd
arg1:      00000000 ...
completion status: 1
```

An explanation of these fields follows.

System Faults

System Fault indicates the occurrence of a page fault and the fault type (such as demand, swap, or cache).

PC (program counter) indicates the address of the program instruction that took the fault.

Virtual address indicates the address that was accessed to cause the fault.

System Calls

PC (program counter) indicates the address of the program instruction that issued the system call.

arg0, arg1 ... arg<n> indicates arguments of a system call. The argument values are shown in both hexadecimal and alphanumeric-equivalent formats. When a value has no alphanumeric equivalent, a dot (.) is printed.

completion status indicates the success or failure of a system call or provides data about the call. For instance, a WRITE system call would usually display a numerical value in this field to indicate the number of bytes that were written during the call.

System Call and System Fault Summaries

System call and fault summaries appear when the Profiler is exited. The system call summary lists the number of calls made by each type of call as well as the total number of system calls made during the monitoring session.

The system fault summary lists the number of faults encountered for each type of fault as well as the total number of faults encountered by the process during the monitoring session.

8.5.2.6 Run Memory Monitor

The Memory Monitor (monregion) monitors the memory regions used by a specified process. To run the Memory monitor, select the > Run Memory Monitor option from the main process monitors menu.

Here are a few sample lines from a memory monitoring session:

```
TEXT    SHARED  RDONLY      size:35    valid:23
00000000: ****    **** **&*    *****|
DATA    PRIVATE  RD/WRT      size:45    valid:16
00400000: ***** **L          *    *****|
00440000: **** *    *    *    *** *|
00480000: *    ***    ***** * *    *** *|
STACK  PRIVATE  RD/WRT  GROWDOWN  size:2    valid:1
bfffffff: * |
```

An explanation of these fields follows:

TEXT, DATA, STACK are the names of three memory regions monitored for any process.

SHARED/PRIVATE indicates whether the region can be shared with other processes or is local to this process.

RDONLY or **RD/WRT** indicates whether the process has write access to the region or the region is read-only.

GROWDOWN indicates that the stack region is monitored according to stack region structure. Since a stack region is structured to begin at high addresses and decrease to low addresses, the region is monitored from high to low. Pages added to a stack region are added at the lowest virtual address rather than the highest. Thus, when a stack region grows, it grows downward.

size indicates the number of virtual pages in the region.

valid indicates the number of physical pages mapped to virtual pages in the region.

Notice the rows of numbers, asterisks (*), ampersands (&), and other alphanumeric characters below the names of the memory regions (TEXT, DATA, and STACK). These characters provide information about the memory pages as follows:

The alphanumeric characters (such as 00000000 and 00400000) directly below the memory regions specify the starting address (in the region) for that line on the screen. Since the activity of the entire region will not display on a single line in the monitor, the monitor breaks the display into several parts.

The asterisks (*) represent a physical page of memory mapped to the region. Every blank space between the *'s represents a page of virtual memory without a physical page mapped to it.

The ampersand (&) indicates the memory page that the PC (program counter) points to.

The L indicates a physical page of memory that is *locked* to a process. A locked page cannot be taken from that process. For example, if the I/O system will need a page for an I/O request, the system will lock a page in memory until the process is finished with that page.

The vertical bar (|) represents the end of the section of memory region shown on that line.

8.5.2.7 Run Process Monitor

The Process Monitor (monproc) monitors CPU use, status, priority, hard and soft fault rates, and current PC (program counter) for a process. To run the Process monitor, select the > **Run Process Monitor** option from the process monitors menu.

Sample output from the Process monitor is as follows:

Process is xns_listen Thu May 25 11:14:53 1989

CPU user time 00 00:00:00.41 CPU system time 00 00:00:06.31
Status: STOP Priority: 14 Username: root
Hard fault rate 00/sec Soft fault rate 00/sec
Physical Memory 196 k Virtual Memory 532 k
Elapsed time: 00 00:52:16:00
PC: 0000efe6 /current
PC: 0000efe6 /last

An explanation of these fields follows:

CPU user and system time displays the amount of CPU time used by the process (user) and the system since the beginning of the monitoring session.

Status displays the process activity (such as sleep or stop) when the CPU looks at it.

Priority displays the priority the system assigns to the process being monitored.

Username displays the username running the process being monitored.

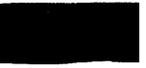
Hard and Soft Fault rates displays the number of hard and soft faults that occurred per second during the sample interval.

Physical and Virtual Memory displays the amount of physical and virtual memory the system assigns to the process being monitored.

Elapsed time displays how long the process has been running.

PC (program counter) displays the address where the program counter was located the last time the monitor polled it. If your program was compiled to include debugger symbols (that could be used with Intergraph's DBG), the monitor can read those symbols and provide more logical values in this field. For instance, the PC might display a more logical address such as "sub1 + 10," where sub1 is the name of a procedure in the program and 10 is the number of bytes offset into sub1.

You will notice two separate color bar graphs when you execute the graphics-based Process monitor. The first bar shows activity for the last sample period; the second bar shows activity for the last 10 sample periods.



Chapter 9: Backing Up and Restoring Files

This chapter contains the following sections:

9.1	Determining a Backup Schedule	9-3
9.2	Devising a Backup Plan According to System Configuration	9-4
9.2.1	Full Network Configuration	9-4
9.2.2	Standalone Configuration	9-5
9.3	Backup Commands and Utilities	9-6
9.3.1	The dd Command	9-10
9.3.2	The tar Command	9-13
9.3.3	The cpio/scpio Commands	9-18
9.3.4	The backup and restore Commands	9-26
9.3.5	The ansitape Utility	9-36
9.3.6	The vmsbackup Utility	9-42
9.3.7	The rtc Utility	9-44
9.3.8	The compress Utility	9-47
9.4	Additional Commands for Floppy Disks	9-51
9.4.1	The format Command	9-52
9.4.2	The <u>mkfs</u> Command	9-54
9.4.3	The cpflop Command	9-56
9.4.4	The to_flop Command	9-56
9.4.5	The fr_flop Command	9-56

You should keep accurate and complete records of hard disk files for the following reasons:

- If the system crashes or if the hard disk must be rebuilt, backup files prevent data loss.
- If you have disk space problems, you may need to store less frequently used files on storage media and load them only when needed.
- Backing up files regularly retains a history of the file system. For example, you may need a file version from a day, a week, or a month ago.
- Certain files or directory structures may require tighter security. For example, if you do not wish for users to access certain files, you can back up those files on a tape or removable disk and remove them from the system.

Backing up files is an involved process that requires careful planning. The system administrator should establish a specific schedule and method of backing up workstation/server files. This chapter provides procedures for backing up and restoring files, as well as establishing an appropriate schedule and method.

Procedures for the following tasks are provided:

- Determining an appropriate backup schedule
- Devising a backup plan according to system configuration

Numerous backup commands and utilities are available under the CLIX operating system. This chapter describes and offers suggestions for using each of the following commands. However, the system administrator must determine which backup command is most appropriate for backing up a particular system.

- dd
- tar
- cpio/scpio
- backup and restore
- ansitape (magnetic tape only)
- vmsbackup (magnetic tape only)
- rtc (magnetic tape only)
- compress (used to compress archive files)

The final section in this chapter describes commands used only for floppy disks. The following floppy commands are discussed:

- format
- mkfs
- mf
- uf
- cpflop
- to_flop
- fr_flop

9.1 Determining a Backup Schedule

You should define a schedule for backing up files so you can maintain complete and accurate records of hard disk files. The backup schedule for each system will vary according to the importance of backups. For example, a system that stores critical data should be backed up more frequently than other systems should be. The following is an example of a schedule that can be used to back up files daily, weekly, and monthly.

Assume that the ACE Engineering Company has three design engineers, each equipped with an InterPro 340 and the EMS design package. This company also has an InterServe 3005 with a Kennedy tape drive mounted on it. ACE engineers realize the importance of quality system backups, so they devised the following plan to periodically back up files from their 340s to the tape drive on the InterServe 3005:

- The daily backup is planned as follows:
 - Each engineer has four tapes for daily backups, one a day for Monday through Thursday. (Friday will be included in the weekly backup.)
 - Each engineer backs up his/her user account daily.
 - Each engineer backs up files at a different predetermined hour to allow tape drive sharing.

- The weekly backup is planned as follows:
 - Each engineer has three tapes for weekly backups, one a week for the first three weeks in the month. (The fourth week will be included in the monthly backup.)
 - Each engineer backs up his/her user account, the */etc* directory, the */usr/lp32/smgr* directory, and the */usr/lp32/vt200/catalog* directory, as the files in these directories contain user-configurable data that may change frequently.

- The monthly backup is planned as follows:
 - Each engineer has one or more tapes for monthly backups, depending on how many monthly backups are needed.
 - Each engineer backs up his */usr* file system and possibly some directories under the root (*/*) directory (such as */etc*).

9.2 Devising a Backup Plan According to System Configuration

The most appropriate backup plan for your system depends on the configuration. Most systems generally fit in one of two configurations:

- The configuration includes a full network with workstations, servers, and VAXes.
- The configuration includes a standalone workstation and a local tape drive.

This section provides an example for devising a backup plan according to your system configuration. It uses the `cpio` command to back up and restore files. For a more complete description of `cpio` and other backup commands and utilities, refer to section 9.3, "Backup Commands and Utilities."

9.2.1 Full Network Configuration

In a full network configuration, possibly including workstations, servers, and VAXes, the backup plan is complex but allows many options. The following is an example of a backup method that uses a tape drive in the network (on a workstation, server, or VAX) to back up workstation/server files.

In the previous subsection, ACE Engineering devised a schedule for backing up files. Now, the company must determine the backup plan most suitable for its system configuration. Since the tape drive is mounted in the InterServe 3005, the ACE engineers will use it as their backup device. They will use the `rtc` (remote tape controller) utility that Intergraph developed so that each workstation can use the tape drive as if it were attached locally.

It is 3 p.m. on Monday and time for John Doe to back up his user account (as he does everyday at this time). First, he uses the `rtc` utility to allocate the tape drive to his workstation. The syntax of `rtc` is as follows:

```
$ rtc -ad -s<nodename> -r<rewdev> -n<norewdev> -t<timeout> <device>
```

- a Allocates a remote device.
- d Deallocates a remote device.
- s Specifies the nodename or address of the host where the remote device resides.
- r Specifies the rewindable device file that references the tape on the remote node.
- n Specifies the nonrewindable device file that references the tape on the remote node.
- t Specifies the length of time (in minutes) that the allocated drive is idle before the connection is terminated. This argument is optional.
- device Specifies the `rtc` device controller specification.

John issues the following command to allocate the remote tape drive to node `ace_pn` for 30 minutes, specifying an automatic-rewind device file:

```
$ rtc -a -sace_pn -r/dev/rmt/mt6 -t30 /dev/rmt/rt0.ctl
```

The following message appears, confirming that the tape drive has been allocated to John's workstation:

```
allocate: /dev/rmt/rt0.ctl ace_pn /dev/rmt/mt6 5
```

Now, John can back up files as if he has a local tape drive. He uses the `cpio` command (described in detail in section 9.3.3) to back up his user directory (`/usr/john`) as follows. Notice that the device file specified in the `cpio` command does not require the `.ctl` extension, since `/dev/rmt/rt0.ctl` was specified when the drive was allocated.

```
$ find /usr/john -print | cpio -ov >/dev/rmt/rt0
```

Once John's user directory has been backed up, he can deallocate the tape drive as follows:

```
$ rtc -d /dev/rmt/rt0.ctl
```

9.2.2 Standalone Configuration

If a system is a single workstation/server with a dedicated tape drive, John will not need to allocate the tape drive because the device is not remote. Instead, he can use the `cpio` command to back up the files in his user directory. The device file in the following example is `/dev/rmt/mt6` to reference a local tape drive at SCSI ID 6.

```
$ find /usr/john -print | cpio -ov >/dev/rmt/mt6
```

9.3 Backup Commands and Utilities

Numerous backup commands and utilities exist under the CLIX operating system. Use the following chart to help determine which utility is best for backing up your system. Each of these commands are described in detail later in this section.

Command	Advantages	Disadvantages	Recommended Uses
dd	Fast	<p>You must know the original filename to recover a file from a backup</p> <p>You cannot write to multiple volumes.</p>	<p>Copying raw binary data</p> <p>Copying to and from device files</p>
tar	Simple	<p>If a file in the saveset becomes corrupted, the remaining files cannot be recovered.</p> <p>You cannot back up device files.</p> <p>You cannot write to multiple volumes. (However, for floppies, you can use <code>to_flop</code> and <code>fr_flop</code>.)</p>	<p>Backing up noncritical data</p> <p>Backing up individual files</p>

Command	Advantages	Disadvantages	Recommended Uses
cpio	<p>You can back up any file, including device files</p> <p>You can direct output to any device, pipe, or file on the system.</p> <p>You can write to multiple volumes.</p>	<p>No verify mode (see scpio)</p>	<p>Performing any level of backup including large-scale backups of critical data</p>
scpio	<p>Includes all advantages of cpio.</p> <p>Error-handling capabilities</p> <p>Asynchronous output and multi-buffering (faster than cpio)</p> <p>Verify mode to confirm integrity of input and archived files</p>		<p>Performing any level of backup including large-scale backups of critical data</p>

Command	Advantages	Disadvantages	Recommended Uses
backup	<p>Good user interface</p> <p>Fair error recovery mechanisms</p> <p>Can write to multiple volume save sets</p>	<p>Cannot back up individual files (only entire file systems)</p>	<p>Writing one or more volumes of tapes</p>
restore	<p>Good user interface</p> <p>Fair error recovery mechanisms</p> <p>Can read multiple volume save sets</p>	<p>Can only recover backup save sets</p>	<p>Recovering backup save set</p>
ansitape	<p>Can read or write tapes</p> <p>Can transfer data between CLIX and VMS</p>		<p>Reading/writing tapes between CLIX and VMS</p> <p>Useful for transferring design files to remote locations with no network link</p>
vmsbackup	<p>Can transfer data between CLIX and VMS</p>	<p>Does not write tapes (only reads them)</p>	<p>Reading VMS-generated backup format tapes</p> <p>Useful for transferring design files to remote locations with no network link</p>

Command	Advantages	Disadvantages	Recommended Uses
rtc	Can access a remote tape drive as though it were local		Reading/writing to tape on a remote machine

9.3.1 The dd Command

The most basic command for transferring data is **dd**. You can use this utility to move raw binary data or files to and from the storage device (hard disk, tape, or floppy disk).

Although you can use **dd** as a backup utility, it lacks a user interface for recovering data. For example, if you use **dd** to write a file to a tape, you must know the original filename, as this information is not maintained in a header.

The **dd** utility is useful for manipulating raw data transferred to a tape from another operating system or for imaging a disk drive by reading input from a raw partition. For example, the following command creates an image of the */usr* file system and sends it to the tape drive referenced by */dev/rmt/mt6*:

```
# dd if=/dev/rdisk/s0u0p7.3 of=/dev/rmt/mt6
```

The string "if=/dev/rdisk/s0u0p7.3" specifies that **dd** will accept input from the raw device located at SCSI ID 0, LUN 0, partition 7, modifier 3. (The input file is the */usr* file system.) The string "of=/dev/rmt/mt6" specifies that **dd** will send the output information to a tape device located at SCSI ID 6. (The output file is the rewindable tape device at SCSI ID 6.)

The following command restores the data backed up with the previous command:

```
# dd if=/dev/rmt/mt6 of=/dev/rdisk/s0u0p7.3
```

Although **dd** does not have the functionality of most other backup utilities, it is fairly fast. Using blocking factors increases performance. (The blocking factor signifies the amount of data buffered in one data block before it is written to the media.)

Caution:

Because data is treated as one large file, if any part of the data becomes corrupted, you probably will not be able to recover any of the backed up data.

Note:

To use **dd** with a cartridge tape, you must add the **conv=sync** option to the command line to preserve a 512 (or multiple) byte boundary as follows:

```
# dd if=/dev/rdisk/s0u0p7.3 of=/dev/rmt/mt6 conv=sync
```

Please refer to the **dd(1)** manual page from the System V Online Documentation product for more information on the **dd** command.

9.3.2 The tar Command

The tape file archiver utility (**tar**) is used to back up a file, directory, or entire file system. However, **tar** does not have any error handling capability. If a file in a **tar** save set becomes corrupted, all files following that file cannot be recovered with **tar**. In addition, **tar** will not back up device files. When attempting to back up files, **tar** skips over device files.

The **tar** command is best used as a secondary method for performing major backups or as a utility to back up noncritical information.

The format for the **tar** command is as follows:

```
$ tar [key] [option] [file]
```

tar is the executable command.
key(s) designates a specific action of the command.
option(s) is a command supplement.
filename(s) designates the files to transfer.

Note:

The **tar** command format does not require a dash (-) in front of the key.

The following command backs up a user account to a tape drive at SCSI ID 6:

```
# tar cvfb /dev/rmt/mt6 20 /usr/john
```

c specifies a create operation.
v specifies verbose mode.
f specifies that a device file (/dev/rmt/mt6) will be supplied.
b specifies that a blocking factor (20) will be supplied. (The blocking factor signifies the amount of data buffered into one block before the data is written to the media.)

To list the files saved on a tape, use the **t** (table) option as follows:

```
$ tar tvf /dev/rmt/mt6
```

The following command recovers the data backed up previously:

```
$ tar xvfb /dev/rmt/mt6 20
```

`tar` output defaults to floppy disk but can be directed to any device, pipe, or file on the system. For example, to back up files to floppy, omit the `f` option and do not specify a device file parameter.

Caution:

The `tar` command is not compatible with floppy disk file systems. The `tar` command will destroy a floppy disk file system by overwriting it.

The following command backs up all files with the `.dgn` extension in the user's current directory:

```
$ tar cv *.dgn
```

To list files saved on a floppy, use the `t` (table) option as follows:

```
$ tar tv
```

To recover all files in this `tar` saveset, use the `x` (extract) option as follows:

```
$ tar xv
```

To recover a single file in this `tar` saveset, use the `x` option and specify the filename to extract as follows:

```
$ tar xv house.dgn
```

The following example appends files to the same floppy disk. The `r` (read) option reads to the end of the tape and appends new files. If the `c` (create) option were used, `tar` would begin writing at the beginning of the tape and thus overwrite existing data.

```
$ tar rv *.sh
```

The following example updates the `tar` saveset. The `u` (update) option copies any file with a more recent date and any new files.

```
$ tar uv *.sh
```

Please refer to the `tar(1)` manual page in the System V Online Documentation product for more information on the `tar` command.

9.3.3 The cpio/scpio Commands

The **cpio** (copy file archives in and out) command can back up any file, including device files. Output from **cpio** can be directed to any device, pipe, or file on the system.

The **scpio** command (streaming **cpio**) is an enhanced version of the **cpio** command for streaming tape drives. It provides two major additions to the features of the **cpio** command: asynchronous input/output and multibuffering, which allows faster processing.

The **cpio** and **scpio** commands are both useful and robust backup utilities. Both utilities are capable of reading each other's savesets, and both utilities can write multiple volumes of tapes. This capability allows more flexibility because you do not need to know how much tape will be needed to store the data.

The input for these utilities must be a list of filenames. Thus, to write a saveset to either of these utilities, you must use a command such as **ls** or (more preferably) **find** and pipe the output to **cpio**. For example, if you wanted to back up the **/usr2** file system to a tape drive at SCSI ID 5, you could use the **scpio** command as follows:

```
# find /usr2 -print | scpio -ov -C10240 -z20 -0/dev/rmt/mt5
```

- find** searches the hard disk for the specified directory or file (in this example, **/usr2**).
- print** writes a list of filenames in the **/usr2** directory in the pipe (**|**), which **scpio** reads from.
- o** copies output from the input list specified (in this case through the **find** command).
- v** specifies that output will display in verbose mode.
- C** specifies the block size that will be used. The block size must be a multiple of 512, between 512 and 63,488. In this example, a block size of 10,240, or 512 x 20, was used.
- z** specifies the number of buffers of the block size that will be allocated. In this example, 20 buffers are allocated, each holding 10,240 bytes. The value specified by **z** must be an integer from 2 to 25. The **z** option enables **scpio** to stream data because a new buffer of data is ready to be flushed before the prior buffer has been written to the media. This option is not available for **cpio**.

- specifies the output device (in this example, */dev/rmt/mt5*). In addition, this option allows you to press <Return> when changing volumes, instead of supplying the device file again. This capability is possible because *scpio* received this information before beginning the backup.

To list the files saved on the tape, use the *scpio* command with the *t* (table) option as follows. The *i* option specifies that input will be received and the *I* option specifies where input will be read from (*/dev/rmt/mt5*).

```
# scpio -ivt -C10240 -z20 -I/dev/rmt/mt5
```

To recover the previously saved data, issue the following command:

```
# scpio -iv -C10240 -z20 -I/dev/rmt/mt5
```

Please refer to the *cpio(1)* and *scpio(1)* manual pages in the System V Online Documentation product for more information on the *cpio* and *scpio* commands.

9.3.3.1 Appending Files to ¼-inch Cartridge Tapes or 9-Track Tapes

In addition, you can use `cpio/scpio` to append files to the end of a cartridge or 9-track tape. The following procedure can be used on any ¼-inch cartridge or 9-track tape drive:

1. Write the original `cpio/scpio` saveset. This example writes the `/usr2` file system to a tape drive at SCSI ID 6:

```
# find /usr2 -print | scpio -ov -C63488 -z20 -0/dev/rmt/mt6
```

2. After the tape is rewound to the beginning, use the following command to skip forward to the end of the tape:

```
# mt -f /dev/rmt/mt6n fseot
```

`mt` passes commands to the tape drive.

`f` specifies that the `mt` command will use the `/dev/rmt/mt6n` tape drive.

`fseot` tells the drive to skip to the end of the recorded portion of the tape.

3. Use the `cpio/scpio` command to write the second saveset to the tape. Since you skipped to the end of the recorded media in step 2, you are essentially appending files to the end of the tape. The following example appends all files in the `/usr/john` directory to the tape:

```
# find /usr/john -print | scpio -ov -C63488 -z20 -0/dev/rmt/mt6
```

Note:

The following command is used to list files saved on tape. However, when you issue this command, only the first saveset is listed.

```
# scpio -it -I /dev/rmt/mt6n
```

To list the second saveset, reissue this command as follows:

```
# scpio -it -I /dev/rmt/mt6n
```

A message similar to the following displays:

```
Reached end of medium on input.  
Change to part 2 and press Return Key [q].  
  <press return> list rest
```

Issue this command as follows for a third time and the second saveset is listed:

```
# scpio -it -I /dev/rmt/mt6n
```

4. If you wish to append a third saveset to the tape, issue the `mt` command again as follows:

```
# mt -f /dev/rmt/mt6n fseot
```

5. Use the `cpio/scpio` command to write the third saveset to the tape. The following example appends all files in the `/usr/jane` directory to the tape:

```
# find /usr/jane -print | scpio -ov -C63488 -z20 -0/dev/rmt/mt6
```

9.3.3.2 Appending Files to 8 mm Helical Tapes

You can also use `cpio/scpio` to append files to the end of an 8 mm tape. The following example is one method for appending files to an 8 mm tape drive:

Note:

Any time you append files to a tape, make sure that you know the exact number of savesets currently on the tape.

1. The following command could be used to write the first saveset to an 8 mm tape:

```
# find /usr/fred -print | scpio -ov -C63488 -z20 -0/dev/rmt/mt6
```

2. List the first saveset by keying in the following command:

```
# scpio -it -I /dev/rmt/mt6n
```

3. Issue the following command to skip to the end of the recorded media:

```
# mt -f /dev/rmt/mt6n fsf 1
```

`mt` passes commands to the tape drive.

`f` specifies that the `mt` command will use the `/dev/rmt/mt6n` tape drive.

`fsf 1` tells the drive to skip forward one saveset.

4. Now that the tape has been forwarded to the end of the recorded media, use the `cpio/scpio` command (as shown in the following example) to write the next saveset to the tape. Since you skipped to the end of the last saveset, you are essentially appending files to the end of the tape.

```
# find /usr/john -print | scpio -ov -C63488 -z20 -0/dev/rmt/mt6
```

5. If you need to append another saveset, use the following command to list each saveset on the tape. (You will need to issue this command as many times as you have savesets on the tape.)

```
# scpio -it -I /dev/rmt/mt6n
```

6. Issue the `mt` command to skip to the end of the recorded media:

```
# mt -f /dev/rmt/mt6n fsf 1
```

7. Append the saveset to the tape by using a command similar to the following:

```
# find /usr/alice -print | scpio -ov -C63488 -z20 -0/dev/rmt/mt6
```

8. Repeat steps 5 through 7 to append any more savesets.

9.3.4 The backup and restore Commands

The **backup** and **restore** commands have been ported to CLIX from Berkeley UNIX. These commands are used in tandem; **backup** is used to write a tape and **restore** is used to recover a tape written with **backup**. These commands are used only for backing up and recovering entire file systems. These utilities have a helpful user interface and good error recovery mechanisms. In addition, **backup** and **restore** can write and read multiple-volume savesets.

The following example illustrates the use of **backup** and **restore**.

A system manager wishes to back up the */usr2* file system. The */usr2* file system resides on the disk drive located at SCSI ID 1, partition 7, modifier 4 (*/dev/dsk/slu0.7.4*) and it is 100,000 blocks in length. The tape drive is located at SCSI ID 5 and the system manager mounted a 2,400-ft. tape on the drive. He issues the following command to back up the file system:

```
# backup Ofunds /dev/rmt/mt5 6250 2350 20 /usr2
```

The **backup** command lists all options first, followed by the arguments.

- 0** specifies the backup level. The level must be an integer between 0 and 9. A level 0 backup will back up all files, while a level 9 backup will back up up the least number of files. The level option has no accompanying argument.
- f** specifies the device that the information will be written to (in this example, */dev/rmt/mt5*).
- u** causes **backup** to write information about the backup level, the file system being backed up, and the time the backup was performed to a tracking file called */etc/dumpdates*.
- n** notifies the operator when **backup** has encountered an error requiring attention. This option has no accompanying argument.
- d** specifies the density the media is being written at (in this example, 6250).

- s** specifies the size of the media, or the length of the tape (in this example, 2,350). Although the physical length of the tape is 2,400 ft., the beginning of tape (BOT) and end of tape (EOT) markers must be accounted for. The size specified is the length of tape that can actually be accessed. You should specify an amount less than the physical length so that backup will not attempt to write past the EOT.
- b** specifies a blocking factor (in this example, 20 KB). The blocking factor must be an integer between 1 and 20.

`/usr2` specifies the file system to be backed up.

Note:

The **X** command-line option specifies an 8mm tape drive.

You can recover the file system or any file in the backup saveset with the `restore` command. To recover the entire backup saveset, key in the following command, where `r` recovers all files from the saveset.

```
# restore -r
```

To recover a file, use the interactive mode of the `restore` command. Once invoked, `restore` simulates the directory structure of the file system and activates the `restore` environment, complete with prompt and basic command ability such as `cd`, `ls`, and `pwd`. You can use these commands to locate the file to recover from the saveset. When you locate the file to recover, use the `a` command to add the file to the list of files to recover and use the `x` command to extract the file from the saveset. The following example illustrates how the `/usr2/john/help` file can be recovered.

```
# restore -i
restore > cd /usr2/john
restore > ls
abcd    check
help    test1
restore > a help
restore > x
```

Please refer to the `backup(1)` and `restore(1)` manual pages in the System V Online Documentation product for more information on the `backup` and `restore` commands.

9.3.5 The ansitape Utility

The **ansitape** utility, an ANSI-standard magnetic tape label program, reads, writes, and creates magnetic tapes conforming to the ANSI standard for magnetic tape labeling. This utility's primary use is to enable you to exchange tapes with those made with VAX/VMS. **ansitape** is useful for transferring design files to remote locations that do not have a network link.

The following command uses **ansitape** to create a tape of an **ansitape** saveset on a tape drive located at SCSI ID 6:

```
$ ansitape -cv mt=/dev/rmt/mt6 rs=512 rf=f cc=e *.dgn
```

c specifies the create operation.

v specifies verbose mode.

mt specifies the output device (in this example, */dev/rmt/mt6*).

rs=512 sets the record size to 512 bytes. This qualifier must be specified exactly as shown in this example or **ansitape** will not transfer design files correctly.

rf=f specifies fixed record format. This qualifier must be specified exactly as it appears in this example or **ansitape** will not transfer design files correctly.

cc=e specifies that carriage control is embedded in the records. This qualifier must be specified exactly as it appears in this example or **ansitape** will not transfer design files correctly.

***.dgn** specifies which files to be transferred. The asterisk is a wildcard. All files with the *.dgn* extension will be transferred.

The following command uses the **t** (table) option to list the contents of an **ansitape** readable tape:

```
$ ansitape -tv
```

To recover an **ansitape** saveset created on a workstation/server, you must use the **x** option as follows to extract the files:

```
$ ansitape -xv
```

To recover an **ansitape** saveset created on a VAX, use the VMS copy command. Ensure that files are contiguous before graphically displaying them.

Please refer to the **ansitape(1)** manual page in the System V Online Documentation product for more information on the **ansitape** command.

9.3.6 The vmsbackup Utility

The `vmsbackup` utility reads VMS-generated backup format tapes. `vmsbackup` cannot write these types of savesets. Like `ansitape`, this utility is useful for transferring design files between the VMS and CLIX operating systems.

The following command reads a VMS backup format tape mounted on a tape drive located at SCSI ID 6:

```
$ vmsbackup -xvf /dev/rmt/mt6
```

x Specifies that files will be extracted.
v Specifies verbose mode.
f Specifies the device file (in this example, `/dev/rmt/mt6`).

The following command uses the `t` (table) option to list the contents of the VMS backup format tape:

```
$ vmsbackup -tv
```

Please refer to the `vmsbackup(1)` manual page in the System V Online Documentation product for more information on the `vmsbackup` command.

9.3.7 The rtc Utility

The rtc (Remote Tape Controller) allows you to use a tape drive on another machine as if it were connected directly to your machine. For example, you can use rtc to allocate the remote tape drive to your workstation/server. Then, you can use a backup utility (such as cpio) to transfer files to or from the tape drive. And finally, you can use rtc again to deallocate the remote tape drive.

For more information on rtc, refer to the *Intergraph Network Core User's Guide*.

9.3.8 The compress Utility

The `compress` command is not used to back up files. However, this command can be helpful for archiving files because it is used to compress a large amount of data into a small area of media space.

The `compress` command can be used with `tar` and `cpio` to compress the contents of a directory into one file. The `compress` command can also be used alone to compress one file to a smaller size. When you compress a directory or file, a new file (with a `.Z` extension) is created containing the compressed file(s).

Note:

To compress a directory, you must be in superuser mode.

The following examples compress the `/usr/chris` directory into an archive file (called `archive_file`) on a storage media such as tape or floppy disk:

```
# tar cvf - /usr/chris | compress > archive_file
```

OR

```
# find /usr/chris -print | cpio -ov | compress > archive_file
```

To decompress a file, use the `compress -d` command. If you used `compress` with a backup utility, you must decompress the file with the same backup utility you used to compress it. The following examples decompress the archive file (`archive_file`) created in the previous examples:

```
# compress -d < archive_file | tar xvf -
```

OR

```
# compress -d < archive_file | cpio -ivmud
```

If you wish to compress a single file (possibly to be stored on storage media later), you can use the `compress` command by itself, as shown in the following example:

```
# compress filename
# ls
filename.Z
```

The following example decompresses this single, compressed file. When you decompress a single file, notice that you should not include the *.Z* extension.

```
# compress -d filename
# ls
filename
```

Please refer to the `compress(1)` manual page in the System V Online Documentation product for more information on the `compress` command.

9.3.9 Erasing Tapes

When used with the `erase` option, the `mt` command (magnetic tape manipulation program) will erase tapes. However, because all Intergraph tape drives perform a full-width erase automatically during write operations, issuing a separate command to erase tapes is unnecessary.

The following command erases a tape mounted in a drive at SCSI ID 6:

```
# mt -f /dev/rmt/mt6 erase
```

When this command is issued, the drive logically disconnects from the SCSI bus to perform the erase. The drive will erase to the end of the tape, rewind, and send *command complete* status back to the host, at which point the host is available to execute new commands.

Note:

Both 1/4-inch and 8 mm tape drives perform the erase procedure as described above. However, 8 mm tapes require between 2 and 2.5 hours to complete an erase operation. Thus, if you wish to initiate an erase operation on an 8 mm tape drive, issue the command as a background process or create another window so that you can continue working while the operation is being executed. Keep in mind, however, that you will not be able to access the drive that is erasing the tape until the operation has completed. The only way to interrupt an erase operation is to reboot the system.

9.4 Additional Commands for Floppy Disks

This section discusses additional commands that can be used to read from and write to floppy disks. These commands include the following:

- `format` (to prepare the disk for storing data)
- `mkfs` (to create a floppy disk file system)
- `mf` (to mount a floppy disk file system)
- `uf` (to unmount a floppy disk file system)
- `cpflop` (to copy the contents of one floppy to another floppy)
- `to_flop` (to write to multiple volumes of floppies)
- `fr_flop` (to read data from multiple volumes of floppies)

However, before these commands are presented, a discussion on how the CLIX operating system accesses floppy disk drives is appropriate. First of all, before a floppy disk can be accessed, it must be formatted for CLIX (using the `format` command). After a floppy disk has been formatted, it can be written using one of two methods. You can create a *file system* on a floppy disk and use it like a CLIX file system (accessing it through UNIX commands like `mv` and `cp`). Or, you can use the floppy disk as a *raw device* (accessing it through backup utilities like `cpio` and `tar`). The raw device method is usually used for archiving data.

The floppy disk device file names the disk type. All disk device files reside in the `/dev/dsk` (for mountable disk types) and `/dev/rdisk` (for raw disk types). Intergraph offers 5¼-inch and 3½-inch floppy disk drives. The device files that represent the 5¼-inch floppy disk drive are `/dev/dsk/floppy` and `/dev/rdisk/floppy`, while the device files that represent the 3½-inch floppy disk drive are `/dev/dsk/ufloppy` and `/dev/rdisk/ufloppy`.

Most systems have only one floppy disk drive. However, some 6000-series systems may have both a 3½-inch and a 5¼-inch floppy disk drive. If your system has only one floppy disk drive, the device files representing that drive will be linked to a file called `/dev/(r)dsk/fl`. This file is called by the various backup utilities when the floppy disk drive is being accessed. If your system has both a 5¼-inch and 3½-inch floppy disk drive, the 3½-inch drive will automatically be assigned as the default (to be linked to `/dev/(r)dsk/fl`). When you wish to access the 5¼-inch drive through a backup utility, you will need to link `/dev/(r)dsk/fl` to `(/dev/(r)dsk/floppy`.

9.4.1 The format Command

Initially, floppy disks require formatting before they can be used as storage media. However, once the storage device has been formatted, you do not need to reformat. Formatting a floppy disk erases all data and rewrites the tracks and sectors to prepare the disk for holding data. Diskettes can be high- or low-density. High-density on Intergraph systems is quad-density, double-sided for floppy disks. Intergraph recommends using high-density diskettes.

	High-Density	Low-Density
5¼-inch	1.2 MB (2400 blocks)	.36 MB (720 blocks)
3½-inch	1.44 MB (28880 blocks)	.72 MB (1440 blocks)

Cautions:

- Formatting destroys all data on a floppy disk.
- A floppy disk formatted for System V must be reformatted to be used in PC-DOS. Reformatting is necessary because System V formats structure the disk differently than PC-DOS disks do.
- A high-density floppy disk should not be formatted as a low-density floppy disk and a low-density disk should not be formatted as a high-density disk.

Follow these steps to format a high-density floppy disk for System V using the format command:

1. Insert the high-density floppy disk in the floppy disk drive.
2. At the system prompt, key in the following:

```
$ /etc/format
```

The following messages appear:

```
Formatting high density floppy,
total 512 blocks -2400
Warning: All data on this floppy will be lost
Insert floppy and hit return when ready
```

When the disk activity light goes out and the system prompt appears, the format procedure is complete.

Follow these steps to format a low-density floppy disk for System V using the **format** command:

1. Insert the low-density floppy disk in the floppy disk drive.
2. At the System V prompt, key in the following:

```
$ /etc/format -l
```

The following messages appear:

```
Formatting low density floppy,  
total 512 blocks -720  
Warning: All data on this floppy will be lost  
Insert floppy and hit return when ready
```

When the system prompt appears, the format procedure is complete.

9.4.2 The mkfs Command

The `mkfs` command (make file system) creates a file system on a floppy disk. Creating a file system allows a floppy disk to be used as a CLIX file system (such as `/usr`). If you do this, standard System V commands can be used to create and manipulate directories and files on the floppy disk.

Follow these steps to create a file system on a floppy disk:

1. Insert a formatted floppy disk in the floppy disk drive.
2. For a high-density floppy disk, key in the following (specifying 2,400 as the number of blocks that a high-density, 5¼-inch floppy disk stores or 2,880 as the number of blocks that a high-density, 3½-inch floppy disk stores):

```
# /etc/mkfs /dev/dsk/fl 2400
```

OR

```
# /etc/mkfs /dev/dsk/fl 2880
```

For a low-density floppy, key in the following (specifying 720 as the number of blocks that a low-density, 5¼-inch floppy disk stores or 1,440 as the number of blocks that a low-density, 3½-inch floppy disk stores):

```
# /etc/mkfs /dev/dsk/fl 2400
```

OR

```
# /etc/mkfs /dev/dsk/fl 2880
```

3. After creating a file system on a floppy, run the `labelit` program to attach a base directory name to the file system. For instance, to run `labelit` on the floppy disk, key in the following, where `mnt1` represents the mounted name of the file system and `mnt2` represents the (user definable) volume name:

```
# /etc/labelit /dev/dsk/floppy mnt1 mnt2
```

4. Like any file system, the floppy disk file system must be mounted before it can be accessed. The `mf` (mount floppy) command can be used to mount the floppy disk file system to the `/mnt` directory (an empty directory used for temporarily mounting additional file systems). Key in the following to mount the floppy disk file system:

```
$ mf
```

5. Once the floppy disk file system has been mounted, change to the `/mnt` as follows to access the floppy disk file system:

```
$ cd /mnt
```

You can now create, manipulate, and delete files located in the floppy disk file system directory.

6. When you have finished accessing the floppy disk, you must use the `uf` (unmount floppy) command to unmount the floppy disk file system. To do this, first change to another directory besides `/mnt`. Then key in `uf` to unmount the floppy file system as follows:

```
$ cd
```

```
$ uf
```

For more information on the `mkfs`, `mf`, and `uf` commands, refer to the manual pages in the System V Online Documentation product.

9.4.3 The `cpflop` Command

The `cpflop` (copy floppy) command copies the entire contents of one floppy disk to another floppy disk, using the same drive. To use this command, key in `cpflop` at the system prompt, as shown in the following example:

```
$ cpflop
Insert source floppy and press <Return> when ready ...
Insert destination floppy #1 and press <Return> when ready ...
1228800 bytes read
1228800 bytes written
```

For more information on the `cpflop` command, refer to the manual page in the System V Online Documentation product.

9.4.4 The `to_flop` Command

The `to_flop` command copies input to the floppy disk drive, prompting you to insert sequential floppy disks as needed. This command can be used with backup utilities such as `tar` and `cpio` to write multiple volumes of floppy disks. The following examples write the `/usr/chris` directory to one or more floppy disks:

```
$ tar cvf - /usr/chris | to_flop

OR

$ find /usr/chris -print | cpio -ov | to_flop
```

To restore files backed-up using `to_flop`, you must use the `fr_flop` utility described in the next section.

For more information on `to_flop` refer to the manual page in the System V Online Documentation product.

9.4.5 The `fr_flop` Command

The `fr_flop` command reads data from one or more floppy disks. This command can be used with backup utilities such as `tar` and `cpio` to restore files located on multiple volumes of floppy disks. The following examples restore the `/usr/chris` directory to the hard disk:

```
$ fr_flop | tar xvf -

OR

fr_flop | cpio -ivmud
```

For more information on `fr_flop` refer to the manual page in the System V Online Documentation product.

- extract** All files on the extraction list are extracted from the backup tape. *restore* will ask which volume the user wishes to mount. The fastest way to extract a few files is to start with the last volume and work toward the first volume.
- setmodes** All directories added to the extraction list have their owner, modes, and times set; nothing is extracted from the tape. This is useful for cleaning up after a restore has been prematurely aborted.
- verbose** The sense of the *v* key is toggled. When set, the verbose key causes the *ls* command to list the i-node numbers of all entries. It also causes *restore* to print information about each file as it is extracted.
- help** List a summary of the available commands.
- quit** *restore* immediately exits even if the extraction list is not empty.

The following characters may be used in addition to the letter that selects the function desired.

- b** The next argument to *restore* is used as the block size of the tape (in kilobytes). If the *-b* option is not specified, *restore* tries to determine the tape block size dynamically.
- f** The next argument to *restore* is used as the name of the archive instead of */dev/rmt/0m*. *rtc(1)* can be used to restore tapes from a remote tape device. If the name of the file is "-", *restore* reads from standard input. Thus, *backup(1)* and *restore* can be used in a pipeline to backup and restore a file system with the following command:
- ```
backup Of - /usr | (cd /mnt; restore xf -)
```
- v** Normally *restore* works silently. The *v* (verbose) key causes it to type the name of each file it treats preceded by its file type.
- y** *restore* will not ask whether it should abort the restore if it gets a tape error. It will always try to skip the bad tape block(s) and continue.

- m** *restore* will extract by i-node numbers rather than by file name. This is useful if only a few files are being extracted to avoid regenerating the complete path name to the file.
- h** *restore* extracts the actual directory, rather than the files that it references. This prevents hierarchical restoration of complete subtrees from the tape.
- s** The next argument to *restore* is a number that selects the file on a multiple backup tape. File numbering starts at 1.

#### FILES

|                |                                                 |
|----------------|-------------------------------------------------|
| /dev/rmt/0m    | the default tape drive                          |
| /tmp/rstidir*  | file containing directories on the tape         |
| /tmp/rstmode*  | owner, mode, and time stamps for directories    |
| /restoresymtab | information passed between incremental restores |

#### SEE ALSO

backup(1), rtc(1).  
newfs(1M), mount(1M) in the *CLIX System Administrator's Reference Manual*.  
mkfs(1M0) in the *UNIX System Administrator's Reference Manual*.

#### DIAGNOSTICS

Complains about bad *key* characters.

Complains if it gets a read error. If *y* has been specified or the user responds "y," *restore* will attempt to continue the restore.

If the backup extends over more than one tape, *restore* will ask the user to change tapes. If the *x* or *i* *key* has been specified, *restore* will also ask which volume the user wishes to mount. The fastest way to extract a few files is to start with the last volume and work toward the first volume.

Numerous consistency checks can be listed by *restore*. Most checks are self-explanatory or can "never happen." Common errors are given below.

*File-name*: not found on tape

The specified *file-name* was listed in the tape directory, but was not found on the tape. This is caused by tape read errors while looking for the file and from using a backup tape created on an active file system.

Expected next file *inumber*, got *inumber*

A file that was not listed in the directory appeared. This can occur when using a backup tape created on an active file system.

Incremental tape too low

When performing incremental restore, a tape that was written before the previous incremental tape or that has an incremental level that is too low was loaded.

Incremental tape too high

When performing incremental restore, a tape that does not begin its coverage where the previous incremental tape left off, or that has an incremental level that is too high was loaded.

Tape read error while restoring *file-name*

Tape read error while skipping over i-node *inumber*

Tape read error while trying to resynchronize

A tape read error occurred. If a file name is specified, its contents are probably partially wrong. If an i-node is being skipped or the tape is trying to resynchronize, no extracted files have been corrupted, though files may not be found on the tape.

Resync restore, skipped *num* blocks

After a tape read error, *restore* may need resynchronize. This message lists the number of blocks that were skipped.

## BUGS

*restore* can become confused when performing incremental restores from backup tapes made on active file systems.

A level zero backup must be performed after a full *restore*. Because *restore* runs in user code, it cannot control i-node allocation; thus, a full *restore* must be performed to get a new set of directories. These directories reflect the new i-node numbering even though the file contents are unchanged.

### 9.3.5 The ansitape Utility

The **ansitape** utility, an ANSI-standard magnetic tape label program, reads, writes, and creates magnetic tapes conforming to the ANSI standard for magnetic tape labeling. This utility's primary use is to enable you to exchange tapes with those made with VAX/VMS. **ansitape** is useful for transferring design files to remote locations that do not have a network link.

The following command uses **ansitape** to create a tape of an **ansitape** saveset on a tape drive located at SCSI 6:

```
$ ansitape -cv mt=/dev/rmt/mt6 rs=512 rf=f cc=e *.dgn
```

**c** specifies the create operation.

**v** specifies verbose mode.

**mt** specifies the output device (in this example, */dev/rmt/mt6*).

**rs=512** sets the record size to 512 bytes. This qualifier must be specified exactly as shown in this example or **ansitape** will not transfer design files correctly.

**rf=f** specifies fixed record format. This qualifier must be specified exactly as it appears in this example or **ansitape** will not transfer design files correctly.

**cc=e** specifies that carriage control is embedded in the records. This qualifier must be specified exactly as it appears in this example or **ansitape** will not transfer design files correctly.

**\*.dgn** specifies which files to be transferred. The asterisk is a wildcard. All files with the *.dgn* extension will be transferred.

The following command uses the **t** (table) option to list the contents of an **ansitape** readable tape:

```
$ ansitape -tv
```

To recover an **ansitape** saveset created on a workstation/server, you must use the **x** option as follows to extract the files:

```
$ ansitape -xv
```

To recover an **ansitape** saveset created on a VAX, use the VMS **copy** command. Ensure that files are contiguous before graphically displaying them.

The following is the **ansitape** manual page from the System V Online Documentation product.

**NAME**

ansitape - ANSI-standard magtape label program

**SYNOPSIS**

ansitape [-] t | x | r | c [vqfaei3] [mt=*device*] [vo=*volume-name*] [rs= [r | *record-size*]] [bs=*block-size*] [rf= [ | f]] [cc= [i | f | e]] *file-name* ...

**DESCRIPTION**

*ansitape* reads, writes, and creates magtapes conforming to the American National Standards Institute (ANSI) standard for magtape labeling.

Primarily, this is useful to exchange tapes with VAX/VMS™ which makes this kind of tape by default.

*ansitape* is controlled by a function key letter (t, x, c, or r). Various options modify the format of the output tape.

The function letters describe the overall operation desired. A minus sign (-) is allowed, but optional, to introduce the first keyword option set. The function is specified with one of the following:

- r Write to a magtape.
- c Create a new magtape. The tape is initialized with a new ANSI volume header. All files previously on the tape are destroyed. This option implies r.
- x Extract all files from the tape. Files are placed in the current directory. Protection is read/write to everyone, modified by the current *umask*(2).
- t List all file names on the tape.

These key letters may follow the function letter:

- v Normally, *ansitape* works silently; the v (verbose) option displays the name of each file *ansitape* treats, preceded by the function letter. It also displays the volume name of each tape as it is mounted. When used with the t option, *ansitape* displays the number of tape blocks used by each file, the record format, and the carriage control option.

- q Query before writing. On write (c or r options), this causes *ansitape* to ask before writing to the tape. On extract operations, *ansitape* displays the CLIX path name and asks if it should extract the file. Any response starting with a “y” or “Y” means yes, and any other response (including an empty line) means no.
- f File I/O is done to standard I/O instead. For example, when writing a tape file that will contain a lint listing, the following could be specified:

```
lint xyz.c | ansitape rf xyz.lint
```

instead of

```
lint xyz.c > /tmp/xyz.lint
ansitape r /tmp/xyz.lint
rm /tmp/xyz.lint
```

When reading, this option causes the extracted files to be sent to *stdout* instead of to a disk file.

- a The tape should be read or written with the ASCII character set. This is the default.
- e The tape should be written with the EBCDIC character set. The mapping is the same one used by the *dd(1M)* program with *conv=ebcdic*. This option is automatically enabled if IBM® format labels are selected.
- i Use IBM format tape labels. The IBM format is similar but not identical to the ANSI standard. The major difference is that the tape will contain no HDR3 or HDR4 records and restricts the names of the files on the tape to 17 characters. This option automatically selects the EBCDIC character set for output. To make an IBM format label on a tape using the ASCII character set, use the option sequence *ia*.
- 3 Do not write HDR3 or HDR4 labels. The HDR3 label is reserved for the use of the operating system that created the file. HDR4 is for overflow of file names longer than the 17 characters allocated in the HDR1 label. Not all systems process or ignore these labels correctly. This switch suppresses the HDR3 and HDR4 labels when the tape will be transferred to a system that cannot support these types of labels.

Function modifiers should be given as a separate argument to *ansitape*. Multiple modifiers may be specified. They must appear after the key-letter options above and before any file name arguments.

**mt=***device*

Select an alternate drive on which the tape is mounted. The default is `/dev/rmt/mt0n`.

**vo=***volume-name*

Specify the name of the output volume. Normally, this defaults to the first six characters of the login name. The string "UNIX™" is used as the default if *ansitape* cannot determine the login name.

**rs=***record-size*

Specify the output record size in bytes. This is the maximum size in the case of variable-format files. This option also turns on the fixed-record-format option. Thus, for variable record sizes with a smaller maximum,

**rs=***record-size* **rf=v**

must be specified. When the record size is manually given, *ansitape* does not read disk files to determine the maximum record length.

**rs=r**

This is a variant of the **rs=** option. It causes *ansitape* to read all disk files for record size, regardless of their size. Normally, files larger than 100K bytes are not scanned for record size. Using this option also implies variable-length records.

**bs=***block-size*

Specify the output block size, in bytes. As many records as will fit are written into each physical tape block. ANSI standards limit this to 2048 bytes (the default), but more or less may be specified. Specifying more may prevent some systems from reading the tape.

**rf=v**

Record format is variable-length. In other words, they are text files. This is the default and normally should not be changed.

**rf=f**

Record format is fixed-length. This is usually a bad choice, and should be reserved for binary files.

- cc=i** Carriage control is implied (default). Unlike CLIX text files where records are delimited by a newline character, ANSI files do not normally include the newline as part of the record. Instead, a newline is automatically added to the record whenever the record is sent to a printing device.
- cc=f** Carriage control FORTRAN. Each line is expected to start with a FORTRAN carriage-control character. *ansitape* does not insert these characters automatically, it merely marks the file as having them. This is of limited usefulness.
- cc=e** Carriage control is embedded. Carriage control characters (if any) are part of the data records. This is usually used with binary data files.

#### Writing ANSI Tapes

The list of files on the command line is written to the tape. A full CLIX path name may be specified. However, only the last path name component (everything after the last /\_ is used as the file name on the tape.

Normally, regular text files are to be exchanged. *ansitape* reads the files one line at a time and transfers them to the tape. The newline character at the end of each line is removed, and the file is written in a variable-length record format. Variable-format files have the length of the longest record specified in a file header. Therefore, *ansitape* will read each input file from disk before the file is written to tape, to determine the maximum record size. The read is skipped if the file is more than 100,000 bytes long. The default carriage control (implied) instructs the other host to restore the newline character before printing the record.

If *ansitape* assumes that the input file is a CLIX text file (FORTRAN or implied carriage control), it will automatically strip the CLIX newline from the end of each record. Stripping is not done with embedded carriage control (**cc=e**) files. If the size of a nontext file (**cc=e**) is not a multiple of the record size, the partial record at the end will be lost.

For binary files, fixed-length records should be used. VAX/VMS normally uses a record length of 512 bytes for things like directories and executable files, but data files may have any record length. Binary files should be flagged for embedded carriage control.

### Reading ANSI Tapes

When reading, the input file list is presumed to be the names of files to be extracted from the tape. The shell wildcard characters asterisk (\*) and question mark (?) may be used. Of course, they must be quoted to prevent the shell from interpreting them before *ansitape* sees them.

None of the options for record format or carriage control need to be specified when reading files. *ansitape* will automatically pick up this information from the header records on the tape and run accordingly. If *ansitape* does not fulfill requirements, the resulting files may be run through *dd(1M)*.

### Multivolume support

When *ansitape* reaches the end of a tape while reading, it requests the next volume with the message "Mount continuation tape and push return." However, the tape is not checked to ensure that the correct volume was mounted.

When *ansitape* reaches the end of a tape it is writing, it requests the next volume as described above. When the new volume is online, *ansitape* initializes it with an ANSI volume header containing a volume name generated from the volume name of the first tape of the set. If the original name is fewer than six characters long, it is padded to six characters with underscores (\_). Then, the last two characters of the name are replaced by a two-digit sequence number. For example, *tap* becomes *tap\_02*, *mylab1* becomes *myla02*, and so forth. The sequence number is the tape's position in the set.

### FILES

/dev/rmt/mt\* half-inch magnetic tape interface  
 /dev/rmt/ms\* quarter-inch magnetic tape interface

### SEE ALSO

*dd(1M)* in the *UNIX System V System Administrator's Reference Manual*.  
*umask(2)* in the *UNIX System V Programmer's Reference Manual*.

### CAVEATS

The *r* (write) option cannot be used with quarter-inch archive tapes, since these tape drives cannot backspace.

The *n**th* occurrence of a file cannot be requested.

File names longer than 80 characters are truncated. This is a limitation of the ANSI labeling standard. If the tape is made without HDR3 and HDR4 labels (3 or i switch), the name is limited to 17 characters.

The record size of files transferred with embedded carriage control must be a multiple of the block size.

### 9.3.6 The vmsbackup Utility

The `vmsbackup` utility reads VMS-generated backup format tapes. `vmsbackup` cannot write these types of savesets. Like `ansitape`, this utility is useful for transferring design files across the VMS and CLIX operating systems.

The following command reads a VMS backup format tape mounted on a tape drive located at SCSI 6:

```
$ vmsbackup -xvf /dev/rmt/mt6
```

**x** Specifies that files will be extracted.  
**v** Specifies verbose mode.  
**f** Specifies the device file (in this example, `/dev/rmt/mt6`).

The following command uses the `t` (table) option to list the contents of the VMS backup format tape:

```
$ vmsbackup -tv
```

The following is the `vmsbackup` manual page from the System V Online Documentation product.

#### NAME

`vmsbackup` - read a VMS backup tape

#### SYNOPSIS

```
vmsbackup [-tx] [-cdevw] [-s setnumber] [-f tapefile] [name ...]
```

#### DESCRIPTION

`vmsbackup` reads a VMS-generated backup tape and writes the files to a CLIX disk. The default operation of the program is to extract every file from the tape and write it to disk. The default may be modified by the following options:

- c** Use complete file names including the version number. A colon and the octal version number will be appended to all file names. This option is useful only when multiple versions of the same file are on a single tape or when a file with the same name exists in the destination directory. By default, version numbers are ignored.
- d** Use the directory structure from VMS.
- e** Process all file name extensions. Since this program is mainly intended to move source code and possibly data from a VMS system to a CLIX system, the default is to ignore all files whose file name extensions specify system-dependent data. The file types that will be ignored unless the **-e** option is specified are as follows:

**exe** VMS executable file  
**lib** VMS object library file  
**obj** RSX™ object file  
**odl** RSX overlay description file  
**olb** RSX object library file  
**pmd** RSX post-mortem dump file  
**stb** RSX task symbol table file  
**sys** RSX bootable system file  
**tsk** RSX executable task file

- f** Use the next argument in the command line as the tape device to be used rather than the default */dev/rmt/0m*.
- s *setnumber*** Process only the given saveset number
- t *tapefile*** Produce a table of contents (a directory listing) on the standard output of the files on tape.
- v** Set verbose mode. Normally *vmsbackup* works silently. The verbose option will cause the file names of the files being read from tape to be written to standard output.
- w** Query the user for file disposition. *vmsbackup* prints the message "extract *file-name* [ny]" and waits for user confirmation that the file is to be extracted. If a word beginning with "y" is given, the file is copied to the file system. Any other input is interpreted as no.
- x** Extract the named files from the tape. The optional *name* argument specifies one or more file names to be searched for on the tape. Pattern matching in the manner of *sh(1)* is attempted using the meta-characters \*, ?, [, and ]. Only files with matching names are processed.

#### FILES

*/dev/rmt/0m*

#### BUGS

The file name match uses the complete VMS file names.

### 9.3.7 The `rtc` Utility

The `rtc` (Remote Tape Controller) allows you to use a tape drive on another machine as if it were connected directly to your machine. For more information on `rtc`, refer to the *Intergraph Network Core User's Guide*.

The following is the `rtc` manual page from the System V Online Documentation product.

#### NAME

`rtc` - remote tape control

#### SYNOPSIS

```
rtc -a [-s system] [-r rewdev] [-n norewdev] [-t timeout] controldev
rtc [-d | -e] controldev
```

#### DESCRIPTION

`rtc` allows a tape drive on another machine to be used as if it resides on the local machine. The following command options allow configuration of the tape drive.

- a Allocate a tape drive on a remote machine.
- e Examine and report the machine where the remote tape resides.
- d Deallocate the tape drive on a remote machine.

The allocate command allows the use of a tape drive on a remote machine. Once allocated, the remote tape drive remains allocated until a timeout occurs (explained later), or the deallocate command is invoked. *Controldev* is the name of a tape control device (i.e., `/dev/rmt/rt0ctl`) which controls the functions of other tape devices in the same group.

The following options are supported:

- |                    |                                                                                       |
|--------------------|---------------------------------------------------------------------------------------|
| -s <i>system</i>   | Specify the remote <i>system</i> where the tape drive resides.                        |
| -r <i>rewdev</i>   | Specify the tape drive <i>rewdev</i> on <i>system</i> as the rewindable tape device.  |
| -n <i>norewdev</i> | Specify the tape drive <i>norewdev</i> on <i>system</i> as the no-rewind tape device. |

- t *timeout*                   Set the idle time allowed to *timeout* minutes. If the tape is idle for *timeout* minutes, it is deallocated following a warning which is printed on the system console. The default *timeout* is 5.
  
- e                               Print the Ethernet address of the remote machine where the control device *controldev* resides.
  
- d                               Close the connection to the remote machine. If the tape drive is being used when the deallocate command is invoked, an error is printed.

If an allocate option is not present, the corresponding environment variable is used if set.

- RTCSYSTEM                    Specifies the remote system (-s option).
  
- RTCREWIND                    Specifies the rewind device (-r option).
  
- RTCNOREWIND                 Specifies the no-rewind device (-n option).
  
- RTCTIMEOUT                  Specifies the idle timeout (-t option).

*rtc* also looks in a series of files to determine the proper action if an option to the allocate command is not present. Given *controldev*, *rtc* looks for a line that begins with the name *controldev*. The line that begins with *controldev* should be followed by a remote system name, separated by space(s) or tab(s). *rtc* then continues looking for a line that begins with that remote system name. Upon finding that, the line should contain a list of options in the following order:

```
rewdev [norewdev [timeout]]
```

Only *rewdev* must be present.

An example file would look like the following:

```
/dev/rmt/rt0ctl ipro1
ipro1 /dev/rmt/mt6 /dev/rmt/mt6n 10
```

The following is the list of files checked.

```
.rtcr
$HOME/.rtcr
/etc/rtcr
```

**EXAMPLES**

This command allocates the `/dev/rmt/mt6` tape drive on the machine `ipro1`.

**FILES**

|                               |                                               |
|-------------------------------|-----------------------------------------------|
| <code>/dev/rmt/rt?.ctl</code> | the control device (used only by <i>rtc</i> ) |
| <code>/dev/rmt/rt?</code>     | the rewind <i>rtc</i> device                  |
| <code>/dev/rmt/rt?n</code>    | the no-rewind <i>rtc</i> device               |

**SEE ALSO**

`cpio(1)`, `mt(1)`, `tar(1)`, `rtc_allocate(3)`.  
`rtc_s(1M)`, `rtc(7S)`, `tc(7S)` in the *CLIX System Administrator's Reference Manual*.

### 9.3.8 The compress Utility

The `compress` command is not used to back up files. However, this command can be helpful for archiving files because it is used to compress a large amount of data into a small area of media space.

The `compress` command can be used with `tar` and `cpio` to compress the contents of a directory into one file. The `compress` command can also be used alone to compress one file to a smaller size. When you compress a directory or file, a new file (with a `.Z` extension) is created containing the compressed file(s).

**Note:**

To compress a directory, you must be in superuser mode.

The following examples compress the `/usr/chris` directory into an archive file (called `archive_file`) on a storage media such as tape or floppy disk:

```
tar cvf - /usr/chris | compress > archive_file
```

OR

```
find /usr/chris -print | cpio -ov | compress > archive_file
```

To decompress a file, use the `compress -d` command. If you used `compress` with a backup utility, you must decompress the file with the same backup utility you used to compress it. The following examples decompress the archive file (`archive_file`) created in the previous examples:

```
compress -d < archive_file | tar xvf -
```

OR

```
compress -d < archive_file | cpio -ivmud
```

If you wish to compress a single file (possibly to be stored on storage media later), you can use the `compress` command by itself, as shown in the following example:

```
compress filename
ls
filename.Z
```

The following example decompresses this single, compressed file. When you decompress a single file, notice that you should not include the `.Z` extension.

```
compress -d filename
ls
filename
```

The following is the **compress** manual page from the System V Online Documentation product.

#### NAME

**compress**, **uncompress**, **<cat - compress and expand data**

#### SYNOPSIS

```
compress [-f] [-v] [-c] [-V] [-d] [-b maxbits] [file ...]
uncompress [-f] [-v] [-c] [-V] [file ...]
<cat [-V] [<f=3>file ...]
```

#### DESCRIPTION

*compress* reduces the size of the named files using adaptive Lempel-Ziv coding. Whenever possible, each file *file* is replaced by one with the form *file.Z*, while keeping the same ownership modes, access and modification times. If no files are specified, the standard input is compressed to the standard output. Compressed files can be restored to their original form using *uncompress*, *zcat*, or **compress -d**.

The **-f** option forces compression of *file*. This is useful for compressing an entire directory, even if some of the files do not actually shrink. If **-f** is not given and *compress* is run in the foreground, the user is prompted as to whether an existing file should be overwritten.

The **-c** option makes *compress* and *uncompress* write to the standard output; no files are changed. The nondestructive behavior of *zcat* is identical to that of **uncompress -c**.

*compress* uses the modified Lempel-Ziv algorithm popularized in "A Technique for High Performance Data Compression", Terry A. Welch, *IEEE Computer*, vol. 17 no. 6 (June 1984), pp. 8-19. Common substrings in the file are first replaced by 9-bit codes 257 and up. When code 512 is reached, the algorithm switches to 10-bit codes and continues to use more bits until the limit specified by the **-b** flag is reached (default 16). *Maxbits* must be between 9 and 16. The default can be changed in the source to allow *compress* to be run on a smaller machine.

After the *maxbits* limit is attained, *compress* periodically checks the compression ratio. If it is increasing, *compress* continues to use the existing code dictionary. However, if the compression ratio decreases, *compress* discards the table of substrings and rebuilds it from scratch. This allows the algorithm to adapt to the next "block" of the file.

Note that the **-b** flag is omitted for *uncompress*, since the *maxbits* parameter specified during compression is encoded within the output, along with a magic number to ensure that neither decompression of random data nor recompression of compressed data is attempted.

The amount of compression obtained depends on the size of the input, the number of bits per code, and the distribution of common substrings. Typically, text such as source code or English is reduced by 50-60%. Compression is generally much better than that achieved by Huffman coding or adaptive Huffman coding, and takes less time to compute.

Under the **-v** option, a message is printed yielding the percentage of reduction for each file compressed.

If the **-V** option is specified, the current version and compile options are printed on **stderr**.

#### DIAGNOSTICS

Exit status is normally 0; if the last file is larger after (attempted) compression, the status is 2; if an error occurs, exit status is 1.

Usage: `compress [-dfvcV] [-b maxbits] [file ...]`  
Invalid options were specified on the command line.

Missing *maxbits*

*Maxbits* must follow **-b**.

*file*: not in compressed format

The file specified to *uncompress* has not been compressed.

*file*: compressed with *xx* bits, can only handle *yy* bits

*File* was compressed by a program that could deal with a larger *maxbits* than the *compress* code on this machine. Recompress the file with smaller *maxbits*.

*file*: already has **.Z** suffix -- no change

The file is assumed to be already compressed. Rename the file and try again.

*file*: filename too long to tack on **.Z**

*File* cannot be compressed because its name is longer than 12 characters. Rename and try again.

*file* already exists; do you wish to overwrite (y or n)?

Respond "y" if the output file should be replaced; "n" if not.

*uncompress*: corrupt input

A SIGSEGV violation was detected which usually means that the input file has been corrupted.

Compression: *xxxx%*

Percentage of the input saved by compression. (Relevant only for *-v*).

- not a regular file: unchanged  
When the input file is not a regular file, (e.g., a directory), it is left unaltered.
- has *xx* other links: unchanged  
The input file has links; it is left unchanged. See *ln(1)* for more information.
- file unchanged  
No savings are achieved by compression. The input remains virgin.

#### **BUGS**

Although compressed files are compatible between machines with large memory, *-b12* should be used for file transfer to architectures with a small process data space (64K bytes or less, as exhibited by the DEC™ PDP™ series, the Intel 80286, etc.)

## 9.4 Additional Commands for Floppy Disks

This section discusses additional commands that can be used to read from and write to floppy disks. These commands include the following:

- `format` (to prepare the disk for storing data)
- `mkfs` (to create a floppy disk file system)
- `mf` (to mount a floppy disk file system)
- `uf` (to unmount a floppy disk file system)
- `cpflop` (to copy the contents of one floppy to another floppy)
- `to_flop` (to write to multiple volumes of floppies)
- `fr_flop` (to read data from multiple volumes of floppies)

However, before these commands are presented, a discussion on how the CLIX operating system accesses floppy disk drives is appropriate. First of all, before a floppy disk can be accessed, it must be formatted for CLIX (using the `format` command). After a floppy disk has been formatted, it can be written using one of two methods. You can create a *file system* on a floppy disk and use it like a CLIX file system (accessing it through UNIX commands like `mv` and `cp`). Or, you can use the floppy disk as a *raw device* (accessing it through backup utilities like `cpio` and `tar`). The raw device method is usually used for archiving data.

The floppy disk device file names the disk type. All disk device files reside in the `/dev/dsk` (for mountable disk types) and `/dev/rdisk` (for raw disk types). Intergraph offers 5¼-inch and 3½-inch floppy disk drives. The device files that represent the 5¼-inch floppy disk drive are `/dev/dsk/floppy` and `/dev/rdisk/floppy`, while the device files that represent the 3½-inch floppy disk drive are `/dev/dsk/ufloppy` and `/dev/rdisk/ufloppy`.

Most systems have only one floppy disk drive. However, some 6000-series systems may have a 3½-inch and a 5¼-inch floppy disk drive. If your system has only one floppy disk drive, the device files representing that drive will be linked to a file called `/dev/(r)dsk/fl`. This file is called by the various backup utilities when the floppy disk drive is being accessed. If your system has both a 5¼-inch and 3½-inch floppy disk drive, the 3½-inch drive will automatically be assigned as the default (to be linked to `/dev/(r)dsk/fl`). When you wish to access the 5¼-inch drive through a backup utility, you will need to link `/dev/(r)dsk/fl` to `(/dev/(r)dsk/floppy`.

### 9.4.1 The format Command

Initially, floppy disks require formatting before they can be used as storage media. However, once the storage device has been formatted, you do not need to reformat. Formatting a floppy disk erases all data and rewrites the tracks and sectors to prepare the disk for holding data. Diskettes can be high- or low-density. High-density on Intergraph systems is quad-density, double-sided for floppy disks. Intergraph recommends high-density diskettes.

|         | High-Density           | Low-Density          |
|---------|------------------------|----------------------|
| 5¼-inch | 1.2 MB (2400 blocks)   | .36 MB (720 blocks)  |
| 3½-inch | 1.44 MB (28880 blocks) | .72 MB (1440 blocks) |

#### Cautions:

- Formatting destroys all data on a floppy disk.
- A floppy disk formatted for System V must be reformatted to be used in PC-DOS. Reformatting is necessary because System V formats structure the disk differently than PC-DOS disks do.
- A high-density floppy disk should not be formatted as a low-density floppy disk and a low-density disk should not be formatted as a high-density disk.

Follow these steps to format a high-density floppy disk for System V using the `format` command:

1. Insert the high-density floppy disk in the floppy disk drive.
2. At the system prompt, key in the following:

```
$ /etc/format
```

The following messages appear:

```
Formatting high density floppy,
total 512 blocks -2400
Warning: All data on this floppy will be lost
Insert floppy and hit return when ready
```

When the disk activity light goes out and the system prompt appears, the format procedure is complete.

Follow these steps to format a low-density floppy disk for System V using the **format** command:

1. Insert the low-density floppy disk in the floppy disk drive.
2. At the System V prompt, key in the following:

```
$ /etc/format -l
```

The following messages appear:

```
Formatting low density floppy,
total 512 blocks -720
Warning: All data on this floppy will be lost
Insert floppy and hit return when ready
```

When the system prompt appears, the format procedure is complete.

## 9.4.2 The mkfs Command

The `mkfs` command (make file system) creates a file system on a floppy disk. Creating a file system allows a floppy disk to be used as a CLIX file system (such as `/usr`). If you do this, standard System V commands can be used to create and manipulate directories and files on the floppy disk.

Follow these steps to create a file system on a floppy disk:

1. Insert a formatted floppy disk in the floppy disk drive.
2. For a high-density floppy disk, key in the following (specifying 2,400 as the number of blocks that a high-density, 5/4-inch floppy disk stores or 2,880 as the number of blocks that a high-density, 3 1/2-inch floppy disk stores):

```
/etc/mkfs /dev/dsk/fl 2400
```

OR

```
/etc/mkfs /dev/dsk/fl 2880
```

For a low-density floppy, key in the following (specifying 720 as the number of blocks that a low-density, 5/4-inch floppy disk stores or 1,440 as the number of blocks that a low-density, 3 1/2-inch floppy disk stores):

```
/etc/mkfs /dev/dsk/fl 2400
```

OR

```
/etc/mkfs /dev/dsk/fl 2880
```

3. After creating a file system on a floppy, run the `labelit` program to attach a base directory name to the file system. For instance, to run `labelit` on the floppy disk, key in the following, where `mnt1` represents the mounted name of the file system and `mnt2` represents the (user definable) volume name:

```
/etc/labelit /dev/dsk/floppy mnt1 mnt2
```

4. Like any file system, the floppy disk file system must be mounted before it can be accessed. The **mf** (mount floppy) command can be used to mount the floppy disk file system to the */mnt* directory (an empty directory used for temporarily mounting additional file systems). Key in the following to mount the floppy disk file system:

```
$ mf
```

5. Once the floppy disk file system has been mounted, change to the */mnt* as follows to access the floppy disk file system:

```
$ cd /mnt
```

You can now create, manipulate, and delete files located in the floppy disk file system directory.

6. When you have finished accessing the floppy disk, you must use the **uf** (unmount floppy) command to unmount the floppy disk file system. To do this, first change to another directory besides */mnt*. Then key in **uf** to unmount the floppy file system as follows:

```
$ cd
$ uf
```

For more information on the **mkfs**, **mf**, and **uf** commands, refer to the manual pages in the System V Online Documentation product.

### 9.4.3 The `cpflop` Command

The `cpflop` (copy floppy) command copies the entire contents of one floppy disk to another floppy disk, using the same drive. To use this command, key in `cpflop` at the system prompt, as shown in the following example:

```
$ cpflop
Insert source floppy and press <Return> when ready ...
Insert destination floppy #1 and press <Return> when ready ...
1228800 bytes read
1228800 bytes written
```

For more information on the `cpflop` command, refer to the manual page in the System V Online Documentation product.

### 9.4.4 The `to_flop` Command

The `to_flop` command copies input to the floppy disk drive, prompting you to insert sequential floppy disks as needed. This command can be used with backup utilities such as `tar` and `cpio` to write multiple volumes of floppy disks. The following examples write the `/usr/chris` directory to one or more floppy disks:

```
$ tar cvf - /usr/chris | to_flop
```

OR

```
$ find /usr/chris -print | cpio -ov | to_flop
```

To restore files backed-up using `to_flop`, you must use the `fr_flop` utility described in the next section.

For more information on `to_flop` refer to the manual page in the System V Online Documentation product.

### 9.4.5 The `fr_flop` Command

The `fr_flop` command reads data from one or more floppy disks. This command can be used with backup utilities such as `tar` and `cpio` to restore files located on multiple volumes of floppy disks. The following examples restore the `/usr/chris` directory to the hard disk:

```
$ fr_flop | tar xvf -
```

OR

```
fr_flop | cpio -ivmud
```

For more information on `fr_flop` refer to the manual page in the System V Online Documentation product.





# Chapter 10: Using the Utility Pages

---

This chapter should be used as a reference for the Utility Pages. Each field on each Utility Page is described briefly.

The Utility Pages are used to configure the workstation or server operating systems, hard disk(s), and peripheral ports. System parameters such as nodename, system date and time, and the Utility Page password should be set when you first set up a system.

This chapter is divided into two sections: the first section describes the workstation Utility menus and the second section describes the server Utility menus. Each section contains illustrations of the Utility menus and descriptions of the options found on each page.

This chapter contains the following sections:

|         |                                    |       |
|---------|------------------------------------|-------|
| 10.1    | Workstation Utility Menus          | 10-3  |
| 10.1.1  | Main Utility Page                  | 10-4  |
| 10.1.2  | Peripheral Configuration Page      | 10-7  |
| 10.1.3  | Operating System Parameters Page   | 10-11 |
| 10.1.4  | Disk Maintenance Utility Page      | 10-13 |
| 10.1.5  | Hard Disk Flaw Data Utility Page   | 10-16 |
| 10.1.6  | Verify Utility Page                | 10-19 |
| 10.1.7  | Format Utility Page                | 10-21 |
| 10.1.8  | Disk Partitioning Utility Page     | 10-23 |
| 10.1.9  | Device Image Copy Utility          | 10-26 |
| 10.1.10 | Rebuild Utility Page               | 10-30 |
| 10.2    | Server Utility Menus               | 10-32 |
| 10.2.1  | Startup Configuration Utility Menu | 10-34 |
| 10.2.2  | Main Utility Menu                  | 10-35 |
| 10.2.3  | Date and Time Menu                 | 10-36 |
| 10.2.4  | Peripheral Configuration Menu      | 10-39 |
| 10.2.5  | Operating System Parameters Menu   | 10-42 |
| 10.2.6  | Disk Maintenance Utilities Menu    | 10-43 |
| 10.2.7  | Disk Partitioning Menu             | 10-46 |
| 10.2.8  | Device Image Copy Utility Menu     | 10-48 |
| 10.2.9  | Rebuild Utility Menu               | 10-54 |



## 10.1 Workstation Utility Menus

To access the workstation Utility menus, you must first reboot the workstation. At the blue Introductory Screen, the cursor appears over the System V icon. Move the cursor within five seconds to avoid booting into the System V operating system. Move the cursor to the Utility icon and select it by tapping the data button. If the Utility Pages have a password, you will be required to key in a password before you can access the Utility Pages.

The following screen buttons are found at the bottom of all Utility Pages except the Verify and Format page:

- **Help.** Displays a message telling you how to access Help for the Utility Page options. To read Help information for a particular parameter, place the cursor over the icon in question and tap the reset button. Information for that specific parameter will display.
- **Utility.** Returns you to the Main Utility Page.
- **System V.** Loads the System V operating system.
- **PC-DOS.** Loads the PC-DOS operating system. This option applies only to 32C workstations.
- **Reset.** Restores the default parameters for the Utility Page that is currently displayed.
- **Load.** Restores the previously saved parameters for the Utility Page that is currently displayed.
- **Save.** Saves the currently displayed parameters for that particular Utility Page. These parameters apply to future workstation boots until they are changed and saved again.

### 10.1.1 Main Utility Page

The main Utility Page, which is accessed directly from the Introductory Screen, provides the means for accessing the remaining Utility Pages. In addition, various system parameters can be set from this page.

The screenshot shows a window titled "Utility" on the left. The main area contains the following fields and options:

- Workstation Password: [ ]
- Timezone: Atlantic (C-6) [ ]
- Daylight Savings: Standard [ ] Daylight [ ]
- Month: October [ ] Day: 18 [ ] Year: 1986 [ ]
- Current time: 14:07:33 [ ]
- Default Operating System: System V [ ] PC-DOS [ ]
- Peripheral Configuration: [ ]
- Operating System Parameters: [ ]
- Disk Maintenance Utility: [ ]
- Disk Partitioning Utility: [ ]
- Device Image Copy Utility: [ ]
- Rebuild Utility: [ ]

At the bottom, there are buttons for Help, Utility, System V, PC-DOS, Reset, Load, and Save.

Figure 10-1. Main Utility Page

#### *Workstation Password*

This password restricts access to the Utility Pages. The Utility Pages allow you to perform disk maintenance tasks such as formatting and partitioning. Thus, you should assign a workstation password when you first receive a system. Only the system administrator should be able to access the Utility Pages.

#### *Timezone*

This field sets the correct timezone for your geographic location. When you first receive a system, you should use this field to assign the proper timezone to the Utility Pages. The list on the following page specifies valid timezones.

Available options for timezones include the following:

|                        |               |
|------------------------|---------------|
| Atlantic (G-4)         | Greenwich + 8 |
| Eastern (G-5)          | Greenwich + 7 |
| Central (G-6)          | Greenwich + 6 |
| Mountain (G-7)         | Greenwich + 5 |
| Pacific (G-8)          | Greenwich + 4 |
| Greenwich - 9          | Greenwich + 3 |
| Alaska - Hawaii (G-10) | Greenwich + 2 |
| Greenwich - 11         | Greenwich + 1 |
| Greenwich - 12         | Greenwich - 1 |
| Greenwich + 11         | Greenwich - 2 |
| Greenwich + 10         | Greenwich - 3 |
| Greenwich + 9          |               |

See the section on Server Utility Menus for a more detailed listing of the international timezones.

**Note:**

When you set the Timezone, Current Time, and Daylight Savings fields on this Utility Page, you are giving the start-up utility program only the information needed to calculate Greenwich Mean Time. To actually view the time corresponding to your timezone, you must set the TZ variable in the /etc/TIMEZONE directory. The default setting for the TZ variable is for Central Standard Time.

*Daylight Savings*

This field allows you to specify whether daylight savings time is currently in effect. This time is saved on an internal clock with battery backup; thus, you should not need to reset this setting at any time.

*Month*

This field allows you to specify and display the correct month. You should make sure that the correct month displays when you first receive a system.

*Day*

This field allows you to specify and display the correct day of the month. You should make sure that the correct day displays when you first receive a system.

*Year*

This field allows you to specify and display the correct year. You should make sure that the correct year displays when you first receive a system.

*Current Time*

This field allows you to display or enter the correct time, in international form, using the following format:

hh mm ss

For example, if the time is 3:29 pm, the entry in this data entry field should be as follows:

15 29 00

*Default Operating System*

This parameter determines which operating system will be automatically loaded when the workstation is booted. After this switch is set, the Save button should be selected to retain this setting each time the workstation is booted. The available settings are **System V** and **PC-DOS**. This option is valid only for 32C workstations since System V is the only available operating system on all other workstations. PC-DOS is available on other workstations only with the Soft PC package (which runs under UNIX).

*Access to Other Utility Pages*

The following screen buttons, when selected, give access to their respective Utility Pages:

- Peripheral Configuration
- Operating System Parameters
- Disk Maintenance Utility
- Disk Partitioning Utility
- Device Image Copy Utility
- Rebuild Utility

### 10.1.2 Peripheral Configuration Page

The Peripheral Configuration Utility Page is used to establish settings for the Screen, Keyboard, Digitizer, and RS-232 Port.

The screenshot shows the 'Peripheral Configuration' utility page. It features a title box on the left and several configuration sections on the right:

- Screen:** Screen Saver is set to 'on'.
- Keyboard:** Key Click is 'on', Membrane Click is 'on', and Bell Tone is 'soft'.
- Digitizer:** Floating Menu Present is 'yes'.
- Serial Port Parameters:**
  - Aux Port Select: sux00
  - Slave Device: 'yes' (checked)
  - Port Type: RS-232
  - Auto XON/XOFF: 'on' (checked)
  - Baud Rate: 9600 bps
  - Word Size: 8 bits
  - Incoming XOFF: 'on' (checked)
  - Parity: disabled
  - Stop Bits: 2
  - Auto RTS/CTS: 'on' (checked)

Navigation buttons at the bottom include: Help, Utility, System V, PC-DOS, Reset, Load, and Save.

Figure 10-2. Peripheral Configuration Page

#### Screen Saver

The Screen Saver feature causes the screen to dim automatically after a period of inactivity. The default setting is yes (to activate the Screen Saver). You will probably wish to leave the Screen Saver turned on to help preserve the screen.

#### Key Click

This setting determines whether the keyboard keys click when pressed. The default setting is on. This setting causes no difference in functionality; it is simply a matter of preference.

#### Membrane Click

This setting determines whether the membrane keys click when pressed. The default setting is on. This setting causes no difference in functionality; it is simply a matter of preference.

### *Bell Tone*

This setting determines whether a bell tone is heard, and if it is, how loud it will be. (The bell tone is used by the vi editor and other software to indicate invalid operations.) The options available in this roll-through box include the following:

- soft
- loud
- none

### *Floating Menu Present*

This setting enables or disables a floating menu on the digitizer. This switch must be set to no (default) if a floating menu is not present.

### *Aux Port Select*

This setting displays the auxiliary port for which all other RS-232 Parameters are established. The following are the three auxiliary ports:

- aux00
- aux01
- aux02

See "Configuring for Peripheral Devices" in the *CLIPPER Hardware Setup and Maintenance Guide* for information on establishing settings for the auxiliary ports.

### *Slave Device*

A slave device only accepts input from the host but cannot send data back to the host. For example, many printers are slave devices. The slave device switch has no function in DOS.

The following settings are available:

- Yes to establish the slave device
- No to establish a no-slave device

### *Port Type*

This setting allows auxiliary port 0 to be set for RS-232 or RS-449.

### *Baud Rate*

Baud rate is a measure of the number of bits per second (bps) transmitted over a serial communication channel. This number is typically about ten times greater than the number of characters transmitted per second.

The baud rate depends on the device connected to the port. The default setting on an Intergraph workstation is 9600.

The following baud rates are available:

|           |          |          |
|-----------|----------|----------|
| 50 bps    | 300 bps  | 2400 bps |
| 75 bps    | 600 bps  | 3600 bps |
| 110 bps   | 1200 bps | 4800 bps |
| 134.5 bps | 1800 bps | 7200 bps |
| 150 bps   | 2000 bps | 9600 bps |

### *Parity*

Parity is a method by which transmitted data can be verified. Even parity indicates that each transmitted word will contain an even number of bits in a logic 1 state. Odd parity indicates that each transmitted word will contain an odd number of bits in a logic 1 state. A special bit referred to as the *parity* bit is either set or cleared upon transmission to meet the parity requirement. When a word is received, if the parity does not check correctly, the data was most likely corrupted during transmission.

The **Disabled** setting does not check or generate parity (default). The **Odd** setting checks and sets the parity to odd for all outgoing data. The **Even** setting checks and sets the parity to even for all outgoing data.

### *Port Type*

This field sets the communication protocol to RS-232 or RS-449 for applicable workstations. (This setting is currently disabled.)

### *Word Size*

Word size is the number of bits that make up a word. Typical values are 5, 6, 7, and 8. Since the ASCII character set uses 7-bit characters, word sizes of 7 and 8 bits are most common. Word size is usually dependent on the device. The default setting is 8 bits.

### *Stop Bits*

Stop bits are an asynchronous indicator that the bits to follow are a valid word. Stop bits are necessary to indicate the beginning of a transmitted word since the sender and receiver of data are not synchronized. Settings of 1 and 2 are available. The default setting is 1 stop bit.

### *Auto XON/XOFF*

Auto XON/XOFF is a protocol used for flow control of serially transmitted data between two peripherals. As data is transmitted to a peripheral that cannot process the data as fast as it is sent, that peripheral can send a stop character to stop the transmission temporarily. When ready, a START character resumes the transmission. Use of this protocol ensures that data will not be lost during transmission. On and off settings are available. The default is on.

Auto XON is a software protocol used by the receiver to ensure that a stop character will be sent when input buffers are nearly full. As the input buffers become clear again, a Start character is sent to resume transmission. On and off settings are available.

### *Incoming XOFF*

Incoming XOFF is a software protocol used by the transmitter to ensure that all START and STOP characters will be honored according to protocol. On and off settings are available. The default is on.

### *Auto RTS/CTS*

RTS/CTS is a protocol used for flow control of serially transmitted data. Instead of transmitting characters for flow control, electrical signals perform the handshaking. The Ready-to-Send (RTS) signal is generated by one peripheral when the peripheral is ready to receive a character. This signal is interpreted as input to the Clear-To-Send (CTS) input of another peripheral. Likewise, RTS from the second peripheral is CTS to the first. On and off settings are available. The default setting is on.

### 10.1.3 Operating System Parameters Page

The Operating System Parameters Utility Page is used to establish default parameters that become effective when the operating system is loaded.

The screenshot shows the 'Operating System Parameters' utility page. On the left is a vertical box labeled 'Operating System Parameters'. The main area is titled 'PC-DOS Information' and contains a toggle for 'Enable Math Coprocessor' with 'yes' selected. Below this is a horizontal line, followed by 'System V Information'. This section has four toggles: 'Boot Single User' (set to 'no'), 'Z-buffer Allocation' (set to 'none'), 'Automatic Dump' (set to 'yes'), and 'Automatic Reboot' (set to 'no'). At the bottom of the main area is a date field '08-00-35-20-12-08' and a 'Node Name' field with a right-pointing arrow. A navigation bar at the very bottom contains buttons for 'Help', 'Utility', 'System V', 'PC-DOS', 'Reset', 'Load', and 'Save'.

Figure 10-3. Operating System Parameters Page

#### *Enable Math Coprocessor*

This setting determines whether the math coprocessor is enabled when the PC-DOS operating system is loaded. The math coprocessor allows the workstation to run PC-DOS programs compiled with floating point math libraries. The default is **yes**. This option is inactive for all 200-series (and above) workstations and servers since these workstations do not have a PC-DOS partition. To run PC-DOS, these workstations can use the Soft PC package. Servers cannot run PC-DOS.

#### *Boot Single User*

This field determines whether the workstation is booted into single-user or multiuser mode. The default is **no** (multiuser mode). If you boot into single-user mode, you will automatically be logged in to the *root* account, and only the *root* file system will be mounted. (*usr* will not be mounted.)

### *Z-Buffer Allocation*

This parameter enables you to allocate 4 MB main UNIX memory to the Graphics Processor (GZ) of 240 workstations for the integrated frame buffer. This option is available only for 240 workstations with 16 MB of main memory. Settings of *left*, *right*, and *both* determine which screen will receive the allocation. (This menu setting is presently disabled.)

### *Automatic Dump*

This yes/no toggle switch can be set to automatically dump all data in memory to the dump device in the System V file system (swap /dev/dsk/s0u0p7.1). If the workstation has a system crash and this switch is set to *yes*, an automatic dump occurs. The dump may provide information pertinent to the cause of the crash. The default is *yes*.

### *Automatic Reboot*

This yes/no toggle switch can be set so that the workstation is automatically rebooted after a system crash. The default is *no*.

### *Ethernet Present*

This yes/no toggle switch can be set to load IEEE 802.3 software when the workstation is booted, thus enabling network communications. The default is *yes*.

### *Node Name*

A nodename is a logical name representing the network address for the system. When you first receive a system, you should assign a nodename in this field. The nodename can be a maximum of eight alphabetic characters, none of which may be uppercase. The hard-wired network address of the workstation automatically displays above the Node Name heading.

### 10.1.4 Disk Maintenance Utility Page

The Disk Maintenance Utility Page is the main utility page from which the following disk Utility Pages can be directly or indirectly accessed:

- Hard Disk Flaw Data Utility Page
- Verify Page
- Format Page

The Disk Maintenance Utility Page is also used to format a floppy disk. Any hard disk settings established on the other hard disk Utility Pages are saved when the hard disk is formatted.

The screenshot displays the Disk Maintenance Utility Page interface. On the left, a box labeled "Disk Maintenance Utility" is visible. The main area is divided into two sections:

- Floppy Disk Formatting:** Includes a "Floppy Density" toggle switch set to "high", an "Elapsed Time" display showing "00:00:00", and a "Format" button.
- Hard Disk Maintenance:** Includes fields for "SCSI ID", "Logical Unit Number", "Disk Capacity" (63960), "Intergraph Part Number" (MESAS 18xx), and "Serial Number" (829053). It also features a "Format Type" toggle switch set to "vms", and buttons for "Profile", "Next Page", and "Park".

At the bottom, there are navigation buttons: "Help", "UTILITY", "System V", "PC-DOS", "Reset", "Load", and "Save".

Figure 10-4. Disk Maintenance Utility Page

#### *Floppy Density*

This toggle switch should be set according to the density of floppy disk you will format using the Utility Page format. Set the switch to high density to format a high-density floppy disk or low density to format a low-density floppy disk.

#### *Format*

Select this switch to format a high- or low-density floppy disk that is inserted in the floppy disk drive. The floppy density toggle switch must be set accordingly.

*Elapsed Time*

This field indicates the amount of time that has elapsed since the floppy format began.

*SCSI ID*

Both the SCSI ID and Logical Unit Number are used to identify or address a hard disk connected to the workstation. The SCSI ID value identifies the hard disk controller. If only one hard disk is connected to the workstation, the SCSI ID is 0 and Logical Unit Number is 0.

*Logical Unit Number*

Both the Logical Unit Number and SCSI ID are used to identify or address a hard disk connected to the workstation. The Logical Unit Number identifies the hard disk. If only one hard disk is connected to the workstation, the SCSI ID is 0 and Logical Unit Number is 0.

*Disk Capacity*

This field displays the number of formatted blocks on the addressed hard disk.

*Intergraph Part Number*

This roll-through box identifies the type of hard addressed disk. The available part numbers are as follows:

- MESA518xx
- FDSK131xx
- FDSK132xx
- FDSK135xx
- FDSK126xx
- FDSK150xx
- FDSK155xx
- FDSK156xx
- FDSK211xx
- FDSK223xx
- FDSK226xx
- FDSK229xx
- FDSK230xx
- User-defined
- Foreign

**Note:**

If **Foreign** displays in this field when you enter this Utility Page, roll through the choices and select the part number appropriate for your hard disk. The correct part number for your disk is given on the Hard Disk Profile Sheet that came with your unit.

*Serial Number*

The serial number is a unique number given to each hard disk. The number currently displayed should be the serial number of the addressed hard disk. If not, enter the correct serial number in this field.

*Format Type*

This toggle switch must be set to **VMS** when an addressed disk that will be used as a VMS disk on a Micro II workstation will be formatted. The default setting is **UNIX**.

*Profile*

This field prints a full profile of the currently addressed hard disk to a printer connected to one of the RS-232 auxiliary ports. This profile includes information on the hardware and format parameters.

*Park*

This setting causes the heads on the addressed hard disk to be parked. Parking the heads is recommended when the system is shut down to maintain the integrity of the data on the disk.

*Next Page*

This icon sends you to the Hard Disk Flaw Data page from which the Verify and Format pages can be accessed.

### 10.1.5 Hard Disk Flaw Data Utility Page

The Hard Disk Flaw Data Utility Page is used to identify, add, and remove hard disk flaws. The flaws are mapped to the addressed hard disk when the hard disk is formatted.

Hard Disk Flaw Data Utility

SCSI ID      Disk Capacity: 63960      Logical Unit Number

No Flaws

Previous Flaw Page      Next Flaw Page

Physical Cylinder: 0      Physical Head: 0      Physical Offset: 0

Logical Block: 0

Add Flaw      Remove Flaw

Format With Flaw Data: yes < no      Verify On Format: yes < no

Verify      Format

Previous Page

Help    Utility    System V    PC-DOS

Reset    Load    Save

Figure 10-5. Hard Disk Flaw Data Utility Page

#### *SCSI ID*

This field displays the SCSI ID value that was set on the Disk Maintenance Utility Page. The value identifies which disk controller is affected by the settings on the other Disk Maintenance Utility Pages.

#### *Disk Capacity*

This field displays the number of formatted blocks on the addressed hard disk.

#### *Logical Unit Number*

This field displays the Logical Unit Number value that was set on the Disk Maintenance Utility Page. The value identifies which disk connected to the disk controller is affected by the settings on the other Disk Maintenance Utility Pages.

*Previous Flaw Page*

This icon returns you to the previous page of flaw data if more flaws exist on the disk than can be listed on the Hard Disk Flaw Data page.

*Next Flaw Page*

This icon moves you to the next page of flaw data if more flaws exist on the disk than can be listed on the Hard Disk Flaw Data page.

*Physical Cylinder*

The physical disk location of a flaw is defined by the physical cylinder, the physical head, and the physical offset. Each of these values is necessary for determining where data is physically located on the disk. Thus, if you are entering flaws by physical location, use this icon to key in the specific cylindrical area of a recording surface (or the physical cylinder part of the address).

*Physical Head*

The physical disk location of a flaw is defined by the physical cylinder, the physical head, and the physical offset. Each of these values is necessary for determining where data is physically located on the disk. If you are entering flaws by physical location, use this icon to key in the read/write head for a particular disk surface (the physical head part of the address).

*Physical Offset*

The physical disk location of a flaw is defined by the physical cylinder, the physical head, and the physical offset. Each of these values is necessary for determining where data is physically located on the disk. If you are entering flaws by physical location, use this icon to key in the byte index offset of the flawed disk (the physical offset part of the address).

*Add Flaw*

This icon allows you to add the flaw location, which was identified by the physical dimensions or logical block location, to the flaw list.

*Logical Block*

A logical block is the address of a flawed area of the hard disk. The operating system reports this address to the screen when a disk block becomes corrupted.

*Remove Flaw*

This icon allows you to remove the flaw location, which was identified by the physical dimensions or logical block location, from the flaw list.

*Format with Flaw Data*

This yes/no toggle switch determines whether the flaw settings stored on the Hard Disk Flaw Data page are included with the format. Setting this switch to yes (default) marks the flaws on the hard disk and retains the flaw list.

*Verify on Format*

This yes/no toggle switch can be set to locate additional flaws on the hard disk when a disk is formatted. Setting this switch to yes (default) checks for additional bad blocks by writing and reading data from the hard disk.

*Verify*

This icon causes the Verify Utility Page to display.

*Format*

This icon causes the Format Utility Page to display.

*Previous Page*

This icon returns you to the Disk Maintenance Utility Page.

### 10.1.6 Verify Utility Page

The Verify Utility Page enables you to verify the addressed disk nondestructively, adding any flaws found to the flaw list.

The screenshot shows a utility page with a large 'Verify' button on the left. To the right, there are three input fields: 'SCSI ID' (with a right-pointing arrow), 'Disk Capacity' (displaying '67960'), and 'Logical Unit Number' (with a left-pointing arrow). Below these are three more fields: 'Current Block' (displaying '0'), 'Elapsed Time' (displaying '00:00:00'), and 'Flaw Count' (displaying '0'). At the bottom of this section are two buttons: 'Verify' and 'Previous Page'. At the very bottom of the page is a navigation bar with buttons for 'Help', 'Utility', 'System V', 'PC-DOS', 'Reset', 'Load', and 'Save'.

Figure 10-6. Verify Utility Page

#### *SCSI ID*

This field displays the SCSI ID value that was set on the Disk Maintenance Utility Page. The value identifies which disk controller is affected by the settings on the other Disk Maintenance Utility Pages.

#### *Disk Capacity*

This field displays the number of formatted blocks on the addressed hard disk.

#### *Logical Unit Number*

This field displays the Logical Unit Number value that was set on the Disk Maintenance Utilities page. The value identifies which disk connected to the disk controller that is affected by the settings on the other Disk Maintenance Utility Pages.

*Current Block*

This field displays the disk block currently being examined, thus indicating the status of the verification.

*Elapsed Time*

This field displays the amount of time that has elapsed since the hard disk verification began.

*Flaw Count*

This field displays the current length of the flaw list for the addressed hard disk.

*Verify*

This icon initiates the Verify procedure. The Verify procedure reads the hard disk and records all flaws to ensure that data is not written over flawed areas. This procedure will not erase or change any data on the disk. You should run the Verify procedure any time you receive bad block messages.

*Previous Page*

This icon returns you to the Hard Disk Flaw Data Utility Page.

### 10.1.7 Format Utility Page

The Format Utility Page enables you to format the currently addressed hard disk and destroys all data currently on the disk.

The screenshot shows a utility page titled "Format". On the left is a large "Format" button. The main area contains several fields:

| SCSI ID       | Disk Capacity | Logical Unit Number |
|---------------|---------------|---------------------|
| >             | 63960         | >                   |
| Current Block | Eapsed Time   | Flow Count          |
| 0             | 00:00:00      | 0                   |
| Test Pattern  | Test Pattern  | Test Pattern        |
| >0xC293       | >0x333        | >0x0                |

Below these fields are three buttons: "Format", "Next Page", and "Previous Page". A warning message states: "All data on addressed disk will be destroyed." At the bottom, there are navigation buttons: "Help", "UTILITY", "System V", "PC-DOS", "Reset", "Load", and "Save".

Figure 10-7. Format Utility Page

#### SCSI ID

This field displays the SCSI ID value that was set on the Disk Maintenance Utility Page. The value identifies which disk controller is affected by the settings on the other Disk Maintenance Utility Pages.

#### Disk Capacity

This field displays the number of formatted blocks on the addressed hard disk.

#### Logical Unit Number

This field displays the Logical Unit Number value that was set on the Disk Maintenance Utilities page. The value identifies which disk connected to the disk controller that is affected by the settings on the other Disk Maintenance Utility Pages.

*Current Block*

This field indicates the status of the format by displaying the disk block that is currently being operated on.

*Elapsed Time*

This field indicates the amount of time that has elapsed since the hard disk format began.

*Flaw Count*

This field displays the current length of the flaw list for the addressed hard disk.

*Format*

This icon initiates the Format procedure.

*Test Pattern*

The entries in the three **Test Pattern** fields define the read, write, and verify data patterns that the hard disk is tested with. Most users do not need to adjust these settings.

*Next Page*

This icon sends you to the Disk Partitioning Utility Page.

*Previous Page*

This icon returns you to the Hard Disk Flaw Data Utility Page.

10.1.8 Disk Partitioning Utility Page

The Disk Partitioning Utility Page is used to specify how you want your hard disk partitioned.

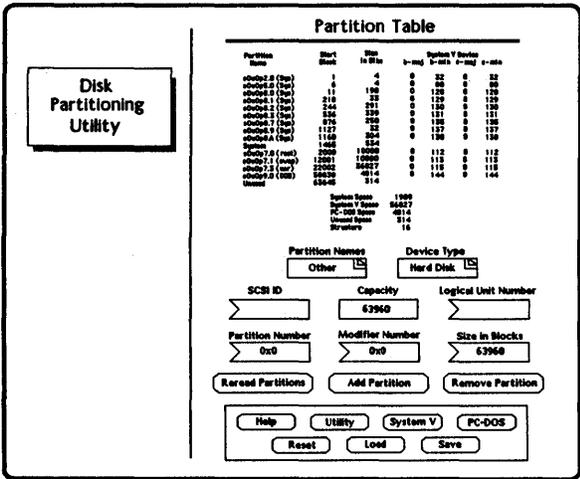


Figure 10-8. Disk Partitioning Utility Page

Partition Table

The Partition Table displays the names of the hard disk partitions and their sizes. Any changes to a partition are reflected in this table.

Partition Names

This roll-through box is used to display the name of the hard disk partition that you want to modify. The following partition names are available:

- System
- root
- swap
- usr
- DOS
- Other

Note:

The partition name value will be automatically set to "Other" when you set the partition number and modifier number for additional partitions.

*Device Type*

This field displays the type of device (either hard disk or floppy disk) that is being addressed.

*SCSI ID*

Both the SCSI ID and Logical Unit Number are used to identify or address a hard disk connected to the workstation. The SCSI ID value identifies the hard disk controller. If only one hard disk is connected to the workstation, the SCSI ID is 0 and Logical Unit Number is 0.

*Disk Capacity*

This field displays the number of formatted blocks on the addressed hard disk.

*Logical Unit Number*

Both the Logical Unit Number and SCSI ID are used to identify or address a hard disk connected to the workstation. The Logical Unit Number identifies the hard disk. If only one hard disk is connected to the workstation, the SCSI ID is 0 and Logical Unit Number is 0.

*Partition Number*

This field displays the partition number associated with the partition shown in the Partition Names roll-through box. When modifying a partition, use this data-entry box to key in the partition number that will be associated with the partition. You can enter the partition number in hexadecimal form (for example, 0x7) or just the partition number (for example, 7). If you key in just the partition number, it will be converted to hexadecimal form when you press <Return>.

*Modifier Number*

This field displays the modifier number associated with the partition shown in the Partition Names roll-through box. When modifying a partition, use this data-entry box to key in the modifier number that will be associated with the partition. You can enter the modifier number in hexadecimal form (for example, 0x4) or just the partition number (for example, 4). If you key in just the partition number, it will be converted to hexadecimal form when you press <Return>.

*Size in Blocks*

This field displays the size of the partition (in blocks) that currently displays in the Partition Name roll-through box. This data-entry box is used to key in the desired size of the partition being modified.

*Reread Partitions*

This icon causes the partition information to be re-read from the selected device.

*Add Partition*

This icon adds the partition displayed in the Partition Name roll-through box and its corresponding size to your hard disk.

*Remove Partition*

This icon removes the partition displayed in the Partition Name roll-through box from your hard disk.

### 10.1.9 Device Image Copy Utility

The Device Image Copy Utility Page allows you to copy portions of partitions or entire partitions on one device to another partition on the same or different device. Partitions can be copied between hard disks, floppy disks, and cartridge tapes. Note that cartridge tapes do not have partitions, but options for skipping to the end of a file or to the end of the tape are provided so that specific areas of the tape can be read from or written to.

#### *Device Type*

This field displays the device to be read from or written to. The available choices are hard disk, floppy, or cartridge tape. The data fields provided for each of these devices are shown in the following illustrations:

The screenshot shows the 'Device Image Copy Utility' window. On the left is a title box with the text 'Device Image Copy Utility'. The main area is divided into 'From' and 'To' columns. Both columns have 'Device Type' set to 'Hard Disk' with a small icon. Below this are fields for 'SCSI ID', 'Logical Unit Number', 'Partition' (set to 0), and 'Modifier' (set to 0). In the center, there is an 'Entire Partition' section with 'yes' selected and 'no' unselected. Below that are 'Block Count' (set to 0) and 'Block Offset' (set to 0). At the bottom of the main area are 'Elapsed Time' (00:00:00) and 'Blocks Remaining' (0). Two buttons, 'Compare' and 'Copy', are located below these fields. At the very bottom of the window is a row of system navigation buttons: 'Help', 'UNIXy', 'System V', 'PC-DOS', 'Reset', 'Load', and 'Save'.

*Figure 10-9. Hard Disk Device Type*

|                                                                                                                                                                                                                                                                          |                                                                               |                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|--------------------------------------------|
| <b>Device Image<br/>Copy<br/>Utility</b>                                                                                                                                                                                                                                 | <b>From</b>                                                                   | <b>To</b>                                  |
|                                                                                                                                                                                                                                                                          | Device Type<br>Floppy <input type="text"/>                                    | Device Type<br>Floppy <input type="text"/> |
|                                                                                                                                                                                                                                                                          | Partition<br>0 <input type="text"/>                                           | Partition<br>0 <input type="text"/>        |
|                                                                                                                                                                                                                                                                          | Modifier<br>0 <input type="text"/>                                            | Modifier<br>0 <input type="text"/>         |
|                                                                                                                                                                                                                                                                          | Block Offset<br>0 <input type="text"/>                                        | Block Offset<br>0 <input type="text"/>     |
|                                                                                                                                                                                                                                                                          | Entire Partition<br>yes <input type="text"/> no <input type="text"/>          |                                            |
|                                                                                                                                                                                                                                                                          | Block Count<br>0 <input type="text"/>                                         |                                            |
|                                                                                                                                                                                                                                                                          | Elapsed Time<br>00:00:00 <input type="text"/>                                 |                                            |
|                                                                                                                                                                                                                                                                          | Blocks Remaining<br>0 <input type="text"/>                                    |                                            |
|                                                                                                                                                                                                                                                                          | <input type="button" value="Compare"/><br><input type="button" value="Copy"/> |                                            |
| <input type="button" value="Help"/> <input type="button" value="Utility"/> <input type="button" value="System V"/> <input type="button" value="PC-DOS"/><br><input type="button" value="Reset"/> <input type="button" value="Load"/> <input type="button" value="Save"/> |                                                                               |                                            |

Figure 10-10. Floppy Disk Device Type

|                                                                                                                                                                                                                                                                          |                                                                      |                                                |                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|------------------------------------------------|--------------------------------------------|
| <b>Device Image<br/>Copy<br/>Utility</b>                                                                                                                                                                                                                                 | <b>From</b>                                                          | <b>To</b>                                      |                                            |
|                                                                                                                                                                                                                                                                          | Device Type<br>Cart. Tape <input type="text"/>                       | Device Type<br>Cart. Tape <input type="text"/> |                                            |
|                                                                                                                                                                                                                                                                          | SCSI ID<br>0 <input type="text"/>                                    | SCSI ID<br>0 <input type="text"/>              |                                            |
|                                                                                                                                                                                                                                                                          | Logical Unit Number<br>0 <input type="text"/>                        | Logical Unit Number<br>0 <input type="text"/>  |                                            |
|                                                                                                                                                                                                                                                                          | Entire Partition<br>yes <input type="text"/> no <input type="text"/> |                                                |                                            |
|                                                                                                                                                                                                                                                                          | Block Offset<br>0 <input type="text"/>                               | Block Count<br>0 <input type="text"/>          | Block Offset<br>0 <input type="text"/>     |
|                                                                                                                                                                                                                                                                          | <input type="button" value="Skip File"/>                             | Elapsed Time<br>00:00:00 <input type="text"/>  | <input type="button" value="Skip File"/>   |
|                                                                                                                                                                                                                                                                          | <input type="button" value="Skip to EOT"/>                           | Blocks Remaining<br>0 <input type="text"/>     | <input type="button" value="Skip to EOT"/> |
|                                                                                                                                                                                                                                                                          | <input type="button" value="Rewind"/>                                | <input type="button" value="Compare"/>         | <input type="button" value="Rewind"/>      |
|                                                                                                                                                                                                                                                                          | <input type="button" value="Retension"/>                             | <input type="button" value="Copy"/>            | <input type="button" value="Retension"/>   |
| <input type="button" value="Help"/> <input type="button" value="Utility"/> <input type="button" value="System V"/> <input type="button" value="PC-DOS"/><br><input type="button" value="Reset"/> <input type="button" value="Load"/> <input type="button" value="Save"/> |                                                                      |                                                |                                            |

Figure 10-11. Tape Device Type

*SCSI ID*

This field displays the SCSI ID of the device to be copied from or to. This field does not appear for floppy disks.

*Logical Unit Number*

This field displays the Logical Unit Number of the device to be copied from or to. This field does not appear for floppy disks.

*Partition*

This field allows you to specify the partition number of the partition to copy from or to. This field does not appear for cartridge tapes.

*Entire Partition*

This option allows you to specify whether to copy an entire partition (yes) or just a portion of one (no). The default setting is no.

*Modifier*

This data entry field allows you to key in the modifier number of the partition that will be copied from or to. This field does not appear for cartridge tapes.

*Block Offset*

This data-entry field allows you to key in the specific block number to begin copying from. For hard disks, this setting should be 5 so that the hard disk flaw data is bypassed.

*Block Count*

This data entry field allows you to specify the total number of blocks to be transferred. If this number exceeds the capacity of the receiving partition, only the number of blocks that can be held by that partition will be transferred.

*Elapsed Time*

This field indicates the amount of time that has elapsed since the data transfer began.

*Blocks Remaining*

This field indicates how many 512-byte blocks remain to be transferred.

*Compare*

This field allows you to compare the data specified in the "From" descriptors to the media specified in the "To" descriptors.

*Copy*

This field allows you to copy the data specified in the "From" descriptors to the media specified in the "To" descriptors. The size of the transfer will either be the entire size of the smaller partition or the size specified in the Block Count field as long as no partition boundaries are crossed.

The remaining screen buttons are used only when copying from or to cartridge tap.

*Skip File*

This field allows you to initiate a forward skip to the next file on the tape at the specified SCSI address.

*Skip to EOT*

This field allows you to initiate a forward skip to the end of the tape on the device at the specified SCSI address.

*Rewind*

This field allows you to rewind the tape on the device at the specified SCSI address.

*Retension*

This field allows you to retension the tape on the device at the specified SCSI address.

### 10.1.10 Rebuild Utility Page

The Rebuild Utility Page assists you in rebuilding the hard disk. From the Rebuild Utility Page, you will create acceptable partitions (using the default partitions or defining your own), load the Rebuild media on the hard disk, and enter the Rebuild environment (System V single-user mode). In single-user mode, the Rebuild menu provides the mechanism for restoring the file systems, loading software, and rebooting the system. For step-by-step instructions on rebuilding, refer to Chapter 11, "Rebuilding and Repartitioning the Hard Disk."

The screenshot shows a window titled "Rebuild" on the left. The main area contains a warning: "Warning: Rebuilding may result in a loss of data!". Below this are three steps:

- Step 1: Create Partition Table**  
Buttons: Default, Custom
- Step 2: Load Rebuild Media**  
Media Type: Microfloppy (selected), Load  
Media Loaded: A progress bar with markers 2 and 3.
- Step 3: Enter Rebuild Environment**  
Button: Boot

At the bottom are buttons for Help and Utility.

Figure 10-12. Rebuild Utility Page

The Rebuild Utility Page is designed to lead you through the process of rebuilding the hard disk. Notice that the page is separated into three steps (defining partitions, loading media, and entering the Rebuild environment). You will not be able to move on to the next step until you have completed the previous step satisfactorily. The left-hand portion of the screen contains important messages informing you of your current status and also identifying your next step.

### *Create Partition Table*

Step 1 requires you to define acceptable hard disk partitions. (This documentation uses the terms *partition table* and *partitions* synonymously.) If your current partitions are not acceptable, you will not be allowed to proceed to step 2 until you redefine them. Messages on the left-hand portion of the screen will indicate specific partition requirements that must be met.

The Default icon uses the Intergraph-defined default partitions and the Custom icon allows you to define your own partitions. Selecting the Custom icon will send you to the Partition Utility Page to redefine hard disk partitions. After you have redefined the partitions, go to the Main Utility Page and then to the Rebuild Utility Page, to continue in the rebuild process.

### *Load Rebuild Media*

Step 2 on the Rebuild Utility Page requires you to load the Rebuild media on the hard disk. You cannot begin this step until partitions acceptable for the hard disk have been defined.

To load the media, select your Rebuild media type (such as floppy) in the Media Type roll-through box. Then select the Load icon. You will be prompted to insert Rebuild Root floppy #2 and press <Return>. After you do so, the system begins loading the media onto the hard disk, and the Media Loaded bar chart indicates the percentage of data that has been loaded. When Rebuild Root floppy #2 finishes, you will be prompted to insert floppy #3 and press <Return>. Once again, the Media Loaded bar chart indicates the percentage of data that has been loaded from the media.

### *Enter Rebuild Environment*

Step 3 on the Rebuild Utility Page requires you to enter the Rebuild environment. You cannot enter the Rebuild environment until you have loaded the Rebuild media on the hard disk.

Once you are ready to enter the Rebuild environment, select the Boot icon. This icon will cause the system to reboot into System V single-user mode and initiate the Rebuild menu. From this menu, you will be able to restore the file systems, invoke *newprod* to load software, and reboot the system. For more information on rebuilding, refer to Chapter 11, "Rebuilding and Repartitioning the Hard Disk."

## 10.2 Server Utility Menus

Follow these steps to access the Server Utility Pages:

1. Power up the InterServe.

OR

Reboot the InterServe by keying in `/etc/reboot` at the superuser (#) prompt or by pressing the reboot button.

2. Respond as directed at the following prompt:

```
Intergraph Corporation
InterServe 200

```

Strike any key within 5 seconds to  
access Startup Utility Menus.

The Startup Configuration Utility menu appears, as shown below.

```

Startup Configuration Utility

```

```
UTility [go to main utility menu]
UNix [boot unix operating system]
Help
```

Key in your selection at the Option prompt.  
Capitals indicate key-in abbreviations.

Option:

3. Key in the following at the Startup Configuration Utility menu to access the Main Utility Menu, from which the other utility menus can be accessed:

Option: **UT**

The following standard conventions apply to all Utility menu options:

- To select a Utility menu option, key in the option name displayed by the menu. The capital letters within the option name represent the minimum key-in required for selecting the option. For example, select the **UTILITY** option from the Startup Configuration Utility menu by keying in the following:

Option: **UT**

- An information field displays beside each menu option:
  - Information enclosed in parentheses indicates a value that you can change by keying in the associated option and a desired value as an argument. The field then displays the new value. For example, keying in the following at the Date and Time Menu displays the value (**JULy**) next to the **MOntH** option.

Option: **MO JUL**

- Information enclosed in square brackets indicates an action performed when the associated option is selected. For example, the information field [**go to main utility menu**] on the Startup Configuration Utility menu indicates that the Main Utility Menu displays when the associated **UTILITY** option is selected.
- All of following standard options appear at the bottom of every Utility menu except the Startup Configuration Utility menu.

#### **UNix**

This icon loads the System V operating system.

#### **SAve**

This icon saves the currently displayed parameters for that particular Utility menu. These parameters apply to future InterServe boots until they are changed and saved again.

#### **LOad**

This icon restores the previously saved parameters for the Utility menu that is displayed.

#### **REset**

This icon restores the default parameters for the Utility menu that is displayed.

#### **Help**

This icon displays a message telling you how to access help for the Utility menus option.

### 10.2.1 Startup Configuration Utility Menu

The following Startup Configuration Utility menu is accessed as described at the beginning of this section.

---

Startup Configuration Utility

---

UTility      [go to main utility menu]  
UNix        [boot unix operating system]  
Help

Key in your selection at the Option prompt.  
Capitals indicate key-in abbreviations.

Option:

The following briefly defines the Startup Configuration Utility menu options.

**UTility**

This option displays the Main Utility Menu.

**UNix**

This option boots the System V operating system.

**Help**

This option displays a message telling you how to access help for the Startup Configuration Utility menu options.

## 10.2.2 Main Utility Menu

The following Main Utility Menu is displayed when the Startup Configuration Utility menu **Utility** option is selected.

---

### Main Utility Menu

---

|                     |                                |
|---------------------|--------------------------------|
| PassWord            | (inactive)                     |
| DAte                | (Mon Jun 12 10:51:01 1989)     |
| Peripheral_Settings | [go to peripheral config menu] |
| Operating_System    | [go to unix config menu]       |
| Disk_Maintenance    | [go to disk maintenance menu]  |
| Disk_Partitioning   | [go to disk partition menu]    |
| Device_image_CoPy   | [go to device copy menu]       |
| REBUILD             | [rebuild the internal disk]    |

#### Standard Options:

UNix   SAve   LOad   REset   Help

Option:

The following briefly defines the Main Utility Menu options.

#### PassWord

This option establishes a password for accessing the Utility menus. Once established, the password is requested when the Startup Configuration Utility menu **Utility** option is selected.

Follow these steps to choose a password:

1. Select the **PassWord** option.
2. At the prompt, key in the desired password.

The **PassWord** field displays **(active)** to indicate that a valid password has been established. Change the password by repeating the procedure.

#### DAte

This option displays the Date and Time Menu with which to change the values in the **DAte** field.

#### Peripheral\_Settings

This option displays the Peripheral Configuration Menu.

**Operating\_System**

This option displays the Operating System Parameters Menu.

**Disk\_Maintenance**

This option displays the Disk Maintenance Utilities Menu.

**Disk\_Partitioning**

This option displays the Disk Partitioning Menu.

**Device\_image\_CoPy**

This option displays the Device Image Copy Utility menu.

**REBUILD**

This option displays the Rebuild Utility menu.

**10.2.3 Date and Time Menu**

The following Date and Time Menu displays when the **DATE** option is selected from the Main Utility Menu. The current date and time is shown below the menu title.

---

Date and Time Menu

---

Mon June 5, 10:52:55 1989

|                  |                           |
|------------------|---------------------------|
| MOnth            | (JUNe)                    |
| DAy              | (5)                       |
| Time_Of_Day      | (10:52:55)                |
| TimeZone         | (CENTral GMT-6)           |
| Daylight_Savings | (Yes)                     |
| YeaR             | (1989)                    |
| UTility          | [go to main utility menu] |

Standard Options:

|      |      |      |       |      |
|------|------|------|-------|------|
| UNix | SAve | LOad | REset | Help |
|------|------|------|-------|------|

Option:

The following briefly defines the Date and Time Menu options.

**Month**

This option sets the month when keyed in with the following format, where **<month>** is the name of the month. The minimal value for **<month>** is the first three letters of the month name (for example, JUN for June).

Option: MO **<month>**

**DAy**

This option sets the date when keyed in with the following format:

Option: DA **<date>**

**Time\_Of\_Day**

This option sets the time when keyed in with the following format:

Option: TOD **<hour> <minute> <second>**

**TimeZone**

This option sets the correct timezone for your geographic location when keyed in with the following format, where **<zone>** is the timezone name. **<zone>** can be the abbreviation (for example, EAST for Eastern) of the timezone name with the syntax for each name given in the following table:

Option: TZ **<zone>**

| <i>Zone Name</i> | <i>Time</i> |
|------------------|-------------|
| ATlantic         | GMT - 4     |
| EASTern          | GMT - 5     |
| CENtral          | GMT - 6     |
| MouNTain         | GMT - 7     |
| PACific          | GMT - 8     |
| YUKon            | GMT - 9     |
| ALAska           | GMT - 10    |
| BERing           | GMT - 11    |
| NEW_Zealand      | GMT - 12    |
| NEW_Caledonia    | GMT + 11    |
| SYDney           | GMT + 10    |
| TOKyo            | GMT + 9     |
| MANila           | GMT + 8     |
| BANkok           | GMT + 7     |
| LHAsa            | GMT + 6     |
| KARachi          | GMT + 5     |

| <i>Zone Name</i> | <i>Time</i> |
|------------------|-------------|
| MAUritus         | GMT + 4     |
| MOScow           | GMT + 3     |
| CAIro            | GMT + 2     |
| PARis            | GMT + 1     |
| LONdon           | GMT         |
| REYkjavik        | GMT - 1     |
| AZOres           | GMT - 2     |
| GREENland        | GMT - 3     |

**Daylight\_Savings**

This option allows you to specify whether daylight savings time is currently in effect when keyed in with the following format, where **<Yes/No>** is Yes to specify daylight savings time and No to specify standard time.

Option: DS **<Yes/No>**

**YeaR**

This option allows you to specify and display the correct year when keyed in with the following format, where **<year>** is the current year (for example, 1989).

Option: YR **<year>**

**UTility**

This option displays the Main Utility Menu.

### 10.2.4 Peripheral Configuration Menu

The following Peripheral Configuration Menu is used to establish settings for the RS-232 port. The Peripheral Configuration Menu displays when the `Peripheral_Settings` option is selected from the Main Utility Menu.

Peripheral Configuration Menu

```

IOP (0)
AUX_select (0)
Port Type (rs-232)
Baud_Rate (9600)
Word_Size (8)
Stop_Bits (1)
Parity (None)
auto_XON/xoff (Yes)
incoming_XOFF (Yes)
auto_RTS/cts (Yes)
Slave_Device (No)
Utility [go to main utility menu]

```

Standard Options:

UNix      SAve      LOad      REset      Help

Option:

The following briefly defines the Peripheral Configuration Menu options.

**IOP <number>**

This option sets the IOP number for which parameters are established.

**AUX\_select <port>**

This option sets the RS-232 auxiliary port, the port for which all other RS-232 parameters are established, where <port> is one of the following:

```

0 for aux00
1 for aux01
2 for aux02

```

**Port Type <type>**

This option sets the port type, where <type> is the type of port to be configured. Only the RS-232 setting is available to InterServe processors.

**Baud\_Rate <rate>**

This option sets the baud rate, where <rate> is the number of bits per second (bps) transmitted over a serial communication channel. This number is typically about ten times greater than the number of characters transmitted per second.

The baud rate depends on the device connected to the port. The default setting on InterServe processors is 9600.

The following baud rates are available:

|           |          |           |
|-----------|----------|-----------|
| 50 bps    | 300 bps  | 2400 bps  |
| 75 bps    | 600 bps  | 3600 bps  |
| 110 bps   | 1200 bps | 4800 bps  |
| 134.5 bps | 1800 bps | 7200 bps  |
| 150 bps   | 2000 bps | 9600 bps  |
|           |          | 19200 bps |

**Word\_Size <size>**

This option sets the word size, where <size> is the number of bits that make up a word. Typical values are 5, 6, 7, and 8. Since the ASCII character set uses 7-bit characters, word sizes of 7 and 8 are most common. Word size is usually dependent on the device. The default setting is 8 bits.

**Stop\_Bits <bits>**

This option sets an asynchronous indicator that the bits to follow are a valid word, where <bits> is the number of stop bits. Stop bits are necessary to indicate the beginning of a transmitted word since the sender and receiver of data are not synchronized. Settings of 1 and 2 are available. The default setting is 1 stop bit.

**Parity <parity>**

This option specifies whether transmitted data will be verified and, if so, the method by which the transmitted data is verified, where <parity> is one of the following:

**None** (default) does not check or generate parity.

**Even** checks and sets the parity to **Even** for all outgoing data. Even parity indicates that each transmitted word will contain an even number of bits in a logic 1 state.

**Odd** checks and sets the parity to **Odd** for all outgoing data. Odd parity indicates that each transmitted word will contain an odd number of bits in a logic 1 state.

A special bit referred to as the **parity** bit is either set or cleared upon transmission to meet the parity requirement. When a word is received, if the parity does not check correctly, the data was most likely corrupted during transmission.

**auto\_XON/xoff <Yes/No>**

This option activates the Auto XON software protocol, used for flow control of serially transmitted data between two peripherals. <Yes/No> is Yes (default) to activate and No to deactivate the protocol.

The Auto XON protocol provides that, as data is transmitted to a peripheral that cannot process the data as fast as it is sent, that peripheral can send a stop character when input buffers are nearly full to stop the transmission temporarily. As the input buffers become clear again, a START character is sent to resume the transmission. Use of this protocol ensures that data will not be lost during transmission.

**incoming\_XOFF <Yes/No>**

This option activates a software protocol used by the transmitter to ensure that all START and STOP characters will be honored according to protocol. <Yes/No> is Yes (default) to activate the protocol and No to deactivate the protocol.

**auto\_RTS/cts <Yes/No>**

This option activates RTS/CTS, a protocol used for flow control of serially transmitted data. <Yes/No> is Yes (default) to activate the protocol and No to deactivate the protocol.

Instead of transmitting characters for flow control, electrical signals perform the handshaking. The Ready-to-Send (RTS) signal is generated by one peripheral when the peripheral is ready to receive a character. This signal is interpreted as input to the Clear-To-Send (CTS) input of another peripheral. Likewise, RTS from the second peripheral is CTS to the first.

**Slave\_Device <Yes/No>**

This option indicates whether a device is a slave device, which can accept input from a host but not send data back. <Yes/No> is Yes to configure the device as a slave device and No to indicate that it is not a slave device.

**Utility**

This option displays the Main Utility Menu.

### 10.2.5 Operating System Parameters Menu

The following Operating System Parameters Menu is displayed when the Main Utility menu `Operating_System` option is selected.

---

Operating System Parameters Menu

---

Single\_User(No)  
Network\_Address           08-00-36-95-6A-00  
Node\_Name                 (jdk\_pn)  
Kernel File Name         (unix)  
UTility                   [go to main utility menu]

Standard Options:

UNix      SAve      LOad      REset      Help

Option:

The following briefly defines the Operating System Parameters Menu options.

#### Single\_User <Yes/No>

This option determines whether the InterServe is booted into single-user or multiuser mode. <Yes/No> is Yes for single-user mode and No (default) for multiuser mode. When you boot into single-user mode, you are automatically logged in to the *root* account, and the only mounted file system is *root*. The */usr* file system must be mounted before you can access it.

#### Network\_Address

This option displays the assigned network address of your InterServe.

#### Node\_Name <name>

This option can be used to assign a nodename to the InterServe, where <name> is a unique name that allows other nodes on the network to identify the InterServe. The nodename can be a maximum of eight alphabetic characters, none of which may be uppercase.

#### UTility

This option displays the Main Utility Menu.

### 10.2.6 Disk Maintenance Utilities Menu

The following Disk Maintenance Utilities Menu displays when the Main Utility Menu **Disk\_Maintenance** option is selected.

---

Disk Maintenance Utilities Menu

---

|                      |                             |
|----------------------|-----------------------------|
| IOP                  | (0)                         |
| SCSI_id              | (0)                         |
| LUN                  | (0)                         |
| Disk Capacity        | (304605)                    |
| Part_Number          | (FDSK155XX)                 |
| Serial_Number        | (03041541)                  |
| Flaw Count           | (6)                         |
| Format_With_Flaws    | (Yes)                       |
| Format_With_Verify   | (Yes)                       |
| Verify_Data_Patterns | (0xC393, 0xB333, 0x0)       |
| Add_Flaw             | [add additional flaw data]  |
| Remove_Flaw          | [remove previous flaw data] |
| List_Flaws           | [display flaw data]         |
| VeriFY               | [verify device flaw count]  |
| FORMAT               | [format the device]         |
| PARK                 | [retract disk r/w heads]    |
| UTility              | [go to main utility menu]   |

Standard Options:

UNix      SAve      LOad      REset      Help

Option:

The following briefly defines the Disk Maintenance Utilities Menu options.

**IOP <number>**

This option sets the IOP number to <number>.

**SCSI\_id <number>**

This option allows you to set the SCSI ID. The SCSI ID is used to identify or address a hard disk connected to the InterServe, where <number> identifies the hard disk controller. If only one hard disk is connected to the InterServe, the SCSI ID is 0.

**LUN <number>**

This option allows you to set the Logical Unit Number, which (along with the SCSI ID) is used to identify or address a hard disk connected to the InterServe, where <number> identifies the hard disk. If only one hard disk is connected to the InterServe, the Logical Unit Number is 0.

**Disk\_Capacity**

This option displays the number of formatted blocks on the hard disk.

**Part\_Number <part number>**

This option sets the part number of the disk, where <part number> identifies the type of hard disk addressed. The available part numbers are as follows:

- MESA518xx
- FDSK131xx
- FDSK132xx
- FDSK135xx
- FDSK126xx
- FDSK150xx
- FDSK155xx
- FDSK156xx
- FDSK211xx
- FDSK223xx
- FDSK226xx
- FDSK229xx
- FDSK230xx
- User-defined
- Foreign

**Note:**

If Foreign displays in this field when you enter this Utility menu, set the part number appropriate for your hard disk. The correct part number for your disk is given on the Hard Disk Profile Sheet that came with your unit.

**Serial\_Number <number>**

This option sets the serial number for the hard disk, where <number> is a unique number given to each hard disk. Check the Hard Disk Profile Sheet that came with the unit to see if the number currently displayed is the serial number of the addressed hard disk. If it is not, be sure to set the correct number.

**Flaw Count**

This option displays the current length of the flaw data list for the addressed hard disk. (See List\_Flaws.)

**Format\_With\_Flaws <Yes/No>**

This option determines whether the data in the flaw data list, displayed by the **List\_Flaws** option, is considered when the hard disk is formatted. <Yes/No> is **Yes** (default) to mark the flaws on the hard disk and retain the flaw list and **No** to disregard the flaw data.

**Format\_With\_Verify <Yes/No>**

This option determines whether additional flaws on the hard disk are checked for during hard disk formatting. <Yes/No> is **Yes** (default) to perform the verification procedure and **No** to format the hard disk without verification. The verification procedure writes data to and reads it from the hard disk to check for additional bad blocks.

**Verify\_Data\_Patterns**

This option specifies that the data patterns displayed are written to and then read from the disk during the verification portion of a format. This process ensures that each disk block is reliable.

**Add\_Flaw <cylinder> <head> <offset>**

OR

**Add\_Flaw <logical block>**

This option adds a flaw location to the flaw data list, displayed by the **List\_Flaws** option. The following are options for this menu selection:

<cylinder> <head> <offset> is the physical disk location of the flaw.

<logical block> is the address of the flawed area on the disk.

See Chapter 11, "Rebuilding and Repartitioning the Hard Disk," for more information on these flaw data formats.

**Remove\_Flaw <cylinder> <head> <offset>**

This option removes a flaw location from the flaw data list, displayed by the **List\_Flaws** option. <cylinder> <head> <offset> is the physical disk location of the flaw. See Chapter 11, "Rebuilding and Repartitioning the Hard Disk," for more information on flaw data formats.

**List\_Flaws**

This option displays a list of flaw data in physical disk location format.

**VeriFY**

This option verifies the addressed disk nondestructively, adding any flaws to the flaw data list displayed by the **List\_Flaws** option.

**FORMAT**

This option formats the currently addressed hard disk.

**Caution:**

All information on the hard disk is lost when it is formatted. Be sure to back up any needed files before selecting this option. Refer to Chapter 11, "Rebuilding and Repartitioning the Hard Disk," for help in determining if the hard disk needs to be formatted.

**PARK**

This option retracts the disk read/write heads.

**Utility**

This option displays the Main Utility Menu.

**10.2.7 Disk Partitioning Menu**

The following Disk Partitioning Menu displays when the **Disk\_Partitioning** option is selected from the Main Utility Menu.

Disk Partitioning Menu

IOP (0)  
SCSI\_id (0)  
Logical\_Unit\_Number (0)  
Disk Capacity (304605)  
List\_Partitions  
Add\_SYStem  
ADD\_Partition  
ReMove\_Partition  
Utility [go to main utility menu]

Standard Options:

UNix      SAve      LOad      REset      Help

Option:

The following briefly defines the Disk Partitioning Menu options.

**IOP <number>**

This option sets the IOP number to <number>.

**SCSI\_id <number>**

This option sets the SCSI ID, which is used to identify or address a hard disk connected to the InterServe. <number> is a number that identifies the hard disk controller. If only one hard disk is connected to the InterServe, the SCSI ID is 0.

**Logical\_Unit\_Number <number>**

This option sets the Logical Unit Number, which (along with the SCSI ID) is used to identify or address a hard disk connected to the InterServe. <number> identifies the hard disk. If only one hard disk is connected to the InterServe, the Logical Unit Number is 0.

**Disk Capacity**

This option displays the number of formatted blocks on the hard disk.

**List Partitions**

This option lists the names and sizes of the hard disk partitions.

**Add\_SYStem**

This option reserves the first 4,000 blocks of hard disk space for the bootable partitions and images created when the hard disk is rebuilt. (See Chapter 11, "Rebuilding and Repartitioning the Hard Disk.")

**ADD\_Partition <partition number> <modifier number> <size>**

This option adds the partition specified with the size indicated to the hard disk. The following arguments must be included:

<partition number> is the number associated with the general partition division. When adding partitions, use the general-purpose numbering convention established by Intergraph. (See Chapter 11, "Rebuilding and Repartitioning the Hard Disk.")

<modifier number> designates a subdivision of the general partition division. When adding the following partitions, you must use the following partition and modifier numbers:

| Partition | Partition Number | Modifier Number |
|-----------|------------------|-----------------|
| root      | 7                | 0               |
| swap      | 7                | 1               |
| usr       | 7                | 3               |
| usr2      | 7                | 4               |

<size> is the size, in blocks, of the partition to be added.

**ReMove\_Partition** <partition number> <modifier number> <size>

This option removes the specified partition. <partition number>, <modifier number>, and <size> are the partition specifications given in the list of existing partitions displayed by the **List\_Partitions** option.

**UTility**

This option displays the Main Utility Menu.

### 10.2.8 Device Image Copy Utility Menu

The Device Image Copy Utility Menu displays when the **Device\_image\_CoPy** option is selected from the Main Utility Menu. This menu allows you to copy portions of a partition or entire partitions on one device to another partition on the same or different device. Partitions can be copied between hard disks, floppy disks, and tapes. Note that tapes do not have partitions. However, options for skipping to the end of a file or to the end of the tape are provided so that specific areas of the tape can be read from or written to.

The data fields displayed on the menu are dependent on the devices selected with the following options.

**Src\_DEVICE\_type** <type> and **Dest\_DEVICE\_type** <type>

These options set the device that will be read from or written to, where <type> is one of the following:

**Hard\_Disk** to specify the hard disk

**Floppy** to specify a floppy disk

**Tape** to specify a tape

The data fields provided for each of these devices are shown in the following illustrations.

*Hard Disk:*

Device Image Copy Utility

from:  
—

Src\_IOP (0)  
 Src\_DEVICE\_type (Hard\_Disk)  
 Src\_SCSI\_id (0)  
 Src\_LUN (0)  
 Src\_Partition (0)  
 Src\_Modifier (0)  
 Src\_Offset (0)

to:  
—

Dest\_IOP (0)  
 Dest\_DEVICE\_type (Hard\_Disk)  
 Dest\_SCSI\_id (0)  
 Dest\_LUN (0)  
 Dest\_Partition (0)  
 Dest\_Modifier (0)  
 Dest\_Offset (0)

Entire\_Partition  
 Block\_Count  
 CoMPare  
 COPY  
 UTility

(Yes)  
 (0)  
 [compare the devices]  
 [copy source to destination]  
 [go to main utility menu]

Standard Options:

UNix      SAve      LOad      REset      Help

Option:

*Floppy Disk:*

Device Image Copy Utility

| <u>from:</u>     |          | <u>to:</u>                   |          |
|------------------|----------|------------------------------|----------|
| Src_DEVICE_type  | (Floppy) | Dest_DEVICE_type             | (Floppy) |
| Src_Partition    | (0)      | Dest_Partition               | (0)      |
| Src_Modifier     | (0)      | Dest_Modifier                | (0)      |
| Src_Offset       | (0)      | Dest_Offset                  | (0)      |
| Entire_Partition |          | (Yes)                        |          |
| Block_Count      |          | (0)                          |          |
| CoMPare          |          | [compare the devices]        |          |
| COPY             |          | [copy source to destination] |          |
| UTility          |          | [go to main utility menu]    |          |

Standard Options:

UNix      SAve      LOad      REset      Help

Option:

**Cartridge Tape:**

Device Image Copy Utility

|                    |        |                              |        |
|--------------------|--------|------------------------------|--------|
| <u>from:</u>       |        | <u>to:</u>                   |        |
| Src_IOP            | (0)    | Dest_IOP                     | (0)    |
| Src_DEVice_type    | (Tape) | Dest_DEVice_type             | (Tape) |
| Src_SCSI_id        | (0)    | Dest_SCSI_id                 | (0)    |
| Src_Skip_File      |        | Dest_Skip_File               |        |
| Src_skip_EOT       |        | Dest_skip_EOT                |        |
| Src_REWInd         |        | Dest_REWInd                  |        |
| Src_Retension_Tape |        | Dest_Retension_Tape          |        |
| Src_Offset         | (0)    | Dest_Offset                  | (0)    |
| Entire_Partition   |        | (Yes)                        |        |
| Block_Count        |        | (0)                          |        |
| CoMPare            |        | [compare the devices]        |        |
| COPY               |        | [copy source to destination] |        |
| UTility            |        | [go to main utility menu]    |        |

Standard Options:

UNix      SAve      LOad      REset      Help

Option:

**Src\_IOP <number>**

This option specifies the IOP number of the device that will be copied from.

**Dest\_IOP <number>**

This option specifies the IOP number of the device that will be copied to.

**Src\_SCSI\_id <id>**

This option specifies the SCSI ID of the device that will be copied from, where <id> is the ID number. This option is not available for floppy disks.

**Dest\_SCSI\_id <id>**

This option specifies the SCSI ID of the device that will be copied to, where <id> is the ID number. This option is not available for floppy disks.

**Src\_LUN <number>**

This option specifies the Logical Unit Number of the device that will be copied from, where <number> is the Logical Unit Number.

**Dest\_LUN <number>**

This option specifies the Logical Unit Number of the device that will be copied to, where <number> is the Logical Unit Number.

**Src\_Partition <partition>**

This option specifies the partition number that will be copied from, where <partition> is the partition number. This option is not available for cartridge tapes.

**Dest\_Partition <partition>**

This option specifies the partition number that will be copied to, where <partition> is the partition number. This option is not available for cartridge tapes.

**Src\_Modifier <modifier>**

This option specifies the modifier number of the partition that will be copied from, where <modifier> is the modifier number. This option is not available for cartridge tapes.

**Dest\_Modifier <modifier>**

This option specifies the modifier number of the partition that will be copied to, where <modifier> is the modifier number. This option is not available for cartridge tapes.

**Src\_Offset <offset>**

This option specifies the specific block number where copying will begin, where <offset> is the block number. For hard disks, this number should be at least 5 so it will bypass hard disk flaw data.

**Dest\_Offset <offset>**

This option specifies the specific block number on the destination device from which the copy will begin, where <offset> is the block number.

**Entire\_Partition <Yes/No>**

This option specifies whether to copy an entire partition. <Yes/No> is Yes to copy an entire partition and No (default) to copy a portion of one.

**Block\_Count <count>**

This option specifies the total number of blocks to be transferred, where <count> is the number of blocks. If the number of blocks specified to be transferred exceeds the capacity of the receiving partition, only the number of blocks that the partition can hold will be transferred.

**CoMPare**

This option initiates a comparison of the data specified in the "From" descriptors to the media specified in the "To" descriptors.

### **COPY**

This option copies the data specified in the "From" descriptors to the media specified in the "To" descriptors. The size of the transfer will either be the entire size of the smaller partition or the size specified in the **Block\_Count** field as long as no partition boundaries are crossed.

### **Utility**

This option displays the Main Utility Menu.

The following options are used only for cartridge tape.

### **Src\_Skip\_File**

This option initiates a forward skip to the next file on the tape at the specified SCSI address.

### **Dest\_Skip\_File**

This option initiates a forward skip to the next file on the tape at the specified SCSI address.

### **Src\_skip\_EOT**

This option initiates a forward skip to the end of the tape on the source device at the specified SCSI address.

### **Dest\_skip\_EOT**

This option initiates a forward skip to the end of the tape on the destination device at the specified SCSI address.

### **Src\_REWind**

This option rewinds the tape on the source device at the specified SCSI address.

### **Dest\_REWind**

This option rewinds the tape on the destination device at the specified SCSI address.

### **Src\_Retension\_Tape**

This option retensions the tape on the source device at the specified SCSI address.

### **Dest\_Retension\_Tape**

This option retensions the tape on the source device at the specified SCSI address.

## 10.2.9 Rebuild Utility Menu

The Rebuild Utility menu assists you in rebuilding the hard disk. From the Rebuild Utility menu, you will create acceptable partitions (using the default partitions or defining your own), load the Rebuild media on the hard disk, and enter the Rebuild environment (System V single-user mode). In single-user mode, the Rebuild options menu provides the mechanism for restoring the file systems. For step-by-step instructions on rebuilding, refer to Chapter 11, "Rebuilding and Repartitioning the Hard Disk."

---

### Rebuild Utility

---

The current partition table is acceptable:

```
ROOT area: 25000
SWAP area: 10000
USR area: 100000
```

```
Do you wish to re-make the partition table (y/n)? [n]:
Do you wish to load from the floppy, microfloppy, tape or
cdrom (f/m/t/c)? [f]:
Do you wish to enter the Rebuild environment (y/n)? [y]:
```

---

The Rebuild Utility Page is designed to lead you through the process of rebuilding the hard disk. Notice that it leads you through three separate steps (defining partitions, loading media, and entering the Rebuild environment). You will not be able to proceed to the next step until you have satisfactorily completed the previous step.

The following sections describe the options on the Rebuild Utility menu:

### *Create Partition Table*

Step 1 requires you to define acceptable hard disk partitions. (This documentation will use the terms *partition table* and *partitions* synonymously.) If your partitions are acceptable, you may proceed to the next step.

If your current partitions are not acceptable for the Rebuild software, you will not be allowed to proceed to step 2 until you redefine them. You can redefine the partition table using Intergraph-defined default partitions, or you can define your own *custom* partitions. If you choose to define your own custom partitions, you will be sent to the Partition Utility menu. For more information on repartitioning, refer to Chapter 11, "Rebuilding and Repartitioning the Hard Disk."

### *Load Rebuild Media*

Step 2 on the Rebuild Utility menu requires you to load the Rebuild media on the hard disk. You cannot begin this step until partitions acceptable for the Rebuild software have been defined.

You will be prompted for the Rebuild media type (such as floppy or microfloppy). Then, you will be prompted to enter each Rebuild Root floppy (or other media type). You are not required to insert Rebuild floppies in any particular order. The only requirement is that all five Rebuild Root floppies are loaded.

### *Enter Rebuild Environment*

Step 3 on the Rebuild Utility Page requires you to enter the Rebuild environment. You cannot enter the Rebuild environment until you have loaded the Rebuild media on the hard disk.

When you respond to the prompt to enter the Rebuild environment, the system will boot into System V single-user mode and initiate the Rebuild options menu. From this menu, you will be able to restore the file systems. For more information on rebuilding, refer to Chapter 11, "Rebuilding and Repartitioning the Hard Disk."







# Chapter 11: Rebuilding and Repartitioning the Hard Disk

---

This chapter contains the following sections:

|          |                                                       |       |
|----------|-------------------------------------------------------|-------|
| 11.1     | Backing Up Files                                      | 11-3  |
| 11.2     | Creating Current Rebuild Floppy Disks                 | 11-4  |
| 11.3     | Booting from the Rebuild Boot Floppy Disk             | 11-7  |
| 11.4     | Entering the Utility Pages                            | 11-8  |
| 11.4.1   | Entering the Utility Pages on a Workstation           | 11-8  |
| 11.4.2   | Entering the Utility Pages on a Server                | 11-9  |
| 11.5     | Verifying Hard Disk Flaws                             | 11-10 |
| 11.5.1   | Verifying a Workstation Hard Disk                     | 11-11 |
| 11.5.2   | Verifying a Server Hard Disk                          | 11-12 |
| 11.6     | Formatting the Hard Disk                              | 11-13 |
| 11.6.1   | Formatting Workstation Hard Disks                     | 11-13 |
| 11.6.2   | Formatting Server Hard Disks                          | 11-14 |
| 11.7     | Selecting the Rebuild Utility Page                    | 11-16 |
| 11.8     | Repartitioning the Hard Disk                          | 11-17 |
| 11.8.1   | Using the Current Partition Table                     | 11-18 |
| 11.8.2   | Repartitioning Using Default Partitions               | 11-19 |
| 11.8.3   | Repartitioning Using Custom Partitions                | 11-20 |
| 11.8.3.1 | Partitioning Overview                                 | 11-20 |
|          | Determining Partition Sizes                           | 11-20 |
|          | Reallocating Space Between Partitions                 | 11-23 |
|          | Understanding Partition Names                         | 11-25 |
| 11.8.3.2 | Creating Nonstandard Partitions                       | 11-28 |
|          | Repartitioning Workstation Hard Disks                 | 11-28 |
|          | Repartitioning an InterServe Hard Disk                | 11-29 |
| 11.9     | Loading the Rebuild Root Media                        | 11-32 |
| 11.10    | Entering the Rebuild Environment                      | 11-33 |
| 11.11    | Restoring the File Systems                            | 11-34 |
| 11.12    | Loading Nucleus Software with newprod                 | 11-42 |
| 11.13    | Preparing Nonstandard Partitions for Use              | 11-52 |
| 11.14    | Loading Application Software with the newprod Utility | 11-53 |
| 11.15    | Restoring Files from Backups                          | 11-54 |

## 11-2 *Rebuilding and Repartitioning the Hard Disk*

---

Rebuilding the hard disk involves (at the minimum) recreating the file system(s). Existing data on a file system is overwritten; thus, rebuilding can be a destructive process. You must rebuild the hard disk under any of the following circumstances. Unless one of these circumstances is true, do not rebuild the hard disk.

- The hard disk is corrupted and you cannot boot your workstation.
- You cannot boot your workstation to single-user mode. (You never reach the blue introductory screen.)
- Bad block messages such as the following appear on the screen:  
  
Disk failed: s0uOp7.1  
read error at block 2933
- You want to change file system types. (For example, you want to convert */usr* from a standard file system to a Fast File System.)
- You want to alter the partition sizes on the hard disk.

This chapter describes how to rebuild the hard disk. Intergraph recommends that you complete the following steps in order to rebuild the hard disk. These steps will ensure that you lose as little data as possible when you rebuild. Each step is described in detail throughout the rest of this chapter.

1. Back up all personal files.
2. Create Rebuild floppy disks with the current software.
3. Boot from the Rebuild Boot floppy (#1)
4. Enter the Utility Pages.
5. Verify hard disk flaws.
6. Format the hard disk (optional).
7. Select the Rebuild Utility Page icon.
8. Repartition the hard disk (optional).
9. Load the Rebuild Root media.
10. Enter the Rebuild environment.
11. Restore the file systems.
12. Load nucleus software with the *newprod* utility.
13. Prepare nonstandard partitions for use.
14. Load application software with the *newprod* utility.
15. Restore files from backups.

## 11.1 Backing Up Files

Before you rebuild the hard disk you should first back up all personal files, including system files (such as */etc/passwd*, */etc/group*, and any *cron* tables), that are unique to the machine you are rebuilding. The rebuild process recreates the file systems on the hard disk and, in the process, overwrites all data on the hard disk. After the rebuild, you will not be able to access any files in the file systems that were recreated. Thus, before you begin rebuilding, back up all personal files that you want to keep.

For procedures for backing up files, refer to Chapter 9, "Backing Up and Restoring Files."

After you back up all personal files that you want to keep, continue to 11.2, "Creating Current Rebuild Floppy Disks."

## 11.2 Creating Current Rebuild Floppy Disks

After you have backed up your personal files, you should create or locate a current set of Rebuild floppy disks. The Rebuild floppy disk set is used to rebuild a corrupted System V hard disk to a bootable state.

New customers receive a set of Rebuild floppy disks with each workstation/server. Each Rebuild floppy disk set is specific to the machine it was shipped with. CLIPPER Rebuild floppy sets contain these diskettes:

- Rebuild Boot (floppy #1)
- Rebuild Root (floppy #2)
- Rebuild Root (floppy #3)

To rebuild, you will use the Rebuild Boot floppy disk appropriate for the machine you are rebuilding to boot and the Rebuild Root floppies to load the rebuild utility.

If your Rebuild floppy disk set is no longer current, you must create a current set. This section provides procedures for creating Rebuild floppy disks. You need three formatted, high-density floppy disks to create a Rebuild floppy disk set.

**Note:**

Do not attempt to make Rebuild floppy disks on the machine that you are rebuilding. Instead, use another machine that is completely operational.

Follow these steps to create a CLIPPER Rebuild floppy disk set:

1. You must be in superuser mode to perform this procedure. Log in and key in `su` at the system prompt as follows:

```
login: username
$ su
#
```

2. Invoke the `newprod` utility to load the rebuild product (ss01003) by keying in the following:

```
newprod ss01003
```

3. Key in the correct entry at the following prompt:

Enter source of installation: n)etwork, f)loppy, t)ape, r)emote cdrom, or local c)drom →

If you key in n for network or r for remote cdrom, the following prompt appears:

Enter network connect nodename or address (08-00-36-XX-XX-XX) →

Key in the network address or the nodename of the delivery node where the new products reside and a valid username and password.

4. Messages similar to the following appear. Follow the instructions as prompted on the screen. To create a complete CLIPPER Rebuild floppy set, select "both" at the first prompt.

---

Installing: REBUILD (ss01003)  
Installing...

Available Rebuild floppy sets:

---

|                              |                                          |
|------------------------------|------------------------------------------|
| 1) 125/32C/220/225 Systems   | (80186 IOP and ROP Graphics)             |
| 2) 200/240 Systems           | (80186 IOP and optional IFB Graphics)    |
| 3) 300/400/3000/4000 Systems | (80386 IOP and optional IFB Graphics)    |
| 4) 2000 Systems              | (Clipper IOP and optional MMG Graphics)  |
| 5) 6000 Systems              | (Clipper IOP and optional EDGE Graphics) |
| 6) None                      |                                          |

---

Which type of rebuild floppies do you want to make (1-6)? 1

Do you want to make a "root" set, a "boot" floppy, both or none? [both]

Downloading the Clipper Rebuild #1 (125/32C/200 Boot) floppy.  
Insert a formatted floppy into the floppy drive and press <RETURN> →

You have a good "125/32C/200 Boot" floppy!  
Label this floppy "Clipper Rebuild #1 (125/32C/200 Boot)"

The "125/32C/220/225 Root" floppy set will require 2 floppies.

Do you want to make Rebuild floppy #2 (y/n)? [y]  
Do you want to make Rebuild floppy #3 (y/n)? [y]

Insert rebuild root floppy #2 and press <RETURN>  
Writing the floppy  
Verifying the floppy  
Rebuild root floppy #2 created successfully

Insert rebuild root floppy #3 and press <RETURN>  
Writing the floppy  
Verifying the floppy  
Rebuild root floppy #3 created successfully

Cleaning up...  
Successful installation: REBUILD (ss01003)  
Product installed in the /usr/ip32/rebuild directory

---

5. After these messages appear, press <Return> to return to the newprod menu. Key in q to exit the newprod utility.
6. Continue to 11.3, "Booting from the Rebuild Boot Floppy Disk."

### **11.3 Booting from the Rebuild Boot Floppy Disk**

After you have created a set of Rebuild floppy disks with the current software, you must boot the workstation from the Rebuild Boot floppy disk. To rebuild the workstation, you may use the set of Rebuild floppy disks delivered with your workstation/server (if the set is not current) or you can use the Rebuild software product to create a set. The previous section describes how to create a set of Rebuild floppies.

To boot from the Rebuild Boot floppy, insert the Rebuild Boot floppy (appropriate for the machine that you will rebuild) in the floppy disk drive and boot the workstation/server. The machine will boot from the Rebuild floppy rather than from the hard disk.

Continue to 11.4, "Entering the Utility Pages."

## 11.4 Entering the Utility Pages

After you have booted from the Rebuild Boot floppy disk, you must enter the Utility Pages. You will use the Utility Pages to verify hard disk flaws, format the hard disk (if desired), repartition the hard disk (if desired), and load the rebuild media. After you have entered the Utility Pages, proceed to 11.5, "Verifying Hard Disk Flaws."

Because the interface is different for workstations and servers, this section separates the procedures for workstations and servers.

### 11.4.1 Entering the Utility Pages on a Workstation

When an Intergraph workstation boots from the Rebuild floppy disk, a blue introductory screen appears. You must move the mouse or cursor within five seconds if you wish to enter the Utility Pages. From the introductory screen, select the **Utility** icon to enter the Utility Pages. If the Utility Pages have been assigned a password, you will be prompted to enter the Utility Page password.

**Note:**

If the Utility Pages do not have a password, you may wish to assign one for security purposes. (Only the system manager should be allowed to access the Utility Pages because this environment allows you to format and repartition the hard disk.) To assign a password to the Utility Pages, select the **Workstation Password** icon on the Main Utility Page and key in the desired password when prompted.

The Utility Pages are a series of screens that allow you to configure the system. At the bottom of each Utility Page screen, notice the **Help** icon. To use the **Help** function, place the cursor on the item for which you want information and tap the reset button.

## **11.4.2 Entering the Utility Pages on a Server**

When an Intergraph server boots from the Rebuild floppy disk, an introductory screen appears. You must press any key within five seconds to enter the Startup Utility Pages. From this screen, key in UT to enter the Utility Pages. If the Utility Pages have been assigned a password, you will be prompted to enter the Utility Page password.

**Note:**

If the Utility Pages do not have a password, you may wish to assign one for security purposes. (Only the system manager should be allowed to access the Utility Pages because this environment allows you to format and repartition the hard disk.) To assign a password to the Utility Pages, key in PW at the Main Utility Page and key in the desired password when prompted.

The Utility Pages are a series of screens that allow you to configure the system. At the bottom of each Utility Page screen, notice the Help option. To use the Help function, key in h followed by the string required to select the option. For example, to read Help information on the Disk Partitioning screen, key in h dp. Notice that the Help function is not case-sensitive.

## 11.5 Verifying Hard Disk Flaws

From the Utility Pages, you will be able to verify the locations of hard disk flaws by running the Verify procedure. This procedure reads the hard disk and records all flaws to ensure that data is not written over flawed areas. It does not erase or change data on the disk.

If you are rebuilding because you wish to repartition the hard disk or convert to a different file system type, you do not need to run the Verify procedure. However, if you are rebuilding the hard disk to correct bad block messages or to correct a corrupted disk, you must run the Verify procedure.

The following is an example of a bad block message:

```
Disk failed: s0uOp07.1 medium error:
read error at block 2933
```

Bad blocks are flawed areas on the hard disk. As the disk drive writes data on the disk, it calculates an Error Correction Code (ECC) for each block and stores the code at the end of each block. When the drive reads data from the disk, it calculates another ECC and compares it to the original stored value. If the codes match, the data stored in that block is healthy and uncorrupted. If the codes do not match, the drive generates an ECC error and the host displays a medium error or a bad block message.

Most Intergraph disks have Automatic Read and Write Reallocation. This means that if the system finds an error that it cannot recover from (save the data, move it to a new location, and experience no data loss), the block will be automatically reallocated to another sector of the disk and the original data will be rewritten there (with no data loss).

However, if the system finds an unrecoverable data error while executing a read or write operation, it will report a medium error. If you receive one or more medium errors, run the Verify procedure on the disk to register and reallocate any other bad blocks that may have developed.

The Verify procedure attempts to read every location on the disk. It checks the data by comparing a computed value of ECC with a value that was computed when the data was written and stored on the disk at the end of each track. If the ECC code does not match, the procedure reads the disk again. One retry is categorized as a soft error. Two or more retries indicate a hard error. Hard errors are flagged as bad blocks. If the procedure finds a bad block, the host reassigns the bad block to a spare location on the disk.

After all bad blocks are found, the disk is clean and ready to go. A bad block indicates that a block of data has an error. As a result, the data in this block may not be recoverable.

Procedures differ for workstations and servers and are described in separate sections.

### **11.5.1 Verifying a Workstation Hard Disk**

If the Verify procedure fails, you will need to reformat your disk, as described in 11.6, "Formatting the Hard Disk." Follow these steps to Verify the hard disk:

1. From the Main Utility Page, select the Disk Maintenance Utility screen button.
2. Select the Next Page screen button to access the Hard Disk Flaw Data Utility Page.
3. Select the Verify screen button to access the Verify page.
4. From the Verify page, check to see that the SCSI ID, Disk Capacity, and Logical Unit Number fields contain the correct settings for the disk you want to verify.

The SCSI ID and LUN for internal hard disks are 0. The disk capacity depends on the internal hard disk size.

5. Select the Verify screen button to begin the Verify process.
6. If the hard disk is corrupted with numerous bad blocks, the Verify process may not complete. If the process is not successful, you must format the hard disk to mark all bad blocks. Proceed to 11.6, "Formatting the Hard Disk."

7. If the verification process completes successfully, you will not be required to complete step six in the rebuild process, "Formatting the Hard Disk," but you may if you wish. Proceed to 11.6, "Formatting the Hard Disk," for more information.

### 11.52 Verifying a Server Hard Disk

If the Verify procedure fails, you will need to reformat your disk, as described in 11.6, "Formatting the Hard Disk." Follow these steps to Verify the hard disk:

1. From the Main Utility Page, key in the following to display the Disk Maintenance Utilities menu:

Option: DM

2. Check to see that the SCSI\_id, Logical\_Unit\_Number, and Disk Capacity fields contain the correct settings for the disk you want to verify.
3. Key in the following to select the VeriFY option, which runs the verification procedure.

Option: VFY

4. If the hard disk is corrupted with numerous bad blocks, the Verify process may not complete. If the process is not successful, you must format the hard disk to mark all bad blocks. Proceed to 11.6, "Formatting the Hard Disk."
5. If the verification process completes successfully, you will not be required to complete step 6 in the rebuild process, "Formatting the Hard Disk," but you may if you wish. Proceed to 11.6, "Formatting the Hard Disk," for more information.

## 11.6 Formatting the Hard Disk

If the Verify procedure fails, you must format the hard disk. Otherwise, formatting is optional. Formatting overwrites all data on the hard disk; therefore, you should not format unless the Verify procedure failed.

Formatting involves structuring the disk so that hardware and software can communicate with the disk. This process writes a test pattern to the disk and reads it to verify the pattern, checks for any errors, and marks locations on the disk (flaws) that cause errors. These flaws are recorded to prevent any future writing to or reading from that location.

Rebuilding without formatting is possible. In many cases, only part of the hard disk is corrupted. You may be able to restore the corrupted file system(s) without formatting the hard disk. If you format, you will lose all data on the disk; if you do not format, you will lose only the data in file systems that you restore. If you do not wish to format the hard disk, proceed to 11.7, "Selecting the Rebuild Utility Page Icon."

Formatting takes approximately 45 minutes for 80-MB hard disks, 30 minutes for 156-MB hard disks, 25 minutes for 180-MB hard disks, 30 minutes for 355-MB hard disks, and 50 minutes for 670-MB hard disks.

**Caution:**

Formatting destroys all data on the disk. Therefore, back up all personal files before formatting.

Formatting procedures for workstations and servers differ and are described separately.

### 11.6.1 Formatting Workstation Hard Disks

Follow these steps to format workstation hard disks:

1. From the Main Utility Page, select the Disk Maintenance Utility Page.
2. When the Disk Maintenance Utility Page appears, select the following settings for an internal hard disk:

SCSI ID = 0  
Logical Unit Number = 0

For an external hard disk, you would set the SCSI ID to 1, 2, or 3, and the LUN would be 0.

3. Match the Intergraph Part Number on the Hard Disk Profile Sheet with the part number in the Disk Type roll-through box.
4. Key in the serial number from the Hard Disk Profile Sheet if it does not display in the Serial Number data entry field.
5. Select the Next Page screen button to access the Hard Disk Flaw Data Utility Page. Select the following settings:

Format With Flaw Data - Yes

Selecting yes for the Format With Flaw data option prevents flaws on the disk from being ignored during the format process.

Verify on Format - Yes

Selecting yes for the Verify on Format option runs the Verify procedure to record any new flaws found on the disk during the format.

6. Select the Format screen button to access the red Format Utility Page.
7. Select the Format screen button to start the procedure. While the format procedure is running, messages revealing the part of the disk being verified and identifying the bad blocks appear on the screen. You do not need to respond to any of these messages. The message "Format Complete" appears when formatting is finished. (Formatting takes approximately 45 minutes for 80-MB hard disks, 30 minutes for 156-MB hard disks, 25 minutes for 180-MB hard disks, 30 minutes for 355-MB hard disks, and 50 minutes for 670-MB hard disks.) Select the Previous Page screen button to return to the Hard Disk Flaw Data Utility Page.
8. Proceed to 11.7, "Selecting the Rebuild Utility Page Icon," for the next step in the rebuild process.

### 11.6.2 Formatting Server Hard Disks

Follow these steps to format an InterServe hard disk:

1. From the Main Utility Page, key in DM to display the Disk Maintenance Utilities menu.
2. Key in the following to set the SCSI ID to 0:

Option: SCSI 0

3. Key in the following to set the logical unit number to 0:

Option: LUN 0

4. Key in the following, where <serial number> is the correct serial number for the disk as the Hard Disk Profile Sheet indicates:

Option: SN <serial-number>

5. Key in the following so that flaw data will be considered during the format:

Option: FWF Y

6. Key in the following for the flaw data list to be verified during the format and for any additional bad blocks to be added to the list and registered:

Option: FWV Y

7. Key in the following to start the format procedure:

Option: FORMAT

While the format procedure is running, messages revealing the part of the disk being verified and identifying the bad blocks appear on the screen. The message "Format Complete" appears when formatting is finished. (Formatting takes approximately 45 minutes for 80-MB hard disks, 30 minutes for 156-MB hard disks, 25 minutes for 180-MB hard disks, 30 minutes for 355-MB hard disks, and 50 minutes for 670-MB hard disks.)

8. Proceed to 11.7, "Selecting the Rebuild Utility Page Icon," for the next step in the rebuild process.

## 11.7 Selecting the Rebuild Utility Page Icon

After verifying and formatting the hard disk, the next step in the rebuild process is selecting the Rebuild Utility Page from the Main Utility Page. Do so by selecting the Rebuild icon on the Main Utility Page (for workstations) or keying in REBUILD at the Main Utility Page (for servers).

The Rebuild Utility Page allows you to repartition, load Rebuild media, and enter the Rebuild environment. Before you can begin restoring the file systems, you must have specified an acceptable partition table, successfully loaded the Rebuild media, and selected the icon to enter the Rebuild environment. If any of these steps has not been completed, you will not be allowed to continue the rebuild process.

Notice that the left portion of the Rebuild Utility Page displays messages concerning your current status and your next step. While you are using the Rebuild Utility Page, refer to this message area for brief instructions.

Proceed to 11.8, "Repartitioning the Hard Disk."

## 11.8 Repartitioning the Hard Disk

From the Rebuild Utility Page, you may repartition the hard disk. This step is not required unless the current partitions are unacceptable for the rebuild procedure (as described in 11.8.1).

Partitioning logically divides the hard disk into separate sections or *devices* for specific areas of user and operating system functions. You must repartition the hard disk only if the current partitions are not acceptable for the Rebuild software. You may also repartition to increase or decrease the size of specific partitions or to add nonstandard partitions. For example, you may need to add a *stash* partition to produce large plots on raster output devices. See the *InterPlot User's Guide* for more information on creating a *stash* partition.

**Note:**

The Rebuild Utility Page refers to the partitions on the disk as the *partition table*. This documentation will use the terms *partitions* and *partition table* interchangeably.

During the rebuild process, you must choose one of the following options concerning hard disk partitions:

- Use the current partitions. (Refer to 11.8.1)
- Repartition using Intergraph-defined default partitions. (Refer to 11.8.2.)
- Repartition using custom partitions. (Refer to 11.8.3.)

### 11.8.1 Using the Current Partition Table

If the current partition table is acceptable for the Rebuild software and you do not wish to alter it, you will not need to repartition the hard disk. Partitions must meet the following requirements before the Rebuild software will accept them:

- The root partition must have at least 25,000 blocks.
- The swap partition must have at least 15,000 blocks.
- The usr partition must exist.

**Note:**

Other software products may have additional partition requirements. The system manager should determine the partition sizes appropriate for the system and the software the system uses. Refer to 11.8.3.1, "Partitioning Overview," for information on factors that you should consider when you establish partition sizes.

Follow these steps if you wish to use the current disk partitions:

1. Access the Rebuild Utility Page.
2. If the messages on the left portion of the screen indicate that the current partitions are not acceptable, you must repartition the hard disk. Proceed to 11.8.2 to repartition using default partitions or 11.8.3 to repartition using custom partitions.
3. If the messages on the left portion of the screen indicate that the partitions are acceptable, proceed to 11.9, "Loading the Rebuild Media."

### 11.8.2 Repartitioning Using Default Partitions

If you are not currently using the (Intergraph-defined) default partitions but wish to, you must repartition the hard disk using the Default option supplied on the Rebuild Utility Page.

**Caution:**

You will lose all data on each partition that you alter.

The following are the default partition sizes that Intergraph has established:

|       | 80 MB  | 156 MB | 200 MB | 355 MB | 584 MB  | 670 MB  |
|-------|--------|--------|--------|--------|---------|---------|
| boot: | 3988   | 7988   | 7988   | 7988   | 7988    | 7988    |
| root: | 25000  | 25000  | 25000  | 25000  | 25000   | 25000   |
| swap: | 27360  | 71000  | 71000  | 71000  | 71000   | 71000   |
| usr:  | 100000 | 200600 | 288053 | 590298 | 1037988 | 1204900 |

**Note:**

The total number of blocks in the partition table and the disk capacity differ. They differ because a one-block header is associated with each partition. The header blocks are included in the disk capacity count and not in the partition table count.

Follow these steps to repartition using default partitions:

1. Access the Rebuild Utility Page.
2. Select the Default icon under the Create Partition Table heading. This menu option will automatically redefine all hard disk partitions according to the defaults.

**Caution:**

You will lose all data in any partition that is altered.

3. If the messages on the left portion of the screen indicate that the partitions are acceptable, proceed to 11.9, "Loading the Rebuild Media."

### 11.8.3 Repartitioning Using Custom Partitions

This section describes repartitioning the hard disk using custom partitions rather than using the Intergraph-defined default partitions described in the previous section. For example, you may want to create a `usr2` or `usr3` partition by taking space from `usr`.

**Caution:**

You will lose all data on each partition that you alter.

- If you are not familiar with the concept of repartitioning the hard disk, read 11.8.3.1, "Partitioning Overview," before you begin the process of actually repartitioning the hard disk.
- If you are familiar with the concept of repartitioning the hard disk, proceed to 11.8.3.2, "Creating Nonstandard Partitions."

#### 11.8.3.1 Partitioning Overview

This section provides information concerning the following topics:

- Determining Partition Sizes
- Reallocating Space Between Partitions
- Understanding Partition Names

##### *Determining Partition Sizes*

Intergraph divides workstation/server hard disks into the following partitions:

- `boot` (system)
- `root`
- `swap`
- `DOS` (32C workstations only)
- `usr`

**Note:**

Intergraph does not automatically place a `stash` partition (for plotting) on hard disks. However, some plotters require `stash` on the plot server. A `stash` may be either a partition on the hard disk or physical memory. Refer to the *InterPlot User's Guide* for information on creating a `stash` partition.

The `boot` partition contains boot images, including the hardware diagnostics test, Utility Pages, communications processor, and System V kernel software. This partition contains 3,988 blocks for workstations with 80-MB hard disks and 7,988 blocks for all other hard disk sizes. You cannot access this partition on the Disk Partitioning Utility Page because you cannot alter its size.

The **root** partition contains a major portion of the System V operating system. It is used mainly for system administrative tasks. By default, this partition contains 25,000 blocks. You should not take space from this partition, and if you receive messages saying that the root (7.0) partition is out of space, you must add space to it.

The **swap** partition is used for swapping portions of memory to the hard disk. By default, it contains 27,360 blocks for 80-MB hard disks and 71,000 blocks for all other hard disk sizes. Space can be removed from the **swap** partition and added to another partition. However, some applications will not run without a minimum amount of swap space. This minimum amount varies among applications.

The **usr** partition contains user directories, most products, and any other directories and files that users access. By default, the **usr** partition contains 100,000 blocks for 80-MB hard disks; 200,600 blocks for 156-MB hard disks; 288,053 blocks for 200-MB hard disks; 590,298 blocks for 355-MB hard disks; 1,037,988 blocks for 584-MB hard disks; and 1,164,276 blocks for 670-MB hard disks. Because this partition is the user's *work* partition, it contains the most free space. You may wish to remove space from **usr** and form a **usr2** partition.

You must have a **stash** on your plot server to plot on most plotting devices. A **stash** may be either a partition on your hard disk or physical memory. (Physical memory may be used only if sufficient physical memory is available on your server. See section 1.13 in the *InterPlot User's Guide*.) The following information illustrates how to calculate the size for a **stash** partition on your hard disk.

Pen plotters and the following raster devices **do not** require a **stash**:

- Shinko 65, 635, and 645
- ILP811
- ILP2217
- Calcomp 5602 and 5602BW
- Versatec V80 (for B-size plots)
- Versatec 8524LR and 8536LR

Use the following guidelines to determine the size of your **stash** partition:

- On a 200-dpi plotter (such as the Versatec V7224), plots require 10 blocks per square inch for processing. Therefore, a 20- X 20-in plot requires 4000 blocks for processing. This conclusion is based on the following calculations:

$$\begin{aligned} 20 \text{ in X } 20 \text{ in} &= 400 \text{ sq in} \\ 400 \text{ sq in X } 10 \text{ blocks per sq in} &= 4000 \text{ blocks} \end{aligned}$$

- On a 400-dpi plotter (such as the Versatec 7436), plots require 40 blocks per square inch for processing. Therefore, a 15- X 15-in plot requires 9000 blocks for processing. This conclusion is based on the following calculations:

$$\begin{aligned} 15 \text{ in X } 15 \text{ in} &= 225 \text{ sq in} \\ 225 \text{ sq in X } 40 \text{ blocks per sq in} &= 9000 \text{ blocks} \end{aligned}$$

**Note:**

A 300-dpi plotter requires 22 blocks per square inch for processing. However, Intergraph's 300-dpi laser printer (ILP811) does not require a **stash** partition.

**Stash** must be allocated in 124-block increments. Therefore, the number of blocks in your **stash** partition must be a multiple of 124. If your **stash** calculation is not a multiple of 124, follow these steps:

1. Divide the number that you have calculated for **stash** by 124.
2. Round the quotient up to the next whole number.
3. Multiply the total by 124.

For example, if you have determined that you need 9000 blocks for **stash**, use this formula:

$$\begin{aligned} 9000/124 &= 72.5806 \quad (\text{Round up to } 73.) \\ 73 \text{ X } 124 &= 9052 \end{aligned}$$

Your **stash** will have 9052 blocks.

The following chart summarizes disk stash requirements:

| <i>Device Type</i>   | <i>Blocks Per Square Inch</i>                |
|----------------------|----------------------------------------------|
| 200-dpi device       | 10 blocks per square inch                    |
| 400-dpi device       | 40 blocks per square inch                    |
| QCR camera (2K mode) | 6400 blocks                                  |
| QCR camera (4K mode) | 24,700 blocks                                |
| Other device types   | $(\text{dpi})^2/4096$ blocks per square inch |

Follow these steps to determine the size of your stash partition:

1. Determine the size (in square inches) of the largest plot you will produce. This example assumes the largest plot is an E-size plot (36 X 44 in).

$$(36 \times 44 \text{ in}) = 1584 \text{ sq in}$$

2. Determine the number of blocks per square inch required for processing. For example, if your plotter is a 400-dpi device, it will require 40 blocks per square inch for processing. Multiply the size of the largest plot by 40 as follows:

$$1584 \text{ sq in} \times 40 \text{ sq in} = 63,360 \text{ blocks}$$

3. Round the number of blocks required for processing to the next highest multiple of 124.

$$\begin{aligned} 63,360/124 &= 510.9677 \\ 511 \times 124 &= 63,364 \end{aligned}$$

Therefore, the stash partition for this example requires 63,364 blocks.

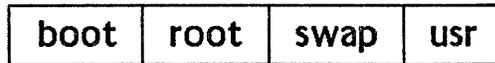
**Note:**

The partition number for stash is *f*, the hexadecimal representation of the decimal number 15. The modifier number for this partition is *0*. Thus, the device file for a stash partition on SCSI ID 0 would be *s0u0pf0*.

### *Reallocating Space Between Partitions*

Before you can take space from one partition and add it to another, the two partitions must be adjacent on the disk.

The order for standard partitions on hard disks is as follows:



For example, you could remove the `usr` partition and add it directly to the `swap` partition (since these partitions are adjacent). However, you could not remove the `usr` partition and add it directly to `root`. You could add the `usr` space to `root` only by adding it to the `swap` partition and then removing it from `swap` and adding it to `root`.

In addition, when you add a partition, the system will search only forward on the disk for space to claim. That is, to remove the `swap` partition and add that space to the `usr` partition, you could not remove the `swap` partition and add a larger `usr` partition. Doing so would cause the system to search for and allocate space from the beginning of the `usr` partition and beyond. Instead, you would need to remove the `swap` and the `usr` partition and then add a larger `usr` partition. In this case, the system would start allocating space from the beginning of the (former) `swap` partition.

**Caution:** Repartitioning the hard disk loses any information in a moved, removed, or resized partition. Therefore, back up all files before repartitioning.

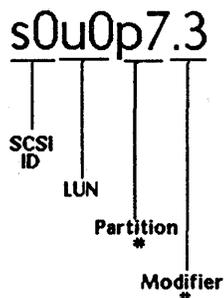
### *Understanding Partition Names*

Partition names appear in the following form on the Disk Partitioning Utility Pages:

sAuBpC.D

The following table defines the components of this partition name:

| <i>Labels</i> | <i>Variables</i>                               |
|---------------|------------------------------------------------|
| s - SCSI ID   | A: SCSI ID number                              |
| u - unit      | B: logical unit (drive) number                 |
| p - partition | C: general-purpose division number (partition) |
|               | D: modifier number (subpartition)              |



The chart on the following pages lists partition numbers for all partitions on Intergraph workstations.

| Disk Partition | 100/200 Series         | 300/3000 Series           |
|----------------|------------------------|---------------------------|
| 2.0            | Flaw Map               | Flaw Map                  |
| 4.0            | Error Map              | Error Logging             |
| 5.0            | Configuration          | Configuration             |
| 8.0            | 80186 IOP Code         | 80386 IOP Code            |
| 8.1            | ROP Code #0            | Not Used                  |
| 8.2            | Blue Screen Code       | Blue Screen Code          |
| 8.3            | UNIX Boot Leader       | UNIX Boot Loader          |
| 8.4            | 80186 IOP Code         | Not Used                  |
| 8.5            | ROP Bios Code          | Digitizer Code            |
| 8.6            | Coprocessor Code       | FPE Exec Code             |
| 8.7            | 80186 CP Code          | 80386 CP Code             |
| 8.8            | Bios Font Data         | Not Used                  |
| 8.9            | ROP Code #1            | Not Used                  |
| 8.a            | Downloaded Diagnostics | Not Used                  |
| 8.b            | ROP Code #2            | Not Used                  |
| 8.c            | ROP Code #3            | IFB Code (MAIN)           |
| 8.d            | ROP Code #4            | IFB Code (ROAM)           |
| 8.e            | ROP Code #5            | IFB Code (ROAM)           |
| a.0            | Not Used               | Downloaded FDM            |
| a.1            | Not Used               | IFB Diagnostics           |
| a.2            | Not Used               | UNIX Engine II Diag       |
| a.3            | Not Used               | UNIX Engine II Diag (IOP) |
| a.4            | Not Used               | IOP+ Diagnostics          |
| a.5            | Not Used               | FPE Diagnostics           |
| a.6            | Not Used               | HC/DIG Diagnostics        |
| a.7            | Not Used               | HC/DIG Diagnostics (IOP)  |
| a.8            | Not Used               | ISI Diagnostics           |
| a.9            | Not Used               | VME Diagnostics           |
| b.f            | Disk Diagnostic Track  | Disk Diagnostic Track     |
| 7.0            | root File System       | root File System          |
| 7.1            | Swap Area              | Swap Area                 |
| 7.3            | usr File System        | usr File System           |
| 7.4            | usr2 File System       | usr2 File System          |
| 7.5            | usr3 File System       | usr3 File System          |
| 9.0            | DOS File System        | DOS File System           |
| f.0            | Plotting Stash         | Plotting Stash            |
| 10.0           | Unused Space           | Unused Space              |
| ff.0           | End Of Disk            | End Of Disk               |

| Disk Partition | Series 2000      | 6000 Series      | 6100/6500 Series   |
|----------------|------------------|------------------|--------------------|
| 2.0            | Flaw Map         | Flaw Map         | Flaw Map           |
| 4.0            | Error Map        | Error Map        | Error Map          |
| 5.0            | Configuration    | Configuration    | Configuration      |
| 8.0            | IO Code          | IO Code          | IO Code            |
| 8.1            | ROP Code #0      | Not Used         | Not Used           |
| 8.2            | Blue Screen Code | Blue Screen Code | Blue Screen        |
| 8.3            | Not Used         | Not Used         | Not Used           |
| 8.4            | Not Used         | Not Used         | Not Used           |
| 8.5            | Not Used         | Not Used         | Not Used           |
| 8.6            | Not Used         | Not Used         | Not Used           |
| 8.7            | Not Used         | Not Used         | Not Used           |
| 8.8            | Not Used         | Not Used         | Not Used           |
| 8.9            | Not Used         | Not Used         | Not Used           |
| 8.a            | Not Used         | Not Used         | Not Used           |
| 8.b            | Not Used         | Not Used         | Not Used           |
| 8.c            | Not Used         | Not Used         | Not Used           |
| 8.d            | Not Used         | EDGE I Code      | EDGE I Code        |
| 8.e            | Not Used         | EDGE II Code     | EDGE II Code       |
| a.0            | FDMDISK          | FDMDISK          | FDMDISK            |
| a.1            | System Board     | EDGE I           | EDGE I             |
| a.2            | Video            | EDGE II          | EDGE II            |
| a.3            | Digitizer        | EDGE III         | EDGE III           |
| a.4            | Not Used         | System Board     | IOI                |
| a.5            | Not Used         | Not Used         | IOI CE             |
| a.6            | Not Used         | VME (Dianostics) | VME (Diagnostics)  |
| a.7            | Not Used         | Digitizer        | Digitizer          |
| a.8            | Not Used         | ISI              | ISI                |
| a.9            | Not Used         | VME Monitor      | VME Monitor        |
| a.a            | Not Used         | SIP              | SIP                |
| a.b            | Not Used         | FIP              | FIP                |
| a.c            | Not Used         | FDDI             | FDDI               |
| a.d            | Not Used         | SR SCSI          | SR SCSI            |
| a.e            | Not Used         | Not Used         | C3+                |
| a.f            | Not Used         | 32 Channel       | 32 Channel         |
| b.0            | Not Used         | EDGE DSP (Mon.)  | EDGE DSP (Monitor) |
| b.1            | Not Used         | Frame Grabber    | Frame Grabber      |
| b.2            | Not Used         | Not Used         | Not Used           |
| b.f            | Disk Diag. Track | Disk Diag. Track | QWIK               |
| 7.0            | root File System | root File System | root File System   |
| 7.1            | Swap Area        | Swap Area        | swap Area          |
| 7.3            | usr File System  | usr File System  | usr File System    |
| 7.4            | usr2 File System | usr2 File System | usr2 File System   |
| 7.5            | usr3 File System | usr3 File System | usr3 File System   |
| 9.0            | Not Used         | Not Used         | Not Used           |
| f.0            | Plotting Stash   | Plotting Stash   | Plotting Stash     |
| 10.0           | Unused Space     | Unused Space     | Unused Space       |
| ff.0           | End Of Disk      | End Of Disk      | End of Disk        |

### 11.8.3.2 Creating Nonstandard Partitions

This section supplies the general steps for creating new partitions. The system manager should decide how much space must be allocated for each partition. For example, to add a stash partition for plotting, refer to 11.8.3.1, "Determining Partition Sizes," to determine how large the stash partition must be and which partition(s) space should come from. Then, follow steps in this section for adding partitions and altering partition sizes. For more information on creating a stash partition, refer to the *InterPlot User's Guide*.

**Caution:** Repartitioning wipes out all data on any altered partition. Therefore, you should back up all files before repartitioning.

#### *Repartitioning Workstation Hard Disks*

Follow these steps to repartition workstation hard disks:

1. Select the Custom icon from the Rebuild Utility Page. This selection will transfer control to the Disk Partitioning Utility Page.
2. Notice the chart displaying the hard disk partitions at the top of the Disk Partitioning Utility Page. Study this chart to determine how you wish to alter the partitions. Keep in mind the following restrictions enforced by the Rebuild software:
  - The root partition must have at least 25,000 blocks.
  - The swap partition must have at least 15,000 blocks.
  - The usr partition must exist.

**Note:**

Other software products may have additional partition requirements. The system manager should determine the partition sizes appropriate for the system and the software the system uses. Refer to 11.8.3.1, "Determining Partition Sizes," for information on factors you should consider when you establish partition sizes.

3. From the Disk Partitioning Utility Page, you can add and remove partitions. Thus, to change partition sizes, you must remove all partitions being altered and then recreate them with new sizes.

To remove a partition, scroll through the Partition Names roll-through box until the name of the partition to remove appears. If the desired name is not in the roll-through box, key in the Partition Number and the Modifier Number (in their respective fields) for the partition to be removed. Select the Remove Partition button to remove the partition currently displaying in the Partition Names roll-through box.

Enter the partition number and modifier number of the new partition. Then, enter the Size in Blocks of the new partition. To add the specified partition, select the Add Partition button. The partition will be added in the first available slot that will accommodate that size.

**Note:**

If you plan to make the size of the last partition equal to the amount of available free space listed in the partition table, you must subtract two blocks from the amount of free space. (The partition header and the "End of Disk" header require one block each.)

4. Notice that the bar chart at the top of the screen reflects adjustments to the disk partitions.
5. After you have defined all hard disk partitions, select the Utility icon to return to the Main Utility Page and then the Rebuild icon to return to the Rebuild Utility Page.
6. Proceed to 11.9, "Loading the Rebuild Media."
7. Keep in mind that you must create a device file and then create and mount a file system on any nonstandard partition that you add (such as usr2 or usr3). These steps are described in 11.13.

*Repartitioning an InterServe Hard Disk*

Follow these steps to repartition an InterServe hard disk:

1. Select the Custom icon from the Rebuild Utility Page. This selection will transfer control to the Disk Partitioning Utility Page.
2. Key in the following to list the partitions currently on the hard disk:

Option: LP

Keep in mind the following restrictions enforced by the Rebuild software:

- The root partition must have at least 25,000 blocks.
- The swap partition must have at least 15,000 blocks.
- The usr partition must exist.

**Note:**

Other software products may have additional partition requirements. The system manager should determine the partition sizes appropriate for the system and the software the system uses. Refer to 11.8.3.1, "Determining Partition Sizes," for information on factors that you should consider when you establish partition sizes.

3. From the Disk Partitioning Utility Page, you can add and remove partitions. Thus, to change partition sizes, you must remove all partitions being altered and then read them with new sizes.

To remove a partition from an InterServe hard disk, first key in the following to list the existing partitions:

Option: LP

Next, write down the partition and modifier numbers for the partitions you will remove (such as 7 and 3 for usr). Refer to 11.8.3.3 for an explanation of partition and modifier numbers.

Then, key in the following command line to remove a partition, substituting the appropriate values in each field:

Option: RMP <partition number> <modifier number>

- RMP is the ReMove\_Partition option, which removes the partition designated in the same line from the disk.
- <partition number> is the number of the partition to be removed.
- <modifier number> is the modifier number of the partition to be removed.

To add a partition to an InterServe hard disk, first key in the following to select the Add\_SYStem option. This option reserves the first 7988 blocks of disk space for the bootable partitions and images that you must create later when rebuilding the hard disk with the Rebuild floppy disk set (described in 11.11, "Restoring the File Systems").

Option: **ASYS**

Key in the following to add each desired partition to the disk:

Option: **ADDP** <partition number> <modifier number> <size>

**ADDP** Selects the **ADD\_Partition** option described in Chapter 8, "Utility Page Reference."

**Note:**

If you plan to make the size of the last partition equal to the amount of available free space listed in the partition table, you must subtract two blocks from the amount of free space. (The partition header and the "End of Disk" header require one block each.)

4. Key in the following to select the **List\_Partitions** option, which lists existing partitions and verifies that the desired partitions have been added:

Option: **LP**

5. After you have defined all hard disk partitions, key in **UT** to return to the Main Utility Page and then **REBUILD** to return to the Rebuild Utility Page.
6. Proceed to 11.9, "Loading the Rebuild Media."
7. Keep in mind that you must create a device file and then create and mount a file system on any nonstandard partition that you add (such as **usr2** or **usr3**). These steps are described in 11.13.

## 11.9 Loading the Rebuild Root Media

The next step in the rebuild process is to load the Rebuild Root media. This step loads minimal file systems on the hard disk. The minimal file systems are stored temporarily in the swap space. When you actually begin restoring the file systems, these file systems will be copied from swap to the appropriate partitions.

Follow these steps to load the Rebuild Root media:

**Note:**

You can rebuild your workstation/server with different media types, including microfloppy (3½-inch diskettes) and floppy disk. The default is microfloppy for 2000/6000-series systems and floppy for all others.

1. Locate (or create) the Rebuild Root media. If your requested media type is floppy disk, locate or create current Rebuild Root floppy disks.
2. Select the appropriate media type (for example, floppy) from the roll-through box in the Load Rebuild Root Media portion of the Rebuild Utility Page.
3. Select the Load icon. When prompted, insert the first Rebuild Root floppy (#2) in the drive and press <Return>.
4. As the media is being loaded on the hard disk, notice the graphic display representing the portion of data that has been loaded from each floppy. This graphic is purely informational; you do not need to respond to it.
5. When floppy #2 has been loaded, remove it from the drive, insert the Rebuild Root floppy (#3) in the drive, and press <Return>.

**Note:**

If a "CRC error" is detected while you are loading the media, try again to load the media. If the media still does not load successfully, recreate the Rebuild Root disk.

6. Proceed to 11.10, "Entering the Rebuild Environment."

### **11.10 Entering the Rebuild Environment**

After loading the Rebuild media, you may enter the Rebuild environment. After you have loaded the Rebuild media on the hard disk, select the Boot icon on the Rebuild Utility Page. This icon will boot the workstation into single-user mode and initiate the Rebuild menu. You do not need to boot from the Rebuild floppy disks because the Rebuild media has already been loaded on the hard disk (in swap space). Thus, you may remove Rebuild media from the disk drive before you boot into the Rebuild environment.

After the workstation has booted into single-user mode, proceed to 11.11, "Restoring the File Systems."

## 11.11 Restoring the File Systems

After you have entered the Rebuild environment, you may restore minimal file systems to the hard disk. Restoring minimal *root* and */usr* file systems prepares the hard disk so that software can be loaded through the *newprod* utility.

The Rebuild environment consists of a series of menus that allow you to perform tasks such as restore the *root* and */usr* file systems, create new file systems (such as */usr2*), install software using the *newprod* utility, reboot the system, and exit the menu.

The menu options you should choose will depend on the extent you need to rebuild. This section describes each option available.

After the workstation/server boots to System V single-user mode, the following screen appears:

```

Welcome to Rebuild

Use the arrow keys to navigate the menus, and then use the
RETURN key to select menu items.

Use the top two rows of the keypad to select the functions that
are shown at the bottom of the display.

TOP TWO ROWS OF THE KEYPAD:

CANCEL(pf1) PREVPAGE(pf2) NEXTPAGE(pf3) (pf4)
(7) HELP (8) END-MENU (9) CONTINUE (-)

Press CONTINUE to go on to the Rebuild Menu

CANCEL PREVPAGE NEXTPAGE (7) (8) END-MENU CONTINUE

```

This screen provides an introduction to the rebuild utility. Notice that the menus require you to use the function keypad to move back and forth between menu levels. To continue to the main rebuild menu, press the CONTINUE (-) key on the function keypad.



To create new *root* and */usr* file systems, select the appropriate file system settings and then press the SAVE key (PF-3). Press the CANCEL key (PF-1) to exit the menu without creating *root* and */usr* file systems. The following settings are available on this menu:

- File system type for *root* and */usr*

Use the CHOICES key (PF-2) to toggle between the *Fast File System* (FFS) and *standard file system* (S51K) types. Fast File Systems contain 8K-byte units, whereas standard file systems contain 1K-byte units. For this reason, Fast File Systems read files more efficiently than standard file systems do. However, the RFS product does not support FFS. Thus, if you use RFS, do not create any file system as a FFS.

- File system volume names for *root* and */usr*

The *volume name* for a file system is the alphanumeric character string that the */etc/labelit* utility uses to label the partition and attach a logical base directory name to the file system. By default, the volume name is the current date. However, it can be any six-character string that you key in.

After the *root* and */usr* file systems have been created, return to the main rebuild menu and download software. Proceed to 11.12, "Loading Nucleus Software with newprod."

### *Rebuild root*

The following menu displays if you choose the "Rebuild root" option from the main rebuild menu:

|                                                                                                                                                                                                                                                |                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1 Rebuild</p> <ul style="list-style-type: none"> <li>Rebuild all</li> <li>&gt; Rebuild root</li> <li>Rebuild usr</li> <li>Make a File System</li> <li>Download software</li> <li>Introduction Page</li> <li>Reboot</li> <li>Exit</li> </ul> | <p>2 Rebuild the root file system</p> <p>File system type: Fast_File_System<br/>File system volume name: 051690</p> <p>Press SAVE to make the root file system.<br/>Press CANCEL to return to the previous menu.</p> |
| <p>CANCEL CHOICES SAVE [ ] HELP F1-D-F1-N1 EXIT</p>                                                                                                                                                                                            |                                                                                                                                                                                                                      |



This menu option creates a minimal */usr* file system. To create the new */usr* file system, select the appropriate file system settings and then press the SAVE key (PF-3). Press the CANCEL key (PF-1) to exit the menu without creating the */usr* file system. The following settings are available on this menu:

- File system type

Use the CHOICES key (PF-2) to toggle between the *Fast File System* (FFS) and *standard file system* (S51K) types. Fast File Systems contain 8K-byte units, whereas standard file systems contain 1K-byte units. For this reason, Fast File Systems read files more efficiently than standard file systems do. However, the RFS product does not support FFS. Thus, if you use RFS, do not create any file system as a FFS.

- File system volume name

The *volume name* for a file system is the alphanumeric character string that the */etc/labelit* utility uses to label the partition and attach a logical base directory name to the file system. By default, the volume name is the current date. However, it can be any six-character string that you key in.

After the */usr* file system has been created, return to the main rebuild menu and rebuild the *root* file system (if needed). Then, proceed to 11.12, "Loading Nucleus Software with newprod."

### *Make a File System*

The following menu displays if you choose the "Make a File System" option from the main rebuild menu:

|                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1 Rebuild</p> <ul style="list-style-type: none"> <li>Rebuild all</li> <li>Rebuild root</li> <li>Rebuild usr</li> <li>&gt; Make a File System</li> <li>Download software</li> <li>Introduction Page</li> <li>Reboot</li> <li>Exit</li> </ul> | <p>2 Make a File System</p> <p>SCSI ID: 0</p> <p>Logical unit number: 0</p> <p>Partition (hex digit): 7</p> <p>Modifier (hex digit): 4</p> <p>File System Type: FFS</p> <p>File System Name: usr2</p> <p>Volume Name: 051690</p> <p>Press SAVE to make the file system.<br/>Press CANCEL to return to the previous menu.</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

CANCEL CHOICES SAVE HELP EXIT

This menu option creates a file system other than *root* or */usr* (for example, */usr2*, */usr3*, etc.). To create the new file system, select the appropriate file system settings and then press the SAVE key (PF-3). Press the CANCEL (PF-1 function key) to exit the menu without creating a file system. The following settings are available on this menu:

- **SCSI ID**

Key in the SCSI ID of the disk on which to create the file system.

- **Logical unit number**

Key in the Logical unit number (LUN) of the disk on which to create the file system.

- **Partition number**

Key in the partition number of the partition on which to create the file system. This number should be a hexadecimal (0-9, a-f) digit. The partition number for all UNIX partitions is 7.

- **Partition modifier number**

Key in the modifier number of the partition on which to create the file system. This number should be a hexadecimal digit. The modifier numbers 0, 1, and 3 are reserved for *root*, *swap*, and */usr*, respectively.

- **File system type**

Use the CHOICES key (PF-2) to toggle between the *Fast File System* (FFS) and *standard file system* (S51K) types. Fast File Systems contain 8K-byte units, whereas standard file systems contain 1K-byte units. For this reason, Fast File Systems read files more efficiently than standard file systems do. However, the RFS product does not support FFS. Thus, if you use RFS, do not create any file system as a FFS.

- **File system name**

Key in the name of the file system (for example, *usr3* or *usr4*). The default file system name is *usr2*. Do not include a leading slash when you key in the file system name.

- File system volume name

The *volume name* for a file system is the alphanumeric character string that the `/etc/labelit` utility uses to label the partition and attach a logical base directory name to the file system. By default, the volume name is the current date. However, it can be any six-character string that you key in.

After the file system has been created, return to the main rebuild menu and download software. Proceed to 11.12, "Loading Nucleus Software with newprod."

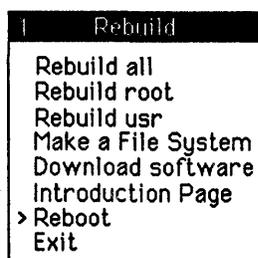
*Introduction Page*

The following menu (which also appears when you first enter the rebuild environment) displays if you choose the "Introduction Page" option from the main rebuild menu:

The screenshot shows a terminal window with a title bar "Rebuild". On the left, a menu is displayed with the following options: "Rebuild all", "Rebuild root", "Rebuild usr", "Make a File System", "Download software", "> Introduction Page", "Reboot", and "Exit". The "Introduction Page" option is selected. The main area of the terminal displays a "Welcome to Rebuild" message followed by instructions: "Use the arrow keys to navigate the menus, and then use the RETURN key to select menu items." and "Use the top two rows of the keypad to select the functions that are shown at the bottom of the display." Below this, it says "TOP TWO ROWS OF THE KEYPAD:" and shows a keypad layout with buttons: "CANCEL (pf 1)", "PREVPAGE (pf 2)", "NEXTPAGE (pf 5)", "(pf 4)", "(7)", "HELP (8)", "END-MENU (9)", and "EXIT (-)". At the bottom of the terminal, a prompt says "Press CANCEL to go on to the Rebuild Menu" and another keypad layout is shown with buttons: "CANCEL", "PREVPAGE", "NEXTPAGE", "HELP", "END-MENU", and "EXIT".

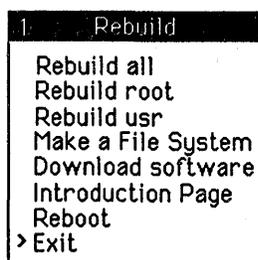
### *Reboot*

If you choose the "Reboot" option from the main rebuild menu, the system will be rebooted. You may reboot after you have restored the file systems and before you load software (so that you will be booted off the *root* partition, rather than *swap*, where the rebuild environment is loaded.) In addition, you must reboot after you load software.



### *Exit*

If you choose the "Exit" option from the main rebuild menu, you will exit the rebuild menus and be placed at the system prompt. To return to the rebuild menu, key in <Ctl-D>.



## 11.12 Loading Nucleus Software with newprod

After you have restored the file systems, you must load nucleus software using the `newprod` utility. Nucleus software includes the software products that must reside on the hard disk before you can run the System V operating system. You do not need to load application software until the system is completely functional.

**Note:**

You have the option of rebooting the system (off the Rebuild Boot floppy) before you load software, so that you will be booted off the *root* partition, rather than the *swap* partition. If you do so, you will need to, once again, boot into the rebuild environment. At that point, the main rebuild menu will not include the "Rebuild all," "Rebuild root," and "Rebuild usr" options.

After you have restored the file systems, you may choose the "Download software" option from the rebuild menu. The following menu appears:

|   |                     |   |              |
|---|---------------------|---|--------------|
| 1 | Rebuild             | 2 | Menu         |
|   | Rebuild all         |   | > XNS        |
|   | Rebuild root        |   | TCP/IP       |
|   | Rebuild usr         |   | Floppy       |
|   | Make a File System  |   | Tape         |
|   | > Download software |   | CDROM        |
|   | Introduction Page   |   | Remote CDROM |
|   | Reboot              |   |              |
|   | Exit                |   |              |

The "Download software" menu invokes the `newprod` utility from the delivery media/source of your choice. Choices include an XNS-based network delivery source, a TCP/IP-based network delivery source, `newprod` floppy disk, `newprod` tape, local CDROM, or remote CDROM.

**XNS network delivery source**

The following menu displays if you choose the "XNS" option from the download software menu:

|                                                                                                                                                                                                                                                       |                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>1 Rebuild</b></p> <ul style="list-style-type: none"> <li>Rebuild all</li> <li>Rebuild root</li> <li>Rebuild usr</li> <li>Make a File System</li> <li>&gt; Download software</li> <li>Introduction Page</li> <li>Reboot</li> <li>Exit</li> </ul> | <p><b>2 Menu</b></p> <ul style="list-style-type: none"> <li>&gt; XNS</li> <li>TCP/IP</li> <li>Floppy</li> <li>Tape</li> <li>CDROM</li> <li>Remote CDROM</li> </ul> |
| <p><b>3 Host Information</b></p> <p>Host Address<br/>(xxxxxxxx.xx-xx-xx-xx-xx-xx): _____</p> <p>User Name: _____</p> <p>Password: _____</p> <p>Press SAVE to load the software.<br/>Press CANCEL to return to the previous menu.</p>                  |                                                                                                                                                                    |
| <p>CANCEL CHOICES SAVE HELP CHID-MENU EXIT</p>                                                                                                                                                                                                        |                                                                                                                                                                    |

This menu allows you to connect to an XNS-based network delivery source (a node that stores products in a deliverable format so that other nodes can connect and install software) and install software through the newprod utility. You must supply the following information at this menu:

- **Host Address**

The network address of the XNS-based host (delivery source) must be entered in the format "XXXXXXXXXX.XX-XX-XX-XX-XX-XX."

- **User Name**

You must supply a valid username for the host.

- **Password**

You must supply a valid password for a user account on the host (if required on the host).

Press the SAVE key (PF-3) to connect to the delivery source and invoke newprod, or the CANCEL key (PF-1) to return to the previous menu.

Once you invoke **newprod**, you must update the following products. These products are *nucleus products*, or the minimum software products required to run the CLIX operating system.

- Delivery Tools (DELTOOLS)
- Workstation/Server Diagnostics (DIAG)
- Environ V Target Shared Library (ENVIRON\_S)
- Gpipe Target Run-Time Shared Library (GPIPE\_S)
- Workstation Network Software (INC)
- System V.3.1 File Systems (SYSTEMV)
- System V.3.1 Boot Images (UNIXBOOT)
- Workstation Graphic Resources (RESOURCES)
- I/Forms Run-Time Package (FORMS\_S)
- RIS Client Support Package (RISCCU)
- Screen Manager (SCREENMGR) (not needed for InterServe systems)
- DEC VT220 Emulation (VT220) (not needed for InterServe systems)

Please refer to the *CLIPPER Software Delivery Guide* for instructions for using **newprod**.

After you have loaded software, reboot the system by selecting the Reboot option from the rebuild menu. Proceed to 11.13, "Preparing Nonstandard Partitions for Use."

**TCP/IP network delivery source**

The following menu displays if you choose the "TCP/IP" option from the download software menu:

The screenshot shows a three-step menu process. Step 1, titled 'Rebuild', lists options: Rebuild all, Rebuild root, Rebuild usr, Make a File System, > Download software (highlighted), Introduction Page, Reboot, and Exit. Step 2, titled 'Menu', lists options: XNS, > TCP/IP (highlighted), Floppy, Tape, CDROM, and Remote CDROM. Step 3, titled 'Network Address Information', contains input fields for Host Node Name, User Name, Password, Host Address (xxx.xxx.xxx.xxx), and Your Address (xxx.xxx.xxx.xxx). Below the fields are instructions: 'Press SAVE to load the software. Press CANCEL to return to the previous menu.' At the bottom is a command bar with buttons: CANCEL, CHOICES, SAVE, HELP, CMD-MENU, and EXIT.

This menu allows you to connect to a TCP/IP-based network delivery source (a node that stores products in a deliverable format so that other nodes can connect and install software) and install software through the `newprod` utility. You must supply the following information at this menu:

- **Host Node Name**

You must supply the nodename of the host (delivery source).

- **User Name**

You must supply a valid username for the host.

- **Password**

You must supply a valid password for a user account on the host (if required on the host).

- **Host Address**

You must supply the network address of the TCP/IP-based host (delivery source). Enter the address in the format "XXX.XXX.XXX.XXX."

- Your Address

You must supply the network address of the machine you are rebuilding. Enter the address in the format "XXX.XXX.XXX.XXX."

Press the SAVE key (PF-3) to connect to the delivery source and invoke newprod, or the CANCEL key (PF-1) to return to the previous menu.

Once you invoke newprod, you must update the following products. These products are *nucleus products*, or the minimum software products required to run the CLIX operating system.

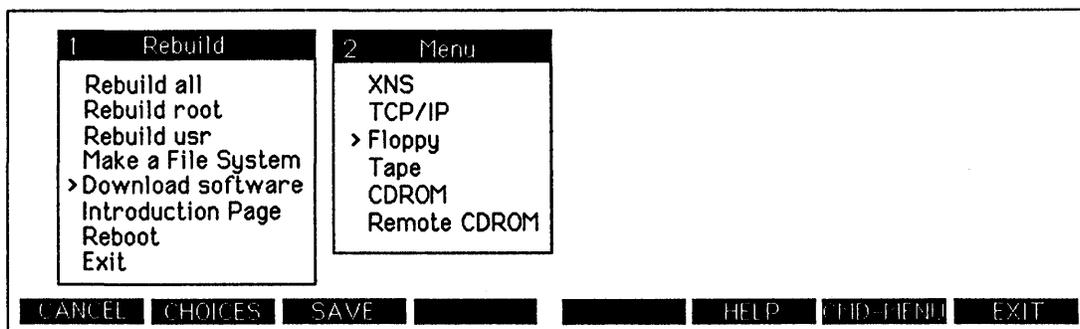
- Delivery Tools (DELTOOLS)
- Workstation/Server Diagnostics (DIAG)
- Environ V Target Shared Library (ENVIRON\_S)
- Gpipe Target Run-Time Shared Library (GPIPE\_S)
- Workstation Network Software (INC)
- System V.3.1 File Systems (SYSTEMV)
- System V.3.1 Boot Images (UNIXBOOT)
- Workstation Graphic Resources (RESOURCES)
- I/Forms Run-Time Package (FORMS\_S)
- RIS Client Support Package (RISCCU)
- Screen Manager (SCREENMGR) (not needed for InterServe systems)
- DEC VT220 Emulation (VT220) (not needed for InterServe systems)

Please refer to the *CLIPPER Software Delivery Guide* for instructions for using newprod.

After you have loaded software, reboot the system by selecting the Reboot option from the rebuild menu. Proceed to 11.13, "Preparing Nonstandard Partitions for Use."

*Floppy disk*

The following messages display if you choose the "Floppy" option from the download software menu:



Working...  
 Changing over to the hard disk root file system.  
 Insert a newprod product floppy and press <RETURN> ---->

This menu option allows you to load software from a local floppy disk drive. When prompted, insert a newprod floppy disk and press <Return>.

Once you invoke newprod, you must update the following products. These products are *nucleus products*, or the minimum software products required to run the CLIX operating system.

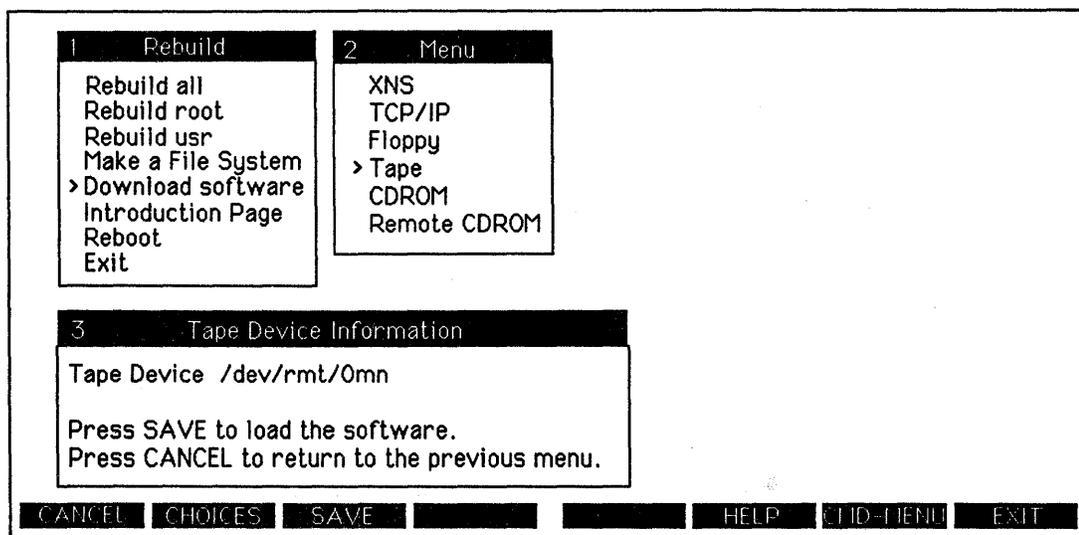
- Delivery Tools (DELTOOLS)
- Workstation/Server Diagnostics (DIAG)
- Environ V Target Shared Library (ENVIRON\_S)
- Gpipe Target Run-Time Shared Library (GPIPE\_S)
- Workstation Network Software (INC)
- System V.3.1 File Systems (SYSTEMV)
- System V.3.1 Boot Images (UNIXBOOT)
- Workstation Graphic Resources (RESOURCES)
- I/Forms Run-Time Package (FORMS\_S)
- RIS Client Support Package (RISCCU)
- Screen Manager (SCREENMGR) (not needed for InterServe systems)
- DEC VT220 Emulation (VT220) (not needed for InterServe systems)

Please refer to the *CLIPPER Software Delivery Guide* for instructions for using newprod.

After you have loaded software, reboot the system by selecting the Reboot option from the rebuild menu. Proceed to 11.13, "Preparing Nonstandard Partitions for Use."

**Tape**

The following menu displays if you choose the "Tape" option from the download software menu:



This menu option allows you to load software from a local tape drive. This menu requires you to supply device information for the tape drive. After you have supplied the correct tape device information, press the SAVE key (PF-3) to invoke newprod, or the CANCEL key (PF-1) to return to the previous menu.

Once you invoke newprod, you must update the following products. These products are *nucleus products*, or the minimum software products required to run the CLIX operating system.

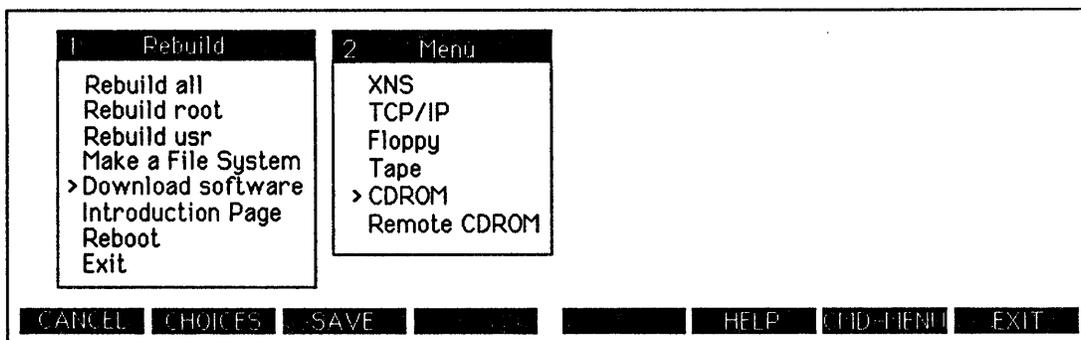
- Delivery Tools (DELTOOLS)
- Workstation/Server Diagnostics (DIAG)
- Environ V Target Shared Library (ENVIRON\_S)
- Gpipe Target Run-Time Shared Library (GPIPE\_S)
- Workstation Network Software (INC)
- System V.3.1 File Systems (SYSTEMV)
- System V.3.1 Boot Images (UNIXBOOT)
- Workstation Graphic Resources (RESOURCES)
- I/Forms Run-Time Package (FORMS\_S)
- RIS Client Support Package (RISCCU)
- Screen Manager (SCREENMGR) (not needed for InterServe systems)
- DEC VT220 Emulation (VT220) (not needed for InterServe systems)

Please refer to the *CLIPPER Software Delivery Guide* for instructions for using newprod.

After you have loaded software, reboot the system by selecting the Reboot option from the rebuild menu. Proceed to 11.13, "Preparing Nonstandard Partitions for Use."

### Local CDROM

The following messages display if you choose the "CDROM" option from the download software menu:



Working...  
 Changing over to the hard disk root file system.  
 Insert a newprod CDROM and press <RETURN>---->

This menu option allows you to load software from a local CDROM disk drive. When prompted, insert a newprod CDROM and press <Return>.

Once you invoke newprod, you must update the following products. These products are *nucleus products*, or the minimum software products required to run the CLIX operating system.

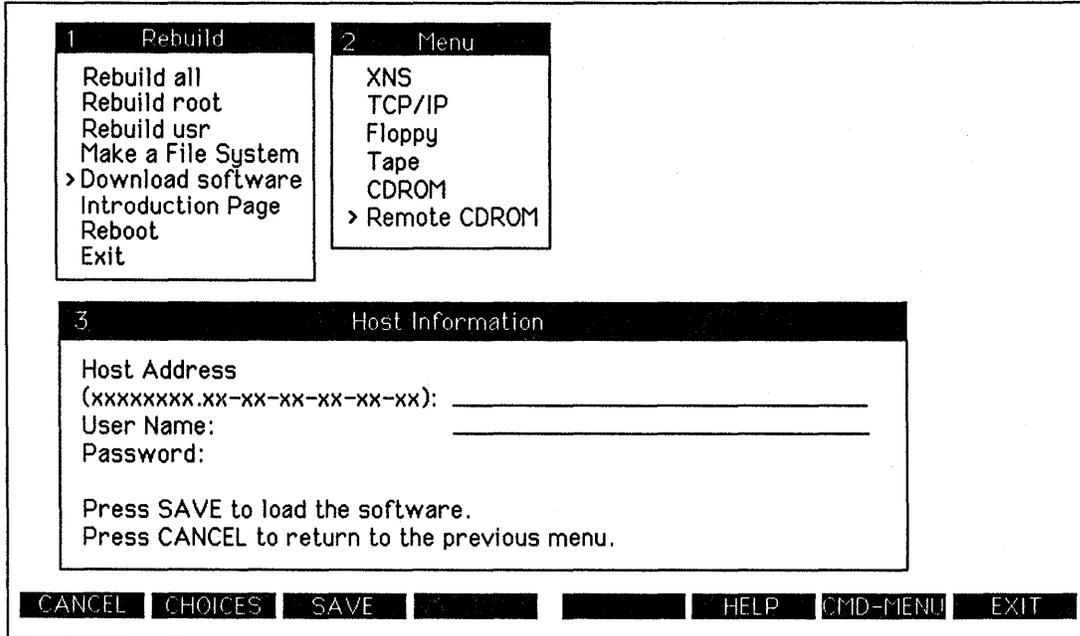
- Delivery Tools (DELTOOLS)
- Workstation/Server Diagnostics (DIAG)
- Environ V Target Shared Library (ENVIRON\_S)
- Gpipe Target Run-Time Shared Library (GPIPE\_S)
- Workstation Network Software (INC)
- System V.3.1 File Systems (SYSTEMV)
- System V.3.1 Boot Images (UNIXBOOT)
- Workstation Graphic Resources (RESOURCES)
- I/Forms Run-Time Package (FORMS\_S)
- RIS Client Support Package (RISCCU)
- Screen Manager (SCREENMGR) (not needed for InterServe systems)
- DEC VT220 Emulation (VT220) (not needed for InterServe systems)

Please refer to the *CLIPPER Software Delivery Guide* for instructions for using newprod.

After you have loaded software, reboot the system by selecting the Reboot option from the rebuild menu. Proceed to 11.13, "Preparing Nonstandard Partitions for Use."

**Remote CDROM**

The following menu displays if you choose the "Remote CDROM" option from the download software menu:



This menu allows you to connect to a remote CDROM drive and install software through the **newprod** utility. You must supply the following information at this menu:

- **Host Address**

The network address of the CDROM node (the workstation/server to which the CDROM drive is connected) must be entered in the format "XXXXXXXX.XX-XX-XX-XX-XX-XX."

- **User Name**

You must supply a valid username for the CDROM node.

- **Password**

You must supply a valid password for a user account on the CDROM node (if required on the CDROM node).

Press the **SAVE** key (PF-3) to connect to the CDROM node and invoke **newprod**, or the **CANCEL** key (PF-1) to return to the previous menu.

Once you invoke `newprod`, you must update the following products. These products are *nucleus products*, or the minimum software products required to run the CLIX operating system.

**Delivery Tools (DELTOOLS)**

Workstation/Server Diagnostics (DIAG)

Environ V Target Shared Library (ENVIRON\_S)

Gpipe Target Run-Time Shared Library (GPIPE\_S)

Workstation Network Software (INC)

System V.3.1 File Systems (SYSTEMV)

System V.3.1 Boot Images (UNIXBOOT)

Workstation Graphic Resources (RESOURCES)

I/Forms Run-Time Package (FORMS\_S)

RIS Client Support Package (RISCCU)

Screen Manager (SCREENMGR) (not needed for InterServe systems)

DEC VT220 Emulation (VT220) (not needed for InterServe systems)

Please refer to the *CLIPPER Software Delivery Guide* for instructions for using `newprod`.

After you have loaded software, reboot the system by selecting the Reboot option from the rebuild menu. Proceed to 11.13, "Preparing Nonstandard Partitions for Use."

### 11.13 Preparing Nonstandard Partitions for Use

The Make a file system option on the rebuild utility now automatically creates a device file, creates a file system, labels the partition, and adds the partition to the */etc/fstab* file (so that the file system will be mounted and checked each time the system boots). Thus, you are no longer required to perform these steps manually.

Proceed to 11.14, "Loading Application Software with the newprod Utility."

## 11.14 Loading Application Software with the newprod Utility

After you have prepared any nonstandard partitions for use, you must load application software. Now that the system is functional and all partitions are mounted, you can load this software on the hard disk.

Follow these steps to invoke newprod and deliver software:

1. Log in to the system and access the superuser account as follows

```
login: username
$ su
#
```

**Note:**

If you restored the */usr* file system(s), user accounts will no longer exist. Log in using the *sys* account. You will need to recreate all user accounts. Refer to 3.2, "Adding User Accounts."

2. Invoke the newprod utility as follows:

```
newprod
```

```
Enter source of installation: n)etwork, f)loppy, t)ape or r)emote cdrom or local c)drom -->
```

If you keyed in *n* for network or *r* for remote cdrom, you will be prompted to enter the network connect string. After rebuilding, you cannot key in the nodename of the delivery node; you must key in the Ethernet address.

3. The newprod menu appears. From this menu, you may *select* (highlight) products to download. Select and load all software products that you wish to use on the system. For instructions on using the newprod utility, refer to the *CLIPPER Software Delivery Guide*.
4. After all products have been successfully downloaded, continue to 11.15, "Restoring Files from Backups."

### **11.15 Restoring Files from Backups**

The final step in the process of rebuilding the hard disk is restoring files from backups.

After restoring the software products with the `newprod` utility, restore all user- and site-specific files that you backed up before beginning the rebuild procedure. For information on restoring files, refer to Chapter 9, "Backing Up and Restoring Files."





## **Appendix A: Keyboard Character Codes (Workstation Only)**

---

This appendix discusses transmitted and received keyboard character codes. This information applies only to Intergraph workstations; it does not apply to InterServe processors.



## Transmitted Character Codes

All light-colored typewriter keys on a workstation keyboard generate the ASCII character codes that correspond to the characters on the keys. The dark typewriter keys generate their respective ASCII control characters.

The table shown below contains the decimal and hexadecimal equivalents for characters on the Intergraph workstation keyboard.

The following keys do not generate character codes.

- Shift
- Alt Mode
- Ctrl
- Caps Lock
- Repeat
- Break (Shift-Select)

The terminal keypad in the upper left-hand corner of the keyboard contains the VT220 editing functions. The following character codes are transmitted from these keys:

| <i>Key</i>  | <i>Code</i> |
|-------------|-------------|
| Find        | Esc [ 1 ~   |
| Insert      | Esc [ 2 ~   |
| Remove      | Esc [ 3 ~   |
| Select      | Esc [ 4 ~   |
| Prev Screen | Esc [ 5 ~   |
| Next Screen | Esc [ 6 ~   |

The DEC VT220 F6 through F20 function keys are mapped to keys A1, B1, and C1 through A15, B15, and C15. Hence, pressing either key A1, B1, or C1, will execute the F6 function.

The Shift key, Alt Mode key, and 2nd F key do not affect the character codes generated by these keys. The keyboard mappings and their corresponding character codes follow.

---

| <i>Key</i> | <i>Keyboard Keys</i> | <i>Character Code</i> |
|------------|----------------------|-----------------------|
| F6         | A1, B1, C1           | Esc [ 1 7 ~           |
| F7         | A2, B2, C2           | Esc [ 1 8 ~           |
| F8         | A3, B3, C3           | Esc [ 1 9 ~           |
| F9         | A4, B4, C4           | Esc [ 2 0 ~           |
| F10        | A5, B5, C5           | Esc [ 2 1 ~           |
| F11        | A6, B6, C6           | Esc [ 2 3 ~           |
| F12        | A7, B7, C7           | Esc [ 2 4 ~           |
| F13        | A8, B8, C8           | Esc [ 2 5 ~           |
| F14        | A9, B9, C9           | Esc [ 2 6 ~           |
| Help       | A10, B10, C10        | Esc [ 2 8 ~           |
| DO         | A11, B11, C11        | Esc [ 2 9 ~           |
| F17        | A12, B12, C12        | Esc [ 3 1 ~           |
| F18        | A13, B13, C13        | Esc [ 3 2 ~           |
| F19        | A14, B14, C14        | Esc [ 3 3 ~           |
| F20        | A15, B15, C15        | Esc [ 3 4 ~           |

The character codes generated by the cursor keys (up, down, left, and right arrows) depend on the state of the Keyboard Setup menu. This menu can be set to either application cursor keys or normal cursor keys. The following are character codes generated under both settings:

| <i>Key</i>  | <i>Normal</i> | <i>Application</i> |
|-------------|---------------|--------------------|
| <i>Code</i> | Esc [ A       | Esc O A            |
|             | Esc [ B       | Esc O B            |
|             | Esc [ C       | Esc O C            |
|             | Esc [ D       | Esc O D            |

Character codes generated by the numeric keypad also depend on the state of the Keyboard Setup menu.

The character codes generated under both normal keypad keys and application keypad keys follow:

| <i>Key</i> | <i>Normal</i> | <i>Application</i> |
|------------|---------------|--------------------|
| 0          | 0             | Esc O p            |
| 1          | 1             | Esc O q            |
| 2          | 2             | Esc O r            |
| 3          | 3             | Esc O s            |
| 4          | 4             | Esc O t            |
| 5          | 5             | Esc O u            |
| 6          | 6             | Esc O v            |
| 7          | 7             | Esc O w            |
| 8          | 8             | Esc O x            |
| 9          | 9             | Esc O y            |
| -          | -             | Esc O m            |
| ,          | ,             | Esc O l            |
| .          | .             | Esc O n            |
| ENTER      | CR or CR LF   | Esc O M            |
|            | Esc O P       | Esc O P            |
|            | Esc O Q       | Esc O Q            |
|            | Esc O R       | Esc O R            |
|            | Esc O S       | Esc O S            |

## Received Character Codes

Vterm receives and transmits data. The received data, which may come from the host or keyboard, is control data that requires interpreting. Control data can be one of two types: control characters and control sequences. Control characters are the keys such as Tab, Line Feed, and Backspace. Control sequences are commonly referred to as escape sequences. Both types of control data along with their corresponding control characters or control sequences and a brief description of the action performed at the terminal are presented here.

### Control Characters

| <i>Control Character</i> | <i>ASCII Code</i> | <i>Action</i>                                                                                                                                                                                  |
|--------------------------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NULL                     | 00                | No action occurs.                                                                                                                                                                              |
| ENQUIRY                  | 05                | Generates and sends the designated <i>answerback</i> message to host. The answerback message can be defined on the Keyboard Setup menu. The default setting for this message is "Hello World." |
| BELL                     | 07                | Makes the keyboard beep if the bell is enabled.                                                                                                                                                |
| BACKSPACE                | 08                | Moves the cursor one character position to the left unless the cursor is at the left margin. If the cursor is at the left margin, no action occurs.                                            |
| TAB                      | 09                | Moves the cursor to the next tab stop or to the right margin if there are no more tab stops.                                                                                                   |
| LINE FEED                | 0A                | Depending on the new line setting, produces either a line feed or a new line.                                                                                                                  |
| VT                       | 0B                | Processes as a line feed.                                                                                                                                                                      |
| FORM FEED                | 0C                | Depending on the formfeed setting, processes as either a screen erase or a line feed.                                                                                                          |
| CARRIAGE RETURN          | 0D                | Moves the cursor to the left margin of the current line.                                                                                                                                       |
| SHIFT OUT                | 0E                | Designates the active character set as G1. If G1 is already active, SHIFT OUT has no effect.                                                                                                   |

|            |    |                                                                                             |
|------------|----|---------------------------------------------------------------------------------------------|
| SHIFT IN   | 0F | Designates the active character set as G0. If G0 is already active, SHIFT IN has no effect. |
| DC1 (XON)  | 11 | No action occurs.                                                                           |
| DC3 (XOFF) | 13 | No action occurs.                                                                           |
| CANCEL     | 18 | Terminates an escape sequence.                                                              |
| SUBSTITUTE | 1A | Terminates an escape sequence.                                                              |
| ESCAPE     | 1B | Introduces escape sequences.                                                                |

### Control (Escape) Sequences

A second type of control data received by `vterm` is the escape sequence. Escape sequences have the following general format:

#### Introducer Intermediate Final

- **Introducer** is the Esc character.
- **Intermediate** is the parameter with 0 as the default for most escape sequences.
- **Final** is the sequence terminator, an ASCII character between @ and ~.

In the list of escape sequences that follow, an `n` is used where a number would normally be placed.

#### Note:

To improve readability, spaces are placed between the components in the escape sequences in this document. Hence, an escape sequence is shown in this document as follows:

Esc [ 2 J

However, the sequence must be keyed in without spaces as follows:

Esc[2J

### Compatibility Level

| <i>Sequence</i>   | <i>Action</i>                     |
|-------------------|-----------------------------------|
| Esc [ 6 1 " p     | Sets the terminal for VT100 mode. |
| Esc [ 6 2 ; 1 " p | Sets the terminal for VT220 mode. |

### Cursor Positioning

| <i>Sequence</i> | <i>Action</i>                                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Esc [ n A       | Moves the cursor n lines up in the same column. Movement is restricted by the top and bottom margins of the scrolling region; the default value is one line.   |
| Esc [ n B       | Moves the cursor n lines down in the same column. Movement is restricted by the top and bottom margins of the scrolling region; the default value is one line. |
| Esc [ n C       | Moves the cursor n columns to the right. Movement is restricted by margin setting; the default value is one column.                                            |
| Esc [ n D       | Moves the cursor n columns to the left. Movement is restricted by the margin setting; the default value is one column.                                         |
| Esc [ n ; n H   | Moves the cursor to line n, column n. The state of the origin mode flag determines the numbering of the lines and columns; the default value is home.          |
| Esc [ n ; n f   | Same as Esc [ n ; n H (preceding).                                                                                                                             |
| Esc D           | Moves the cursor down one line in the same column; if the cursor is at the bottom margin, the screen scrolls upward.                                           |
| Esc E           | Moves the cursor to the first position on the next line; if the cursor is at the bottom margin, the screen scrolls upward.                                     |
| Esc M           | Moves the cursor up one line in the same column; if the cursor is at the top margin, the screen scrolls downward.                                              |

Esc 7 Saves attributes and the cursor position; the following attributes are saved:

- Cursor position
- Autowrap state
- Character attributes (such as block, visible)
- Character set state
- Origin mode state
- Selective erase state

Esc 8 Restores attributes and the cursor position saved using Esc 7; if an Esc 7 was not previously issued, the cursor moves home, the origin mode is reset (origin is upper left), no character attributes are assigned, and autowrap resets if no characteristics were saved.

### Character Set Selection

| <i>Sequence</i> | <i>Action</i>                                              |
|-----------------|------------------------------------------------------------|
| Esc ( 0 - G0    | Designates graphics character set as the G0 character set. |
| Esc ( B - G0    | Designates ASCII character set as the G0 character set.    |
| Esc ) 0 - G1    | Designates graphics character set as the G1 character set. |
| Esc ) B - G1    | Designates ASCII character set as the G1 character set.    |

Refer to the Control Characters section on page A-6. See the SHIFT OUT and SHIFT IN ASCII control characters to set the active character area to G0 or G1.

## Terminal Modes

| <i>Sequence</i> | <i>Action</i>                                                                                                                 |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------|
| Esc [ 2 0 h     | Designates the control character LF as CR LF.                                                                                 |
| Esc [ 2 0 1     | Designates the control character LF as LF and the control character CR as CR.                                                 |
| Esc [ 4 h       | Sets insert mode.                                                                                                             |
| Esc [ 4 1       | Sets replacement mode.                                                                                                        |
| Esc [ ? 1 h     | Generates cursor key codes in normal mode.                                                                                    |
| Esc [ ? 1 1     | Generates cursor key codes in application mode.                                                                               |
| Esc [ ? 3 h     | Sets 132-column mode.                                                                                                         |
| Esc [ ? 3 1     | Sets 80-column mode.                                                                                                          |
| Esc [ 12 h      | Turns off local echo.                                                                                                         |
| Esc [ 12 1      | Turns on local echo.                                                                                                          |
| Esc [ ? 2 s h   | Makes the cursor visible.                                                                                                     |
| Esc [ ? 2 s 1   | Makes the cursor invisible.                                                                                                   |
| Esc [ ? 5 h     | Sets reverse video (light background, dark text).                                                                             |
| Esc [ ? 5 1     | Sets normal video (dark background, light text).                                                                              |
| Esc [ ? 6 h     | Designates row 1 as the first row of the scrolling region. (Home is the upper left-hand region of the defined scroll region.) |
| Esc [ ? 6 1     | Designates row 1 as the first row on the window. (Home is the upper left-hand region of the window.)                          |
| Esc [ ? 7 h     | Activates autowrap mode. (Movement to the next line may cause a scroll depending on the state of the scroll mode.)            |
| Esc [ ? 7 1     | Deactivates autowrap mode.                                                                                                    |

|               |                                                                                                                                    |
|---------------|------------------------------------------------------------------------------------------------------------------------------------|
| Esc [ > 4 h   | Causes incoming formfeeds to be interpreted as a clear display and move to home.                                                   |
| Esc [ > 4 1   | Causes incoming formfeeds to be interpreted as linefeeds.                                                                          |
| Esc [ > 9 1   | Causes text on screen to be preserved when modes switch between 80 and 132 column.                                                 |
| Esc [ > 9 h   | Causes clear screen and home functions to be performed when the column mode is changed.                                            |
| Esc [ > 1 1 h | Causes erase commands to be executed relative to the defined scroll region.                                                        |
| Esc [ > 1 1 1 | Causes erase commands to be executed relative to the entire window.                                                                |
| Esc [ > 1 2 h | Puts spaces between the current position and the next tab stop when a tab is placed.                                               |
| Esc [ > 1 2 1 | Moves the cursor directly to the next tab stop without affecting the characters between the current position and the new position. |
| Esc [ > 1 4 h | Activates local echo.                                                                                                              |
| Esc [ > 1 4 1 | Deactivates local echo.                                                                                                            |
| Esc =         | Sets the keypad translation to application mode.                                                                                   |
| Esc >1        | Sets the keypad translation to normal mode.                                                                                        |

The escape sequences that set up terminal modes can contain up to 30 intermediate characters separated by semicolons. For example, the following sequence sets the number of columns to 132, sets the reverse video to on, and sets the autowrap to on.

Esc [ 3;5;7 h

### Tab Stops

| <i>Sequence</i>          | <i>Action</i>                             |
|--------------------------|-------------------------------------------|
| Esc H                    | Sets a tab stop at the current column.    |
| Esc [g<br>or<br>Esc [ Og | Clears a tab stop at the cursor position. |
| Esc [ 3 g                | Clears all tab stops.                     |
| Esc [ > 5g               | Sets a tab stop at every eighth column.   |

### Character Attributes

Esc [ n ; n ; ... n m

The character attributes sequence can contain up to 30 intermediate characters separated by semicolons. If no parameter is given, the default is assumed to be zero. The intermediate characters are defined as follows:

|    |                    |
|----|--------------------|
| 0  | All attributes off |
| 1  | Bold               |
| 4  | Underscore         |
| 7  | Reverse video      |
| 22 | Not bold           |
| 24 | Not underscored    |
| 27 | Normal video       |
| 30 | Black text         |
| 31 | Red text           |
| 32 | Green text         |
| 33 | Yellow text        |
| 34 | Blue text          |
| 35 | Magenta text       |
| 36 | Cyan text          |
| 37 | White text         |
| 40 | Black text block   |
| 41 | Red text block     |
| 42 | Green text block   |
| 43 | Yellow text block  |
| 44 | Blue text block    |
| 45 | Magenta text block |
| 46 | Cyan text block    |
| 47 | White text block   |

**Erase Attributes**

Esc [ n " q

The intermediate character defaults to 0. The meanings of the intermediate character are as follows:

- 0 All attributes off
- 1 Nonerasable character
- 2 Erasable character

The above character attributes pertain to the erase commands only (Esc .. K and Esc .. J).

**Line Attributes**

| <i>Sequence</i> | <i>Action</i>                                                                                                                                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Esc \# 3        | Sets line to the top half of the double-high, double-wide font.                                                                                                                                                                       |
| Esc \# 4        | Sets line to the bottom half of the double-high, double-wide font; the same character must be used on both lines for a complete image; if the line previously had a single width, all characters to the right of the center are lost. |
| Esc \# 5        | Sets line to single-high, single-wide.                                                                                                                                                                                                |
| Esc \# 6        | Sets line to single-high, double-wide.                                                                                                                                                                                                |
| Esc \# :        | Sets line to the top half of double-high, single-wide font.                                                                                                                                                                           |
| Esc \# ;        | Sets line to the bottom half of double-high, single-wide font; as in double-high, double-wide, the same character must be typed on consecutive lines to form the entire image.                                                        |

## Editing

| <i>Sequence</i> | <i>Action</i>                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------|
| Esc [ n L       | Inserts n lines at the cursor position; the default is one line.                                       |
| Esc [ n M       | Deletes n lines at the cursor position; the default is one line.                                       |
| Esc [ n @       | Sets inserted character attributes to normal and inserts n blank characters at the cursor position.    |
| Esc [ n P       | Deletes n characters starting with the character at the cursor position; the default is one character. |

## Erasing

| <i>Sequence</i>         | <i>Action</i>                                                                                                                    |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| Esc [ n X               | Erases the next n-1 characters and the character at the cursor position; the default erases only the character under the cursor. |
| Esc [ K or<br>Esc [ 0 K | Erases from the cursor position to the end of the line (including the cursor position).                                          |
| Esc [ 1 K               | Erases from the cursor position backward to the beginning of the line (including the cursor position).                           |
| Esc [ 2 K               | Erases an entire line.                                                                                                           |
| Esc [ J or<br>Esc [ 0 J | Erases from the cursor position to the end of the window (including the cursor position).                                        |
| Esc [ 1 J               | Erases from the cursor position to the top of the window (including the cursor position).                                        |
| Esc [ 2 J               | Erases the entire display (subject to the erase extents mode).                                                                   |
| Esc [ ? n K             | Same as the Esc [ K sequences except that only erasable characters are erased.                                                   |
| Esc [ ? n J             | Same as the Esc [ J sequences except that only erasable characters are erased.                                                   |

**Set Margins**

| <i>Sequence</i> | <i>Action</i>                                                                                                                                                                                                                                                                                                            |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Esc [ n ; n r   | Sets top and bottom margins for the scrolling region and moves the cursor to home (relative to the window); the first n is the number of the first line in the scrolling region; the second n is the number of the last line in the scrolling region. Both default to top and bottom of the window respectively, if set. |

**Note:**

No protection is provided for setting scrolling regions that are not in the visible virtual terminal window.

**Set Terminal LEDs**

| <i>Sequence</i> | <i>Action</i>                                                                                                                                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Esc [ n q       | This sequence may have up to 30 intermediate characters separated by semicolons; the default is 0.<br><br>0 Turns off all LEDs.<br>1 Turns on LED 1.<br>2 Turns on LED 2.<br>3 Turns on LED 3.<br>4 Turns on LED 4. |

**Set Cursor Attributes**

This sequence may have up to 30 intermediate characters separated by semicolons. The default is 0.

|           |                                     |
|-----------|-------------------------------------|
| Esc [ 0 v | Sets cursor to visible.             |
| Esc [ 1 v | Sets cursor to invisible.           |
| Esc [ 2 v | Sets cursor to underline.           |
| Esc [ 3 v | Sets cursor to reverse video block. |

| <i>Sequence</i>  | <i>Action</i>                                          |
|------------------|--------------------------------------------------------|
| Esc [c or Esc [Z | Requests information about the virtual terminal setup. |

- Esc [ ? 62; 1; 2; 6c Specifies the following:
- VT220 term type (62)
  - 132 columns (1)
  - Printer port (2)
  - Selective erase (6)
- Esc [ 5 n Requests information about the virtual terminal status.
- Esc [ 0 n Replies to the virtual terminal status request; specifies that no malfunction occurred.
- Esc [ 6 n Requests information about the cursor position.
- Esc [ r ; c R Replies to the cursor position request; specifies that the cursor is at r line and c column.

## Appendix B: Read/Write Compatibility of Floppy Disks with Intergraph Workstations and IBM Personal Computers

---

This appendix discusses the read/write compatibility of floppy disks with Intergraph workstations versus their compatibility with IBM personal computers.

### Notes:

- The IBM PC/AT computer and the Intergraph workstation with PC-DOS running have total read/write compatibility with floppy diskettes. Another machine can read from or write to floppies formatted on either machine.
- IBM PC/AT and IBM PC machines are less than 100 percent reliable when reading from and writing to each other's floppies.
- Likewise, Intergraph workstations with PC-DOS running and IBM PC machines are less than 100 percent reliable when reading from and writing to each other's floppies.

Reading and writing to the same floppy disk on the two IBM machines or between the Intergraph workstations with PC-DOS running and an IBM PC are possible. With the **VERIFY ON** option, a read/write success rate of approximately 95 percent has been achieved between the IBM PC/AT and the IBM PC and between the Intergraph workstation with PC-DOS running and the IBM PC. Using the editor of the machine(s) in question, enables verification by appending the **VERIFY ON** command to the *autoexec.bat* file.

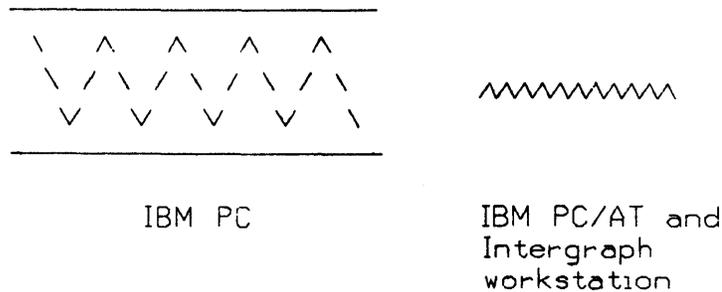
### Note:

Adding the **VERIFY ON** option to the *autoexec.bat* file doubles the time required for all input and output. If you choose not to add **VERIFY ON** to the *autoexec.bat* file, you may use the **/v** option of the DOS copy command.

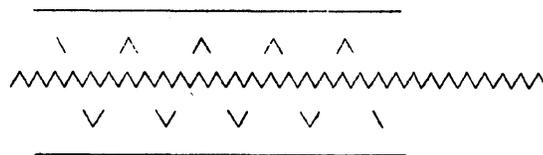


The following is a detailed explanation of why reading/writing to floppies between the IBM machines and between the Intergraph workstation and the IBM PC is difficult.

The following diagram represents a portion of a track on a floppy diskette. Representations of the data patterns written by the three machines in question appear in the track. The pattern written by the floppy drives of the IBM PC/AT and the Intergraph workstation is narrower than the pattern written by the IBM PC.



For example, suppose you format a floppy on an IBM PC and use the PC to write data to that floppy. Then, you take the same floppy to an IBM PC/AT or an Intergraph workstation and write to it from that machine. The following diagram indicates how a track on the floppy might look after both machines have written to it.



As the diagram shows, data remains on the track from the previous IBM PC write operation. The IBM AT or Intergraph workstation that wrote to the track most recently will probably have no trouble reading back the overlaid information because of the narrower range of the track's floppy read/write head. However, if this floppy is taken back to the IBM PC, erroneous data may be read because of the wider range of its floppy drive read/write head.

During write operations, the **VERIFY** option causes data to be written and then verified by re-reading it. **VERIFY** retries the write operation until it reads back exactly what was written or until it reaches some fixed number of reattempts.

The safest way to ensure floppy data integrity is to avoid mixing write operations between the two IBM machines or between the IBM PC and the Intergraph workstation(s). However, as previously mentioned, with **VERIFY ON**, you can achieve a success rate of approximately 95 percent when reading and writing to floppy disks between these machines.

## Appendix C: System V/PC-DOS Utilities (Workstation Only)

---

Intergraph supplies three UNIX utilities that allow file exchange between System V and PC-DOS: `dtu`, `utd`, and `dls`. These utilities are delivered with the `systemv` product and stored in the `/usr/bin` directory. For more information on these utilities, see the *CLIX Programmer's and User's Reference Manual*.

By default, any manipulations of PC-DOS files automatically default to the floppy disk drive (a:) unless the hard disk drive specification, `c:`, is used. For example, if you want to copy a file from the System V operating system to PC-DOS, the utility attempts to copy the System V file to a PC-DOS floppy disk if one is present in the floppy disk drive. This appendix provides descriptions and examples illustrating how to use these utilities.



## dtu (DOS to UNIX)

The System V `dtu` command copies files from a PC-DOS file system to a System V file system.

The following two formats are used to issue the `dtu` command:

- `# dtu <dosfile> <unixfile>`

For example, to copy the PC-DOS file `dos.dat` to the System V file `/usr/bill/dos`, key in the following:

```
dtu dos.dat /usr/bill/dos
```

- `# dtu <dosfile> [dosfile2...] <unixdir>`

For example, to copy the PC-DOS files `bill.dat` on the a: drive and `bill.txt` on the c: drive to the System V directory `/usr/bill`, key in the following:

```
dtu a:bill.dat c:bill.txt /usr/bill
```

### Note:

The `dtu` utility accepts the DOS wildcard characters "\*" and "?". You must use the quotation marks (" ") around the characters as follows so that the UNIX shell does not expand them before invoking `dtu`.

```
dtu a:"*" .dat /usr/bill
```

## Redirecting the Output of dtu

The following is an example of how to redirect the output of the `dtu` utility:

```
dtu a:file.c | pg
```

This example allows the user to display the contents of a DOS file without actually overwriting it.

### Note:

If you are copying executable files, use the `-p` option of `dtu` as in the following example. This option does not translate DOS carriage returns to UNIX linefeeds.

```
dtu -p a:bill.dat /usr/bill
```

## utd (Unix to DOS)

The System V `utd` command copies files from a System V file system to a PC-DOS file system. As in the previous section, the PC-DOS `a:` drive is assumed unless another drive is specified.

### Copying Individual Files

The following two command formats are used to copy individual files from System V to DOS using the `utd` command:

- `# utd <unixfile> <dosfile>`

For example, to copy the System V file `/usr/bill/text` to the PC-DOS file `bill.txt`, key in the following:

```
utd /usr/bill/text bill.txt
```

- `# utd <unixfile> [unixfile2...] <dosdirectory>`

For example, to copy the System V files `/usr/bill/text` and `/usr/bill/data` to the PC-DOS directory `bill.dir`, key in the following:

```
utd /usr/bill/text /usr/bill/data bill.dir
```

#### Note:

If you are copying executable files, use the `-p` option of `utd` as in the following example. This option provides for no translation of UNIX linefeeds to DOS carriage returns.

```
utd -p /usr/bill/text bill.dir
```

### Copying Directories

To copy all the files in one UNIX directory to a DOS directory (and have all the files retain their original names), use one of the following methods:

```
utd <unixdirectory>/* <dosdirectory>
```

OR

```
cd <unixdirectory>
utd * <dosdirectory>
```

For example, key in the following to copy all System V files in */usr/bill* to the PC-DOS directory *bill*:

```
utd /usr/bill/* /bill
```

OR

```
cd /usr/bill
utd * /bill
```

Either command will copy each file in the directory. Individual files are stored separately (instead of in one large file) with each file retaining its original name.

#### **Redirecting the Input of utd**

The following is an example of how to redirect the input of the *utd* utility:

```
pr -f *.c | utd a:prfile
```

## dls (DOS Listing)

The `dls` command allows you to list the contents of a PC-DOS directory from System V. The PC-DOS `a:` drive is assumed unless another drive is specified. This command uses the following syntax:

```
dls [-altrR] <name1> [<name2> ...]
```

The following options are available for the `dls` command:

`-a` lists all entries, including hidden files, system files, dot (`.`), and dot dot (`..`).

`-l` lists in long format, giving the following information:

- Filename
- File conditions denoted with a character flag if the condition is present and a blank if it is not. The following are character flags:

`r` denotes a read-only file.

`h` denotes a hidden file.

`s` denotes a system file.

`v` denotes a volume label.

`d` denotes a directory.

`a` specifies that the entry has been archived.

- Date and time of last modification.
- Index number of the beginning file allocation table (FAT) entry for the file.
- Size of the file in bytes.

`-t` sorts entries by time of last modification with most recent time first (instead of sorting alphabetically).

`-r` reverses sorting order.

`-R` recursively lists subdirectories as each is encountered.

To list the PC-DOS directory `bill.dir` on the `a:` drive, key in the following:

```
dls bill.dir
```

**Note:**

The `dls` utility accepts the wildcard characters "\*" and "?". You must use the quotation marks (" ") around the characters so that the UNIX shell does not expand them before invoking `dls`.

For example, the following command lists all files with the `.c` extension in the current UNIX working directory:

```
dls *.c
```

In addition, the following command lists all files with the `.c` extension on the floppy disk:

```
dls a:*.c
```







## Glossary

---

This glossary defines unfamiliar terms used in the *CLIPPER System Administrator's Guide*.



**12-Button Cursor**

An absolute tracking input device used to move the cursor on the workstation screen.

**1600/6250-bpi Tape Drive**

A dual-density streaming magnetic tape drive that reads from and writes to ANSI-standard nine-track, half-inch magnetic tape in 6250- or 1600-bpi (bits per inch). It is a direct reel-to-reel servo tape drive consisting of a tape transport mechanism, tape motion control electronics, dual-density read/write circuitry, and data-formatting electronics.

**Absolute Pathname**

The pathname used to access any file in the file system. An absolute pathname begins at root (/) and works down through the hierarchy; the last name is the name of the file or directory.

**Active Process**

The VT220 window that controls the Process ID field, the message fields, the window control strip, and the keyboard. The active process has a highlighted window icon strip.

**addusers Utility**

System administrative utility that automates the process of adding user accounts.

**Alphanumeric Character**

An alphabetic or numeric character.

**ANSI Color Escape Sequences**

Command sequences that allow the text color and text blocks (the area surrounding each character) to be changed for VT220 window display.

**ansitape Command**

An ANSI-standard magnetic tape label program that reads, writes, and creates magnetic tapes conforming to the ANSI standard for magnetic tape labeling. The primary use for this utility is to enable tapes to be exchanged with those made with VAX/VMS. **ansitape** is useful for transferring design files to remote locations that do not have a network link.

**ASCII**

Acronym for American National Standard Code for Information Interchange. ASCII is a code that uses seven bits to represent graphic and control characters. Generally, this guide will refer to an ASCII file as a text file.

**Auxiliary Port**

RS-232-compatible serial port. Intergraph workstations/servers have three auxiliary ports labeled aux00, aux01, and aux02.

### **Background Process**

A process initiated by a command line that ends in an ampersand (&) character. If a process is executing in the background, you do not have to wait for it to complete before issuing additional shell commands. (A process not executed in the background is a foreground process.)

### **Backup**

A file or directory copy, usually saved on a storage device such as a tape or floppy disk. Backup files prevent data loss if the system crashes.

### **backup Command**

A Berkeley command ported to the Intergraph workstation. This command is used with the Berkeley restore command; backup writes a tape and restore recovers the tape. backup and restore are used only for backing up and recovering entire file systems. These utilities have a helpful user interface and good error recovery mechanisms. In addition, backup and restore can write and read multiple-volume savesets.

### **Bad Blocks**

Flawed areas on the hard disk. As the disk drive writes data to the disk, it calculates an Error Correction Code (ECC) for each block and stores the code at the end of each block. When the drive reads data from the disk, it calculates another ECC and compares it to the original stored value. If the codes match, the data stored in that block is healthy and uncorrupted. If the codes do not match, the drive generates an ECC error and the host displays a medium error (for 156-MB CDC disks) or a bad block message (for 355- and 670-MB Maxtor disks).

### **Baseline**

Software delivered free-of-charge with an Intergraph workstation or server. There are two baseline categories: nucleus and supplemental. Baseline nucleus software must reside on the workstation/server before the System V operating system can run. Baseline supplemental software is delivered free, but does not need to be loaded on the workstation/server.

### **Baud Rate**

Transmission speed, or the measure of the number of bits per second (bps) transmitted over a serial communication channel. This number is typically about ten times greater than the number of characters transmitted per second. The baud rate for a workstation depends on the device connected to the port. The default setting on an Intergraph workstation is 9600.

**Block Device**

A disk or tape drive that stores information in blocks of characters. The device file for a block device is stored in the */dev/dsk* directory. (If a device is not a block device, it is a character device.)

**Block Major Number**

The number that represents the disk SCSI address and unit. The block major is an argument in the *mknod* command, which creates a device file.

**Block Minor Number**

The number that represents the disk partition and modifier. The block minor is an argument in the *mknod* command, which creates a device file.

**Boot**

To power up a workstation/server, perform diagnostics tests, and load software. Also referred to as a *cold boot*.

**Boot Partition**

The partition that contains boot images, including the hardware diagnostics test, utility pages, communications processor, and System V kernel software. This partition contains 3,988 blocks for workstations with 80-MB hard disks and 7,988 blocks for all other hard disk sizes. You cannot access this partition on the Disk Partitioning Utility page because you cannot alter its size.

**Bootable Images**

The System V Boot Images, Plotter Boot Images, Diagnostics, Startup Utilities Menus, and Ethernet CP Images. The workstation/server bootable images are loaded in memory and executed when the system is powered up.

**Bourne Shell (sh)**

The standard System V shell. The system administrator's root account uses the Bourne Shell (sh).

**Cartridge Tape Drive (optional)**

A high capacity 1/4-inch tape drive that uses standard, readily available, 600-foot DC600A or equivalent tapes to store 60 MB formatted data. The recording format used is QIC-24, which is an industry standard.

**Catalog Function**

The feature that saves a customized virtual terminal window. If you used the Setup menus and *regis* color combinations to customize a window, you can save the window characteristics so that you can use the window regularly. The Catalog function allows you to save the characteristics of that virtual terminal. You can then add the cataloged file to the Workstation pull-down menu to access the cataloged window regularly.

**CDROM (Compact Disk Read Only Memory)**

A storage medium used to deliver software products. One delivery CDROM holds approximately 600 MB of data, allowing you to load software from only a few CDROMs rather than from many tapes or floppy disks. CDROM is *read-only*; you cannot write data to it.

**Character Device**

A raw device (usually a terminal, printer, or modem) that displays characters one at a time. The device file for a character device is stored in the */dev/rdisk* directory. (If a device is not a character device, it is a block device.)

**Checkbox**

Menu devices used to select from a list of items.

**Clearinghouse**

A utility that provides a means of translating the nodename to the node address.

**CLIPPER Processor**

The main processor for Intergraph workstations and servers.

**CLIX**

The operating system that Intergraph workstations/servers are based on. Intergraph ported AT&T's UNIX System V operating system version 5.3.1 to the CLIPPER processor and enhanced it with Berkeley and Intergraph extensions. This enhanced operating system is commonly called the CLIX operating system.

**CLIX Resource Monitor (CRM)**

A system monitor for the CLIX operating system. System managers and end users may use CRM to monitor either the system or individual processes in the system. Programmers may also use CRM during program development to analyze program behavior. CRM is delivered as part of the CLIPPER System Software that comes free with all Intergraph CLIPPER workstations and servers.

**Coaxial Cable**

The primary cable through which workstations/servers and hosts communicate.

**Collapse Icon**

The window manipulation icon used to compress the window area so that only the delete icon, the window name field, and the collapse icon display.

**Command Line**

A string of commands and arguments that execute the desired function. Usually, the user keys this string in at the system shell prompt.

**Command Recall**

A shell feature that allows you to recall previous commands without re-entering the command. The *history* file in the user's home directory stores all previously executed commands.

**Communications Processor (CP)**

The device necessary for Intergraph workstations/servers and Intergraph VAXs to qualify and function as nodes on an IEEE 802.3 Local Area Network.

**Console Device**

A serial line terminal connected to a host computer.

**Console Window**

A mechanism used to display the System V operating system messages. The console window emulates a VT220 terminal.

**core Files**

Image files written by System V when problems such as memory violations, illegal instructions, bus errors, and user-generated quit signals occur.

**cpio Command**

A System V command used to copy the specified input to a specified output device. (cpio stands for copy file archives in and out.)

**cron (Commands Run On Notice)**

The command that executes commands at specified dates and times.

**CRT Bias**

A voltage applied to the CRT to establish a reference level for operation. CRT bias adjusts the intensity of the raster display. Horizontal lines appearing across the display screen are signs that the CRT bias needs adjusting.

**C Shell (csh)**

The BSD standard shell ported to the Intergraph workstation/server. C Shell uses syntax similar to C; supports command-line substitution; and incorporates a history mechanism, job control facilities, and interactive filename and username completion.

**Cursor**

The arrow, block, or underscore that displays on the screen and indicates the point of input (from the mouse, 12-button cursor, or keyboard).

**Data Entry Field**

The field on a menu used to accept user-supplied data.

#### **dd Command**

The backup utility used to move raw binary data or files to and from the storage device (hard disk, tape, or floppy disk). This utility is useful for manipulating raw data transferred to a tape from another operating system or for imaging a disk drive by reading input from a raw partition.

#### **def\_proc File**

A file that specifies system and user processes to be invoked automatically. The `/usr/tp32/smgr/def_proc` file specifies processes for all users, while the `/usr/<login>/def_proc` specifies processes for only one user.

#### **Delete Icon**

A window manipulation icon used to remove a window from the screen. You should stop the process owning the window before you delete the window. Deleting the window will usually stop the process.

#### **Delivery Tools**

A software product that contains a group of utilities that simplifies software delivery to the workstation/server. The utilities provided in the delivery tools are `newprod`, `menuprod`, `getmenus`, `fr_flop`, `dates`, `makenode`, and `compress`. Each of these utilities is linked to `/usr/bin` so that the full pathname does not need to be keyed in to invoke the utility.

#### **Device File**

A file in the `/dev` directory that allows you to access a hard disk partition or storage device such as a tape drive. Device files are also referred to as *special files*.

#### **Diagnostics**

Disk-resident tests performed on the hardware of the workstation/server when the system is powered up. During the diagnostic tests, messages reporting test results appear on the screen.

#### **Digitizer Processor Board**

The board that controls all digitizing table, 12-button cursor, and floating menu input/output. The I/O processor board (Intel 80186 or 80386) handles all digitizer processor data for the workstation/server in System V (workstation only).

#### **Digitizing Table**

A flat surface used for placing paper menus or tracing maps, drawings, and photos.

**Directory**

A logical structure that contains filenames and subdirectories. To better organize your files, you can group related files together in a common directory.

**Disk Maintenance Utilities Page**

The main hard disk utility page from which the following hard disk utility pages can be directly or indirectly accessed: Hard Disk Flaw Data Utility page, Verify page, and Format page.

**Disk Partitioning Utility Page**

The Utility page used to specify how a hard disk is to be partitioned.

**Double Tap**

To quickly press and release a cursor button twice.

**ed**

A UNIX line editor that manipulates text line-by-line.

**EMACS**

A screen editor that contains three components: EMACS-tc, MicroEmacs, and MousEmacs. The EMACS product is delivered free-of-charge with each workstation/server.

**EMUX (Ethernet Multiplexer)**

A device that can be used on a transceiver and on an Intergraph VAX to allow up to eight transceiver cables to be connected.

**Environ V Library**

The C language function library containing Intergraph-developed routines that programmers can use to develop their own applications.

**Error Boxes**

Areas that display system messages you must act on before continuing a process. Error boxes visibly inform you of fatal errors, warnings, or system messages.

**Etherjack**

An optional 15-pin wall outlet used to connect an office transceiver cable to a standard transceiver cable.

**Ethernet Address**

The unique identifier hard-wired on each node. The Ethernet address is needed for each node to identify and communicate with another node on the network. The Ethernet address is also referred to as the *network address*.

**Ethernet Port**

The IEEE 802.3 interface for the Intergraph workstations/servers. This port allows 15-pin transceiver cable plugs to attach to this port.

**Fast File System (FFS)**

A way of organizing data stored on a magnetic disk drive to enhance speed for disk read and write operations. The new file system clusters sequentially accessed data and provides two block sizes to allow fast access to large files while not wasting large amounts of space for small files. A Fast File System accesses files up to 10 times faster than the standard UNIX file system does.

**features.com File**

A product information file that provides a general product description, delivery constraints, delivery prompts, and related documentation. This file resides in each product's directory.

**File Management Utility (fmu)**

A software program that enables an Intergraph workstation/server to communicate with any node on an IEEE 802.3 network.

**File System**

A directory structure that allows users to access partitions. Examples of file systems include */usr*, */usr2*, and *root (/)*. There are two types of file systems: Fast File Systems (FFS) and standard file systems. The *newfs* command creates a FFS and the *mkfs* command creates a standard file system. Recreating a file system destroys any data in the existing file system. The Rebuild process creates a file system automatically.

**find Command**

A command for locating filenames that match qualifications specified on the command line.

**fixes.com File**

A product information file that contains information regarding any changes to the latest Intergraph software delivery.

**Floating Menu**

A flat, rectangular tablet that allows you to attach a paper menu to its surface and then move the floating menu freely about the digitizing table without needing to reinitialize the paper menu. The floating menu is an optional peripheral for the InterPro and InterAct workstations. A floating menu comes with the digitizing table that is purchased separately for InterView workstations.

### **Floppy Disk Drive**

A storage device, located in the monitor head, used to access and archive software for System V and PC-DOS. The floppy disk drive uses standard 5.25-inch floppy disks. It can use both high-density (1.2 MB) and low-density (360K) floppy disks.

### **Format**

A process that destroys all data on the hard disk. The format process involves structuring the disk so that hardware and software can communicate with it. This process writes a test pattern on the disk and reads it to verify the pattern, checks for errors, and marks disk locations that cause errors. These flaws are recorded to prevent future writing to or reading from that location.

### **fr\_flop Command**

A filter command that allows you to retrieve data from a set of floppy disks created with the `to_flop` utility.

### **Frame Buffer Board**

A board that stores the 6 MB data that defines all displayed pixels. This board also transforms this data into the red, green, and blue analog video signals sent to the monitor.

### **fsck Command**

The command that checks file system consistency and interactively repairs it. This System V program is executed when the system is powered on or rebooted if the unit was not shut down properly.

### **Function Key Menu**

A menu composed of the membrane keys at the top of the keyboard. These keys have been predefined for specific commands.

### **Function Keys**

The 48 membrane keys located across the top portion of the keyboard. These keys are mapped for VT220 emulation and IBM-PC emulation.

### **getty (/etc/getty)**

The second process invoked in the system initialization procedure. `getty` reads the `/etc/gettydefs` file to determine terminal settings and spawns the `login` process.

### **Graphics Resources**

Software that provides video look-up tables (VLTs), text fonts, symbols, and standard system error messages. Graphics Resources are used in programs such as `Menubuilder` and `Procasso`, and are available to use in application programs being developed.

### **Group Name/Number (GID)**

The identifier in the */etc/passwd* file that assigns new users to groups. The GID may range from 1 to 65,353. All user accounts created with *addusers* default to a GID of 500; those created with *sysadm* default to a GID of 1. Users with the same GID are able to access one another's files based on the second set of read, write, and execute permissions.

### **Hard Disk Drive**

An internal storage device included with each Intergraph workstation/server. Intergraph provides 156-, 355-, 584-, and 670-MB hard disk drives for its workstations/servers. The 156-, 355-, and 670-MB hard drives are 5.25-inch Winchester disk drives. The 584-MB drive is an 8-inch Winchester disk drive.

### **Hard Disk Flaw Data Utility Page**

The Utility page used to identify any hard disk flaws and to record flaws that need to be added (or removed) when the hard disk is formatted. The flaws are mapped to the addressed hard disk during formatting.

### **Help Facility**

An online reference facility that provides quick access to application command definitions.

### **Hierarchical Menu**

A menu type found in standalone and VAX-based applications. The menus are called up in the working area of the screen.

### **Hidden File**

An ordinary file that begins with a period (.) character. The period (.) prevents the file from being listed with the *ls* command. You can, however, list all files including hidden files with the *ls -a* command.

### **Home Directory**

The directory associated with a user's account. The home directory is normally the directory that the user logs in to.

### **I/O Processor Board**

The PC board used by pre-6000-series Intergraph systems. This board controls input/output functions for both System V and PC-DOS, initiates power-up and reboot diagnostics, provides the clock/calendar for the date and time, and is the main processing unit for PC-DOS functions. For 6000-series Intergraph workstations and servers, these processes are performed by the CLIX kernel.

**IEEE 802.3 Local Area Network (LAN)**

The physical layer of cables that provide the hardware required to run network software. This software allows Intergraph workstations/servers and Intergraph VAXs to communicate.

**init (/etc/init)**

The first process invoked upon system initialization. **init** specifies the run level (as defined by the */etc/inittab* file) and invokes the **getty** process.

**InterAct Workstation**

An Intergraph graphics workstation with an attached digitizing table.

**Intergraph Network Core Monitor (INCMON)**

A network monitor utility that can assist in diagnosing network problems or can gather information relating to the current network load and performance.

**Internet Address**

The 32-bit address assigned to each host that uses the TCP/IP, NQS, NFS, or RFS products.

**InterPro Workstation**

An Intergraph graphics workstation without a digitizing table.

**InterServe Processor**

Intergraph's nongraphics server.

**InterView Workstation**

Intergraph's dual-monitor graphics workstation with a separate digitizing table.

**Introductory Screen**

The screen first seen at boot-up or reboot. This screen displays the Intergraph logo and four screen buttons: System V, PC-DOS, Utility, and Help.

**Kernel**

The lowest level of the System V operating system. The kernel allocates system resources, maintains file systems, manages memory, controls workstation access, and manipulates the workstation/server hardware.

**Keyboard**

The peripheral input device composed of terminal emulation keys, function keys, typewriter keys, and a numeric keypad. The keyboard is used for entering text and numbers, issuing special keyboard commands, and selecting function-key menu commands. The keyboard functions according to the requirements of the active emulation. This emulation can be an InterAct emulation, an alphanumeric terminal, or an IBM PC.

**Korn Shell (ksh)**

An enhanced version of the Bourne shell (the standard for System V). Korn shell (ksh) is the default for user accounts created with **addusers**.

**Left (Command) Button**

The left button on the 12-button cursor used to invoke hierarchical menus (single tap), place a 3D data point (double tap), or perform the same functions as the left button of the MicroSoft mouse (IBM PC emulation).

**Local Window**

A window connected to the local System V operating system rather than to a network host.

**Login Account**

A user account. This is the account the user is placed in when he/she logs in to the system. Each line or record in the */etc/passwd* file represents a different login account.

**Login Directory**

A user's home directory. When a user logs in to the system, he/she is placed in this directory automatically. If the user account was created with **addusers** or **sysadm**, the default login directory is */usr/<login>*.

**lost+found Directory**

The directory where files *found* by the **fsck** utility are placed. The **fsck** program runs automatically when the system boots if the system was not shut down properly. This program finds and stores files not linked orderly to the file system.

**Logical Unit Number (LUN)**

A means of identifying or addressing a hard disk connected to the workstation/server. If only one hard disk is connected to the workstation/server, the Logical Unit Number is 0.

**Macro**

A string of keystrokes or commands invoked with a single keystroke or command. For example, a series of keystrokes can be mapped to one key so that the key invokes a desired process.

**mail/mailx**

System V communication programs used to send messages and files to users across the network.

**Main (UNIX) Processor/Dual Port Memory Board**

The board that houses the main processor (Intergraph's CLIPPER) and serves as the central processing unit for System V operations.

**makenode Utility**

A software delivery utility that loads products in deliverable format on a workstation/server specified as a delivery source. This delivery source can then deliver products across the network to other nodes.

**Manual Page**

A System V command description from the System V Online Documentation product (sysvdoc). Each man page discusses a separate command or file. To read an online man page, key in `man <command>`. For example, to read the documentation for the `ls` command, key in `man ls` (assuming that the `sysvdoc` product has been loaded on your system).

**Menubuilder**

The software package used for creating and editing screen menus used in both VAX-based and standalone applications. Symbols and icons created with Procasso can be accessed and used in menus created with Menubuilder.

**Menu Devices**

Display devices that allow you to make choices and manipulate menus using the mouse and keyboard input. The devices are found in pop-up menus and the Utility Pages.

**Menuprod**

A menu-driven interface for downloading workstation software products through the `newprod` utility.

**Message Strip**

The area at the top of the workstation screen. The message strip contains the workstation icon, primary screen icon, message area, and process ID field. The message strip is initialized when the Screen Manager software is loaded.

**Middle (Data) Button**

The middle button on the 12-button cursor used to select from any menu on the screen, select an icon, manipulate windows, place a 2D data point, identify graphical elements, select from a screen tutorial, activate windows (processes), or select from an introductory screen.

**Modifier Number**

The subpartition number. For example, the partition number for the `usr` partition is 7.3. The 7 is the hard disk division number for UNIX and the 3 is the modifier number for the `usr` partition.

**Modify Icon**

The window manipulation icon used to change the size and location of a window.

**mount Command**

The command that connects a partition to an existing file system. A partition must be mounted on a file system before you can access any data on the partition. The `/etc/mount` command is used to mount a partition.

**Multiuser Mode**

The operating system mode established when file systems are mounted with the screen manager and the VT220 software. This is the default mode when the system is booted.

**netaddr Command**

The command used to obtain the node address of an Intergraph VAX or workstation.

**Network Address**

The unique identifier hard-wired on each node. The network address is needed for each node to identify and communicate with another node on the network. This address is also referred to as the *Ethernet address*.

**Network Software**

The network communication software, or INC product, that allows a workstation/server to communicate with other workstations/servers and host computers across an IEEE 802.3 network.

**newprod Utility**

The utility for installing Intergraph software products on the hard disk of an Intergraph workstation/server.

**Nodename**

A name that can be assigned to a node address of a network device so that it is easier to remember. The nodename for Intergraph workstations/servers can have a maximum of 8 alphanumeric characters while all other devices on the network allow up to 14 alphanumeric characters.

**Nucleus Software**

Software that must reside on the hard disk before the System V operating system can run. This software is delivered free-of-charge with each Intergraph workstation/server.

**Numeric Keypad**

The part of the workstation keyboard used for entering numbers and special keyboard commands.

### **Operating System**

The set of programs that control the workstation/server's overall performance. Intergraph workstations/servers are based on the CLIX operating system, which is AT&T's UNIX System V version 5.3.1 operating system with additional Berkeley extensions. The System V operating system is based on three main components: the kernel, the shell, and the System V file system.

### **Operating System Parameters Page**

The Utility page used to establish default parameters that are effective when the operating system is loaded.

### **Ordinary File**

Any file that stores information such as programs, design files, raster graphics, and ASCII text. Commands and executables are also ordinary files.

### **Parity**

A method by which transmitted data can be verified. Even parity indicates that each transmitted word will contain an even number of bits in a logic 1 state. Odd parity indicates that each transmitted word will contain an odd number of bits in a logic 1 state. A special bit referred to as the parity bit is either set or cleared when it is transmitted so the parity requirement is met. If the parity does not check correctly, the data was most likely corrupted during transmission.

### **Partition**

A logical division of a hard disk. Internal hard disks are divided into boot, root, swap, and usr partitions. Each partition is used for a specific area of user and operating system functions. Intergraph delivers workstations and servers with predefined partitions. However, you may alter partition sizes to meet individual system requirements.

### **Partition Number**

A number identifying a partition. For example, the partition number for the usr partition is 7.3. The 7 is the hard disk division number for UNIX and the 3 is the modifier number for the usr partition.

### **Passwd File (/etc/passwd)**

The file that contains a record for each user account in the system. Each record in this file contains the following fields: username, password, user identification number, group identification number, miscellaneous information, home directory, and default login shell.

### **Pathname**

A name that identifies files in the System V file system. The pathname indicates a file or directory's location in the file system hierarchy and provides a way of accessing that file or directory. The two types of pathnames are absolute and relative.

**Peripheral Configuration Page**

The Utility page used to establish settings for the screen, keyboard, plotter port and RS-232 port.

**Permissions**

The set of protection codes associated with each file (whether ordinary or directory). These protection codes determine which system users can access the file. Read (r), write (w), and execute (x) permissions are specified for the file owner, the user group the file owner belongs to, and other users.

**Plotter Port**

The port used to connect an Intergraph V80 plotter or a color thermal printer. Some other devices supported on this port include QCR and PCR film recorders and a laser printer.

**PID**

See *Process Identification Number*.

**Pipe**

The software utility that uses the output of one command as the input to another command.

**Pop-To-Bottom Icon**

The window manipulation icon used to place a window beneath all other windows on the screen.

**Pop-To-Top Icon**

The window manipulation icon used to place a window on top of any other window on the screen.

**Pop-up Menu**

The screen menu displayed constantly until a screen button that cancels or exits the menu is selected.

**Power-Up**

To flip the power switch of the workstation/server to the 1 (on) position.

**Powerdown**

To shut the workstation/server down, wait for the system halted message, and flip the power switch of the workstation to the 0 (off) position.

**Primary Screen Icon**

An arrow icon next to the workstation icon on dual-monitor workstations. This icon is used to switch the primary screen from one screen to the other. All windows and processes appear on the primary screen by default.

**Process Identification Number**

The sequential number that System V assigns to a process when it is initiated.

**Process Name**

The name of the process running in a particular window.

**Prompt**

The character or set of characters that indicate that the process is waiting for input from the user. The standard System V ksh prompt is the \$ symbol. The standard System V superuser prompt is the # symbol.

**Pull-down Menus**

Menus activated from the workstation icon or some window control strip options. Pull-down menus can either be single- or multicolumn.

**Raster Operations Processor (ROP) Board**

The board that performs window protection and clipping, vector generation, polygon fill, raster text generation, block raster/clip operation, cursor control, and hardcopy processing.

**Reboot**

To reload workstation/server software. A warm boot (to boot the system when it is already running) just reloads the software while a cold boot (to turn the system on) performs diagnostic tests in addition to reloading the software.

**Rebuild Floppy Disks**

Floppy disks required to rebuild the hard disk. The Rebuild Floppy Disk set consists of four diskettes: Rebuild Boot and Root for 100/32C/200 workstations/servers and Rebuild Boot and Root for 300/400/3000/4000 workstations/servers. To rebuild one workstation/server, you need only the Rebuild Boot and Root diskettes that correspond to your workstation/server model number. The Rebuild Boot diskette is used to boot the workstation/server into single-user mode, while the Rebuild Root diskette is used to restore the file systems.

**Rebuild Process**

The procedure required to rebuild a corrupted hard disk, repartition, or convert to Fast File System. This procedure includes booting into single-user mode with the Rebuild Boot diskette, restoring file systems with the Rebuild Root diskette, loading software through the **newprod** utility, and rebooting back into System V multiuser mode from the hard disk.

**Redirection**

The process of specifying where to send input and output to. The > and < symbols are used to redirect I/O in a command line.

**regis Color Commands**

The command used to customize VT220 windows. This command allows you to change the background color, text color, and highlighted text color.

**Relative Pathname**

The pathname of a file or directory in relation to the current working directory. A pathname can be specified relative to the current working directory, thereby minimizing the length of the pathname that is entered.

**Repaint Icon**

The window manipulation icon used to refresh a window. If this icon does not appear in the window icon strip, the window is refreshed automatically.

**restore Command**

A Berkeley command ported to the Intergraph workstation. This command is used with the Berkeley backup command, where backup writes a tape and restore recovers the tape. backup and restore are used only for backing up and recovering entire file systems. These utilities have a helpful user interface and good error recovery mechanisms. In addition, backup and restore can write and read multiple-volume savesets.

**Restricted Shell**

A shell that provides a controlled environment for a user. For example, the sysadm and setup accounts are initiated under a restricted shell.

**Right (Reset) Button**

The right button of the 12-button cursor. The reset button is used to cancel a command action, reject an element selection, reset a command, place a 3D tentative data point (double tapping), or, in IBM PC emulation, perform the functions of the MicroSoft mouse's right button.

**Roll-Through Box**

The menu device used to cycle through available choices.

**Root Directory**

The base directory (/) of System V and PC-DOS.

**Root Partition**

The partition that contains a major portion of the System V operating system. It is used mainly for system administrative tasks. By default, this partition contains 25,000 blocks. If you receive messages saying that the root (7.0) partition is out of space, you must add space to this partition.

**RS-232 Port**

The port used to connect standard RS-232 devices such as printers, modems, and plotters. This port can also be used to connect the workstation/server to an Intergraph VAX.

**runcd Command**

A shell script used to mount a CDROM and initiate the CDROM Menu. You can invoke **newprod** or **makenode** from the CDROM menu. To execute this shell script, key in **runcd** at the **#** prompt.

**Run Level**

A number describing the state of the operating system. The run level is used with the **init** command. The following lists possible run levels:

- 0 shutdown
- 1 single-user
- 2 multiuser, NFS support
- 3 multiuser, RFS support
- 4 not used
- 5 not used
- 6 reboot

For example, the command **init 6** reboots the system.

**Saveset**

All files backed up at a certain point in time using a backup command or utility. For example, if you used the **cpio** command to back up the **/usr** file system, the **cpio saveset** will be the **/usr** file system.

**scpio Command (streaming cpio)**

An enhanced version of the **cpio** backup command for streaming tape drives. It provides two major additions to the features of the **cpio** command: asynchronous input/output and multi-buffering, which allows faster processing. Furthermore, **scpio** includes a verify mode that confirms the integrity of input and archived files.

**Screen Buttons**

Menu devices used to initiate a menu action.

**Screen Manager**

System V software that controls processes on Intergraph graphics workstations.

**Screen Menus**

The six types of menus used to activate processes and initiate commands when using System V: menubar, pull-down, hierarchical, window, pop-up, and function key.

**Scroll**

To move up or down multiple lines on the screen.

**SCSI Port**

An industry-standard interface that allows devices, such as a hard disk drive or cartridge tape drive, to communicate with an Intergraph workstation I/O processor board.

**Server**

The InterServe processors, which are plot, compute, and communication server nodes. These processors allow multiple workstation users on an IEEE 802.3 (Ethernet) network to share files, programs, and printing and plotting devices. Servers use an alphanumeric terminal for a console device. They do not have graphics capabilities.

**Shell**

The System V command language interpreter. The shell maintains the two-way communication flow between the user and the kernel, manages information, and serves as a programming language.

**Shell Script**

A program composed of shell commands.

**Shutdown**

The procedure that you should run before you turn off an Intergraph workstation or server. If you do not shut down properly, data may be corrupted.

**Single-User Mode**

The operating mode where only the root file system is mounted. The Screen Manager and VT220 software is not loaded. The prompt for single-user mode is #.

**Spawn**

To initiate a process.

**Special File**

A device file. Special files allow you to access a hard disk partition or storage device such as a tape drive. These files reside in the */dev* directory.

**Standard File System**

Standard (S51K) file systems contain 1K-byte units, whereas Fast File Systems contain 8K-byte units.

**Stash**

A partition or physical memory required on your plot server to plot on most plotting devices. Physical memory may be used only if sufficient physical memory is available on the server.

**Storage Device**

A peripheral device such as tape drive or floppy disk drive used to archive files.

**String**

A sequence of characters.

**Superuser Account**

A privileged account required for system administrative utilities such as **addusers** (for adding user accounts to the system) and **newprod** (for installing software). Superuser is the account of the root user.

**Superuser Mode**

The mode that allows a user to read or modify any file on the system and bypass the normal file security checks. This mode is distinguished with a # prompt.

**Supplemental Software**

Software products that you received free-of-charge with your Intergraph workstation/server; these products are not required to run the System V operating system.

**Swap Partition**

The partition used for swapping portions of memory to the hard disk. By default, this partition contains 27,360 blocks for 80-MB hard disks and 71,000 blocks for all other hard disk sizes. Space can be taken from the swap partition to add to another partition. However, some applications will not run without a minimum amount of swap space. This minimum amount varies among applications.

**sysadm Utility**

System V utility that provides a menu interface for performing system administrative functions such as backing up files, setting up a new system, adding user accounts, and removing user accounts.

**sys\_proc (/usr/ip32/smgr/sys\_proc)**

A file that defines system processes to be invoked automatically. All processes invoked by this file will run as the *root* user. By default, the console window is the only process invoked by this file.

**System V**

A standard delivery product on the workstation/server. This product includes the basic file system, which consists of two **cpio** **save** **set** **image**s. One image is for the *root* file system, the source of the file system structure, and the other is for the *usr* file system.

**Tapping a Button**

To press and immediately release a button.

**tar Command**

The tape archive command used to back up any files except special files.

**Terminal Emulation Keys**

Membrane keys in the upper left-hand corner of the workstation keyboard reserved for special terminal-related functions such as the editing commands found on a VT220 terminal.

**Terminal Window**

A window connected to a host computer on a network.

**to\_flop Command**

A filter command that allows a stream of data, greater than a floppy disk's capacity, to be written to sequential floppy disks.

**Toggle Switch**

The menu device used to switch between two alternatives.

**Transceiver**

A device that transmits signals between nodes. The transceiver has a tap for the coaxial cable and a connector for the transceiver cable(s).

**Transceiver cable**

The cable that runs from the interface on the transceiver to the interface on the Intergraph workstation/server (Ethernet port). The transceiver cable is a shielded cable that has 15-pin D-connectors on each end; each connector has a slide lock.

**tty**

A psuedo-terminal. (tty is an abbreviation for *teletypewriter*.)

**Typewriter Keys**

Keys that provide all the upper- and lower-case letters, numerals, and symbols found in a standard ANSI typewriter layout.

**UID**

See *User Identification Number*.

**uname -n Command**

The command used to obtain the nodename of an Intergraph VAX, workstation, or server.

**UNIX Operating System**

An operating system developed by AT&T Technologies.

**umount Command**

The command that unmounts a previously mounted file system contained on the block special device or directory.

**User Account**

A login account. This is the account the user is placed in when he/she logs in to the system. Each line or record in the */etc/passwd* file represents a different login account.

**User Identification Number**

A number that the */etc/passwd* file associates with a user account. This number is located in the third field of the */etc/passwd* file.

**usr Partitions**

The partitions (such as *usr*, *usr2*, and *usr3*) that contain all user directories, most products, and any other directories and files that users access.

**Utility Pages**

The menus used to configure the workstation or server operating systems, hard disk(s), and peripheral ports.

**uucp Command**

The command used to exchange files between workstations/servers. (*uucp* stands for UNIX system-to-UNIX system copy.) With this command, you can place files directly in another user's directory.

**Verify Program**

The procedure that reads the hard disk and records all flaws to ensure that data is not written over flawed areas. It does not erase or change the data on the disk.

**vi Editor**

A UNIX screen-oriented display editor based on the UNIX *ed* line editor.

**Virtual Screen**

A screen that acts as its own workstation. InterPro (single-screen) workstations can be configured for one or two virtual screens. Configuring an InterPro workstation for two virtual screens enables you to run multiple processes on each virtual screen. Thus, even though the InterPro workstation has one physical screen, you can use two virtual screens.

**visit**

The program that provides the ability to log in to another network node. This program can be used only for purposes other than running graphics applications.

**vmsbackup Command**

The utility that reads VMS-generated backup format tapes. Vmsbackup cannot write VMS-generated savesets. This utility is useful for transferring design files across the VMS and CLIX operating systems.

**VT220 Emulation Software**

Software that allows the workstation to act as a DEC VT220 alphanumeric terminal. This software allows application programs to transfer standard ANSI escape sequences to and from the workstation.

**vterm Command**

The command that creates local and terminal VT220 windows. This command has many options that can be used to determine window characteristics.

**Window Control Strip**

The list of choices displayed beneath the message strip at the top of the screen. This strip displays only if the active process calls for it. The menu's display depends on the active process. A selection from this menu issues a command or displays another type of menu.

**Window Icon Strip**

The bar across the top of each window. This bar contains window manipulation icons and the window name.

**Window Manipulation Icons**

Icons located in the Window Icon Strip. These icons allow you to move a window, resize a window, pop a window to the top of a stack, pop a window to the bottom of a stack, collapse a window, move a collapsed window, restore a collapsed window, delete a window, repaint a window, clear the window screen, copy and paste text, or reset the window terminal.

**Window Menus**

Menus inside movable windows. Their appearance depends on the program that is running. Window menus can be moved, collapsed, and deleted, but their size and proportions cannot be changed.

**Windows**

Rectangular areas on virtual screens for graphics output and locator input. Each window is owned by a process currently executing under the System V operating system.

**Working Directory**

The directory in which a user is currently working.

**Workstation**

An Interpro, InterAct, or InterView graphics computer.

**Workstation Icon**

The icon used to activate the Workstation menu.

**Workstation Menu**

The menu initiated with the Workstation icon. This menu allows you to create a terminal window and connect to a host, create a local window to access the System V operating system, initiate an Intergraph-developed process, and edit the Workstation menu.







## Index

---



---

- .def\_\_proc, 5-3, 5-35
- .env file, 4-1, 4-9
- .exrc file, 4-1, 4-13
- .profile file, 4-1, 4-6, 4-8
- .smgr\_\_pd, 5-26
- /
- /dev/dsk
  - creating device file for, 7-3
- /dev/rmt, 9-5, 9-10, 9-11, 9-13 to 9-18, 9-20, 9-21, 9-25
- /etc/format, 9-27
- /etc/gettydefs, 2-5, 2-6, 2-8
- /etc/init, 2-3
- /etc/init.d, 2-9
- /etc/inittab, 2-3, 2-4, 2-6
- /etc/labelit, 7-10
- /etc/mkfs, 7-10
- /etc/mknod, 7-8
- /etc/motd, 4-6
- /etc/mount, 7-12
- /etc/newfs, 7-11
- /etc/passwd, 3-5, 3-7, 3-13, 3-14, 3-16, 4-4
  - fields in, 4-4
- /etc/profile, 4-1, 4-6
- /etc/rc2, 2-5
- /usr/tmp, 7-14

**3**  
32-channel RS-232 interface assembly

**A**  
accessing

- directory files, 3-20
- ordinary files, 3-21

accounting

- initializing at bootup, 2-9

accounts

- adding passwords, 3-13
- adding user accounts, 3-5
- customizing variables in user accounts, 4-8
- removing passwords, 3-16
- removing user accounts, 3-10
- /etc/passwd file, 4-4

- active group
  - changing, 3-24
- adding cron command file, 4-15
- adding partition, 11-28
  - on server, 7-28
  - on workstation, 7-27
- adding user accounts, 3-5
  - comparison of addusers and sysadm, 3-5
  - manually, 3-5
  - using addusers, 3-6
  - using sysadm, 3-11
- adding processes to workstation menu, 5-26
- address
  - node or network, 1-7, 10-12, 10-42
- addusers
  - advantages, 3-5
  - using to add user accounts, 3-6
- administering
  - adding user accounts, 3-5
  - maintaining file systems, 7-1
  - removing user accounts, 3-10
  - setting up initial system, 1-1
- administrator
  - system administrator account, 3-3
- aliases
  - defining, 4-9
- ANSI color escape sequences, 5-22
- ansitape, 9-20
- assigning nodename, 1-6, 10-12, 10-42
- assigning passwords, 3-13
- asynchronous input/output
  - supported in scp for backups, 9-13
- at, 4-17
- attaching
  - external devices, 2-6
  - window to printer, 5-15
- auto xon/xoff
  - setting on workstation, 10-10
- automatic dump, 10-12
- automatic reboot, 10-12
- auxiliary ports
  - setting RS-232 parameters
    - server, 10-39
    - workstation, 10-8
- available disk space, 7-14

**B****background**

changing screen color, 5-33

changing pattern, 5-32

backing up entire file system, 9-8, 9-11, 9-13

backing up files, 9-6

**backups**

ansitape example, 9-20

commands and utilities, 9-6

compatibility with VAX/VMS, 9-21

defining schedule, 9-3

examples using dd, 9-10

examples using tar, 9-11, 9-12

multiple volumes using scp/cpio, 9-13

raw binary data, 9-10

support of asynchronous input/output in, 9-13

dd command, 9-10

tar command, 9-11, 9-12

to local tape drive using cpio, 9-5

to remote tape drive using rtc, 9-4

using ansitape, 9-20

using backup and restore, 9-18, 9-19

using scp/cpio, 9-13, 9-14

using vmsbackup, 9-21

backup command, 9-18

**backup plan**

full network configuration, 9-4

standalone configuration, 9-5

**bad blocks**

explanation, 11-10

messages, 11-10

**batch**

run command at specific time, 4-17

run commands at regular intervals, 4-15

**baud rate**

setting on server, 10-40

setting on workstation, 10-9

**bell**

setting tone

workstation, 10-8

bgpattern, 5-32

**binary data**

moving using dd command, 9-10

- blocking factors with dd, 9-10
- boot
  - booting with rebuild disks, 11-7
- boot partition, 11-20
- booting single user
  - server, 11-41
  - workstation, 11-11
- bootup
  - files executed at, 2-1
  - starting accounting and lp, 2-9
- bourne shell, 3-7, 4-5
- buffered screens
  - setting up, 5-11

## C

- calendar, 4-14
- canceling a print request, 6-5
- cartridge tapes
  - with dd command, 9-11
- catalog file
  - creating window using, 5-24
- catalog menu, 5-24
- changing
  - file ownership, 3-23
  - file protection codes, 3-20
  - group access to file, 3-24
  - background pattern, 5-32
  - number of virtual screens, 5-30
  - screen color, 5-33
  - your active group, 3-24
  - partition sizes, 11-28, 11-20
- character codes
  - keyboard, A-1
    - decimal and hexadecimal equivalents, A-3
    - received, A-6
    - transmitted, A-3
- checklist for initial setup tasks, 1-1
- chgrp, 3-25
- chmod, 3-22
- chown, 3-23
- clearinghouse database, 1-6

- 
- CLIX Resource Monitor (CRM), 8-1
    - adding to def\_proc, 8-3, 5-34
    - adding to pull-down menu, 8-3
    - changing defaults
      - system monitors, 8-9
    - forms, 8-6
    - installing, 8-3
    - invoking, 8-4
    - linking to /usr/bin, 8-3
    - menus, 8-10, 8-24
    - monitoring
      - all aspects of specific process, 8-29
      - all system processes, 8-16
      - CPU use, 8-15
      - input/output, 8-15
      - memory use, 8-14
      - memory use by process, 8-28
      - page faults, 8-26
      - system calls and faults, 8-26
  - Process Monitors, 8-19
    - command line, 8-19
    - exiting, 8-19
    - invoking, 8-19
    - memory monitor, 8-28
    - process monitor, 8-29
    - profiler, 8-26
    - selecting process to monitor, 8-24
    - selecting profiler options, 8-25
  - running curses-based
    - process monitors, 8-10
    - system monitors, 8-7
  - running graphics-based
    - system monitors, 3-9
  - System Monitors, 8-7
    - command-line options, 8-8
    - command-line syntax, 8-7
    - curses-based, 8-10
    - exiting, 8-7
    - graphics-based, 8-9
    - invoking from command line, 8-7
    - Monitor Parameters, 8-11
    - Show Memory Usage, 8-17
    - Show Open Files, 8-17

- Top CPU, 8-15
- Top Fault, 8-12
- Top I/O, 8-15
- Top Memory, 8-14
- Top System, 8-16
- topcpu, 8-15
- topfault, 8-12
- topio, 8-15
- topmem, 8-14
- topsys, 8-16
  - using FMLI interface, 8-7
- collapsing window to symbol, 5-7
- color
  - changing screen color, 5-33
- color window
  - creating with ANSI color escape sequence, 5-22
  - creating with regis, 5-18, 5-19
  - creating with vterm, 5-6, 5-18
- command recall, 4-10 to 4-12
  - defining, 4-11, 4-12
  - setting temporary editor, 4-12
- commands
  - addusers, 3-9
  - ansitape, 9-20
  - at, 4-17
  - backup, 9-18
  - calendar, 4-13
  - chgrp, 3-25
  - chmod, 3-23
  - chown, 3-23
  - conv=sync for dd, 9-11
  - cpflop, 9-31
  - cpio, 9-13
  - cron, 4-15
  - crontab, 4-16
  - dd, 9-10
  - df, 7-13
  - dls, C-6
  - dtu, C-3
  - du, 7-14
  - fr\_flop, 9-31
  - format, 9-27
  - init, 2-3
  - labelit, 7-11
  - mkfs, 7-10, 9-29
  - mknod, 7-3 to 7-8

---

- mount, 7-12
- newfs, 7-11
- newgrp, 3-24
- passwd, 3-13
- recalling from command line, 4-10
- regis, 5-18
- restore, 9-19
- rmusers, 3-10
- rtc, 9-22
- showconfig, 2-6
- scpio, 9-13
- su, 3-3
- sysadm, 3-3, 3-8
- tar, 9-11
- to\_flop, 9-31
- topcpu (CRM), 8-15
- topfault (CRM), 8-12
- topio (CRM), 8-16
- topmem (CRM), 8-14
- topsys (CRM), 8-16
- utd, C-4
- vmsbackup, 9-21
- vterm, 5-4
- compatibility
  - copying files from DOS to UNIX, C-3
  - copying files from UNIX to DOS, C-4
  - with IBM PC, B-1
- configure
  - background pattern, 5-32
  - number and location of message strips, 5-31
  - number of virtual screens, 5-30, 5-31
- configure option on workstation menu, 5-26
  - add, 5-26
  - delete, 5-28
  - modify, 5-28
  - screens, 5-30
- configure a line printer, 6-1, 6-3
- console login, 3-3
- console window
  - creating, 5-36
  - customizing, 5-36
- copy file archive in and out (cpio), 9-13
- copying files from DOS to UNIX, C-3
- copying files from UNIX to DOS, C-4

- core files
  - removing, 7-15
- cpio, 9-13
- creating color window
  - using ANSI, 5-22
  - using regis commands, 5-18, 5-19
- creating device file, 7-3 to 7-8, 9-31
- creating file system on new partition, 7-7
- creating rebuild floppy disks, 11-3
- CRM (see CLIX Resource Monitor), 8-1
  - adding to def\_\_proc file, 8-3
  - adding to workstation menu, 8-3
- cron, 4-15
- crontab, 4-16
- csh, 3-7
- Curses
  - CRM interface, 8-10
- customizing
  - adding default processes to .def\_\_proc, 5-34
  - adding default processes to def\_\_proc, 5-35
  - adding default processes to sys\_\_proc, 5-36
  - console window
    - through sys\_\_proc, 5-36
  - creating window that logs you in, 5-7
  - creating windows, 5-4
  - display characteristics, 5-12
  - function key menu, 5-16
  - keyboard characteristics, 5-13
  - login prompt in gettydefs, 2-5
  - number and location of message strips, 5-30, 5-31
  - number of virtual screens, 5-30
  - saving customized window, 5-24
  - setting up reminder service, 4-14
  - terminal characteristics, 5-10
  - /etc/passwd file, 4-4, 3-16
  - /etc/profile file, 4-6
  - background pattern, 5-33
  - login environment, 4-1
  - screen color, 5-34
  - workstation menu
    - adding cataloged window, 5-27
    - adding processes to, 5-26
    - connecting to remote host, 5-27
    - deleting processes on, 5-28
    - modifying processes on, 5-28

**D**

- data capture
  - creating log file, 5-7
- dd, 9-10
- dedicated printer, 6-1
- default file permission modes, 3-20
  - setting with umask, 3-21
- default processes
  - setting with `.def__proc`, 5-35
  - setting with `def__proc`, 5-34
  - setting with `sys__proc`, 5-36
- defining aliases, 4-9
- defining macros, 5-16
- `def__proc`, 5-34
- deleting
  - suggestions for freeing disk space, 7-14
- delimiters
  - off, 2-6
  - respawn, 2-6
- design files
  - backing up using ansitape, 9-20
  - backing up using vmsbackup, 9-21
- device files
  - creating, 7-3
  - creating for scsi 0, lun 0, 7-4
  - creating for scsi 1, lun 0, 7-5
  - creating for scsi 2, lun 0, 7-6
  - creating for scsi 3, lun 0, 7-7
- Device Image Copy Utility Page
  - server, 10-48
  - workstation, 10-26
- device parameters, 2-8
- df, 7-13
- directory files, 3-22
  - accessing based on file permission modes, 3-22
- disk
  - creating device file, 7-3 to 7-8
  - floppy
    - file system, 9-29
    - formatting, 9-31
  - Format Utility Page
    - server, 10-43
    - workstation, 10-21
  - formatting, 11-13
  - formatting floppy disks for PC DOS, 9-27
  - freeing space, 7-14
  - monitoring space, 7-14

- partitioning, 11-17
  - Utility Page options on server, 10-46
  - Utility Page options on workstation, 10-26
- reading hard disk for errors, 11-10
- rebuilding using nonstandard partitions, 11-17
- SCSI ID on Utility Pages
  - server, 10-43
  - workstation, 10-14
- understanding partition names, 11-25
- verify flaws, 11-10
- verifying and adding flaw data
  - server, 10-11
  - workstation, 10-12
- Disk Maintenance Utility Page
  - server, 10-43
  - workstation, 10-13
- Disk Partitioning Utility Page
  - server, 10-46
  - workstation, 10-23
- disk space
  - amount free (df), 7-13
  - amount used (du), 7-14
  - freeing, 7-14
  - monitoring, 7-14, 7-17
- disk space used, 7-14
- display
  - amount of disk space used, 7-14
  - available disk space, 7-13
- display characteristics, 5-12
- display setup menu, 5-12
- dls, C-6
- DOS
  - copying files from DOS to UNIX, C-3
  - copying files from UNIX to DOS, C-4
  - listing contents of DOS directory, C-6
- dtu, C-3
- du, 7-14
  
- E
- editor
  - setting default editor, 4-10
- emulation
  - vt100, vt220, 5-4
- erasing
  - data on floppy disk, 9-27
- escape sequences
  - keyboard control characters, A-7

execute access, 3-21  
external devices  
  attaching, 2-6

## F

fast backups  
  dd, 9-10

### Fast File Systems

  /etc/newfs, 7-11  
  changing from standard to fast file, 7-11  
  creating, 7-11  
  creating from rebuild menu, 11-33  
  mounting, 7-12

### fields

  in file protection code (rwx), 3-20  
  in /etc/passwd file, 4-5

### file access, 3-20

  categories of users, 3-19  
  directory files, 3-20  
  modifying, 3-21  
  ordinary files, 3-21

### file owner, 3-22

  changing, 3-22

### file protection codes (rwx)

  changing, 3-21  
  fields, 3-21, 3-22  
  meaning for directory files, 3-36  
  meaning for ordinary files, 3-37

### file size

  displaying with du, 7-14

### file system

  backing up, 9-10, 9-11, 9-13  
  changing from standard to Fast File, 7-11  
    creating fast file system, 7-11  
  commands and utilities for backing up, 9-10, 9-11, 9-13  
  creating on new partition, 7-9  
    Fast File System, 7-11  
    standard file system, 7-10  
  defining plan for backing up, 9-4  
  difference between fast and standard, 7-9  
  Fast File System  
    changing to, 7-11  
    creating Fast File System, 7-11

files

- backing up and restoring, 9-18
  - to local tape drive, 9-5
  - to network tape drive, 9-4
  - using backup and restore, 9-18
  - using dd, 9-10
  - using scp/cpio, 9-13
  - using tar, 9-11
- labeling, 7-11
- mounting, 7-12
- restoring, 11-33
- using ansitape to back up, 9-20
- using vmsbackup to back up, 9-21

files

- .env, 4-9
- .exrc, 4-13
- .profile, 4-8
- /etc/gettydefs, 2-6, 2-7, 2-8
- /etc/inittab, 2-6, 2-7
- /etc/profile, 4-6
- access, 3-20
- changing ownership, 3-23
- changing group, 3-25
- copying from DOS to UNIX, C-3
- copying from UNIX to DOS, C-4
- core
  - removing, 7-15
- def\_proc and CRM, 8-3
- displaying size, 7-14
- executed at login, 4-1
- executed at system startup, 2-1
- executed in multiuser mode, 2-9
- getty, 2-5, 2-7, 2-8
- gettydefs, 2-6, 2-7, 2-8
- init.d, 2-9
- inittab, 2-4, 2-6, 2-7
- Korn shell environment file, 3-7
- permission modes, 3-20
  - changing, 3-22
- profile file, 4-6
- protection codes, 3-20
  - changing, 3-22
- screen manager default processes files, 5-34 to 5-36
- startup processes, 2-1
- terminal definition (gettydefs), 2-5
- user's login environment file (.profile), 4-8

find

- freeing disk space, 7-15
- using with cpio/scpio, 9-13

- first
  - tasks for setting up system, 1-1
- flaws
  - adding on Utility Pages
    - server, 10-43
    - workstation, 10-16
  - adding to flaw list using verify
    - server, 10-43
    - workstation, 10-19
  - format with flaw data
    - server, 10-43
    - workstation, 10-18
  - verify disk flaws, 11-10
- floppy disks
  - backups with tar, 9-11
  - copy one floppy to another, 9-31
  - creating rebuild disks, 11-3
  - density, 9-27
  - file system, 9-29
  - formatting, 9-27
  - mkfs command, 9-29
  - mount file system, 9-29
  - preparing for use, 9-27
  - read/write compatibility with IBM PC, B-1
  - restore from multiple, 9-31
  - unmount file system, 9-30
  - write to multiple, 9-31
- FMLI
  - CRM interface, 8-5
- forcing password change, 3-14
- Format Utility Page
  - workstation, 10-21
  - server, 10-43
- formatting, 11-13
  - explanation, 11-13
  - floppy disks, 9-27
  - server, 11-14
  - Utility Page
    - server, 10-43
    - workstation, 10-21
  - workstation, 11-13
- free disk space
  - displaying, 7-13
- freeing, 7-14
- function keys
  - assigning macros, 5-16

## **G**

- general setup menu, 5-10
  - creating log file, 5-7
- getting started, 1-1
- getty, 2-5, 2-6, 2-7, 2-8
- gettydefs, 2-6, 2-7, 2-8
- GID, 3-7, 3-9, 4-5
- global variables
  - setting up in /etc/profile, 4-6
- group, 3-7, 3-9, 4-5, 3-24
  - changing group access, 3-24
  - changing your active group, 3-24

## **H**

- hard disk
  - adding flaws on Utility Page
    - server, 10-43
    - workstation, 10-16
  - adding flaws to flaw list
    - server, 10-43
    - workstation, 10-19
  - creating device file, 7-3 to 7-8
    - external
      - creating device file, 7-3 to 7-8
  - Format Utility Page
    - server, 10-44
    - workstation, 10-21
  - formatting, 11-13
  - internal
    - creating device file, 7-3 to 7-8
  - partitioning
    - Utility Page options on server, 10-44
    - Utility Page options on workstation, 10-23
  - reading for errors, 11-10
  - rebuilding using nonstandard partitions, 11-28
  - understanding partition names, 11-25
  - verify, 11-10
  - verifying
    - server, 10-43
    - workstation, 10-19
- Hard Disk Flaw Data Utility Page
  - workstation, 10-16
- history
  - setting up in Korn shell, 4-10
- home directory, 3-7, 3-9
- hue, saturation, and lightness chart, 5-21

## I

- icon
  - creating symbol for collapsed window, 5-7
- imaging
  - using dd command to back up, 9-10
- inactive window, 5-6
  - creating, 5-6
- init, 2-3, 2-10
- init.d, 2-9
- initial tasks, 1-1
- initialization files, 4-1
- initializing
  - from multiuser mode, 2-9
- inittab, 2-6, 2-7
- interface
  - CRM, 8-5
- internal hard disk
  - creating device file for, 7-3 to 7-8
- invoking
  - CRM, 8-4
- iop
  - setting on server, 10-37

## K

- key clicks
  - setting, 10-7
- keyboard
  - assigning macros, 5-16
  - character codes, A-1
    - decimal and hexadecimal equivalents, A-3
    - received, A-6
  - control characters
    - escape sequences, A-7
  - customizing, 5-13
- keyboard
  - setting key clicks, 10-7
- keyboard characteristics setup menu, 5-13
- Korn shell
  - environment file, 3-7, 4-5
- ksh, 3-7, 4-5

## L

- labeling file system, 7-11
- labelit, 7-11
- lightness, saturation, and hue chart, 5-21
- line printer, 6-1
- linking files to rc2.d, 2-5, 2-9
- listing contents of DOS directory, C-6

- loading software
  - after rebuild, 11-41
- local backup, 9-5
- location of message strips
  - changing, 5-31
- log file
  - creating, 5-7
  - creating from setup menu, 5-11
- logical unit number, 10-14
- login
  - adding login accounts, 3-5
  - assigning passwords, 3-13
  - creating window to log you in, 5-7, 5-11
  - customizing global variables, 4-6
  - customizing .env file, 4-9
  - customizing users login environment, 4-1
    - environment, 4-1
    - files executed in, 4-9
  - file
    - /etc/passwd, 4-4
  - process, 4-3
  - removing login accounts, 3-10
  - removing passwords, 3-16
    - from root, 3-19
- login directory, 3-7, 3-9, 4-5
- logging in as root, 3-3
- lost+found directory
  - removing files, 7-16
- lp
  - initializing at boot-up, 2-9
- LUN, 10-14
  - setting on Utility Pages
    - server, 10-42
    - workstation, 10-14

## M

- macro setup menu, 5-16
- macros
  - defining, 5-16
- Main Utility Menu
  - workstation, 10-3
  - server, 10-33
- major and minor numbers
  - for SCSI 0, LUN 0, 7-4
  - for SCSI 1, LUN 0, 7-5
  - for SCSI 2, LUN 0, 7-6
  - for SCSI 3, LUN 0, 7-7
- manager
  - system manager account, 3-3

---

**managing**  
  adding passwords, 3-13  
  forcing password change, 3-14  
  monitoring system with CRM, 8-1  
  removing passwords, 3-16  
    from root, 3-17  
    from Utility Pages, 3-19  
  setting up initial system, 1-1

**manipulating directories**  
  based on file permission modes, 3-21

**manipulating ordinary files**  
  based on file permission modes, 3-22

**math coprocessor**  
  enabling, 10-11

**menu**  
  deleting options, 5-28  
  modifying options, 5-28  
  adding options, 5-26

**menus**  
  catalog, 5-24  
  display setup, 5-12  
  general setup, 5-10  
  keyboard characteristics, 5-13  
  macro menu, 5-16  
  printer setup, 5-15  
  tab setup menu, 5-14

**message strips**  
  configuring number and location of, 5-31

**mkfs, 7-9, 9-29**

**mknod, 7-3 to 7-8**

**modify option**  
  workstation menu, 5-26

**modifying**  
  inittab file, 2-6, 2-7  
  gettydefs file, 2-6, 2-8

**monitoring processes, 8-1**

**monitoring a print request, 6-4**

**monitoring disk space, 7-13**

**mounting, 7-12**  
  a fast file system, 7-12  
  a standard file system, 7-12

**moving message strips, 5-31**

**multiuser initialization, 2-9**

**multiuser mode**  
  files executed in, 2-9

**multivolume backup**  
  using backup and restore, 9-18  
  using scp/cpio, 9-13

## N

- naming file system, 7-10, 7-11
- network backup, 9-4
- newfs, 7-11
- newgrp, 3-24
- newprod
  - after rebuild, 11-41
- node address, 10-12
  - setting nodename from Utility Pages
    - server, 10-40
    - workstation, 10-12
- number of message strips
  - changing, 5-30

## O

- off delimiter, 2-6
- operating system
  - files executed at startup, 2-1
  - monitoring, 8-1
  - restoring/rebuilding, 11-1
  - setting default from Utility Page, 10-6
  - setting parameters on Utility Page
    - server, 10-40
    - workstation, 10-11
- Operating System Parameters Utility Page
  - server, 10-40
  - workstation, 10-11
- ordinary files
  - accessing based on file permission modes, 3-22
- other user, 3-21
- owner, 3-21
  - changing, 3-23

## P

- parameters
  - device, 2-8
- parity
  - setting on server, 10-38
  - setting on workstation, 10-9
- partition, 11-20
  - adding
    - on server, 11-29
    - on workstation, 11-28
  - changing default sizes, 11-28
  - connecting to existing file system, 7-12
  - copying
    - server, 10-46
    - workstation, 10-26

- creating device file, 7-3 to 7-8
  - for SCSI 0, LUN 0, 7-4
  - for SCSI 1, LUN 0, 7-5
  - for SCSI 2, LUN 0, 7-6
  - for SCSI 3, LUN 0, 7-7
- creating file system on, 7-10
  - Fast File System, 7-11
  - standard file system, 7-10
- creating nonstandard sizes
  - on server, 11-29
  - on workstation, 11-28
- default sizes, 11-19
- mounting, 7-12
- overview, 7-20
- raw
  - backing up, 9-10
- reallocating space between partitions, 11-23
- removing partition
  - on server, 11-29
  - on workstation, 11-28
- root, 11-21
- boot, 11-20
- standard naming conventions, 11-25
- stash
  - determining size of, 11-21
- swap, 11-21
- dos, 11-21
- usr, 11-21
- stash, 11-21
- types of, 11-20
- understanding names of, 11-25
- using nonstandard partition sizes, 11-28

partition table, 10-23

partitioning

- specifying from Utility Pages
  - server, 10-44
  - workstation, 10-23

partitioning hard disk, 11-17

- creating default, 11-19
- explanation, 11-20
- restoring file systems, 11-33

passwd, 3-13

password

- Utility Page
  - setting on server, 10-33
  - setting on workstation, 10-4

- passwords, 3-13
  - changing, 3-13
  - creating, 3-13
  - forcing, 3-14
  - removing, 3-15
    - from root, 3-15
    - from Utility Pages, 3-18
  - restrictions, 3-13
- PATH
  - setting search path, 4-8
- pattern
  - changing background pattern, 5-32
- PC
  - read/write compatibility of floppy disks, B-1
- PC-DOS
  - copying files from DOS to UNIX, C-3
  - copying files from UNIX to DOS, C-4
  - exchange between System V, C-1
  - formatting floppy disks, 9-27
  - listing contents of DOS directory, C-6
  - setting as default operating system, 10-6
- Peripheral Configuration Utility Page
  - server, 10-37
  - workstation, 10-7
- permission modes
  - changing, 3-22
  - description, 3-20
- pipe
  - into cpio/scpio, 9-13
- plotting
  - determining size of stash partition, 11-21
- preparing floppies to back up data, 9-27
- printer
  - attaching to window, 5-15
  - setting up a dedicated, 6-1
- printer setup menu, 5-15
- printing from a dedicated printer, 6-4
- privileges
  - system management, 3-3
- problem
  - bad block messages, 11-2
  - can't boot workstation, 11-2
  - which option to choose from Rebuild menu, 11-33 to 11-50
- procedures
  - adding password, 3-13
  - adding user accounts, 3-5
    - addusers, 3-6
    - sysadm, 3-8

---

- backup and restore commands, 9-6 to 9-9
- command recall from command line, 4-10
- creating device file, 7-3 to 7-8
- creating file system, 7-9
  - Fast File System, 7-10
  - standard file system, 7-9
- creating rebuild floppy disk set, 11-3
- creating windows, 5-4
- customizing login environment, 4-1
- customizing login prompt, 2-5
- displaying disk space used, 7-14
- displaying free disk space, 7-13
- forcing password change, 3-14
- formatting, 11-13
- formatting floppy disk, 9-27
- maintaining system, 7-13, 7-17
- rebuilding from rebuild floppy disks, 11-1
- rebuilding using nonstandard partitions, 11-20
- removing password, 3-16
  - from root, 3-17
  - from Utility Pages, 3-19
- removing user accounts, 3-10
- repartitioning after format, 11-20
- setting global variables, 4-6
- setting up initial system, 1-1
- steps for creating nonstandard partition
  - on server, 11-29
  - on workstation, 11-28
- verify hard disk, 11-10

process ID

- moving message strip locations, 5-31

Process Monitors (CRM), 8-19

processes

- adding to `.def__proc` file, 5-35
- adding to `def__proc` file, 5-34
- adding to `sys__proc` file, 5-36
- adding to workstation menu, 5-26
- getty, 2-5
- init, 2-3
- login process, 4-3
- login, 4-3
- rc2, 2-9

products

- removing to allocate disk space, 7-16

profile, 4-6

protection codes (rwx)

- changing, 3-22
- description, 3-20

## R

- raw data
  - backups, 9-10
- read access, 3-20
- read/write compatibility of floppy disks, B-1
- read, write, and execute access, 3-20
  - modifying, 3-22
- reading VMS backup tapes using vmsbackup, 9-21
- reallocating space between partitions, 11-23
- reboot
  - automatic after system crash, 10-12
- rebuild
  - floppy disks
    - booting from, 11-7
    - menu options, 11-34 to 11-50
    - restoring files, 11-42
- rebuild floppy disks, 11-3
  - options, 11-4, 11-5
- rebuilding, 11-1
  - conditions requiring, 11-2
  - creating rebuild floppy disks, 11-3
  - loading software using newprod, 11-41
  - overview of steps, 11-2
  - entire disk, 11-34
  - root file system, 11-35
  - usr file system, 11-36
  - to change to default partition sizes, 11-20
- recall
  - buffered screens, 5-11
  - command recall, 4-10
- recalling commands, 4-10
- reconfigure
  - number of virtual screens, 5-30
- redirection
  - to scp/cpio command for backups, 9-13
- regis, 5-15
- reminder service, 4-14
- remote
  - backups to remote tape drive, 9-4
- remote tape controller command, 9-4, 9-5, 9-22
- removing partition
  - on server, 11-29
  - on workstation, 11-28
- removing files to allocate disk space, 7-14 to 7-16
- removing passwords, 3-16
  - from root, 3-17
  - from Utility Pages, 3-19

---

- removing products, 7-16
- removing user accounts, 3-10
  - using rmusers, 3-10
  - using sysadm, 3-11
  - comparison of rmusers and sysadm, 3-10
  - manually, 3-10
- repartitioning
  - see partitioning, 11-20
- repartitioning for nonstandard partitions, 11-20
- respawn delimiter, 2-6
- restore command, 9-19
- restoring files, 9-1
- rmusers, 3-10
- root account, 3-3
  - removing password from, 3-17
- root partition, 11-21
- RS-232 interface assembly, 2-6
- RS-232 parameters
  - auxiliary port settings
    - server, 10-37
    - workstation, 10-8
- rtc, 9-4, 9-5, 9-22
- run levels, 2-3
  - processes invoked in run level 2, 2-3
- running command at specific time, 4-14
- running commands at regular intervals, 4-15
- rwX permissions modes, 3-20
  - changing, 3-22

## S

- saturation, hue, and lightness chart, 5-21
- saving window configuration, 5-24
- scpio, 9-13
- screen
  - changing color, 5-33
  - configuring number of virtual screens, 5-30
- screen manager, 5-2
  - invoking default processes from def\_\_proc, 5-34
  - invoking default processes from .def\_\_proc, 5-35
  - invoking default processes from sys\_\_proc, 5-36
- screen saver
  - setting, 10-7
- SCSI 0, LUN 0
  - creating device file for, 7-4
- SCSI 1, LUN 0
  - creating device file for, 7-5
- SCSI 2, LUN 0
  - creating device file for, 7-6

- SCSI 3, LUN 0
  - creating device file for, 7-7
- SCSI ID
  - setting on Utility Pages
    - server, 10-41
    - workstation, 10-14
- security
  - root account, 3-3
- server
  - backing up to remote tape drive using, 9-4
  - Utility Pages for, 10-30
- setting
  - default processes, 5-34 to 5-36
  - display characteristics, 5-12
  - tab stops, 5-14
  - search path, 4-7, 4-8
  - up reminder service, 4-14
  - up terminal characteristics, 5-10
  - up initial system, 1-1
- setting nodename on Utility Pages
  - server, 10-40
  - workstation, 10-12
- setting bell tone, 10-8
- setting up a line printer, 6-1, 6-3
- setup menus
  - catalog, 5-24
  - display menu, 5-12
  - general, 5-10
  - keyboard characteristics, 5-13
  - macro, 5-16
  - printer, 5-15
  - tabs, 5-14
- shells, 3-7, 3-9, 4-5
- showconfig command, 2-6
- single user
  - booting from Utility Page
    - server, 10-40
    - workstation, 10-11
- size
  - displaying file size, 7-14
- smgr
  - customizing, 5-2
- smgr\_vlt, 5-33
- software
  - loading after rebuild, 11-41
  - removing software products, 7-16

---

space  
  allocating to each partition, 11-20  
  monitoring disk space used and free, 7-13, 7-14  
  reallocating to another partition, 11-23

special files, 7-3 to 7-8

starting accounting and lp, 2-10

startup  
  files executed, 4-1  
  files executed at system startup, 2-1

stash partition, 11-21  
  determining size of, 11-21

stop bits  
  setting on server, 10-38  
  setting on workstation, 10-10

streaming cpio, 9-13

su, 3-3, 2-8

submitting a print request, 6-4

superuser, 3-3  
  accessing, 3-3

swap partition, 11-21

symbols  
  creating symbol for collapsed window, 5-7

sysadm  
  adding user accounts, 3-8  
  advantages, 3-6  
  removing user accounts, 3-11

system configuration  
  devising backup plan, 9-4

system maintenance  
  removing user accounts, 3-10

system management  
  adding user accounts, 3-5  
  customizing login environment, 4-1  
  files for user accounts, 4-1  
  files executed at system startup, 2-1  
  forcing password change, 3-14  
  maintaining system, 7-13 to 7-17  
  monitoring disk space, 7-14 to 7-16  
  passwords, 3-13  
  removing user accounts, 3-10  
  setting up system, 1-1  
  system manager account, 3-3

System Monitors (CRM), 8-7

system security

- adding passwords, 3-13
- forcing password change, 3-14
- removing passwords, 3-16
  - from root, 3-17
  - from Utility Pages, 3-19
- root account, 3-3

system setup

- adding user accounts, 3-5
- initial tasks, 1-1

system startup

- files executed, 2-1
- in multiuser mode, 2-9
- setting global variables in /etc/profile, 4-6

systemv

- removing products to reclaim disk space, 7-14 to 7-16

T

tab setup menu, 5-14

tape archive utility (tar), 9-11

tape drive

- allocating remote drive, 9-4, 9-22
- backing up to, 9-4, 9-5, 9-22
- backing up to using backup and restore, 9-18
- using scp/cpio, 9-13

tapes

- exchanging workstation and VAX, 9-20, 9-21
- using rtc command, 9-4, 9-5, 9-22

tar, 9-11

temporary files

- removing, 7-14

terminal

- getty process, 2-5
- setting default terminal type, 2-5

terminal characteristics

- setting up, 5-10

terminal settings, 2-8

termio, 2-8

timezone

- setting on Utility Page
  - server, 10-34
  - workstation, 10-4

topcpu (CRM), 8-15

topfault (CRM), 8-12

topio (CRM), 8-15

topmem (CRM), 8-14

topsys (CRM), 8-17

**troubleshooting**

- rebuild options, 11-33 to 11-50
- rebuilding using nonstandard partitions, 11-20
- reasons to rebuild, 11-2
- when you receive bad block messages, 11-2

**U****UID, 4-5****ulimit, 4-6****umask, 4-8****user accounts**

- /etc/passwd file, 4-4
- adding, 3-5
- adding passwords, 3-14
- adding with addusers, 3-6
- adding with sysadm, 3-8
- customizing .env file, 4-9
- customizing variables in .profile file, 4-8
- removing, 3-10
- removing passwords, 3-16

**user types: owner, group, other, 3-20****users**

- adding user accounts, 3-5
- assigning passwords, 3-13
- customizing login environment, 4-1
- group identification code (GID), 4-5
- identification code (UID), 4-5
- removing passwords, 3-16
- removing user accounts, 3-10
- user accounts file, 4-4

**usr partition, 11-21****utd, C-4****Utility Pages**

- format, 10-23, 10-43
- reference for workstations and servers, 10-1
- removing password, 10-4, 10-35
- server, 10-30
  - Device Image Copy Utility Page, 10-46
  - Disk Maintenance Utility Page, 10-41
  - Disk Partitioning Utility Page, 10-44
  - Hard Disk Flaw Data, 10-43
  - Main Utility Menu, 10-33
  - Peripheral Configuration Utility Page, 10-7
  - Startup Configuration Utility Menu, 10-32
  - using, 10-31
  - verify hard disk, 10-43

workstation

- Device Image Copy Utility Page, 10-26
- Disk Maintenance Utility Page, 10-13
- Disk Partitioning Utility Page, 10-23
- Format Utility Page, 10-21, 10-44
- Hard Disk Flaw Data, 10-16
- Main Utility Page, 10-3
- Operating System Parameters Page, 10-11, 10-40
- Peripheral Configuration Utility Page, 10-37
- Verify Utility Page, 10-19,

utilities

- System V and PC-DOS, C-1

V

VAX/VMS

- backing up to remote VAX tape drive, 9-21
- reading VMS-generated backup tapes, 9-21

Verify

- explanation, 11-10
- on server, 11-12
- on workstation, 11-11

Verify Utility Page

- server, 10-43
- workstation, 10-19

virtual screens, 5-30

- changing number of, 5-30

vlt

- changing default screen color, 5-33

VMSbackup, 9-21

vt200

- customizing, 5-2, 5-4

vt220, 5-2

- options for creating windows, 5-4
- setup menu, 5-8 to 5-16

vterm, 5-4

- command-line options, 5-4 to 5-8

W

watcher (CRM), 8-19

window icon

- creating symbol for collapsed state, 5-7

window size

- customizing, 5-7

**windows**

- attaching printer to, 5-15
- cataloging, 5-24
- creating, 5-4
  - a collapsed window, 5-7
  - a color window, 5-6
    - using ANSI, 5-22
    - using regis commands, 5-18, 5-19
  - a log file of activity, 5-7
  - a specific title, 5-7
  - a window that uses symbol when collapsed, 5-7
  - a window with small text, 5-6
  - an inactive window, 5-6
  - console, 5-36, 5-6
  - from command line, 5-4
  - that logs you in automatically, 5-7
  - through workstation menu, 5-8
  - to uncollapse when it receives output, 5-6
  - using catalog file, 5-24
  - with specific origin and size, 5-6, 5-7
- creating log file, 5-7
- customizing defaults in `.def_proc` file, 5-35
- customizing defaults in `def_proc` file, 5-34
- customizing defaults in `sys_proc` file, 5-37
- saving window configuration, 5-24
- setting display characteristics, 5-12
- setting up terminal characteristics, 5-10
- vterm options, 5-4 to 5-6

**word size**

- server, 10-38
- workstation, 10-9

**workstation menu**

- commands associated with menu options, 5-27
- configure, add option, 5-26
- configure, background, 5-32
- configure, delete option, 5-28
- configure, modify option, 5-28
- configure, screens option, 5-30
- creating windows, 5-4
- customizing
  - adding processes to, 5-26
  - deleting processes on, 5-28
  - modifying processes on, 5-28

**write access, 3-20**

**X**

xon/xoff

setting on server, 10-39

setting on workstation, 10-10

**Z**

z-buffering, 10-12