Stardent

Installation/ Administration Guide

Stardent Computer Inc.

Part Number: 340-0119-01

CONTENTS

Section 2 Configuration

11 System Overview

12 NVRAM Variables

Section 3 Administration

23 System Processes

24 File System Administration

25 User Services

26 System Security and Tuning

27 Software Installation

28 System Crash Procedure

List of Figures

List of Tables

PREFACE

This document covers installation, configuration and administration of the the Stardent 1500/3000 Graphics Supercomputer, the Stardent 1500/3000 Computational Server and the Stardent 1500/3000 Expansion Cabinet. It shows you how to get your Stardent 1500/3000 installed and running; it describes how to configure the system; it describes how to perform ongoing tasks such as creating file systems, adding users, creating backup tapes, and shutting down the system; and it describes how to install software updates. It also includes, as an appendix, a complete manual for the Stardent 1500/3000 PROM.

The guide assumes that you are familiar as an end user with the UNIX operating system. Its focus is on specific tasks and procedures, not on exhaustive descriptions of command options, file formats, and so on. For that information, we encourage you to supplement your use of this guide with the following:

- The *Commands Reference Manual*, for full explanations of system administrative commands.
- The *Programmer's Reference Manual*, for file formats and utilities.
- The *Network File System Manual*, for NFS system administration.
- Any generally available books on UNIX system administration. O'Reilly and Associates, Inc., Newton, Mass. publishes books on a variety topics relating to UNIX System Administration. Specifically for networking, we recommend the O'Reilly book, "Managing UNIX mail" and D.E. Comer's *Internetworking With TCP/IP: Principles, Protocols and Architecture*, Prentic Hall, N.J., 1987.

Stardent

INSTALLATION

Section 1

UNPACKING

CHAPTER ONE

This chapter describes how to unpack the Stardent 1500/3000 Graphics Supercomputer, Stardent 1500/3000 Server or Stardent 1500/3000 Expansion Cabinet. It also includes general warnings.

NOTE
Two people are needed for unpacking and placement.

The term System Module refers to the cabinet that houses the Stardent 1500/3000 processors, memory, etc. The User Interface Module consists of the monitor, junction box, keyboard, mouse and other input devices. The User Interface Cable is a thick 50 or 200 foot cable with multiple connectors at each end. The User Interface Cable is used to connect the Stardent 1500/3000 System Module to the User Interface Module's junction box.

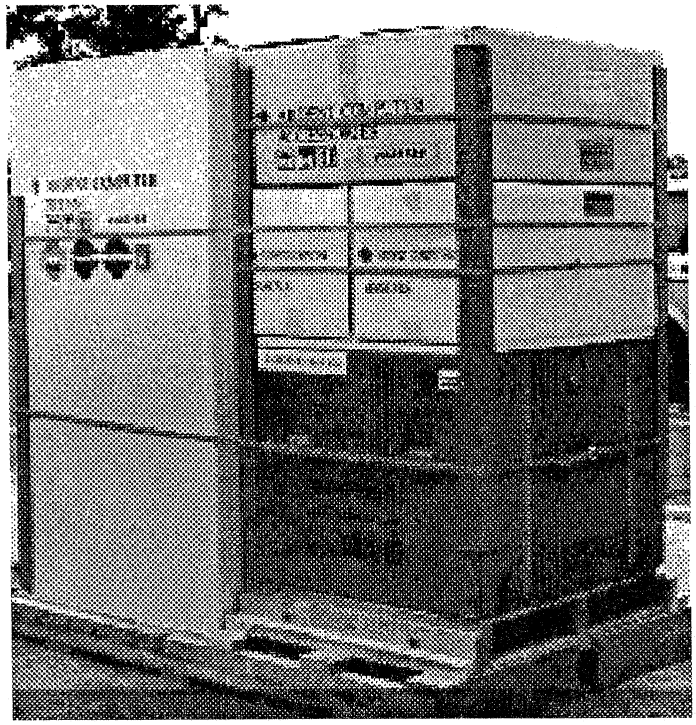
- The Stardent 1500/3000 Graphics Supercomputer is shipped in several containers that together hold the System Module, the monitor, other components of the User Interface Module, cables, documentation and cartridge tapes. The containers are shipped on one large pallet.
- The Stardent 1500/3000 Server is shipped on one large pallet. Containers hold the System Module, documentation and cables.
- The Stardent 1500/3000 Expansion Cabinet and cables are shipped together on one large pallet-mounted container. External mass storage peripherals are shipped in separate boxes.

Special care must be taken in unpacking the System Module or Expansion Cabinet. The next section gives unpacking instructions. Before unpacking the monitor for the Stardent 1500/3000 Graphics Supercomputer, remove the accessory and documentation boxes from the shipping pallet. Then remove the monitor box from the top. Two people can now lift the monitor and move it into place.

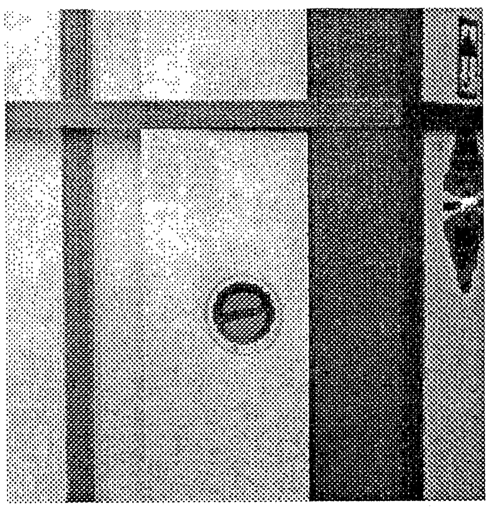
Unpacking the System Module or Expansion Cabinet

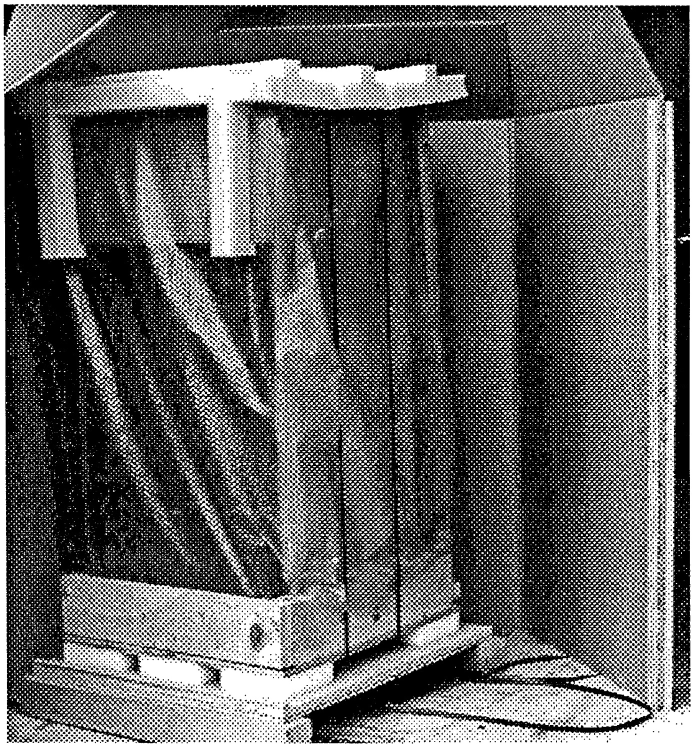
The photographs on the next pages show how to unpack the System Module or Expansion Cabinet. Allow at least 8 feet of clearance to remove the System Module or Expansion Cabinet from its shipping pallet.

Before unpacking the System Module or Expansion Cabinet, inspect the box for obvious shipping damage. Report suspected problems to Ardent Customer Support.

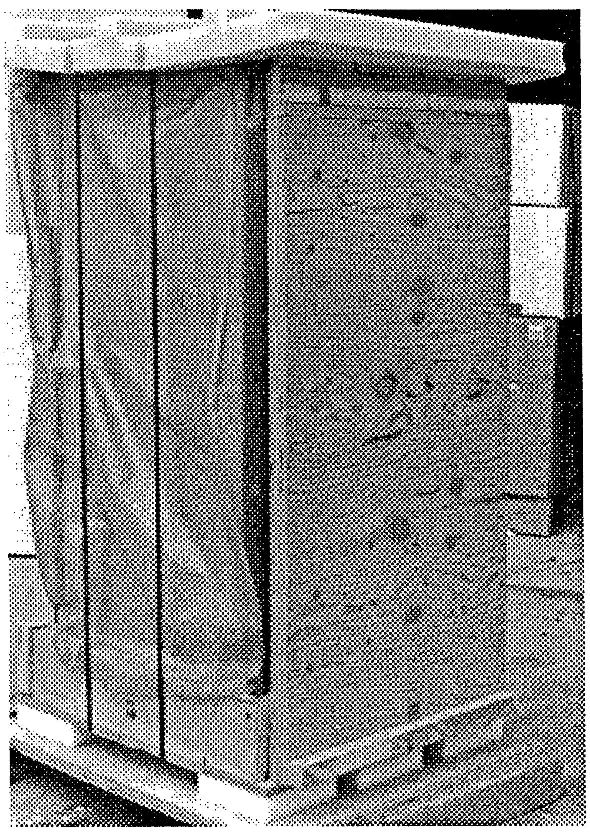


Loosen and remove the plastic screw-on fasteners that hold the box together. Retain the fasteners.





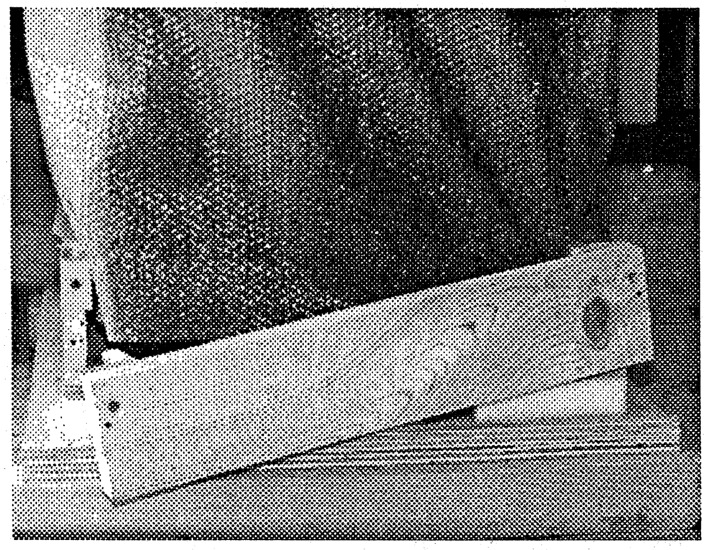
Cut the shipping straps and remove the box.



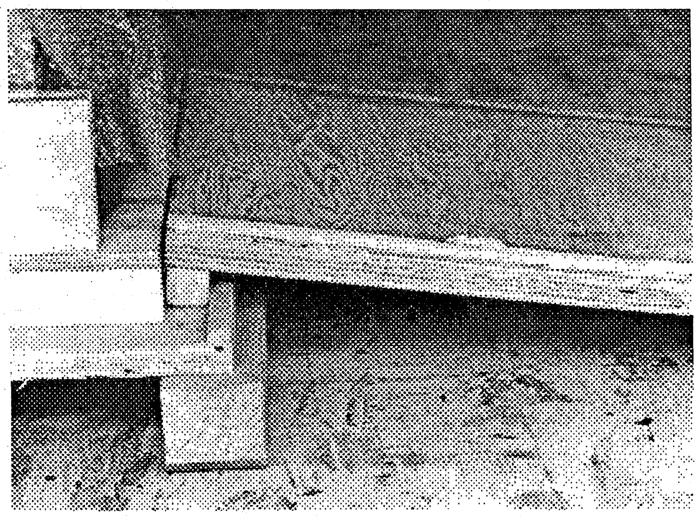
Cut any remaining shipping straps. Locate the wooden ramp that is to be used to remove the System Module or Expansion Cabinet from the cushioned pallet. The ramp is fastened to the pallet with 2 bolts. Using a 14mm ratchet or adjustable wrench, remove the bolts before removing the foam cap from the top of the System Module or Expansion Cabinet.

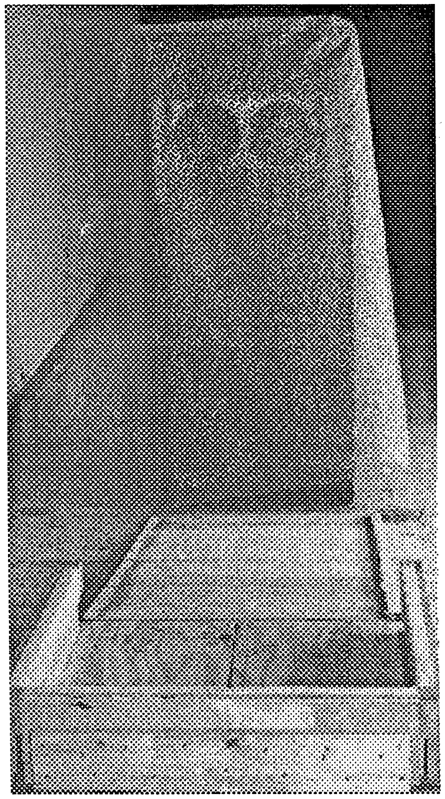
Unpacking the System Module or Expansion Cabinet
(continued)

Locate the two bolts on the opposite side of the pallet from the ramp. These 2 bolts hold a wooden support that must be removed before the System Module or Expansion Cabinet can be unloaded. Using a 14mm ratchet or adjustable wrench, remove the 2 bolts and the wooden support they secure.

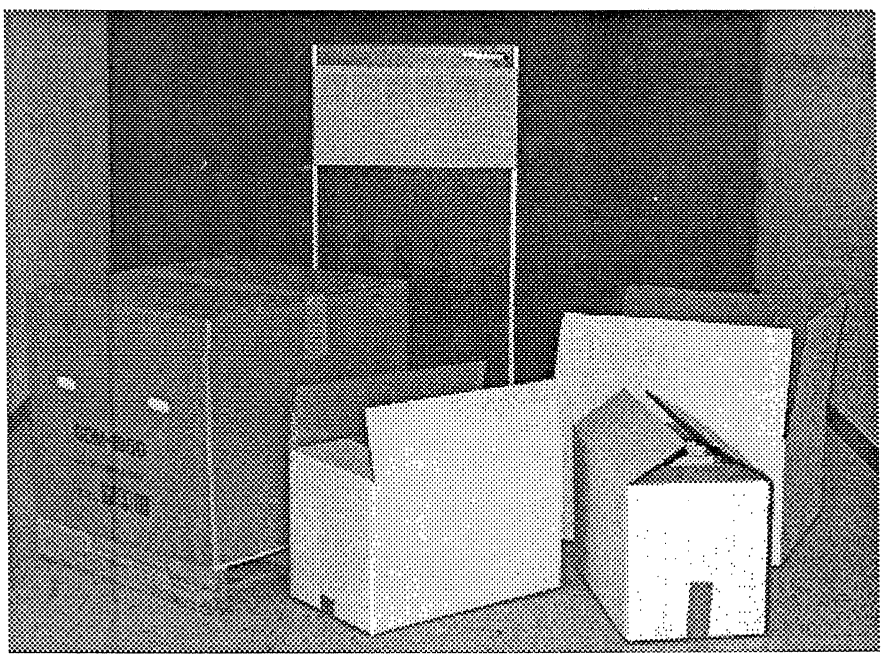


Put the wooden ramp in place where the wooden support was just removed. At the top of the ramp is a lip that should be placed with a corresponding lip on the cushioned pallet.





Carefully slide the System Module or Expansion Cabinet down the ramp. (The unit has built-in casters.) *The System Module or Expansion Cabinet should slide smoothly, with no bumps or jolts.*



Once the System Module or Expansion Cabinet is on the floor, roll it into place.

Unpacking the System Module or Expansion Cabinet (continued)

Inventory

Each system is shipped with a packing list and a configuration sheet. Check the two lists to ensure that they are consistent and that they correspond to the equipment you have received.

Warnings

The front door of the Stardent 1500/3000 System Module should remain closed and locked during system operation.

During operation each slot in the card cage should be filled with a circuit board or board filler.

The System Module should be a minimum of 12 inches from the wall.

The power should be turned off completely before anything is removed from the System Module cabinet.

The circuit breaker should be able to handle the expected load. (See the *Site Preparation Guide* for power specifications.

Circuit boards and devices should not be removed without consulting the procedures in the *Field Service Manual.*

Air vents should not be blocked.

INSTALLING THE GRAPHICS SUPERCOMPUTER

CHAPTER TWO

This chapter describes how to place and connect the Stardent 1500/3000 Graphics Supercomputer. It assumes that a site has been prepared for the system according to the information in the *Site Preparation Guide.*

Placement

Follow the instructions in the chapter entitled *Unpacking* to unpack and inventory the system. To allow sufficient air flow, place the Stardent 1500/3000 System Module at least 12 inches from the back wall. To provide clearance for the back door, place the System Module about 25 inches from the back wall.

You may locate the monitor up to 50 or 200 feet from the System Module, depending upon the length of the User Interface Cable shipped with the system. Consult the *Site Preparation Guide* for instructions on routing the User Interface Cable.

NOTE
The term System Module refers to the cabinet that houses the Stardent 1500/3000 processors, memory, etc. The User Interface Module consists of the monitor, junction box, keyboard, mouse and other input devices. The User Interface Cable is a thick 50 or 200 foot cable with multiple connectors at each end. The User Interface Cable is used to connect the Stardent 1500/3000 System Module to the User Interface Module's junction box. (See Figures 2-2 and 2-3.)

Card Cage Configuration

The Stardent 1500/3000 card cage has been configured at the factory according to the quantities and types of CPU, memory, I/O and graphics boards you ordered. Figure 2-1 shows the following general card cage configuration options:

- Standard
- Dual I/O, Single Graphics
- Dual I/O, Dual Graphics
- Dual I/O, No Graphics (Server configuration option)

In the figure each label (cpu, memory, etc.) corresponds to a card cage board type. If more than one label is listed for a given card cage slot, that means that more than one board type can be placed in the slot. The horizontal bars below the labels refer to order of placement for that board type. For instance, each configuration shows the CPU board in slot 5 with a single bar. That means that for single CPU board configurations, the CPU board is placed in slot 5. For multiple processor configurations the second CPU board is placed in slot 2, and so on.

The next sections describe how to install the Stardent 1500/3000 Graphics Supercomputer (options 1-3 in Figure 2-1). See Chapter 3 for instructions on installing the Stardent 1500/3000 Server (option 4 in Figure 2-1).

Cabling: Single User Interface

Follow the instructions in this section if your system has a single Stardent 1500/3000 monitor and either one or two I/O boards. (Options 1 or 2 in Figure 4-1.)

1. Routing the User Interface Cable. The User Interface Cable is a thick, 50 or 200 foot cable with multiple connectors at each end. If you plan to locate the Stardent 1500/3000 System Module and Stardent 1500/3000 monitor in separate rooms you may need to route the User Interface Cable through ductwork, ceiling or false floors. When you do so take care that sources of electro-magnetic fields, such as light ballasts and motors, are avoided. (The User Interface cable connectors are 2.2 inches wide at the System Module end and 2.9 inches wide at the User Interface end. Make sure conduits are wide enough to accommodate connectors.

NOTE
If you ordered a stereo monitor with your system, a 50 or 200 foot Stereo Interface Cable has been shipped to you. We recommend that you tape the Stereo Interface Cable to the User Interface Cable with electrician's tape so the two cable can be routed together through ceilings, floors or ductwork. Follow the cabling instructions in this chapter, then refer to the chapter entitled *Stereo Monitor* for information on connecting the Stereo Interface Cable and using the monitor in stereo mode.

Connecting the User Interface Cable. The User Interface Cable has well-marked connectors at each end for attachment to the Stardent 1500/3000 System Module and junction box (the small black box with multiple ports shipped with the system). There are 6 connectors at the System Module end and 4 connectors at the junction box end. Attach the User Interface Cable to the System Module and junction box according to the illustration in Figure 2-2 and the information in Table 2-1. If you have a dual I/O system, use the I/O board in slot 1, as shown in Figure 4-2.

2. Junction box to monitor connections. Connect the junction box to the monitor according to the illustration in Figure 2-2 and the information in Table 2-1. There are three coaxial cables (R-G-B) and a short AC cord.

3. System Module AC line cord. Plug the AC line cord for the System Module into the lower right rear of the System Module. A power switch is located just above the line cord socket. Make sure it is off.

Table 2-1. Stardent 1500/3000 Cable Connections

User Interface Cable to System Module	
Connector Label	Connect To
RS-232C	RS-232 port C on I/O board
RS-232D	RS-232 port D on I/O board
Keybd-mouse-spkr	Keybd-mouse-spkr port on I/O board
RED	RED connector on graphics board
GREEN	GREEN connector on graphics board
BLUE	BLUE connector on graphics board
User Interface Cable to Junction Box	
Connector Label	Connect To
System In	System in port on junction box
RED	RED Inport on junction box
GREEN	GREEN In port on junction box
BLUE	BLUE In port on junction box
Junction Box to Monitor	
Connector Label	Connect To
RED Out	RED connector on monitor
GREEN Out	GREEN connector on monitor
BLUE Out	BLUE connector on monitor
Short AC cord	AC receptacle on monitor

A strain relief bracket is located just to the right of the System Module's AC line cord receptacle. Loosen the Phillips screw that holds the strain relief bracket in place, then turn the bracket so that it secures the AC line cord. Tighten the screw.

Plug the other end of the System Module line cord into an appropriate wall outlet. (See the *Site Preparation Guide* for power specifications.) If possible, test the outlet with a voltmeter.

4. Junction box AC line cord. Plug the AC line cord for the User Interface Module into its socket on the junction box and into an appropriate wall outlet.

5. Other input devices. Attach knob box and tablet cables (if you are installing those accessories). See the chapter entitled *Knob Box and Tablet* for details.

6. System Module cable routing. Route cables and cords behind the metal cable and cord guard on the back of the System Module. (See Figure 2-4 for location of the metal guard.) You may now close the back door of the System Module.

You may now connect other components (network, terminals, printers, modems, external mass storage devices) or boot the system. See the the remaining chapters in this section for instructions on installing mass storage peripherals and booting the system and see the *Configuration* section of this guide for instructions on installing and configuring other peripheral components.

Cabling: Dual I/O Board, Dual Graphics

If you ordered a system with two Stardent 1500/3000 monitors, (option 3 in Figure 2-1), two I/O boards and two graphics boards have been installed in the Stardent 1500/3000 card cage and two User Interface Cables and two junction boxes have been shipped to you.

At the System Module end, connect one of the User Interface Cables to the I/O board in slot 1 and the graphics board in slot 4. Connect the other User Interface Cable to the I/O board in slot 8 and the graphics board in slot 7. (Refer to Figure 4-3.) Follow the steps in *Cabling: Single User Interface* above for the rest of the instructions.

Mounting the Junction Box

The junction box can be placed behind the monitor on a desk or table, or it can be mounted as shown in Figure 2-5. Use screws to mount the junction box bracket shown in the figure, then slide the junction box onto the bracket.

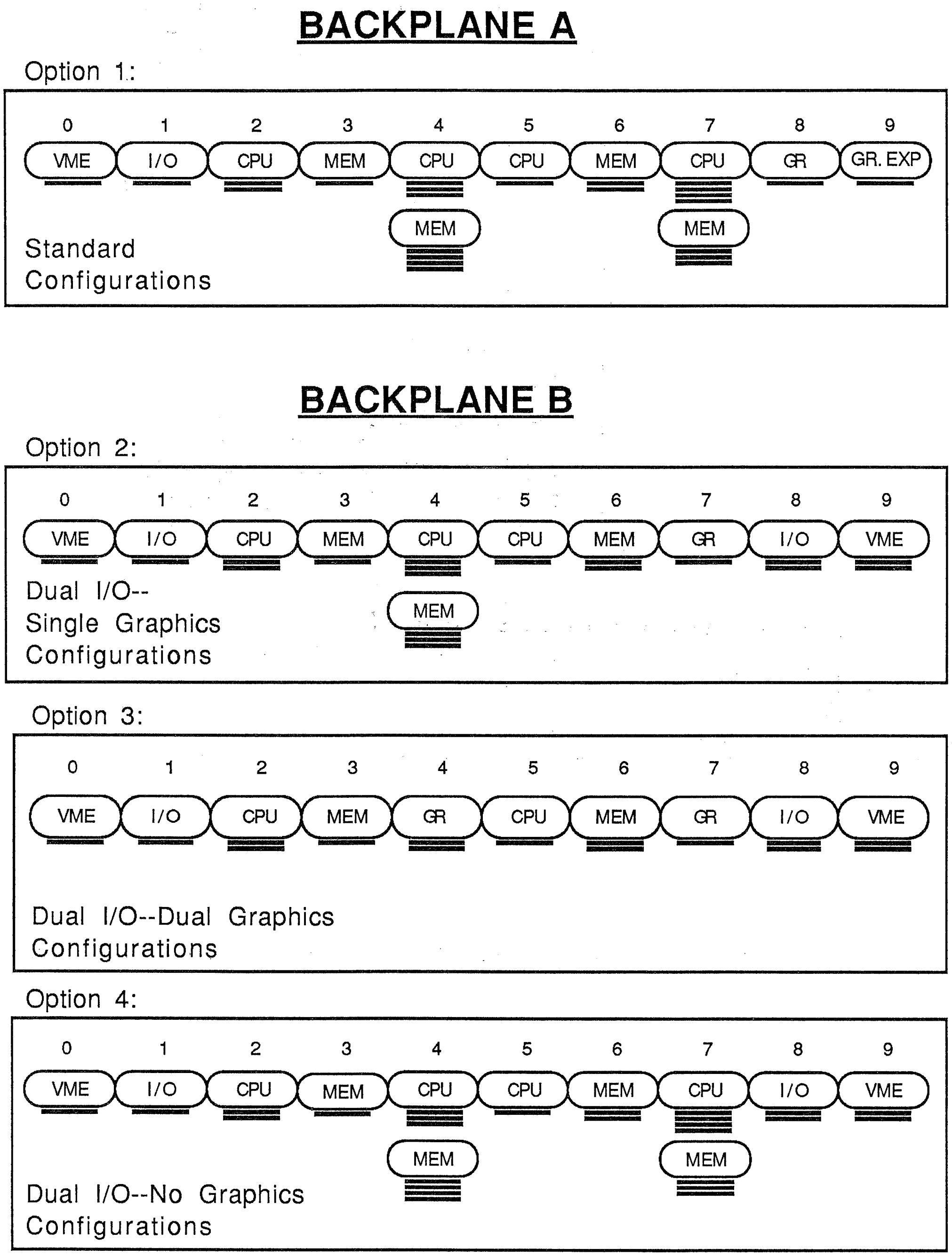


Figure 2-1. Stardent 1500/3000 Card Cage Configuration Options

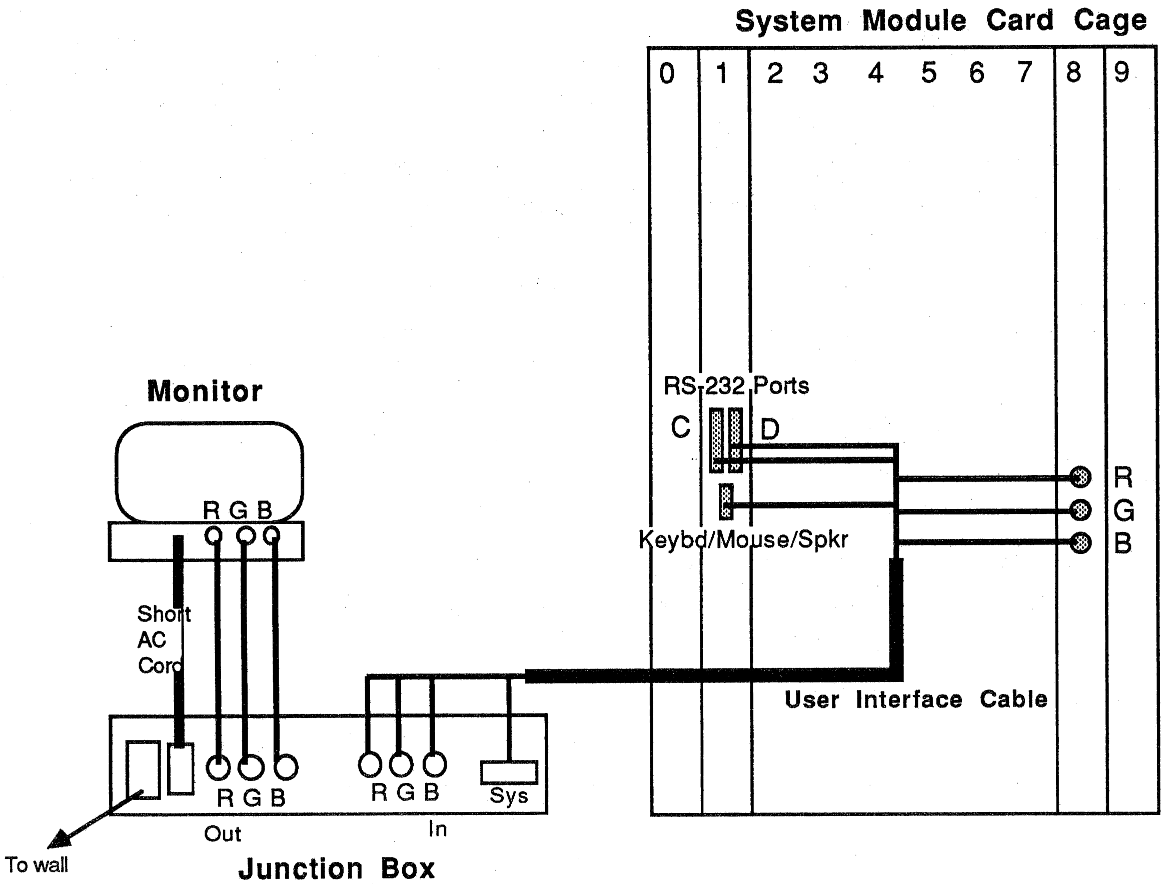


Figure 2-2. Stardent 1500/3000 Cable Connections: Single User Interface

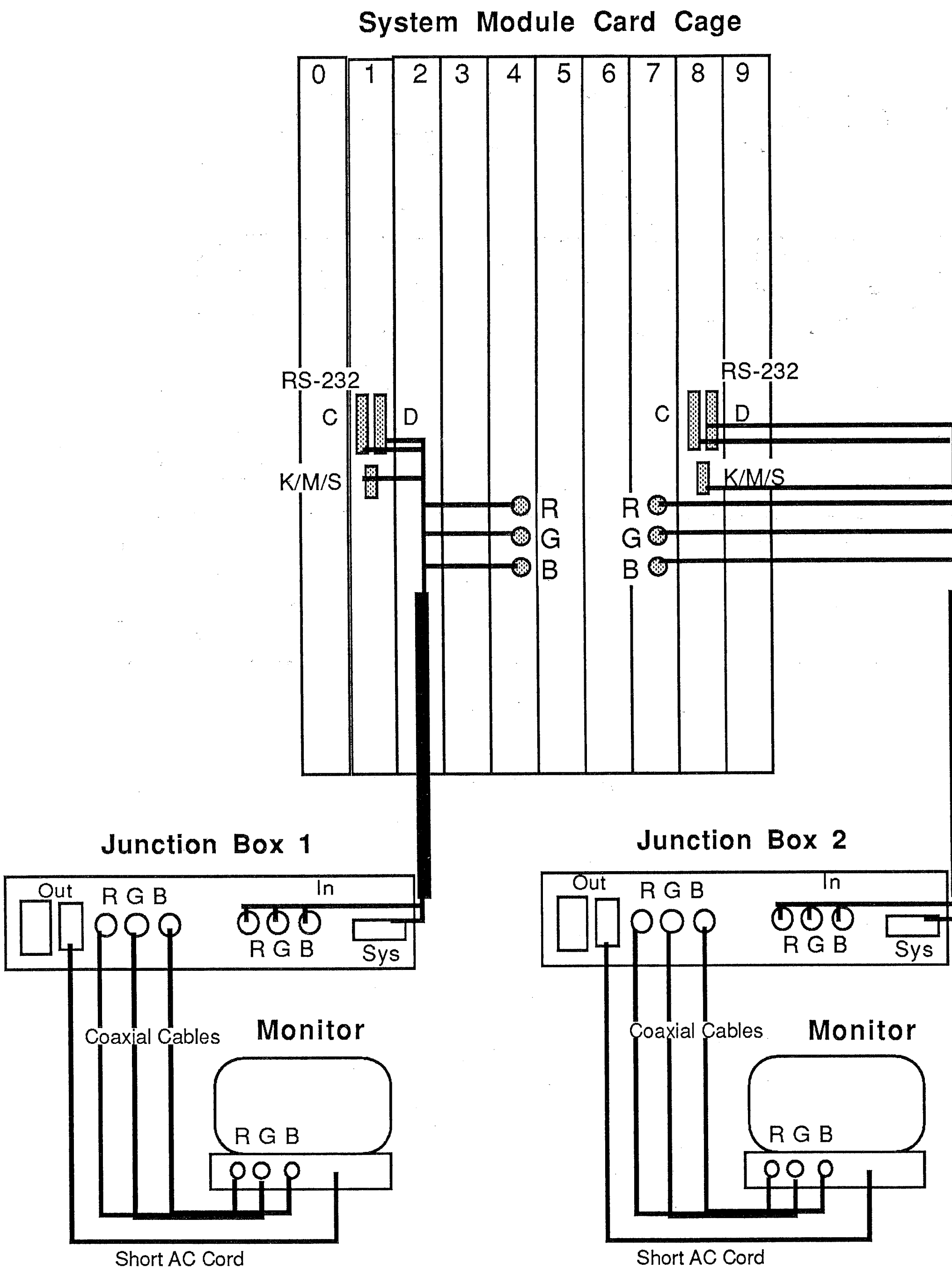


Figure 2-3. Stardent 1500/3000 Cable Connections: Dual User Interface

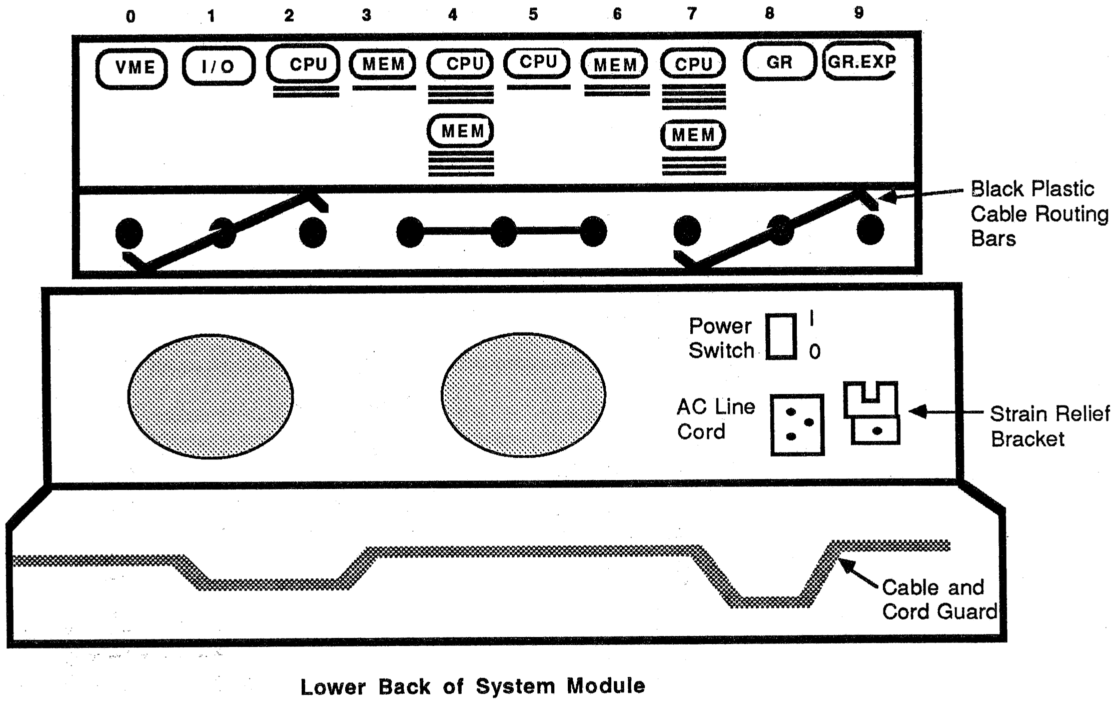


Figure 2-4. Stardent 1500/3000 System Module Cable Routing

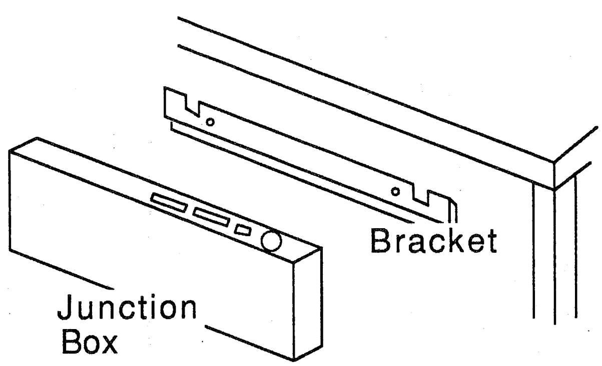


Figure 2-5. Junction Box Mounting

INSTALLING THE SERVER

CHAPTER THREE

Follow the instructions in chapter entitled *Unpacking* to unpack and inventory the system. To allow sufficient air flow, locate the Stardent 1500/3000 Server System Module at least 12 inches from the back wall. To provide clearance for the back door, locate the System Module about 25 inches from the back wall. Once the Stardent 1500/3000 Server System Module is in place, plug the AC line cord into the lower right rear of the System Module. Plug the other end of the line cord into an appropriate wall outlet. (See the *Site Preparation Guide* for power specifications.) If possible, test the outlet with a voltmeter.

Connecting the System Console

You must supply a VT100-compatible terminal to be used as the system console. The terminal should support

- 8 bit characters, one stop bie, no parity
- CR, NL, LF, TAB, and BS characters
- 9600 and 2400 baud operation.

Make sure the terminal is configured for 9600 baud use.[1] Connect the terminal to the RS-232 Diagnostic Communications Port (DCP) on the Server's boot master CPU board. (An RS-232 cable is provided with your Server.)

[1] During normal operations the system console can always operate at 9600 baud. A 2400 baud link is only required in the unlikely event that the PROM fails during power-up and the front panel key switch is in the diagnostic mode. In that case the NVRAM variable *baud* defaults to 2400, and messages are sent out the DCP at 2400 baud. At that point you must reconfigure the terminal to be able to communicate with your Stardent 1500/3000 server. Press the BREAK and RETURN keys in sequence, repeatedly, until you get recognizable output on the screen. Pressing the BREAK and RETURN keys cycles you through the terminal's available baud rates until 2400 is reached. Once you get a PROM prompt at 2400 baud, you can reset the NVRAM variable *baud* to 9600. When you power-up again, you are in 9600 baud mode.

Table 3-1 shows boot master CPU card cage assignments and Figure 3-1 shows the location of the DCP.

Table 3-1. Boot Master CPU Board Assignments

Number of CPU Boards	Slot Containing Boot Master CPU
1	5
2	2
3	4
4	4

Once the System Module and console are in place you may connect other components (network, terminals, printers, modems, external mass storage devices) or boot the system. See the remaining chapters in this section for instructions on installing mass storage peripherals and booting the system and see the *Configuration* section of this guide for instructions on installing and configuring other peripheral components.

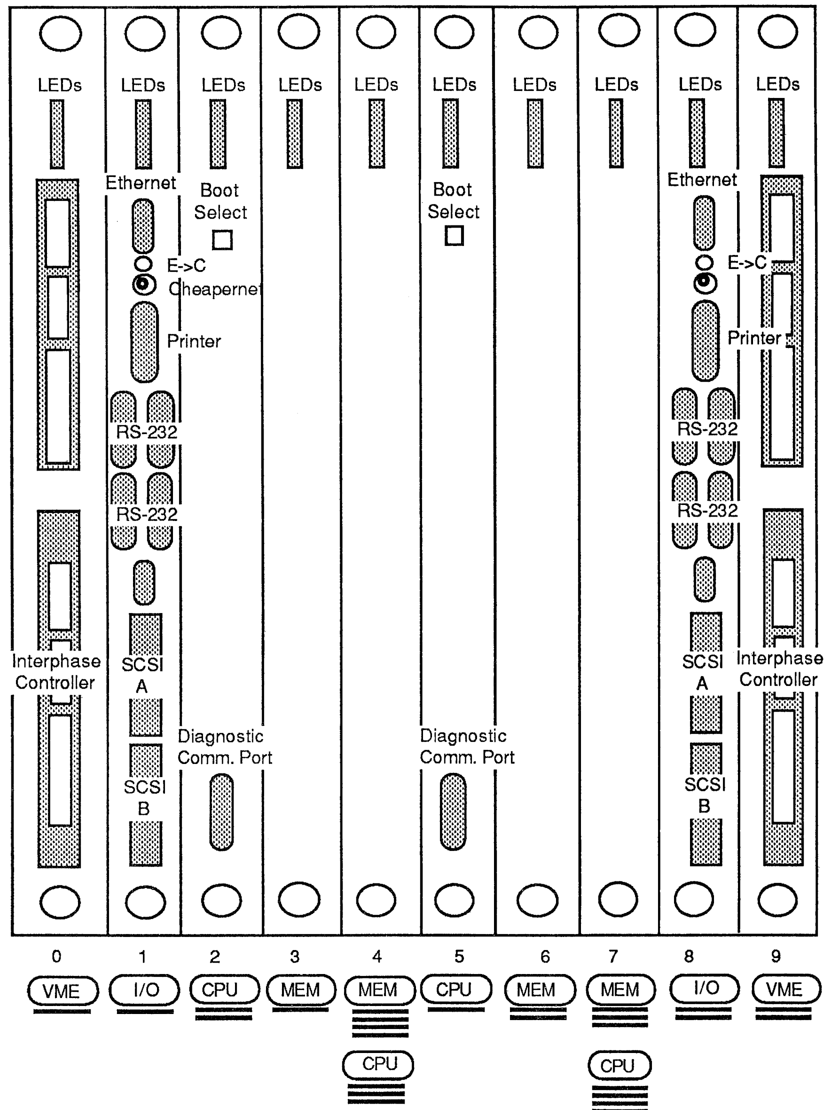


Figure 3-1. Stardent 1500/3000 Server Backpanel

INSTALLING THE EXPANSION CABINET

CHAPTER FOUR

The Stardent 1500/3000 Expansion Cabinet (Stardent Model N-ENC/100 or N-ENC/200) is used to mount external peripherals for the Stardent 1500 and 3000 Visualization Systems and Departmental Supercomputers. The cabinet can support

- a VME card cage
- a Half-inch tape drive
- Dual SMD Disk Drawer(s)
- Quad SCSI Disk Drawer(s).

This chapter shows how to place and install the Expansion Cabinet, gives guidelines for mounting peripherals and describes cabling, front door removal, anti-tip protection, and the cabinet's AC power controller. Chapters 5 through 8 explain how to install the peripheral drawers themselves.

WARNING

If your Expansion Cabinet has a VME card cage, always power on the VME card cage before powering on the system. If the power on order is reversed the system may not boot or operate correctly.

Never pull more than one peripheral drawer out at a time.

Be careful not to bend pins when you connect cables.

Installing the Expansion Cabinet

The Expansion Cabinet is similar in dimension and appearance to the Stardent 1500/3000 System Module. It is 50.5 inches high, 22.5 inches wide, and 36 inches deep.

The Expansion Cabinet is shipped in one large pallet-mounted container. If you ordered a VME Card Cage, the card cage and cables are shipped in the same pallet-mounted container. The Half-Inch Tape Drive, Quad Disk Drawer and the Dual SMD Disk Drawer and associated cables are shipped individually in separate boxes.

Follow the general instructions in the chapter entitled *Unpacking* to unpack your Expansion Cabinet and follow the instructions in Chapters 5-8 of this document to unpack peripherals. Check your shipping order and configuration sheets to make sure that you have received a full system.

The steps on the next two pages describe how to place and connect the Expansion Cabinet.

Step 1.

Move the Expansion Cabinet into place on the right of the System Module, facing front. Figure 4-1 shows the relative location of the two units with front panels flush and the Expansion Cabinet located at least 12 inches from the back wall.

Step 2.

Locate the flat metal coupler and the sets of brackets, bolts, and Phillips screws (4 each) that have been shipped on the inside of the Expansion Cabinet. The coupler provides strain relief and keeps the Expansion Cabinet and System Module a fixed distance apart (6 inches). Figure 4-2 shows the appearance of the metal coupler.

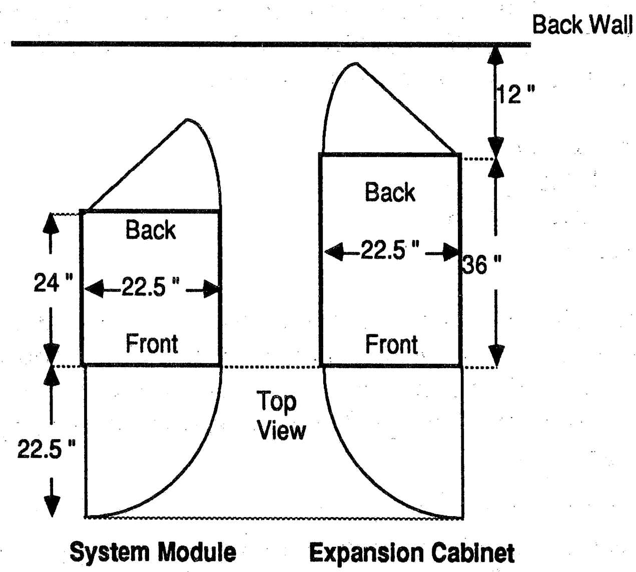


Figure 4-1. Expansion Cabinet and System Module Footprint

Step 3.

Use bolts to attach the coupler brackets to the System Module and the Expansion Cabinet. Each cabinet has holes at the bottom to accommodate the bolts. See Figure 4-2 for proper placement of the brackets.

Step 4.

Move the Expansion Cabinet and System Module into place about 6 inches apart and with front panels flush. Place the metal coupler on top of the coupler brackets and adjust the position of the Expansion Cabinet and System Module so the holes in the coupler match up with the holes in the tops of the coupler brackets.

Step 5.

Use the Phillips screws to attach the coupler to the coupler brackets.

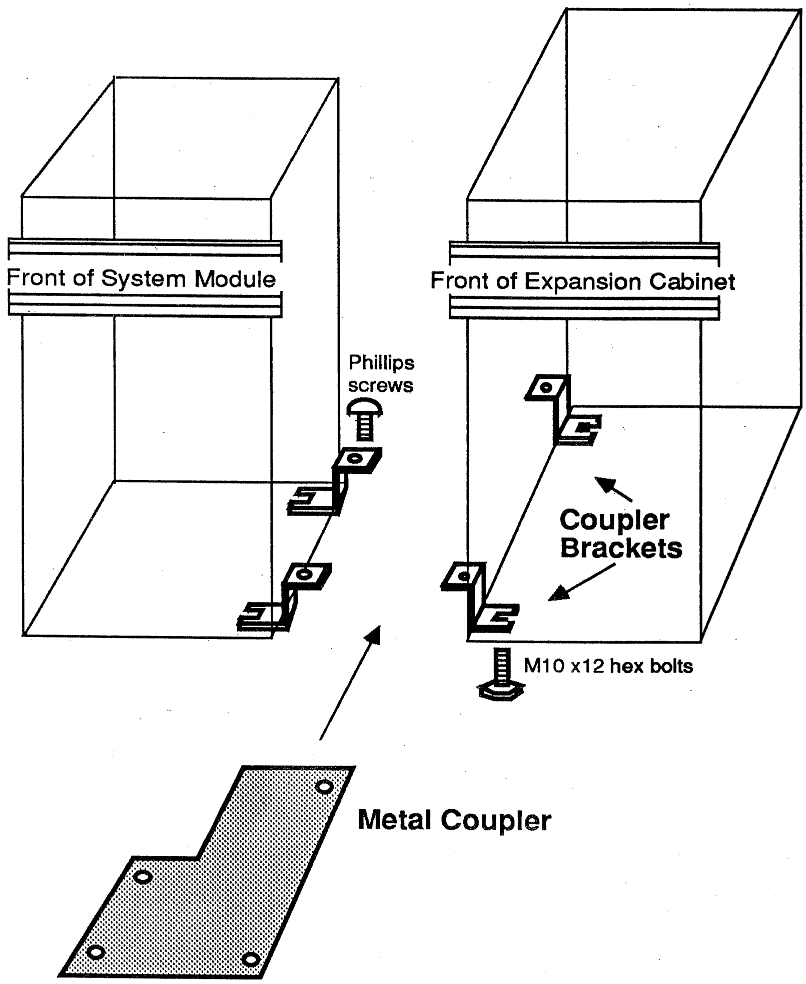


Figure 4-2. Expansion Cabinet Installation

Step 6.

Follow the instructions in Chapters 5-8 to install peripherals in the Expansion Cabinet.

Step 7.

Once all the peripherals have been installed, connect the Expansion Cabinet's AC line cord to the AC Power Controller at the lower back of the Expansion Cabinet and to an appropriate wall outlet.

Once the System Module and Expansion Cabinet are installed you can connect other components or boot the system. See *Booting and Shutting Down the System* in this guide for booting instructions and see *Disks* if you need to format disks. Please read the caution at the beginning of this chapter before you boot the system.

Power Options

The Expansion Cabinet can be configured for 120 Volt/20 Amp or 240 Volt/16 Amp operation. You choose a power option at the time the cabinet is ordered, consistent with the power requirements of the peripherals you plan to install. As indicated below, the total current requirement of the Expansion Cabinet (20 Amps or 16 Amps) constrains the number and type of peripherals you can install.

The Expansion Cabinet uses the same line cord as the Stardent 1500/3000 System Module: 115V/30A or 230V/20A. See the *Site Preparation Guide* for line cord specifications.

Mounting Peripherals in the Expansion Cabinet

The Expansion Cabinet accepts up to 25 EIA units of 19-inch rack-mounted peripherals. Two of the EIA units are required for the AC power controller; the other 23 units are available to mount peripherals.

When you order an Expansion Cabinet with peripherals, the Cabinet arrives with peripheral mounting rails already installed, so the location of the peripherals in the Expansion Cabinet is predetermined. Any subsequent changes in your Expansion Cabinet configuration must be consistent with the following constraints:

• *Space Constraint:* Total space used by the peripheral modules cannot exceed 23 EIA rack units.

• *Power Constraint:* Total current requirement for installed peripherals cannot exceed 20 Amps for the 120 Volt option or 16 Amps for the 240 Volt option.

Table 4-1 gives current and height specifications for individual peripheral modules. Note that current requirements for the VME Card Cage are a function of the number of boards installed. Table 4-1 gives an upper bound based the current maximum configuration of one VME repeater and four InterphaseV/4200 Controller boards; Table 4-2 gives requirements on a per-board basis.

• *Location Constraint:* The 1/2-inch tape drive must be placed at the top of the Expansion Cabinet chassis. The VME card cage must be located at the bottom of the chassis, just above the AC Power Controller. To assure optimal performance other peripheral drawers should be spaced equally in the cabinet.

• *Quantity Constraint:* The Expansion Cabinet supports a maximum of one 1/2-inch tape drive and one VME card cage. You can install more than one dual disk drawer or quad disk drawer as long as the total number of modules does not exceed six (The AC power converter has six sockets), and you install a maximum of one SCSI Quad drawer per Stardent 1500/3000 SCSI controller.

Figure 4-3 gives example configurations that meet the power, space and location constraints mentioned above.

Table 4-1. Expansion Cabinet Peripherals; Current and Height Constraints

Module	Current (Amps) 120V	240V	Height (EIA Units)
HP 88780A 1/2" Tape Drive	3.0	1.6	5
Quad Disk Drawer	3.5	2.0	3
Dual Disk Drawer	6.0	3.5	3
VME Card Cage (VME Repeater + 4 Interphase 4200 boards)	3.2	1.6	12
Total Requirement	<=20	<=16	<=23

Cabling

Specific cabling requirements for each peripheral module are contained in the appropriate chapter of this guide. Here are general guidelines:

- Route AC line cords from the peripheral module down the left side cable plenum of the Expansion Cabinet (facing rear) to sockets on the AC Power Controller. A single line cord connects the AC Power controller to the wall.

- Route signal cables down the right side cable plenum of the Cabinet to the VME card cage or directly to the System Module.

- Use the grey rubberized cable ducts located along the sides of the Expansion Cabinet chassis to help keep cables in place.

Figure 4-4 shows the cable routing plan.

Table 4-2. VME Card Cage Current Requirements

Item	Current (Amps) 120V	Current (Amps) 240V
VME Card Cage with VME Repeater	.8	.4
Interphase 4200 boards (each)	.6	.3

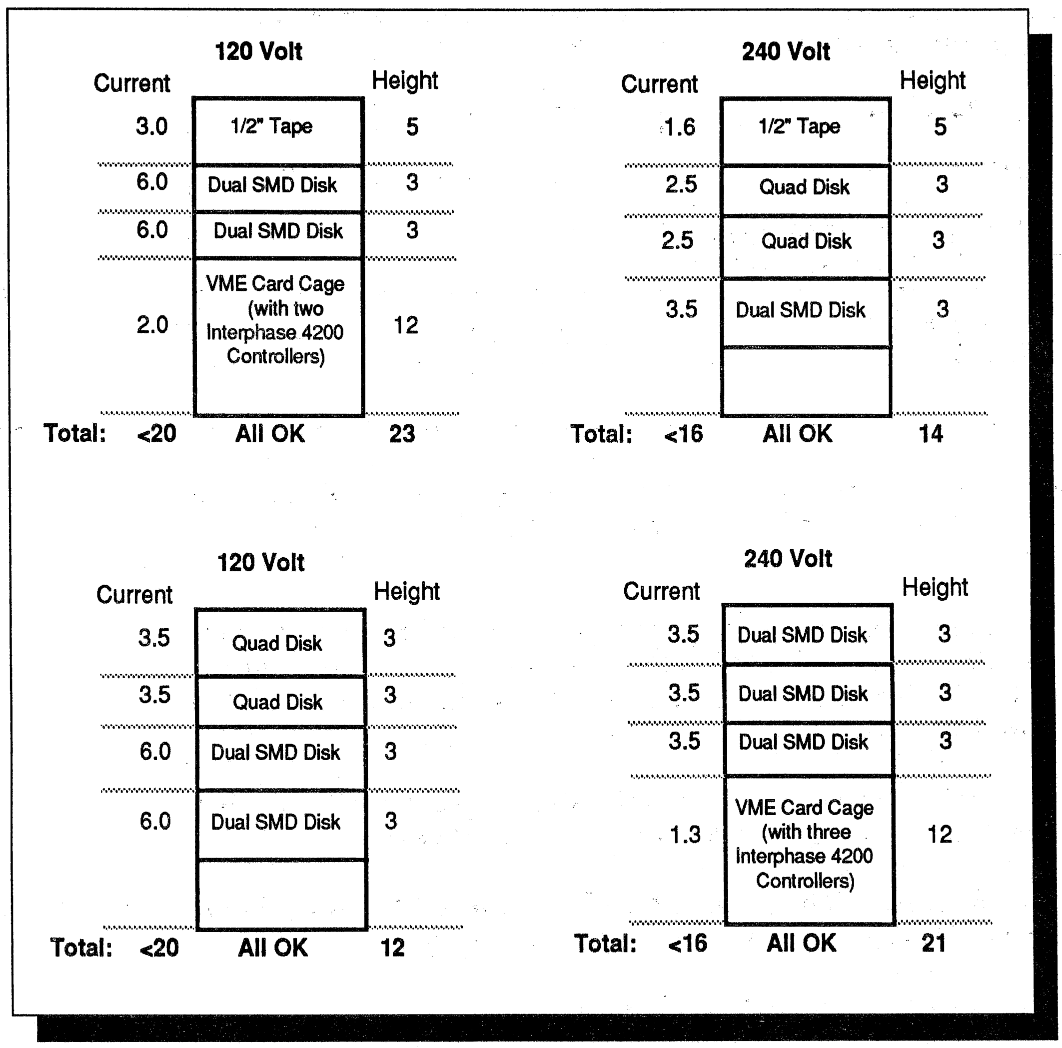


Figure 4-3. Example Expansion Cabinet Configurations

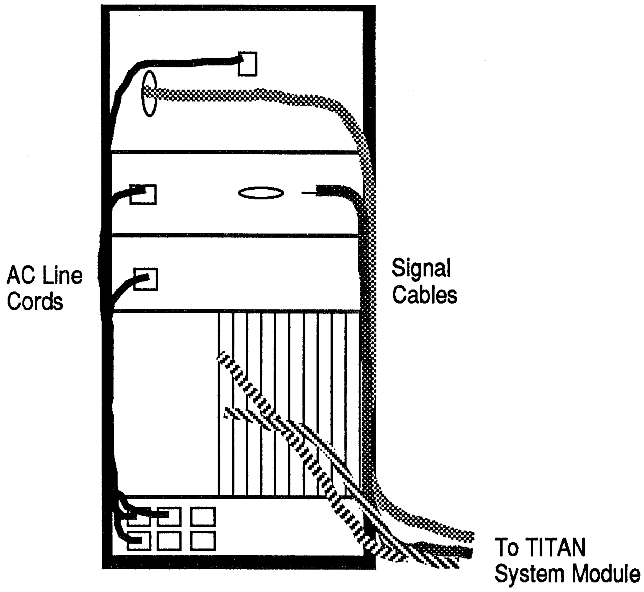


Figure 4-4. Expansion Cabinet Cable Routing Plan (Rear View)

Front Door Removal

Removing the front door of the Expansion Cabinet allows improved access during installation. To remove the front door, open the door as wide as possible and lift it vertically. To reinstall, align the door hinges with the holes in the upper and lower hinge brackets in the Expansion Cabinet chassis. See Figure 4-5 for reference.

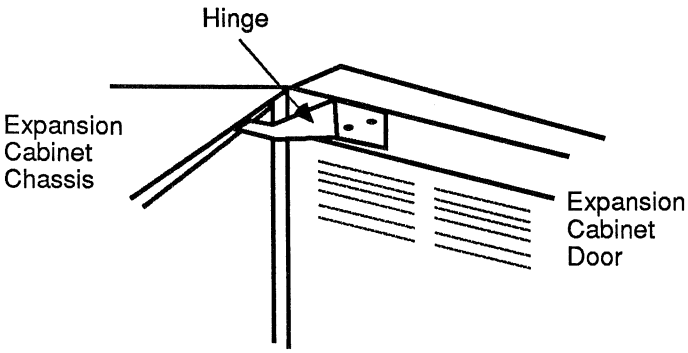


Figure 4-5. Expansion Cabinet Front Door Removal

Anti-Tip Protection

The Expansion Cabinet has an anti-tip bar that should be extended when you install peripherals in the upper portion of the Cabinet and when you pull out the 1/2-inch tape drive to gain access. To use the anti-tip bar

WARNING
Never pull more than one peripheral drawer out at the same time. The anti-tip bar can support the weight of only one loaded peripheral drawer.

(1) Open the front door of the Expansion Cabinet.

(2) Locate the anti-tip bar at the bottom of the Expansion Cabinet.

(3) Pull until the bar is fully extended.

(4) Rotate the anti-tip bar's support column 90 degrees so that the circular base is flush with the floor.

(5) Tighten the locking nut on the end of the bar so the circular base does not slip.

See Figure 4-6 for reference.

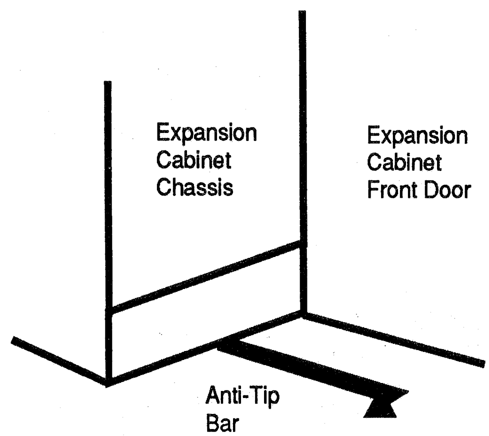


Figure 4-6. Anti-Tip Bar

AC Power Controller

The AC Power Controller accepts power from an AC wall outlet and provides power in turn to each of the peripherals in the Expansion Cabinet. The AC Power Controller is pre-installed at the bottom of each Expansion Cabinet. It is 19-inch rack mounted and requires 2 EIA units of vertical space.

Figure 4-7 shows the back of the Power Controller. The six identical AC power receptacles on the left side of the controller accept power cords from each of the peripheral modules. To connect the power controller's AC line cord, plug the cord in, then secure it with its strain relief bracket. The strain relief bracket is located just above the line cord receptacle, as shown in Figure 4-7. Loosen the two screws on the bracket, then rotate the bracket 180 degrees to hold the cord in place. Finally, route the AC line cord to the correct wall outlet. The AC Power Controller uses the same line cord as the Stardent 1500/3000 System Module. See the *Site Preparation Guide* for line cord specifications.

NOTE
For the 120V/20A option a NEMA L5-30 plug is used (nominal rating 115V/30A). For the 240V/16A option a NEMA L6-20 plug is used (nominal rating 230V/20A).

Each AC Power Controller can accept 120 Volt/20 Amp or 240 Volt/16 Amp power. Your choice of line cord determines the power option for the Expansion Cabinet. All peripherals in the Cabinet must use the same power option; the AC controller does no voltage conversion.

The AC Power Controller has a power-on indicator light panel and a circuit breaker switch on its back panel.

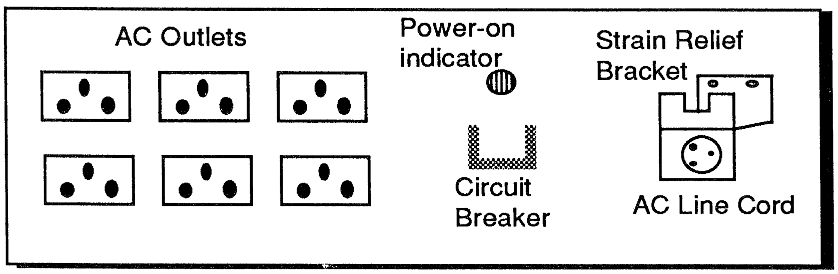


Figure 4-7. AC Power Controller Back Panel

INSTALLING THE QUAD DISK DRAWER

CHAPTER FIVE

The 5 1/4-inch Quad disk drawer accommodates up to four SCSI disk drives (380 MB or 760 MB) with power supplies and connectors. Figure 5-1 shows the front panel of the drive.

WARNING
Never pull out more than one peripheral drawer at a time.

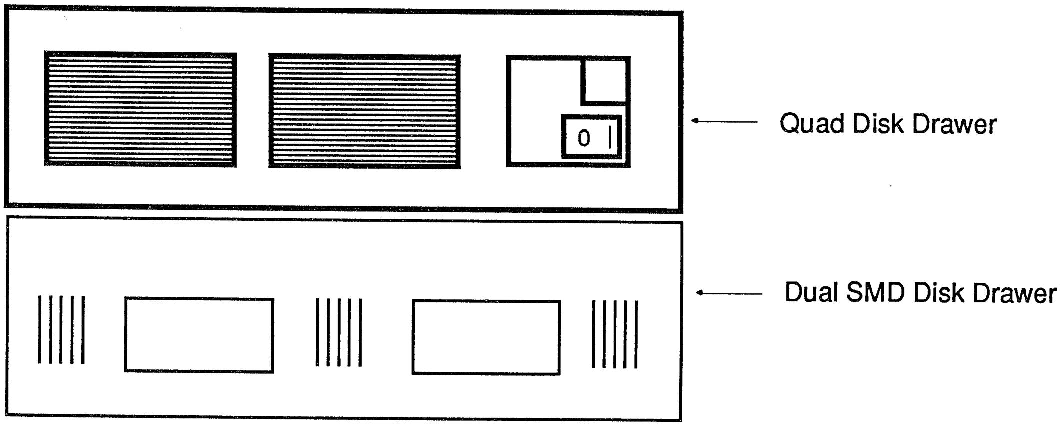


Figure 5-1. Quad Disk Drawer and Dual SMD Disk Drawer

Quad Drawer Installation

Follow this procedure to install the drive drawer in the Expansion Cabinet.

NOTE
Consult the *Field Service Manual* if you need to install mounting rails.

Step 1: Note.

Two people are needed for this procedure. Make sure the Expansion Cabinet power is off and that the cabinet is unplugged at the AC power controller.

Step 2.

Unpack the drive drawer. To unpack the drawer, open the shipping box, remove the four styrofoam corner protectors and carefully lift the drawer out. **The drawer weighs about 80 pounds when fully configured.**

Step 3.
Open the front door of the Stardent 1500/3000 Expansion Cabinet. Remove the front door, if you wish, to improve access. Extend the anti-tip bar. (See Chapter 4 for instructions.)

Step 4.
Slide the appropriate rack mounting rails until they are fully extended and lock into place. The rails, as shown in Figure 5-2, should already be installed in the Expansion Cabinet chassis. If you need to install mounting rails yourself, consult the *Field Service Manual.*

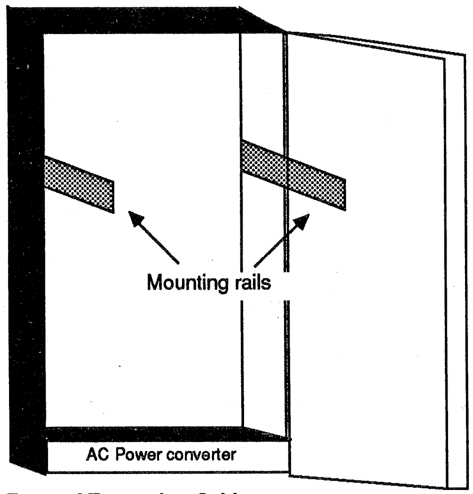


Figure 5-2. Quad Drawer Mounting Rails

Step 5.
With two people holding the drawer, mount the drawer on the mounting rails.

Step 6.
To slide the drawer back into the Expansion Cabinet, release the locking latches on the mounting rails and push the drawer in until it is fully retracted.

Step 7: Check Interlock Mechanism.
Check the back of the expansion cabinet to see if it has a drawer interlock cable. The cable prevents more than one drawer from being pulled out at once. If your Expansion Cabinet has an interlock cable, attach it to the peripheral drawer.

Step 8: Cabling.

The drives in the Quad Drawer are all daisy-chained together and thus require only one SCSI signal cable. Plug the signal cable into the back of the drive drawer as shown in Figure 5-3, then route the cable down the right side of the Expansion Cabinet and over to the System Module. Plug the AC line cord into the back of the drawer as shown in the figure and route the cable down the left side of the Expansion Cabinet to the AC power controller. The Quad Drawer also requires a SCSI terminator, as shown in the figure.

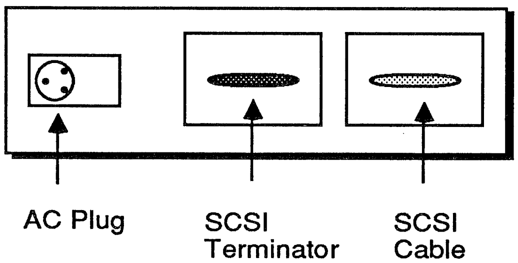


Figure 5-3. Back of Quad Disk Drawer

Step 9: Configuration.

The disk drives should already be configured for use. The disk device number and associated UNIX device references are listed in the chapter entitled *System Overview*. If you wish to add a device to the drawer, need to recode device numbers or need to format disks, consult the *Field Service Manual*. Note that each Stardent 1500/3000 SCSI controller (0 or 1) can be connected to a maximum of seven SCSI devices, including internal devices. Use a maximum of one Quad drawer per SCSI controller.

INSTALLING THE HALF-INCH TAPE DRAWER

CHAPTER SIX

Follow this procedure to install the HP88780A 1/2-inch reel-to-reel tape drive with SCSI interface.

WARNING
Never pull out more than one peripheral drawer at a time.

NOTE
Consult the *Field Service Manual* if you need to install mounting rails.

Step 1: Note.
Two people are needed to install the tape drive in the Stardent 1500/3000 Expansion Cabinet. **The drive weighs about 120 pounds.** Make sure the Expansion Cabinet is unplugged at the AC Power Controller.

Step 2.
Unpack the tape drive according to the manufacturers instructions.

Step 3.
Consult the HP manual shipped with the tape drive for notices, warnings, and general operating procedures.

Step 4.
Open the front door of the Expansion Cabinet. Remove the front door to improve access. (See Chapter 4 for instructions.)

Step 5.
Extend and place the anti-tip foot according to the instructions given in Chapter 4.

Step 6.
Slide the uppermost rack mounting bars until they are fully extended and click into place.

Step 7.
With two people holding the drive, carefully install the drive on the mounting rails, and make sure it engages in the mounting rail before pushing it into the Expansion cabinet. Make sure the tape is upright and the control panel is to the right as you face the Expansion Cabinet. See Figure 6-1.

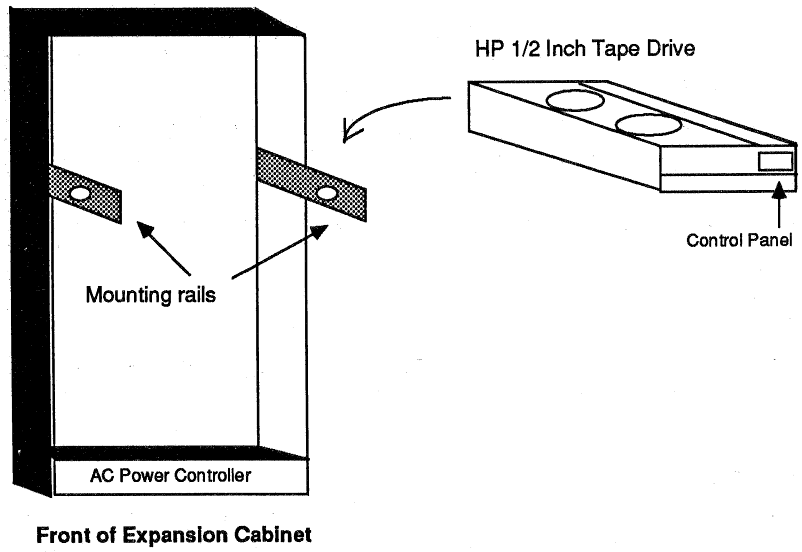


Figure 6-1. Installing the 1/2-Inch Tape Drawer

Step 8.

To slide the tape drive into the Expansion Cabinet, release the locking latches on the side of the mounting rails, and slide a movable clip into place so that the locking latch stays released. Push the drive drawer into the Expansion Cabinet until is is fully retracted.

Step 9: Check Interlock Mechanism.

Check the back of the expansion cabinet to see if it has a drawer interlock cable. The cable prevents more than one drawer from being pulled out at once. If your expansion cabinet has an interlock cable, attach it to the peripheral drawer.

Step 10: Cabling.

The Half-inch tape drive requires a SCSI signal cable, a SCSI terminator and an AC power cord. Each is supplied with the tape drive itself. Connect the SCSI cable to the right-hand SCSI port on the tape drive; connect the terminator to the left-hand SCSI port on the drive. Route the SCSI cable down the right side of the Expansion Cabinet and over to port A or B on the Stardent 1500/3000 System Module I/O board. Remember that SCSI port A corresponds to UNIX controller number 0 and SCSI port B corresponds to controller 1.

Route the tape drive's AC line cord down the left side of the Expansion Cabinet to one of the six AC power receptacles on the AC Power Controller.

Setting the Tape Drive ID

Tape drive ID 7 should be assigned to the drive. Follow this procedure.

Step 1

Make sure the tape drive is powered-up, but off-line.

Step 2.

Locate the control buttons on the front panel of the drive, as shown in Figure 6-2.

Step 3.

Set the Tape ID as follows:

Press OPTION to enter option mode.

Press NEXT until ADDR or ID appears.

Press ENTER.

Press NEXT or PREV until you reach ID number 7.

Press ENTER. The number 7 appears for a minute, then the display reverts to ADDR or ID.

Press OPTION or RESET to leave option mode.

The tape drive is now configured for use. Follow the instructions in the HP 88780A User's Guide to use the tape drive with your system.

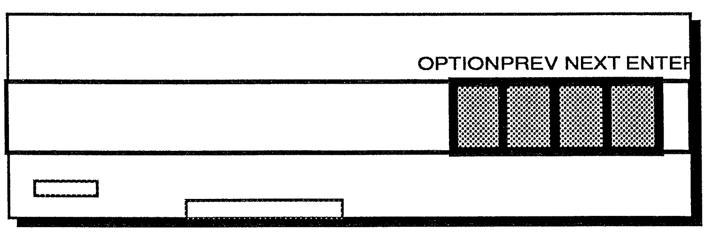


Figure 6-2. HP 88780A Tape Drive Front Panel

INSTALLING THE DUAL SMD DISK DRAWER

CHAPTER SEVEN

The Dual SMD Disk Drawer holds up to two 8-inch SMD disk drives, the Fujitsu M2382K, a fan and a power supply. Each pair of disk drives is controlled by an Interphase V/4200 controller board installed in the VME Card Cage or in the Stardent 1500/3000 System Module's VME Expansion Board. For a description of the VME Card Cage see Chapter 8.

WARNING
Never pull out more than one peripheral drawer at a time.

NOTE
Consult the *Field Service Manual* if you need to install mounting rails.

To install the Dual SMD Disk Drawer in the Stardent 1500/3000 Expansion Cabinet, follow this procedure.

Step 1: Note.
Two people are needed for this procedure. A Phillips screwdriver is required. Make sure the Expansion Cabinet is unplugged at the AC Power Controller.

Step 2.
Unpack the drive drawer. To unpack the drawer remove the top cover of the shipping box, remove the cardboard tube that surrounds the drive drawer, remove the top styrofoam packing material, and lift the drawer out. **The drawer weighs about 100 pounds when fully configured.** Note that the control and data cables for the disk drives are already attached to the drive drawer.

Step 3.
Open the front door of the Expansion Cabinet. Remove the front door, if you wish, to improve access. Extend the anti-tip bar. (See Chapter 4 for instructions.)

Step 4.
With two people holding the drawer, mount the drawer on the mounting rails located in the Expansion Cabinet chassis. (See Figure 7-1.)

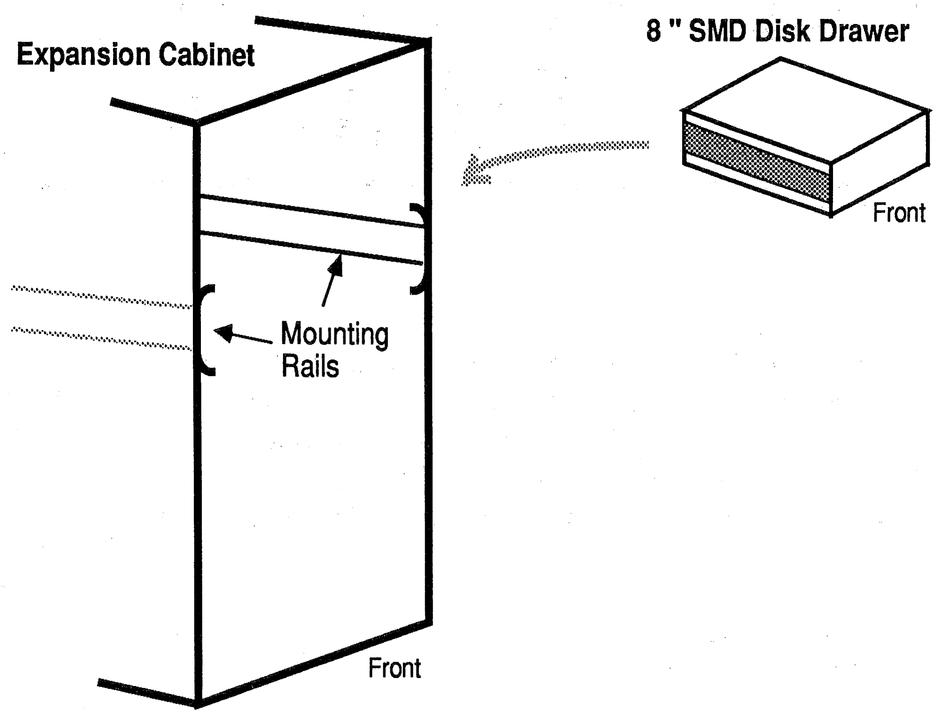


Figure 7-1. Installing the Dual SMD Disk Drawer

Step 5.

As you slide the drawer into the Expansion Cabinet, make sure that the control and data cables for the disk drives are fed out the back of the Expansion Cabinet.

Step 6: Configure the disk drives.

Before securing the drive drawer in the Expansion Cabinet, check the configuration of the disk drive DIP switches as shown in Figure 7-2 and described in Table 7-1. You do not need to remove the metal cover of the disk drive to check the DIP switches. Note that as you face the drive drawer from the front of the Expansion Cabinet, Drive 0 is on your right and drive 1 is on your left.

(The disk drives also has terminators that should already have been corrected configured at the factory. If you need to check the disk drive terminators, see the *Field Service Manual.*)

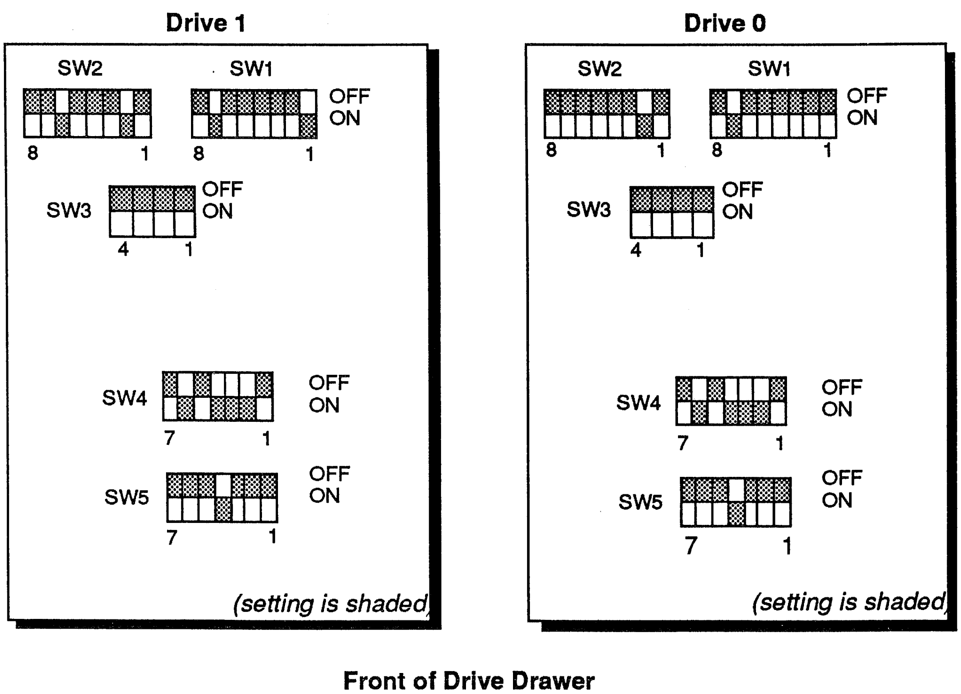


Figure 7-2. SMD Disk Drive DIP Switches

Table 7-1. SMD Disk Drive DIP Switch Settings

Switch Block	Drive 1	Drive 0
SW1:	1 and 7 ON, others OFF	7 ON, others OFF
SW2:	2 and 6 ON, others OFF	2 ON, others OFF
SW3:	1-4 OFF	1-4 OFF
SW4:	2,3,4,6 ON, others OFF	2,3,4,6 ON, others OFF
SW5:	4 ON, others OFF	4 ON, others OFF

Step 7.

Push the drive drawer in until it is fully retracted. Attach the two Phillips screws that secure the drive drawer to the Expansion Cabinet chassis and the two screws that secure the front panel of the drive drawer. Consult Figure 7-3 for reference.

Step 8: Check Interlock Mechanism.

Check the back of the expansion cabinet to see if it has a drawer interlock cable. The cable prevents more than one drawer from being pulled out at once. If your expansion cabinet has an interlock cable, attach it to the peripheral drawer.

Step 8: Cabling.

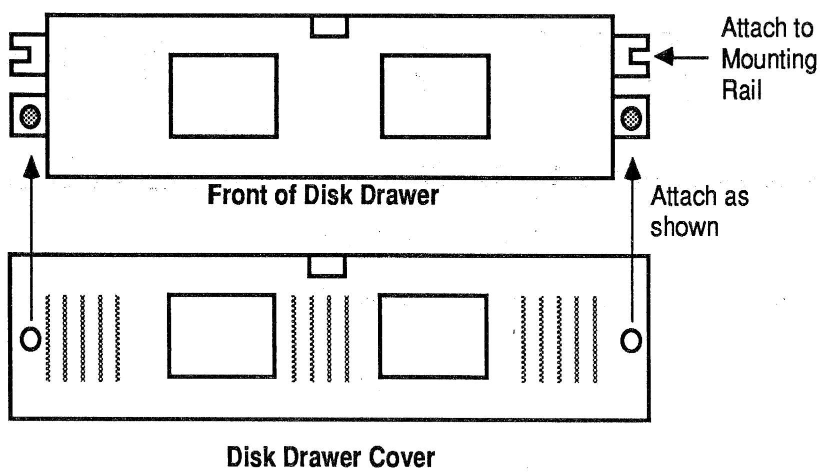


Figure 7-3. Front of 8-Inch SMD Disk Drawer

Route the 60-position signal cable and the 26-position data cable(s) down the right side of the Expansion Cabinet to the VME card cage or directly over to the System Module's VME Adapter Board (if you have no VME card cage in your Expansion Cabinet). See Chapter 8 for cabling instructions for the VME card cage configuration; consult the *Field Service Manual* if your configuration does not include a VME card cage.

To add or replace a drive in the drawer consult the *Field Service Manual.* For information on formatting disks see the chapter entitled *Disks* in this guide.

INSTALLING THE VME CARD CAGE

CHAPTER EIGHT

The Expansion Cabinet's VME card cage allows your Stardent 1500/3000 system to support more than four SMD disk drives. (Up to four SMD disk drives can be supported directly via controllers in the Stardent 1500/3000 System Module's VME Expansion Board, Slot 0 of the System Module card cage.)

WARNING
Always power on the VME card cage before powering on the Stardent 1500/3000 system. If the power on order is reversed the system may not boot or operate correctly.

When you order a VME card cage as part of your Expansion Cabinet configuration, the Expansion Cabinet is shipped to you with the following VME equipment already installed:

- A VME card cage with 12 board slots, numbered 1-12 from the left as you face the back of the Expansion Cabinet. The first slot on the left (Slot 1) is reserved for a VME repeater board. The Interphase V/4200 Controller boards you ordered should be installed in slots 2-5. (Slots 6-12 are currently unused.)

- A secondary HVE repeater board which is cabled to a primary HVE repeater in the System Module VME Adaptor Board (each HVE repeater is labeled primary or expansion (secondary)). The board is shipped installed in the VME card cage and with cables attached.

- 1-4 Interphase V/4200 controller boards installed in the VME card cage.

To use the VME card cage you must

(1) Confirm your Interphase V/4200 Controller configuration.

(2) Connect cables from the SMD disk drawer(s) to the card cage.

(3) Connect cables from the HVE repeater in the card cage to the corresponding repeater in the Stardent 1500/3000 System Module VME Adaptor Board (slot 1).

(4) Connect the VME Card Cage's AC line cord to a corresponding receptacle in the Expansion Cabinet's AC Power Controller.

The next sections show how to do each task.

Confirm Interphase V/4200 Controller Configuration

The Interphase controllers installed in the VME card cage should already be configured for use. For UNIX references, the controller in Slot 2 of the card cage is Controller 2, the controller in Slot 3 is Controller 3, and so on. Figure 8-1 shows the back of the VME card cage with slots identified, and Figure 8-2 gives Interphase Controller addresses, UNIX controller numbers, and format program controller numbers.

If you need to install Interphase controllers, or are changing your configuration, consult the *Field Service Manual.*

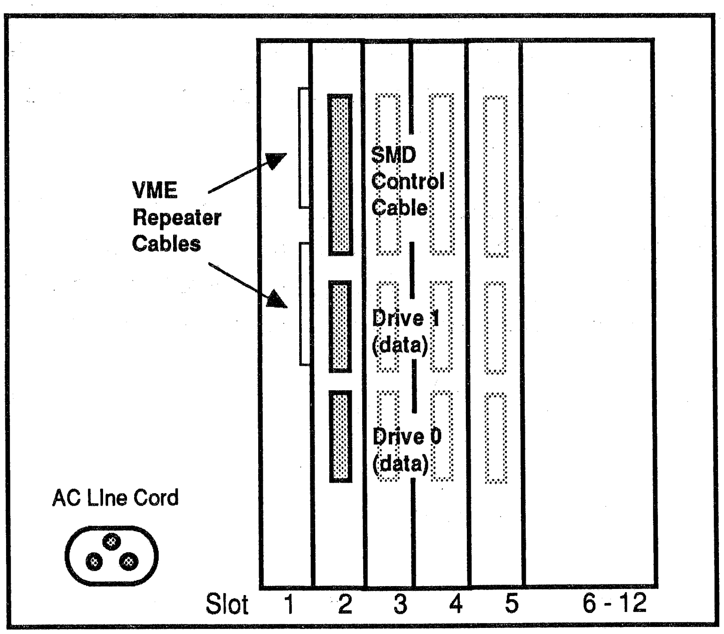
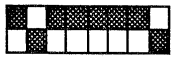
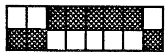
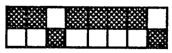
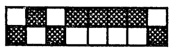


Figure 8-1. Back of VME Card Cage

VME Card Cage Slot	Inpterphase Controller Switch Settings	VME Address	UNIX Name	Format Controller
2	on / off 	0x400	c2dXsX	0
3	on / off 	0x600	c3dXsX	1
4	on / off 	0x800	c4dXsX	2
5	on / off  1 … 8	0xa00	c5dXsX	3

(setting is shaded)

Figure 8-2. Interphase Controller Switch Settings and Controller Numbers

Connect Disk Drawer to VME Card Cage

Chapter 7 describes how to install the 8-inch SMD Disk Drawer. Follow these steps to route cables to the VME card cage once the disk drawer has been installed.

Step 1.

Route the 60-pin control cable to the matching 60-pin port on the correct VME card cage board (see Figure 8-1). Make sure the cable orientation is downward.

Step 2.

As you face the back of the Expansion Cabinet, the SMD disk drive on your left is Drive 0. Route the 26-pin data cable from Drive 0 to the LOWER 26-pin port on the correct VME card cage board. Route the 26-pin data cable from Drive 1 (the right hand drive as you face the back of the Expansion Cabinet) to the UPPER 26-pin port on the VME card cage board. See Figure 8-1 for reference. Make sure the cable orientation is downward.

Connect VME Card Cage to System Module

Follow these steps to cable the VME card cage to the VME Adaptor Board in the System Module.

Step 1: Note.
Make sure the System Module is powered-down and unplugged. A Phillips screwdriver is required for this procedure.

Step 2.
If you are using an already-installed VME Adaptor Board, remove the board from the System Module card cage according to the instructions in the *Field Service Manual*. A new front panel is required to install the HVE repeater clamps and cables properly. If you plan to use a new VME Adaptor Board, unpack the board carefully. If the HVE repeater is not yet installed in the VME Adaptor Board, unpack the HVErepeater.

NOTE
If you encounter any problems using the VME Adaptor Board, check to see that the board's DIP switches are all in the OFF position.

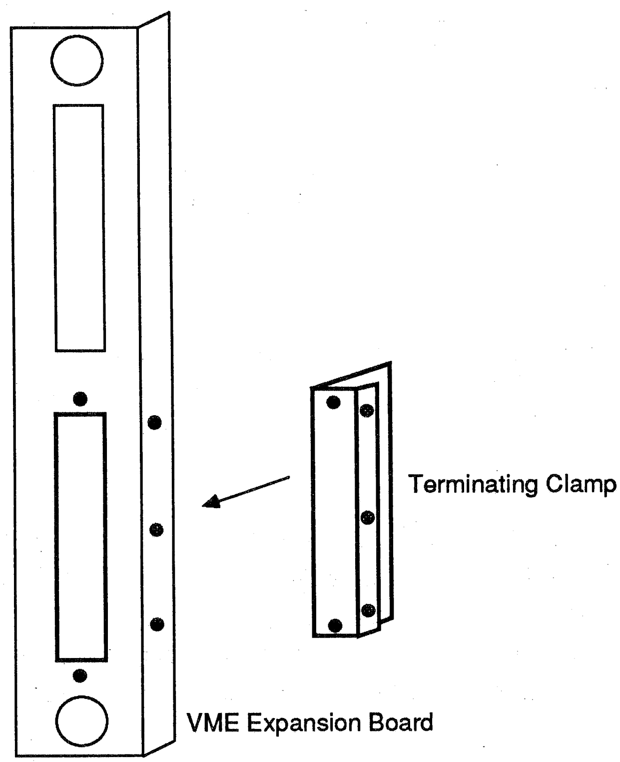


Figure 8-3. Metal Terminating Clamps, VME Expansion Board

Step 3.
Locate the metal terminating clamp on the outer edge of the VME Adaptor Board (see Figure 8-3). Remove the clamp. As shown in the figure, there are five Phillips screws; two on the edge and three toward the face of the board. If your VME Expansion Board does not have a clamp (it may have a metal slot guard instead), a clamp should have been shipped to you under separate cover. If so, remove the slot guard and locate the clamp.

Step 4.
If the HVE repeater board is not yet installed, install it in the lower bay of the VME Adaptor Board, component sides up on each board. As shown in Figure 8-4, the 50 and 60-pin ports should face toward the port edge of the board. Secure the board with the four black lever at the corners of the bay opening.

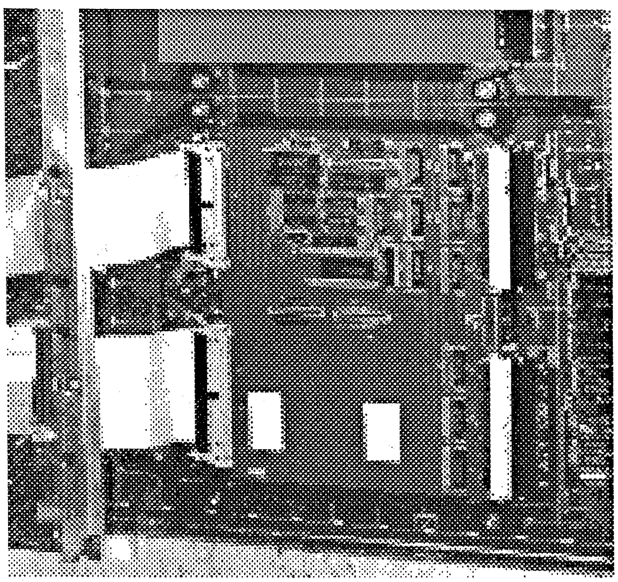


Figure 8-4. VME Expansion Board with Interphase Controller

Connect VME Card Cage to System Module (continued)

Step 5.
Feed the 50 and 60-pin connectors from the Expansion Cabinet's VME card cage (slot 1) through the opening on the edge of the VME Adaptor Board where you just removed the metal clamp.

Step 6.
Attach the 60-pin cable to the matching 60-pin port on the HVE repeater board in the VME Adaptor Board. Attach the 50-pin cable to the 50-pin port. The cables are keyed visually, and pin 1 (the red-coded end of the cable) faces downward when the VME Adaptor Board is installed in the System Module card cage.

Step 7.
Reinsert the clamp you removed in Step 3. Use the clamp to terminate the copper shielding on the two cables. Secure the clamp with the five Phillips screws mentioned in Step 3.

Step 8.
Reinstall the VME Adaptor Board according to the instructions in the *Field Service Manual.*

Connect AC Line Cord

Attach the VME Card Cage AC line cord to the back of the card cage and to one of the AC receptacles on the Expansion Cabinet's AC Power Controller.

The VME Card Cage is now fully installed as cabled. If all other peripherals are installed you can proceed to power-up the system. For booting instructions see *Booting the Shutting Down the System* in this guide and for configuration instructions see the section entitled *Configuration.*

INSTALLING THE TWO- GIGABYTE TAPE DRIVE

CHAPTER NINE

Follow this procedure to install the Exabyte EXB-8200 cartridge tape drive with SCSI interface.

Note. Plan to locate the Exabyte tape drive on top of the System Module or Expansion Cabinet or on a nearby table or desktop.

Step 1. Unpack the tape drive.

Step 2: Device ID Number. The tape drive is shipped with its DIP switches set as device number 6. You can confirm the DIP switch settings by looking through the small panel on the rear of the drive (See Figure 9-1). Plan to connect the tape drive to SCSI controller 1 on the System Module I/O Board and use device number 6. If you wish to use another device ID, consult the *Field Service Manual* for details on how to configure the device number. The UNIX special device files created for the drive are:

NOTE
Because the device number (6) has been set at the factory you do not need to remove the small panel and change the DIP switches. To remove the panel requires a star-shaped torque screwdriver.

/dev/rmt/c1d6h (high density, rewind on close)

or

/dev/rmt/c1d6hn (high density, no rewind on close)

Step 3: Stardent 1500/3000 Power Off. The Stardent 1500/3000 System Module **must** be turned completely off before cabling the tape drive to the Stardent 1500/3000. See the *Installation and Administration Guide* for details on how to shut down the system.

WARNING
Never turn Stardent 1500/3000's power off until shutdown to state 0 (the PROM) is complete. Turning the power off prematurely risks file system corruption.

Step 4: SCSI Cable. Attach a SCSI cable to the upper SCSI port on the rear of the tape drive (See Figure 9-1). Connect the other end of the cable directly to SCSI Controller 1 (Port B on the Stardent 1500/3000 System Module I/O Board) or as the last device in a daisy chain (i.e., with a Half-Inch Tape Drawer or Quad Disk Drawer). To add another device to the tape drive in a daisy chain, remove the external terminator on the lower SCSI port and connect a SCSI cable to the desired device. Keep in mind that a SCSI controller can handle a maximum of 7 SCSI devices.

Step 5: Power Cord. Connect the Exabyte AC power cord to the receptacle on the rear of the drive and to an appropriate AC power source.

Step 6: Stardent 1500/3000 Power On. The Stardent 1500/3000 must be powered on for the Exabyte to perform its auto-test. Bring the system up to PROM mode, but **do not boot the UNIX system.** Booting the system before the tape drive performs its auto-test prevents the system from recognizing the Exabyte's presence.

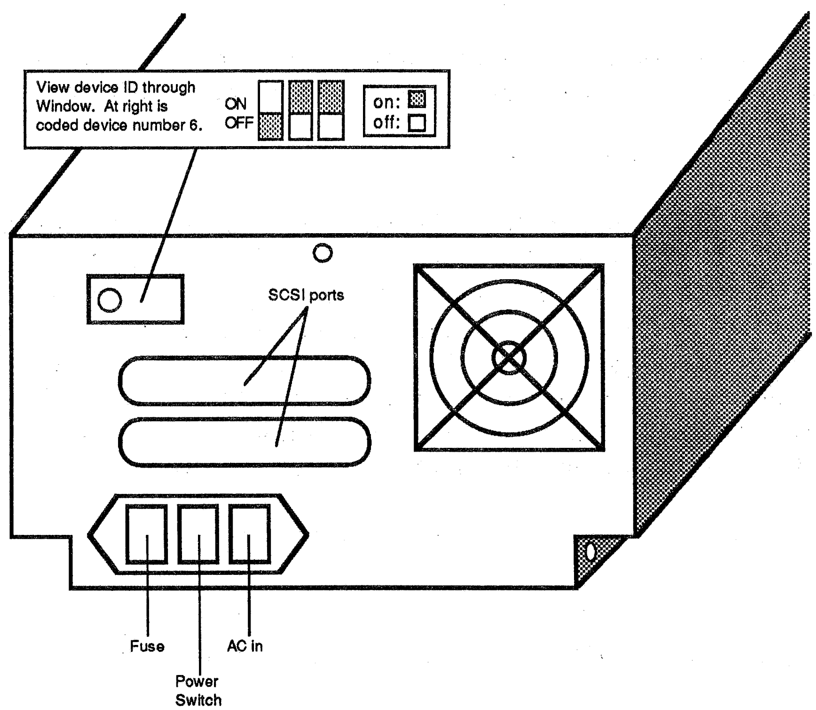


Figure 9-1. Exabyte Tape Drive Rear Panel

Step 7: Exabyte Power On. Turn on the tape drive using the power switch on the rear of the drive. The drive performs its auto-test, which lasts 1-2 minutes. At the completion of the auto-test, both LED's turn off.

Step 8: Insert Tape Cartridge. Press the door open button. The door will open in 10 to 15 seconds. Insert a tape and close the door. After a few seconds the green LED will appear, indicating that the tape drive is ready.

Step 9: Boot the System. See the chapter entitled *Booting and Shutting Down the System* for instructions on how to boot the UNIX system.

Operating Notes

Here are several important notes about operating the Exabyte tape drive.

- If the door-open button is pressed while the amber LED is off, the Exabyte drive will ignore the request. Pressing the door-open button will **not** stop the tape during a write sequence.
- When you push the door-open button with a tape installed it takes about 15 seconds for the door to open.
- When you push the door-open button and the tape is not at the beginning the tape will rewind before the door opens. Rewinding may take a long time.

Other items of interest:

- The drive must be cleaned after every 30 hours of use, or problems will occur while reading or writing to tapes. Obtain a cleaning kit from the Exabyte Corporation (Exabyte Part Number 180123).
- Tape cartridges may be purchased from the Exabyte Corporation. Several sizes are available, and large quantities reduce the price significantly. An 8mm tape cartridge with 2048 MB capacity is Exabyte Part Number 180104.

BOOTING AND SHUTTING DOWN THE SYSTEM

CHAPTER TEN

Stardent 1500/3000 is installed initially by an Stardent field service representative who makes sure that the options you ordered are connected properly and working and that the system software is operational. When installation is complete Stardent 1500/3000 is in automatic boot mode.

This chapter shows you how to boot Stardent 1500/3000 automatically and manually. It also describes how to set the system date and time, change system run levels, reset the system, and shut the system down.

Booting the System Automatically

If your system has been set to automatic boot mode you can use the following procedure to boot Unix from a power-off condition. If you follow the procedure given here and Unix does not boot automatically, follow the instructions in the next section to boot the system.

Step 1. Make sure your Stardent 1500/3000 is completely connected and plugged in. If this has not yet been done, see the other chapters in the *Installation* section of this guide or your Stardent representative.

Step 2. A plastic cartridge has been placed in the cartridge tape drive to protect the heads during shipment. Remove the cartridge by pushing it into the drive. The drive's spring mechanism releases the cartridge so you can pull it out.

Step 3. Place the power switch on the lower right rear of the System Module in the "on" position.

Step 4. Place the lower key switch on the front of the System Module in the "normal" position. Insert the key into the upper key switch.

Step 5. Power up the monitor or system console terminal.

Step 6. Power up the System Module by turning the upper key switch to the "on" position.

Step 7. The system should boot automatically. Within a few minutes you should either see a *login:* prompt appear on the monitor or system console terminal screen, or the # prompt that appears in Unix single user mode.

If the single user mode prompt appears, it means that the *initdefault* entry in your system's */etc/inittab* file has been set to 1 (for single user mode). If so, you can enter multi-user mode simply by typing

```
# init 2
```

(See *System Processes* in this guide for more information on the */etc/inittab* file.

To boot the system from a power-on condition (upper key switch in the "on" position), make sure the lower key switch is in the "normal" position and turn the upper key switch to the "reset" position. The spring-loaded switch slips back to the "on" position when you let go.

If the root file system is in an inconsistent state, *fsck* repairs it and then causes an automatic reboot of the system.

Booting the System Manually

You may need to boot Stardent 1500/3000 manually for any of the following reasons:

- The system has been set to manual boot mode.
- A problem prevents the automatic boot procedure from working correctly. In that case you may need to perform diagnostic tests from the PROM, and then reboot the system manually.
- The Stardent 1500/3000 operating system needs to be loaded from tape containing a software update.
- You want to bring the Stardent 1500/3000 system up in single-user mode to perform special administrative functions.

You boot the system manually from the Stardent 1500/3000 PROM. The PROM prompt appears when any of the following is true:

- You power on the machine with the reset control key switch in the "diagnostic" position.
- You interrupt the automatic boot procedure by pressing <CTRL> C.
- The automatic boot procedure aborts, due to certain system failures.
- The NVRAM variable *bootmode* is set to *m*, for manual boot.

To boot the system manually use the following procedure.

Step 1: Check the values of the NVRAM boot variables. The default boot actions taken by the PROM are controlled by the PROM environment variables. These variables are kept in non-volatile memory (NVRAM) and specify the default boot device, boot file name, and so on. Since the variables are stored in NVRAM they persist across Stardent 1500/3000 power shut-downs.

To list the PROM environment variables, type **n** (for nvram) at the PROM prompt.

Table 10-1 gives the default values of the NVRAM boot variables (also called PROM boot environment variables).[1] For the boot procedure you can ignore the other variables listed in response to the **n** command.

If each of the variables is set at its default value, proceed immediately to the next step. If not, set the value with the command

[1] An additional NVRAM boot variable, *bootmode,* does not influence the way in which you boot the UNIX system from the PROM, but does affect what happens when you boot from a power-off condition. If *bootmode* is set to *a* (for automatic), automatic booting proceeds when you power-on the system. If *bootmode* is set to *m* (for manual), the PROM prompt appears when you power-on the system.

Table 10-1. NVRAM Boot Variables - Default Values

bootfile	unix
path	scsi(1,5,0)
rootdev	scsi(1,5,0)
swapdev	scsi(1,5,1)
secondary	scsi(1,5,8)sash
dumpdev	scsi(1,5,1)

`prom 1>` **n** *variablename=value*

For example, to set bootfile to the value unix, type

`prom 0>` **`n bootfile=unix`**

For more on setting NVRAM boot variables, see the chapter entitled *NVRAM Variables* and *Boot* in the *PROM Manual* section of this guide.

Step 2: Check the setting of the lower key switch on the Stardent 1500/3000 System Module front panel. To boot to the multi-user level place the switch in the "normal" position; to boot to the single-user level place the switch in the "diagnostic" position.

Step 3: Issue the appropriate boot command. If the environment variables have been set correctly, you can boot the system with the command

`prom 1>` **`b`**

(To boot to single user mode you can issue the command

`prom 1>` **`b -s`**

This overrides the setting of the lower key switch and orders a boot to single user mode.)

Step 4: Check the Date and Time. Once the system is booted, check the date and time setting by issuing the command

`#` **`date`**

If you need to set the date or time see *Date* below in this chapter.

Booting the System from the DCP Port

Follow these instructions to boot the UNIX system from a terminal connected to the Diagnostics Communication Port (DCP).

If the DCP port terminal is **not** to be the system console, follow the manual boot procedure given in this chapter.

If you want the DCP port terminal to be the console, you must do two things:

(1) Set the value of the NVRAM variable console to *dcp:*

```
prom 1> n console=dcp
```

(2) Issue a boot command. If the terminal baud rate is 9600 baud, you can issue the simple boot command

```
prom 1> b
```

If the terminal baud rate is *not* 9600, you must issue a full boot command, specifying the terminal baud rate as well as the boot instructions:

```
prom 1> boot scsi(1,5,0)unix -b{baudrate}
```

Please see *Boot* in the *PROM Manual* section of this guide for more on *boot* command syntax.

Setting the Date and Time

Having Stardent 1500/3000's date and time set correctly is essential for accurate record keeping and tracking. Follow these steps to set the date and time.

1. Check the date. When you boot Stardent 1500/3000 for the first time you should check that the date and time are accurate. You can do that by issuing the *date* command:

```
# date
Mon Jan 7 02:40:32 PST 1987
#
```

The date and time should be correct. If they are not, change them as follows:

2. Set the time zone. The environment variable *TZ* determines the time zone for your Stardent 1500/3000. You should receive Stardent 1500/3000 with a correct time zone setting; but if you do not, or if the machine is later moved to a different zone, you need to set the time zone.

If you have Bourne shell users on your system, set the time zone for them in the system profile, */etc/profile*. Just change the lines

```
TZ=value
export TZ
```

in the file. The rules for assigning values for TZ are in the *sh*(1) section of the *Commands Reference Manual*. For instance,

```
TZ=PST8PDT
```

refers to the Pacific Time Zone and Daylight Savings Time.

The system profile is not executed for C-shell users on the system, so setting the time zone is a little different. Users must set the time zone variable in individual *.login* files. They should add the line

```
setenv TZ value
```

to the file. *value* can be assigned using the same rules as for the shell environment variable TZ. Alternatively, the value can be taken from the filenames in the directory */etc/zoneinfo*.

3. Set the date. If you are logged on in single-user mode, use the *date* command to set the date and time. (In multi-user mode, as super-user, change the date and time using the *datetime* command. See the *Commands Reference Manual* for details.) Use the *date* command as follows:

```
# date mmddhhmmyy
```

The first *mm* is the month number (01 to 12). *dd* is the day of the month. *hh* is the hour, on the 24 hour clock. The second *mm* is the minute number. The optional *yy* is the last two digits of the year number.

For instance, the command line

```
# date 0107024087
```

yields the following result:

```
# date 0107024087
# date
Wed Jan 7 02:40:12 PST 1987
#
```

Changing Run Levels

If your system is in single user mode (run level 1, s or S), you can change to multiuser mode (run level 2) by typing

```
# init 2
```

From multiuser mode you can change to single use mode by becoming super-user and typing

```
# init 1
```

For more on changing run levels and on doing a full shutdown of the system, see *Shutting Down the System* below in this chapter.

Resetting the System

This section describes methods of resetting Stardent 1500/3000. Figure 10-1 summarizes the information.

Power-off Reset

Moving the upper front panel key switch to the "off" position turns Stardent 1500/3000's power off. Before turning the power off, make sure the system has been shut down properly. (See *Shutting Down the System* below in this chapter.

To restart the system after power-off, place the lower key switch in the desired position as described below. Turn the upper key switch to the "on" position.

Hard Reset

When you do a hard reset of the system, the system is immediately halted and control is returned to the PROM. A hard reset may corrupt the file system or cause you to lose recently created files, so be sure that the system has been shut down properly or that you have no alternative before doing a hard reset.

To do a hard reset, place the lower front panel key switch in the "normal" position and turn the upper key switch to the spring-loaded "reset" position.

Once control has been returned to the PROM, the UNIX system reboots automatically if the current value of the NVRAM variable *bootmode* is *a* (automatic). Otherwise the system remains in PROM mode and a manual boot is required to return you to the UNIX system.

If *bootmode* has been set to *a* but you wish to remain under PROM control upon hard reset, turn the lower key switch back to the "diagnostic" position within two seconds of doing the reset.

Soft Reset

A soft reset is generally used for debugging purposes. When you order a soft reset of the system, the system interrupts the program that is running and returns control to the PROM. To return to the program after a soft reset you type the PROM command **g** or **go**.

Issuing a soft reset does not cause damage to the system or the program running, providing the **go** command is used to restart after debugging is complete.

To do a soft reset, place the lower front panel key switch in the "diagnostic" position and and turn the upper key switch to the "reset" position. Alternatively, hold down the <CTRL> and <ALT> keys on the key board and type <DEL>.

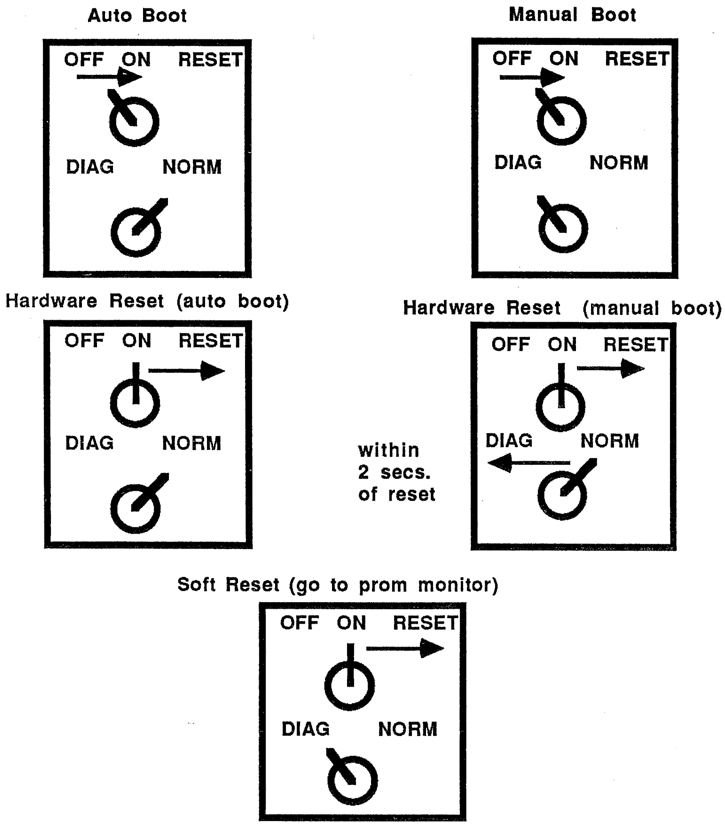


Figure 10-1. Resetting the System: Keyswitch Operations

Shutting Down the System

The term *shutdown* means to change the state, or run level, of the system. You must be super-user to shut the system down.

The simplest way to shut the system down safely and go to the PROM is to type

```
# sync
# halt
```

If you wish to use the PROM for OS debugging use a soft reset to enter the PROM. (See above, *Resetting the System*.)

The shutdown Command

shutdown(1M) is the general command to change the state, or run

WARNING
Never turn Stardent 1500/3000's power off until shutdown to state 0 (the PROM) is complete. Turning the power off prematurely risks file system corruption.

level of the system. You must be super-user to use the command. Here is the syntax of the command:

shutdown [-y] [-g*graceperiod*] [-i*initstate*]

where *graceperiod* is in seconds and *initstate* is the new state to be entered.

The *shutdown* command executes the file *etc/shutdown,* a shell script containing shutdown instructions and messages. Within the */etc/shutdown* script, the init process is used to perform the level change, and any appropriate */etc/rcN* files are executed as the system enters the new state.

Stardent 1500/3000 is shipped with a default */etc/shutdown* script. You can modify the script if you wish, but that shouldn't be necessary for most applications.

Complete Shutdown

To prevent corruption of the root file system, it is important that Stardent 1500/3000's power **NEVER** be shut off until the shutdown process is complete.

To do a complete shutdown with the *shutdown* command you use the *initstate* option 0. This changes Stardent 1500/3000's state to zero, the power-down (PROM) state. Here are some example command lines:

shutdown -i0
Shuts the system down to the power-down (PROM) state. The system sends a warning message and a final message, waits 60 seconds, and then sends a message asking for confirmation. When the shutdown is complete a you are returned to the PROM and it is safe to turn the power off.

shutdown -g90 -i0
Same as the first command listed, except that the system waits 90 seconds between the final message and the confirmation request.

shutdown -y -i0
Same as the first command, except that the *-y* option pre-answers the confirmation question affirmatively. You do not have to confirm before the system is completely

shutdown. If you issue this command, your screen looks something like this:

```
# shutdown -y -i0
System going down in 60 seconds.  Please log off!

System going down immediately.

The system is down.


prom #>
```

Shutting Down to Single-User Mode

Single-user mode is used to install or remove software utilities, backup or restore files and file systems, and to check file systems. You can bring the system down to single user mode by issuing the command

```
# init 1
```

as super user; or you may issue a *shutdown* command. The *shutdown* command executes the file */etc/shutdown* (which contains the *init* command) and and executes the file */etc/rc1* if it exists.

Here are some example command lines:

shutdown -i1 Brings the system down to the single-user run level, unmounting all file systems except the root file system and killing all processes except those related to the console.

shutdown -is Brings the system down to the single-user run level, but does not unmount file systems and only kills processes spawned by *init*.

shutdown -iS Brings the system down to the single-user run level, but does not kill processes nor unmount file systems.

The *-y* and *-ggraceperiod* options can also be used when shutting the system down to the single-user level. For instance

```
# shutdown -y -i1
```

shuts the system down to single-user run level 1 without asking for confirmation. You see the following message appear,

```
INIT: SINGLE USER MODE
#
```

indicating that you may now proceed with your administrative tasks.

When you are ready to return the system to multi-user state, you can issue the *shutdown* command with run level 2 (multi-user) specified, or shut the system down completely and reboot.

Caution about Shutdowns

If there are multiple users on your system, it is important that they be notified when a shutdown is imminent. You can do this in the following ways.

- If you are scheduling a shutdown for a particular time, notify users by entering a message to that effect in the */etc/motd* (message-of-the-day) file.
- Use the *wall*(1) (write-to-all-users) command to warn users when a shutdown is imminent.

Stardent

CONFIGURATION

Section 2

SYSTEM OVERVIEW

CHAPTER ELEVEN

This chapter is a brief system administrator's look at the Stardent 1500/3000 graphics supercomputer or Stardent 1500/3000 Server. It gives an overview of the hardware components and software environment, lists key files and directories used in system administration, and describes how to record video images from the Stardent 1500/3000 Graphics Supercomputer monitor screen.

Hardware Overview

This section describes what you need to know as a system administrator about Stardent 1500/3000 hardware.

Front Panel

The front panel of the Stardent 1500/3000 System Module is located behind a decorative, corrugated facade that slides up and down. The front panel has two key switches, two run lights, and a cartridge tape drive. These items are described below.

NOTE
Throughout this document the term "System Module" refers to the cabinet that houses the Stardent 1500/3000 processors, memory, etc.

Key Switches. The front panel has two key switches. The upper switch is the power and reset switch; the lower switch is the reset control switch. The spring-loaded power switch has three positions: "off", "on", and "reset". The "off" position is to the left, the "on" position is to the right, and the spring-loaded "reset" position is further to the right (the key snaps back to the "on" position after reset).

The reset control switch has two positions: the "diagnostic" position and the "normal" position. *Booting the System* in this guide describes how to use the key switches to reset the system.

Lights. Behind the Stardent logo on the front panel is a green power light (LED). It monitors the state of the Stardent 1500/3000 power supply and is always on when Stardent 1500/3000 is powered. The amber LED on the front panel is used for diagnostic purposes. It monitors the boot process as follows:

Preboot	LED is on.
Booting UNIX	LED flashes slowly while UNIX is loaded.
Running normally	LED flashes quickly.
Problem	LED is on steadily. Examine the console for error messages and the CPU board LEDs for failure information.

PROM

The Stardent 1500/3000 PROM (programmable read-only memory) is used for initial boot of Stardent 1500/3000, system diagnostics, diagnostic and boot commands, and any function external to the UNIX operating system. When the system is first powered up (in manual boot mode) the PROM outputs a card cage inventory and a prompt similar to the following:

```
Titan Monitor, Version 6.3, Thu May 26, 10:28:31 PDT 1988
Board Inventory:
slot    status          revid          board type
  1     0x012c001a      24             I/O
  2                     3              CPU, cpuid: 1
  3     0x01ff0020      0              Memory, 32 Mbyte
  5                     3              CPU, cpuid: 2
  6     0x01ff0020      0              Memory, 32 Mbyte
  8     0x012c9000      0              Graphics
  9                     0              Graphics Expansion
prom #>
```

At the PROM prompt you can boot the Stardent 1500/3000 operating system, boot hardware diagnostics, boot the format program or perform diagnostic procedures. See the chapters entitled *Booting the System* and *Disk Drives* for details.

I/O Drawers

Two I/O drawers are located behind the front panel. They are accessible from the front of the System Module when its hinged door is open. (Note that the hinged door has a lock along its right edge). The drawer on the left contains the primary and one optional SCSI disk drive. The drawer on the right contains the 1/4-inch cartridge tape drive and an optional SCSI disk drive. (The cartridge tape and disk drives are discussed in the next sections.)

Refer to the *Field Service Manual* for instructions how to remove the I/O drawers.

Devices

Stardent 1500/3000 supports various mass storage devices:

- Internal 5 1/4–inch SCSI disk drives (1-3)
- Internal 1/4–inch SCSI cartridge tape drive
- Quad Disk Drawer with up to 4 SCSI 5 1/4–inch disk drives
- 1/2-inch SCSI reel-to-reel tape drive
- 2-Gigabyte SCSI cartridge tape drive
- 1-Gigabyte SMD disk drives (8–inch)

To identify each device the system requires a controller number and a device number.

The figures and tables on the next several pages show controller and device numbers for all currently supported disk and tape devices. The tables also give full UNIX device names for each device and, where appropriate, the device names to be used from the PROM. Note that the PROM only supports references to the SCSI disk drives and the 1/4-inch SCSI cartridge tape drive. For a full discussion of UNIX device names, see *File System Administration* in this guide.

Table 11-1. Internal SCSI Device Names and Numbers

Device	Location	Ctlr #	Dev. #	UNIX Reference	PROM Ref.
1/4" Cartridge Tape:	Rt. I/O drawer	0	6		
QIC-11 low-density, rewind on close	Rt. I/O drawer	0	6	/dev/rmt0 or /dev/rmt/c0d6l	not supported
QIC-24 high-density, rewind on close	Rt. I/O drawer	0	6	/dev/rmt16 or /dev/rmt/c0d6h	scsi(0,6,*x*)
QIC-11 low-density, no rewind on close	Rt. I/O drawer	0	6	/dev/rmt4 or /dev/rmt/c0d6ln	not supported
QIC-24 high-density, no rewind on close	Rt. I/O drawer	0	6	/dev/rmt20 or /dev/rmt/c0d6hn	not supported
Primary SCSI disk:	Lt. I/O drawer	1	5	/dev/dsk/c1d5s*x*	scsi(1,5,*x*)
1st optional SCSI disk:	Rt. I/O drawer	0	5	/dev/dsk/c0d5s*x*	scsi(0,5,*x*)
2nd optional SCSI disk:	Lt. I/O drawer	1	4	/dev/dsk/c1d4s*x*	scsi(1,4,*x*)

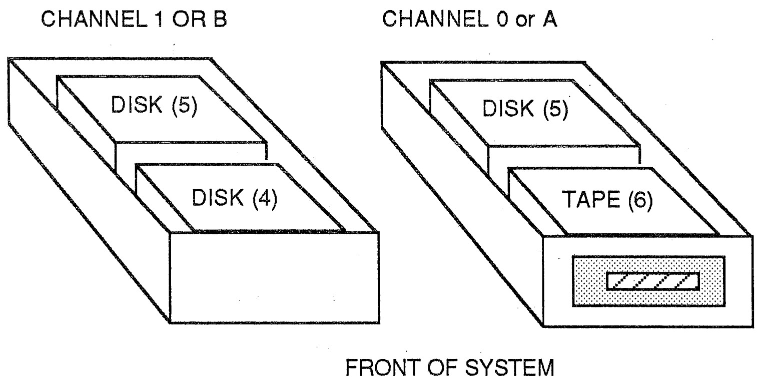


Figure 11-2. Internal SCSI Device Locations

Internal SCSI Device Notes:

(1) The PROM only supports high density, rewind-on-close tape devices.

(2) The italicized *x* in the Table 11-1 UNIX and PROM device references (last two columns of the table) refers to a specific disk partition or tape section.

Table 11-2. SCSI Quad Disk Drawer Device Names and Numbers

Controller, Location	Ctlr #	UNIX Device Name	PROM Ref.
Attached to SCSI A on I/O board in card cage slot 1	c0		
Left front		/dev/dsk/c0d1s*x*	scsi(0,1,*x*)
Left rear		/dev/dsk/c0d2s*x*	scsi(0,2,*x*)
Right rear		/dev/dsk/c0d3s*x*	scsi(0,3,*x*)
Right front		/dev/dsk/c0d7s*x*	scsi(0,7,*x*)
Attached to SCSI B on I/O board in card cage slot 1	c1		
Left front		/dev/dsk/c1d1s*x*	scsi(1,1,*x*)
Left rear		/dev/dsk/c1d2s*x*	scsi(1,2,*x*)
Right rear		/dev/dsk/c1d3s*x*	scsi(1,3,*x*)
Right front		/dev/dsk/c1d7s*x*	scsi(1,7,*x*)
Attached to SCSI A on I/O board in card cage slot 8	k0		
Left front		/dev/dsk/k0d1s*x*	scsi(2,1,*x*)
Left rear		/dev/dsk/k0d2s*x*	scsi(2,2,*x*)
Right rear		/dev/dsk/k0d3s*x*	scsi(2,3,*x*)
Right front		/dev/dsk/k0d7s*x*	scsi(2,7,*x*)
Attached to SCSI B on I/O board in card cage slot 8	k1		
Left front		/dev/dsk/k1d1s*x*	scsi(3,1,*x*)
Left rear		/dev/dsk/k1d2s*x*	scsi(3,2,*x*)
Right rear		/dev/dsk/k1d3s*x*	scsi(3,3,*x*)
Right front		/dev/dsk/k1d7s*x*	scsi(3,7,*x*)

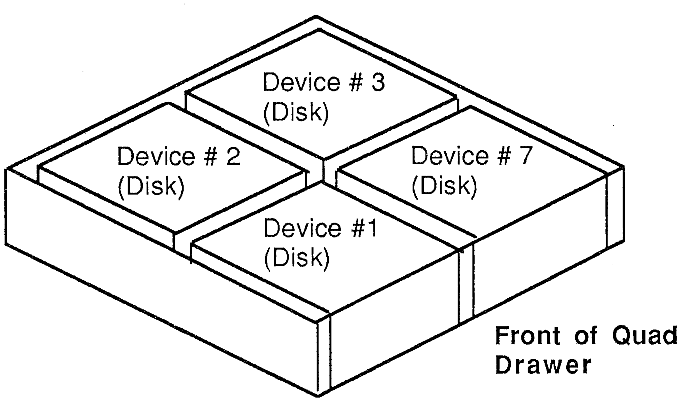


Figure 11-3. SCSI Quad Drawer Device Locations

Quad Disk Drawer Device Notes:

(1) The PROM controller numbers differ from UNIX controller numbers. PROM controller range from 0 to 3 whereas UNIX controller numbers have the values c0, c1, k0 and k1.

(2) The k controller numbers are only used in dual I/P board configurations.

(3) The italized *x* in the Table 11-1 UNIX and PROM device references (last two columns of the table) refers to a specific disk partition or tape section.

Table 11-3. Half-Inch Tape Device Names and Numbers

Controller, Description	Ctlr #	UNIX Device Name
Attached to SCSI A on I/O board in card cage slot 1	c0	
High density, rewind on close		/dev/rmt/c0d7h
High density, no rewind on close		/dev/rmt/c0d7hn
Low density, rewind on close		/dev/rmt/c0d7l
Low density, no rewind on close		/dev/rmt/c0d7ln
Attached to SCSI B on I/O board in card cage slot 1	c1	
High density, rewind on close		/dev/rmt/c0d7h
High density, no rewind on close		/dev/rmt/c0d7hn
Low density, rewind on close		/dev/rmt/c0d7l
Low density, no rewind on close		/dev/rmt/c0d7ln
Attached to SCSI A on I/O board in card cage slot 8	k0	
High density, rewind on close		/dev/rmt/k0d7h
High density, no rewind on close		/dev/rmt/k0d7hn
Low density, rewind on close		/dev/rmt/k0d7l
Low density, no rewind on close		/dev/rmt/k0d7ln
Attached to SCSI B on I/O board in card cage slot 8	k1	
High density, rewind on close		/dev/rmt/k1d7h
High density, no rewind on close		/dev/rmt/k1d7hn
Low density, rewind on close		/dev/rmt/k1d7l
Low density, no rewind on close		/dev/rmt/k1d7ln

1/2-Inch Tape Device Notes:

(1) The k controller numbers are only used in dual I/P board configurations.

2-Gigabyte Tape Device Notes:

(1) The k controller numbers are only used in dual I/P board configurations.

(2) To avoid confusing device names, do not connect the 2-gigabyte tape drive to controller 0 (port A) on the I/O board in slot 1.

Table 11-4. Two-Gigabyte Tape Device Names and Numbers

Controller, Description	Ctlr #	UNIX Device Name
Attached to SCSI B on I/O board in card cage slot 1	c1	
High density, rewind on close		/dev/rmt/c1d6h
High density, no rewind on close		/dev/rmt/c1d6hn
Attached to SCSI A on I/O board in card cage slot 8	k0	
High density, rewind on close		/dev/rmt/k0d6h
High density, no rewind on close		/dev/rmt/k0d6hn
Attached to SCSI B on I/O board in card cage slot 8	k1	
High density, rewind on close		/dev/rmt/k1d6h
High density, no rewind on close		/dev/rmt/k1d6hn

Table 11-5. Dual SMD Disk Drawer Device Names and Numbers

Controller, Description	Ctlr #	Device #	UNIX Device Name
Attached to VME board in card cage slot 0			
right hand drive	c2 and up	d0	/dev/dsk/c2d0s*x* /dev/dsk/c2d0s*x* ...
left hand drive	c2 and up	d1	/dev/dsk/c2d1s*x* /dev/dsk/c3d1s*x* ...
Attached to VME board in card cage slot 9			
right hand drive	k2 and up	d0	/dev/dsk/k2d0s*x* /dev/dsk/k3d0s*x* ...
left hand drive	k2 and up	d1	/dev/dsk/k2d1s*x* /dev/dsk/k3d1s*x* ...

Dual SMD Disk Device Notes:

(1) Controller numbers c2, c3, c4, etc. and k2, k3, k4, etc. are all reserved for SMD device controllers. Assign controller numbers consecutively.

(2) The italized *x* in the Table 11-1 UNIX and PROM device references (last two columns of the table) refers to a specific disk partition or tape section.

(3) Each SMD drive controller can handle up to 2 daisy-chained drives. It you are setting up a disk-striped configuration in which each controller is connected on only one drive, use the device numbers as given in the table on this page (d0 or d1) but change the device controller number every time you add a new controller.

The following sections give specific information about the disk drives and the internal SCSI tape drive. For disk and tape installation and adminstrative procedures, see the *Installation* section of this guide and the chapter entitled *Disk Drives*.

1/4–inch SCSI Cartridge Tape Drive

The 1/4-inch cartridge tape is used for software distribution and backups. Stardent 1500/3000 uses a a SCSI controller-based 1/4-inch tape drive.

Reading and Writing the Cartridge Tape

NOTE
To avoid potential read errors, we recommend that you use high density tapes for reading and writing.

Be especially careful when writing tapes to Stardent 1500/3000. The only tape format that can be written on the SCSI 1/4–inch tape drive is the *high* density format (3M DC600A or Sony QD-600A). The DC300A tape can be *read* (on the low-density driver */dev/rmt0* or */dev/rmt/c0d6l),* but *cannot be written* on Stardent 1500/3000. In particular, *do not* use Sun tapes for backups or any writing.

Note that the file copy command *dd*(1M) has conversion features that allow you to read tapes written on other systems. See the *Commands Reference Manual* for details.

For your reference, the tape standard is ANSI standard X3B5/85: "Unrecorded Magnetic Mini-Tape Cartridge For Standard Interchange." Here are the most important components of this standard:

Size	.25 inch (6.30 mm)
Flux Transitions	12,500 ftpi (495ftpmm)
Coercivity	550 Oersteds (44,000 Ampers/Meter)

Stardent Computer Corp. distributes its software on the 3M CD600A cartridge tape, which complies with this standard. The 3M CD600A tape holds 125 MB of data in the QIC-24 tape format.

Loading and Unloading the Cartridge Tape

To load a cartridge tape in the drive:

(1) Face the cartridge tape drive.

(2) Orient the tape so that the metal plate is down, the writing is on the top, and the roller, capstan, and tape opening are to the left.

(3) Push in the tape. The tape travels to the back of the drive and the tape lock (on the lower left of the tape) snaps into position to hold in the tape.

Once the tape is placed in the drive the green light located on the tape drive lights up. The drive tensions the tape and positions the tape so that the beginning-of-tape (BOT) marker is under the read/write tape head. At this point, and whenever the tape is in the drive and in the BOT position, the green light on the front panel turns off.

To remove the cartridge tape from the drive:

(1) Wait until the green light is off.

(2) Press on the tape.

(3) The tape lock snaps out of the way and the tape partially ejects. You can then remove the tape from the drive.

Disk Drives

Stardent currently supplies the following disk drives for use with Stardent 1500/3000:

PRIAM 738: A 5 1/4–inch SCSI disk drive with capacity 380 MB unformatted (320 MB formatted).

MAXTOR XT-8760S: A 5 1/4–inch SCSI disk drive with capacity 760 MB unformatted (640 MB formatted).

Fujitsu K2382M: An 8–inch SMD disk drive with capacity 1 GB unformatted.

Disk Specifications

Table 11-6 contains disk drive specifications.

Table 11-6. Stardent 1500/3000 Disk Specifications

Item	PRIAM 738	MAXTOR XT-8760S	Fujitsu K2382M
Controller name	scsi	scsi	VME
Controller number	c0,c1,k0 or k1	c0,c1,k0 or k1	c2 and up or k2 and up
Sector size	1K	1K	2K
Sectors/track	18	28	22
Tracks/cylinder	15	15	27
Sectors/cylinder	270	416	594
Cylinders	1222	1629	745
mkfs parameters:			
blks/track	9	8	11
tracks/cyl	15	13	27

Disk Formats and Partitions

Each disk drive contains a volume header and a number of disk partitions (as specified in the volume header). The volume header resides at the beginning of the disk and contains the information summarized in Table 11-7.

As a system administrator you interact with the disk volume header to obtain partition sizes for creating a file system (the *mkfs*(1M) command), or to repartition the disk. See *File System Administration* for file system creation and *Disk Drives* for information about partitioning the disk.

Table 11-7. Disk Volume Header Information

Item	Description
disk device parameters	device-dependent parameters
volume header file system	small, flat file system used for standalone programs that can be booted directly from the PROM or from the standalone shell, *sash*
disk partition table	list of up to 12 partitions numbered 0-11, each of which contains this information: partition number number of (1024-byte) blocks in the partition first block in the partion partition type (file system, volume header, bad track forwarding (track replacement), raw, diagnostic, or entire volume)

UNIX Disk Partition References

UNIX (and PROM) references to disks require a disk partition number. Table 11-8 lists some important partitions on the default primary boot disk (*/dev/dsk/c1d5sx*).

Table 11-8. Default Primary Disk Partitions

Function	Partition Number	Partition Name
root file system partition	0	/dev/dsk/c1d5s0
swap partition	1	/dev/dsk/c1d5s1
entire disk partition	5	/dev/dsk/c1d5s5
volume header	8	/dev/dsk/c1d5vh
track replacement partition	9	/dev/dsk/c1d5s9
entire volume partition	10	/dev/dsk/c1d5s10
diagnostics partition	11	/dev/dsk/c1d5s11

Disk Drives in this guide contains a reprint of the full default partition table for the primary boot disk. For overview purposes, simply note the following:

- Partition numbers are purely logical; they do not refer to actual physical location on the disk. For instance, partition 8, the volume header, always begins at block 0 of the disk.

- There is a direct association between disk partitions and UNIX file systems; each file system resides on a disk partition. Your choice of which partitions you use for user files or data is a function of available disk partitions and their sizes. As you create file systems and assign particular uses to them, keep the following in mind:
    - The root file system should not be used for any user files or data.
    - Partitions 8-11 on each disk are reserved for the purposes given in Table 11-8 above.
    - The default partitions configured for your system are sufficient for many applications. You may have no need as a system administrator to repartition the disk.

Computer Boards

The Stardent 1500/3000 System Module card cage contains one or more CPU boards, one or more memory boards, one or two I/O boards, one or two graphics boards, and (optionally) graphics expansion and VME expansion boards. Refer to Figure 2-1 in Chapter 2 of this guide for the proper location of the boards in the card cage as a function of configuration option:

- Single I/O board
- Dual I/O board, single graphics board
- Dual I/O board, dual graphics boards
- Dual I/O board, no graphics (Server)

Each board has a small set of LEDs near the top of the board. On each board the top (first) LED is green. It becomes active after power-up when the board passes a go/no-go test. It should remain on for the entire time the system is operational.

On each remaining board all of the remaining LEDs are red. The second LED on each board is a general problem indicator. If it is on while the top LED is off, there is a definite problem. If both the first and second LEDs are on, there may or may be a problem. If both LEDs are on, a needed hardware check has probably not been done.

On the CPU and memory boards LEDs 3 and 4 are bus activity lights. LED 3 is an S-bus activity light and LED 4 is an R-bus activity light. If you observe these LEDs while the system is running, you can judge levels of activity on the two system buses. The last six LEDs on each board are for more detailed diagnostic purposes.

For more on the CPU board LEDs see the *PROM Manual* section of this guide. For detailed descriptions of the boards themselves, see the *Hardware Reference Manual.*

Diagnostics Communication Port

Stardent 1500/3000 has an asynchronous communication port available for diagnostic purposes. The port is accessible through the RS-232 jack on the back of the master CPU board. (See below for identification of the master CPU board.) You can connect a VT100-compatible terminal directly via a null modem cable, or you can connect a modem (the default is 2400 baud). The terminal should support

- 8 bit characters, one stop bit, no parity
- CR, NL, LF, TAB, and BS characters
- 2400 and 9600 baud operation.

To use a modem or terminal attached to the DCP you need to ensure baud rate compatibility between the port and the modem or terminal. You can do that in any of the following ways.

(1) Set the NVRAM variable *baud* to the baud rate of the terminal or modem. (See *NVRAM Variables* in this guide for instructions.)

(2) Use devices that operate at the default rate of 2400 baud.

(3) When the terminal or modem is connected and communications are established, press the BREAK and RETURN keys in sequence until you get recognizable PROM information on the screen. The BREAK-RETURN sequence cycles you through the feasible set of port baud rates.

Once you have reached the PROM over the DCP, you can run diagnostic tests or boot the Stardent 1500/3000 operating system. See *Booting the System* in this guide for booting instructions.

Master CPU Board

NOTE
The boot master CPU is only a master processor from the perspective of the PROM. From a UNIX and general architectual perspective, there is no difference among processors. All functionality is symmetric.

In multiple processor configurations one of the CPU boards is designated as the boot master. It is the first board booted when the system is powered up and it is the board whose DCP should be used for general diagnostic purposes.

Once the PROM is operational you can determine the master CPU board by the activity of the board LEDs. During normal operations the LEDs on the master CPU board flash up and down, while the LEDs on any other CPU boards flash in unison.

Table 11-9 shows the default card cage location of the master CPU board as a function of system configuration. It also lists the CPU ID of each board. (The CPU ID is used for a variety of internal system functions.[1])

Table 11-9. Default Master CPU Board Assignments

Number of CPU boards	Slot Containing Default Master CPU	CPU ID of Master CPU
1	5	2
2	2	1
3	4	0
4	4	0

If you wish to change the master CPU you can do so by powering down and disabling the current master CPU board.

The PROM prompt always shows the CPU ID of the current master CPU board. If you see the prompt

```
prom 1>
```

for instance, the boot master CPU is the CPU board with CPU ID 1.

You cannot change the boot master CPU without physically disabling or removing the master board, but the PROM does let you communicate with other CPU boards. Use the command

[1] To determine the mapping between CPU ID and card cage slot, take the octal representation of the card cage slot, drop the 4's digit, and reverse the remaining two digits. The resulting number is the CPU ID. Thus, for instance, the CPU board in slot 5 has octal representation:

    1 0 1

Dropping the 4's digit leaves

    0 1

and reversing the remaining digits leaves

    1 0

which is the binary representation of the number 2. Thus, the CPU ID of the CPU board in slot 5 is 2.

```
prom 0> x newCPUID
```

to communicate with other CPU boards, where *newCPUID* is the CPU ID of the board with which you wish to communicate.

Preventive Maintenance

Preventive maintenance consists of retensioning the cartridge tape, cleaning the cartridge tape head, and cleaning the air filter.

Cartridge tape retensioning. The cartridge tape should be retensioned after every 2 hours of use, to compensate for the over- and under-stretching that is inevitable when tapes are used. From the UNIX system use the *mt*(1) command:

```
# mt -f /dev/rmtxx reten
```

where */dev/rmtxx* is the name of a cartridge tape *special device file* as listed in Table 1 in *System Overview* in this guide and described in *File System Administration*.

Cartridge tape head cleaning. The cartridge tape head should be cleaned after every few hours of use. Follow these steps:

(1) Dock the head, if necessary, to gain access. To dock the head, press inward on the tape carriage plate where the cartridge is normally inserted. Stop pushing when the travel is restricted and the latching mechanism releases. The carriage plate slides out automatically (about 3/8 inch). The head is now docked.

(2) Use a foam swab soaked in head cleaning solution to clean the head and the tape cleaner blades. Use only lint-free swabs. **Never clean the capstan with the head cleaner, as severe damage may result.** Return the head to the ready (to load tape) position by pushing the carriage plate until its motion is restricted from any further travel.

Alternatively, you can clean the head with the Perfect Data QIC II Drive Head Cleaning Kit. (Wangtec Part No. 102791-21). The kit consists of a tape cartridge form factor with a cleaning pad and an arm for moving the cleaning pad across the head. Apply a few drops of cleaning solution to the pad, then insert the cleaning cartridge into the drive. Move the handle up and down to clean the head surface. After cleaning, remove the cartridge and proceed with normal use.

Air Filter Cleaning. An air filter is located inside the front door of the System Module, near the bottom. The spongy filter is held in place by a textured boundary strip on the door. The filter should be cleaned when dirty (about every 6 months for an average environment). Check the filter more often if your environment is unusually dusty. Follow these steps to clean the filter.

(1) Power down Stardent 1500/3000 and open the front door. Gently pull the air filter to remove it.

(2) Rinse the filter with water and shake it out to dry.

(3) Put the filter back in its place on the inside of the System Module front door. Press around the edges to secure it.

Software Environment

The Stardent 1500/3000 operating system is Stardent Computer's adaptation of the UNIX operating system. It includes a full implementation of AT&T's UNIX System V Release 3 with Berkeley 4.3 UNIX extensions, high performance enhancements, and support for Stardent 1500/3000's integrated graphics.

System administrative procedures are derived from these sources according to Table 11-10:

Table 11-10. Stardent 1500/3000 Administration: Sources

Stardent 1500/3000 Administration: Sources	
Stardent Computer	
	Hardware configuration and operation
	Initial operating system installation
	Operating system updates
	Boot procedure
	Disk partitioning, formatting, file system type selection
	Cartridge tape procedures
	Prom monitor
	System crash dump procedure
	Root file system reconstruction
	Networking (NFS)
AT&T V.3 UNIX System	
	User administration
	Terminal line and terminal type administration
	File system creation, checking, mounting
	Back-up procedure
	System multi-user boot sequence
	System shut down
	System security
	Line printer spooling
	uucp
	Mail (mailx)
Berkeley 4.3 UNIX System	
	C-shell
	Networking (rlogin, rsh, /etc/hosts, ...)
	Mail (aliases, sendmail,...)

Kernel Environment

The Stardent 1500/3000 operating system supports all AT&T V.3 kernel calls. It supports BSD 4.3 kernel calls for TCP/IP, Berkeley mail, and selected Berkeley features such as job control and symbolic links.

User Environment

The user environment consists mainly of AT&T V.3 commands. They are located in the usual places: */bin*, */usr/bin*, and */etc*. BSD 4.3 derived commands are located in */usr/ucb*.

Compilation Environment

NOTE
Because the Stardent 1500/3000 operating system is based on System V UNIX, not all BSD system calls are available. You can check on the availability of particular system calls by using the *man* command.

The Stardent 1500/3000 provides compilation environments for both AT&T V.3 and BSD 4.3 programs. Programs can be compiled to use either set of system calls or a combination thereof. You can make the selection on the command line. For instance, the C compiler accepts

```
cc -43 {standard cc arguments}
```

to specify the Berkeley system call libraries and header files.

Separate *lib* and *include* directories exist for AT&T V.3 and BSD 4.3. The AT&T V.3 directories are in their usual places: */usr/include, /usr/lib,* and */lib*. The BSD directories are in */lib/bsd, /usr/include/bsd,* and */usr/lib/bsd*.

Where to Find Things

Here is a general list of directories and files needed for system administration. The directory and file hierarchy is illustrated in Figure 11-4.

/bin	Contains user commands.
/dev	Contains special device files and directories (see *File System Administation).*
/etc	A collection of files used in system administration.
/lib, /usr/lib	Contains AT&T-derived library files and specialized commands.
/lib/bsd, /usr/lib/bsd	Contains BSD-derived library files.
/tmp	Contains temporary files. This is one of the directories that needs to be monitored for excessive growth. (See *File System Administation).*
/usr/adm	Contains logs of system error messages. */usr/adm/messages* contains general system error messages; specialized messages are in other */usr/adm* files.

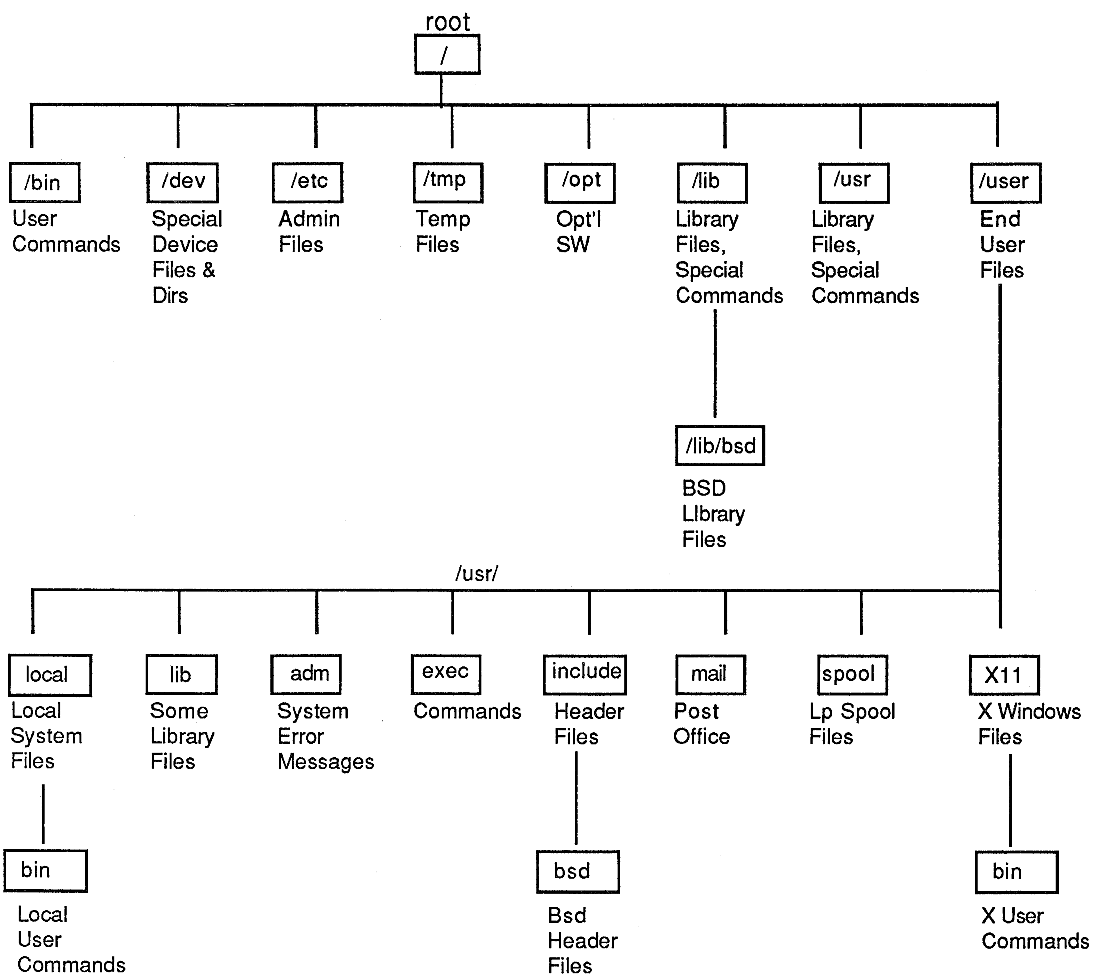


Figure 11-4. UNIX Administration Directories and Files

/usr/bin	Contains executable commands.
/usr/include	Contains AT&T-derived header files for C programs.
/usr/include/bsd	Contains BSD-derived header files for C programs.
/usr/mail	The local post office directory.
/usr/spool	Contains the lp spool, mail, uucp, and cron subdirectories.
/usr/X11	Contains X Window System files.
/usr/X11/bin	Contains X Window System commands.

/etc/cron. You can use the program */etc/cron* to execute commands at specified dates and times. */etc/cron* is a *daemon*, a program that runs in the background when needed, with no active intervention. When activated by a shell script, */etc/cron* scans the files in */usr/spool/cron/crontabs* regularly to see what commands should be executed. *crontab* files have the following format:

```
min hour daymo month daywk command
```

Here is a sample entry from a *crontab* file.

```
00 17 * * 1      /etc/fsstat >/dev/console 2>/dev/console
```

The entry means "At 5:00pm in the evening on mondays during any month of the year, check to see if there are any file systems that need to be fscked."

Video Recording

Stardent 1500/3000 has an RS-170 interface that allows you to record video images from the Stardent 1500/3000 screen. With the proper equipment you can record continuous motion scenes or single frames that are then converted into an animation sequence.

Equipment

To record video images from Stardent 1500/3000 you need these pieces of equipment:

- A Stardent 1500/3000 graphics expansion board as part of your system configuration.
- An NTSC encoder to convert separate RGB channels into a composite NTSC-encoded signal.[2]
- A sync generator, to feed a synchronizing signal to Stardent 1500/3000 and to the encoder.[3]

[2] Stardent 1500/3000 supports several video formats: NTSC (National Television Standards Committee), PAL (Phase Alternating Line), and Betacam™. For PAL recording substitute PAL equipment for the NTSC equipment listed here and use the *pal* command instead of the *rs170* command described below under "How to Record Video." For Betacam™ recording the arrangement is somewhat different. See "Betacam™ Recording," below.

[3] Stardent 1500/3000 does not support "Sync On Green." A separate sync generator is always needed.

(The encoder and sync generator may be combined in one unit.)

- A 1/2, 3/4, or 1-inch video tape recorder (VTR), to record the NTSC signal onto tape. For animation recording a VTR capable of frame-by-frame recording is needed.
- A video monitor to allow you to view the video sequence as it is being recorded by the VTR.
- For frame-by-frame animation, an animation controller to accept single frames from Stardent 1500/3000 and send them in sequence to the video recorder.

RS-170 Specification

Stardent 1500/3000 meets the RS-170 specification for video recording through ports on its graphics expansion board. RS-170 is a low resolution (broadcast) video standard with signals 15.7Khz horizontal and 30hz interlaced vertical (about 640 pixels by 480 lines). To record from the high resolution (1280 pixel by 1024 line) Stardent 1500/3000 screen you have the option of restricting the recorded image to a 640 pixel by 480 line area of the screen or of using a *high resolution scan-converter* to convert an entire 1280x1024 screen image into RGB and NTSC output. The high resolution scan-converter approach is more costly, but it does allow you record the output of the entire 1280x1024 screen.

Video Tape Recording Formats

Video tape recording formats vary in quality and price. You can record video images from Stardent 1500/3000 in any of the following formats:

VHS: VHS is a 1/2-inch format that is recorded in the composite mode. [4] Signal horizontal resolution is approximately 210 lines. [5] VHS is appropriate for applications where image quality is not critical and when no copies (multi-generation recordings) of the master tape are required.

3/4-inch U–Matic: U-Matic is recorded in the composite mode but is more stable and less noisy than VHS. Signal horizontal resolution is about 250 lines. U-Matic is considered to be an industrial

[4] Composite is the NTSC signal that combines all sync, timing, color and luminance into one signal.

[5] This is a measure of signal clarity, not of the actual scan lines recorded.

standard and is suitable as a master tape for duplication of distribution copies.

3/4-inch SP U–Matic: SP U-Matic is fully interchangeable with standard U-Matic but utilizes a metal particle tape to increase resolution while decreasing chroma noise and ringing. Signal horizontal resolution is about 330 lines. In general terms, third generation SP U-Matic is of the same quality and first generation standard U-Matic.

SP Betacam™: SP Betacam™ is a 1/2-inch format (not the same as 1/2-inch home Beta) that is recorded in component mode (3 discreet signals are recorded on tape). By eliminating the NTSC composite process many aritfacts are eliminated, i.e., ringing, chroma noise, chroma crawl. Also, resolution is dramatically increased to approximately 360 lines because luminance is recorded separately.

Composite Recording

Figure 11-5 shows the basic signal flow for composite NTSC video recording. All video signals are connected by 75-Ohm coaxial cables with BNC connectors. This is the basic signal flow for all VHS, 3/4-inch U-Matic and 3/4-inch SP U-Matic arrangements.

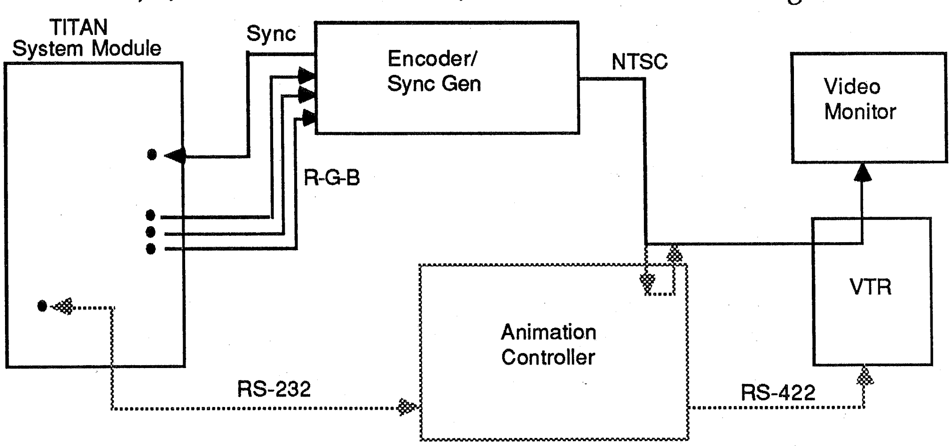


Figure 11-5. Video Recording Equipment Placement

As illustrated, the Stardent 1500/3000 graphics expansion board outputs RGB signals, which are sent to a combined encoder and sync generator. The encoder in turn produces a composite NTSC signal that is sent to the video recorder and to a video monitor.

The sync generator built into the encoder unit provides sync and timing signals for the video signal.

Sync is sent to Stardent 1500/3000 so that its output is synchronized with the encoder.

Animation Recording

The dashed lines in Figure 1-4 represent the control cables and additional signal connections required for recording animation sequences.

To record animation sequences you need an animation controller and a VTR that is able to do single-frame recording, in addition to the encoder and sync generator.

Figure 11-6 shows one possible arrangement for recording animation sequences. Links from Stardent 1500/3000 to the animation controller and from the animation controller to the VTR control the recording of individual frames. From Stardent 1500/3000, control is maintained via an RS-232 line connected to port A or B on the System Module I/O board. From the animation controller to the VTR, control is maintained over an RS-422 link. Note that Stardent 1500/3000 requires custom software to communicate with the animation controller over the RS-232 link.

The NTSC signal is routed from the encoder to the animation controller and then on to the VTR and video monitor. In the arrangement shown, loop-throughs within the animation controller and the VTR are used to route the NTSC cable.

The illustration in Figure 1-5 is a general guideline for animation recording, not a specification. For specifics, rely on wiring diagrams for the animation controller you plan to use.

Betacam™ Recording

Figure 11-7 shows basic signal flow for component video recording (e.g. Betacam™). All video signals are connected by 75-Ohm coaxial cables with BNC connectors. The dashed lines show additional connections for animation recording.

In this case component signals (Y, R-Y, B-Y) are recorded on tape and a transcoder is used to convert Stardent 1500/3000's RGB output to Y, R-Y, B-Y.

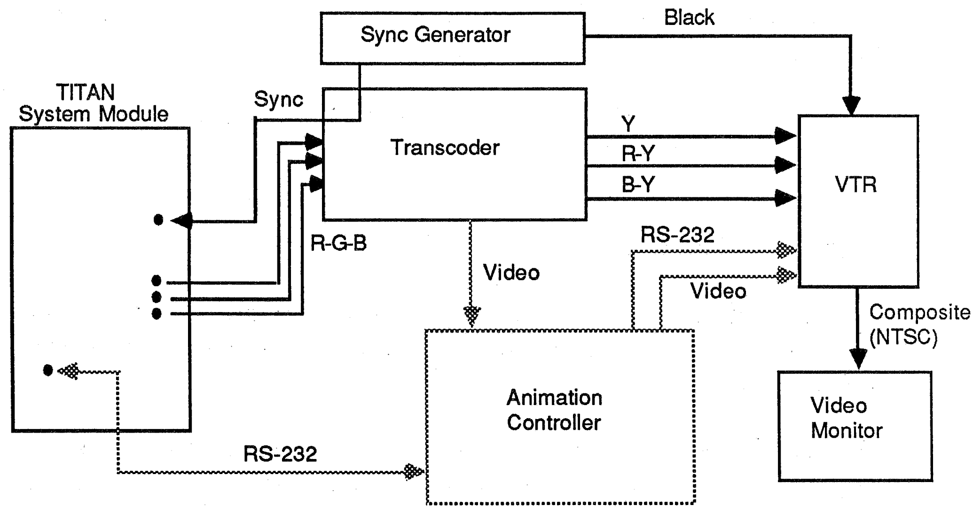


Figure 11-6. Component Video Recording Equipment Placement

How to Record Video

To make video recordings use the UNIX system command *rs170*(1).[6] When you issue the command

```
# rs170 on
```

Stardent 1500/3000 begins outputting the video image present in the 640 pixel by 480 line region in the upper left hand corner of the Stardent 1500/3000 screen.[7] Please note these points:

(1) The Stardent 1500/3000 monitor screen is disabled when you issue the *rs170 on* command, because the command changes the video timing rate of the Stardent 1500/3000 hardware. [8] To see the image being recorded you must use a separate video monitor as shown in Figure 1-4. With the video monitor you can see the image being recorded and the effects of interaction (e.g. with the knob box).

[6] If you are using the PAL format for video recording, substitute the command *pal on* or *pal off* for the

[7] Unless you are using a high resolution scan-converter, in which case the entire screen image is recorded.

[8] Scrambled fragments of the image appear on the Stardent 1500/3000 monitor while in RS170 mode. This is normal.

(2) While you are recording, you can issue commands from the Stardent 1500/3000 monitor or from another terminal. You do not see the results of the command on the Stardent 1500/3000 monitor screen, however.

(3) Because you are recording only the image in the 640x480 upper left hand corner of the Stardent 1500/3000 screen, the image should be in a window scaled to that size. The Doré call to display graphics in that area is

```
DoDevice("ardentx11","-geometry 640x480+0+0")
```

(For information on the Doré Library, see the *Technical Overview* and the *Doré Programmer's Guide*.)

To stop a video recording sequence, issue the command

```
# rs170 off
```

You can issue the command from the Stardent 1500/3000 monitor or from another terminal. If you use the Stardent 1500/3000 monitor, try to make sure that the mouse pointer has not been moved during the video recording sequence. If it has, you may need to search for a UNIX window before the *rs170 off* command is accepted.

NVRAM VARIABLES

CHAPTER TWELVE

NVRAM (non-volatile random access memory) variables are environment variables for the PROM. The NVRAM variables (listed in Table 12-1) serve as parameters for the boot process, determine monitor characteristics, store network addresses, and determine the baud rate for Diagnostic Communications Port communications.

Table 12-1. NVRAM Variables [1]

Variable	Function	Default	Mod. Status
Boot:			
bootmode	auto or manual boot	a	*
bootfile	program to be booted	unix	
path	device containing bootfile	scsi(1,5,0)	
rootdev	disk partition for root file system	scsi(1,5,0)	
secondary	secondary boot program	scsi(1,5,8)sash	
swapdev	disk partition for swap and page	scsi(1,5,1)	
dumpdev	device for system crash dumps	scsi(1,5,1)	
Monitor:			
screen_font	monitor font choice	2	
screen_back	background color	4411cc (hex)	*
screen_fore	character color	ffff00 (hex)	*
monitor	monitor type	std	*
Network:			
etheraddr	machine's Ethernet address	(individual)	
netaddr	machine's Internet address	(individual)	**
hostname	machine name	(individual)	**
Communications:			
baud	direct connect port baud rate	9600	*

Administrative tasks consist of displaying and setting the values of NVRAM variables. For more on NVRAM variables see the *PROM Manual* section of this guide.

NVRAM Variables: Tasks

1. Display/set the values of NVRAM variables. To see a list of all current NVRAM variables issue the command

```
# nvram
```

from the UNIX system or from the PROM (the PROM also accepts the abbreviation **n**). To set the value of an NVRAM variable use the command

nvram *variable=value*

from the UNIX system or the PROM (again, the PROM accepts **n** as an abbreviation.)

2. Set NVRAM boot variables. We recommend that you leave the boot variables at their default values, with the optional exception of *bootmode*. *bootmode* determines whether or not the system autoboots upon power-up or reset. Choose *a* for autoboot and *m* for manual. If you choose manual the PROM prompt appears upon power on or reset; if you choose autoboot, the UNIX system is booted upon power-up or reset. [2]

To set the variable *bootmode* to manual issue this command at the PROM prompt:

---

[1] The "modification status" column in the table shows whether or not the file requires modification. If the modification status box is empty, no modification is recommended. (Commands fall in this category.) If the box has a single asterisk (*), modification is optional. If the box has a double asterisk (**), the file or variable must be examined (and modified if necessary).

[2] In autoboot mode the system boots to single or multi-user level according to the "initdefault" entry in the file */etc/inittab*. If the line

is:2:initdefault:

appears at the beginning of */etc/inittab* the system boots to multi-user mode (run level 2); if the line

is:1:initdefault:

appears the system boots to single user mode.

```
prom 1> n bootmode=m
```

3. Set NVRAM monitor variables. Table 12-1 shows the four NVRAM monitor variables. *screen_font* need not be changed. *monitor* should be set when the system is initialized for the first time. Choose *std* for standard monitor and *stereo* for stereo monitor.

screen_back and *screen_fore,* respectively, determine the background and character color for the monitor. The values are coded as

RRGGBB

where each pair of letters is a hex number (0-ff) showing the intensity of the color red, green, or blue. For instance, the value *ff0000* shows the maximum intensity of red, with no green or blue present, *f600e1* shows high intensities of red and blue, with no green, present, and *23aa89* shows a combination of the three colors. The values of *screen_back* and *screen_fore* can be changed at will.

4. Set NVRAM network variables. Three NVRAM variables, *etheraddr, hostname,* and *netaddr,* are required for configuration of an Ethernet (or Cheapernet) network.

Each Stardent 1500/3000 has already been assigned an Ethernet address (maintained in the NVRAM variable *etheraddr*) that looks something like this:

```
00-00-7a-80-00-07
```

Do not change the address unless you are an experienced network administrator doing special network customization.

hostname should be changed to the name of your machine. Choose a short, alphanumeric name.

netaddr defines an Internet address for the system. See *Network* in this chapter for information about obtaining an Internet address and about uses of the variable *hostname.*

DISK DRIVES

CHAPTER THIRTEEN

For a general overview of the Stardent 1500/3000 disks, see *System Overview* in this guide. This chapter describes how to format, partition and stripe Stardent 1500/3000-supported SCSI and SMD disk drives. Table 13-1 lists the commands, files, and variables needed for disk configuration.

Table 13-1. Disk Configuration

Item	Function
Utility: standalone format program	 format disks, from PROM
Command: dvhtool	 read, modify disk volume header
Files: /dev/dsk /dev/rmt*	 (block) disk device files raw disk device files
NVRAM Variables: path rootdev swapdev dumpdev	 device containing boot file disk partition for root file system disk partition for swap and page device for system dumps

If you need to install a SCSI or SMD disk drive, see the *Field Service Manual* or your Stardent field service representative. You can use the diagnostic program *periph.diag* to check any SCSI disk drives you have just installed and to check the current device configuration. Load the diagnostics tape shipped with your system (see *System Overview* in this guide for instructions on how to use the tape drive) and type

```
prom 1>  b scsi(0,6,4)
```

to boot the peripheral diagnostics program from tape.

The program brings up a menu that allows you to check the current SCSI device configuration and perform a suite of diagnostic tests before returning you to the PROM prompt.

Formatting Disks

WARNING
Before you format a disk, make sure that you have no need for any data on the disk. Formatting a disk destroys all data on the disk.

Disk formatting is required when you configure a new disk for the system and, in rare cases, when a disk has been damaged. The Stardent 1500/3000 format program formats both SCSI and SMD disk drives.

To boot the format program (which resides in section 2 of the Stardent distribution boot tape), bring up the PROM from a power-off condition by turning the lower key switch on the front panel to the "diagnostic" position and turning the machine on. Load the Stardent software distribution boot tape (tape 1 of 2) according to the instructions given in *System Overview* in this guide. Then, at the PROM prompt, type

```
prom 1> b scsi(0,6,2)
```

You see some initial messages on the screen, then the format program prompts you for type of disk interface:

```
Format Utility

Known Disk Interface Types:
        (1)   SCSI bus
        (2)   Interphase (SMD)
Select Interface Type (1-2, 0 to quit):
```

Type 1 to format a SCSI disk or 2 to format an SMD disk.

At this point the format program branches according to whether you request to format a SCSI or SMD drive. The next two steps describe how to use the format program in each case.

Formatting SCSI Disk Drives

After you select the SCSI disk interface type the format program prompts you for controller and unit number. (Unit number is the same as device number.) Refer to *System Overview* in this guide for correct controller and unit numbers. For instance, to format the primary boot disk (controller 1, device or unit number 5) answer as follows:

```
Select Interface Type (1-2, 0 to quit): 1
controller number? 1
unit number? 5

Device is a MAXTOR XT-8760S
```

Notice that once you have selected a particular disk drive, the format program prints the make and model of the drive, followed by a question.

```
Use default parameters? y
```

Unless you are absolutely sure that you need special device parameters, always accept the default. If the drive has already been formatted on a Stardent 1500/3000 and the default paramerters are different from the current parameters, the following message and question appear on the screen:

```
New parameters invalidate partition table.
Reinitialize the partition table? y
```

You *must* answer "yes."

The program next asks you if you wish to dump and/or modify the device parameters. **It is not necessary to do so**. The format program automatically supplies parameters that are correct for the device you selected. If you choose to dump the device parameters you see output similar to this:

```
dump device parameters (y/n)? y
Error parameters:
        Error recovery modes:
                Error correction: Enabled
                Recovered Errors: Report
                Transfer on error: Enabled
                Automatic Write Forwarding: n
                Automatic Read Forwarding: n
        Retry Count = 16
        Correction Span = 0
        Head Offset Count = 0
        Data Strobe Offset Count = 0
        Recovery Time Limit: No Retries
Disconnect/Reselect Parameters:
        Buffer Full Ratio = NA
        Buffer Empty Ratio = NA
        Bus Inactivity Limit = NA
        Disconnect Time LImit = NA
        Connect Time Limit = NA
Configuration/Sparing Parameters:
        Disk Geometry: 1222 Cylinders, 15 Tracks/Cylinder
        Track Configuration = 19 blocks/track, 1024 bytes/block
        Sparing Parameters:
```

```
                Tracks per Defect Zone = 15
                Alternate Tracks per Defect Zone = 15
                Alternate Tracks per Volume = 0
        Interleave = 30
        Track (Spiral) Skew = 0
        Cylinder Skew = 0
modify device parameters (y/n)?
```

Table 13-2 shows the parameters supplied by the format program for the PRIAM 738 and MAXTOR XT-8760S SCSI drives. If you do need to change a parameter answer "yes" to the modify question. The program then prompts you for any parameter changes that are permitted for the drive. To change a parameter type in the new value (or *yes* if it is a yes/no question); to maintain the current value type RETURN (or *no* if it is a yes/no question). To reiterate, however, **it is not necessary to change any parameters.**

If you change the track configuration, the alternate tracks per defect zone, or the alternate tracks per volume parameter, you see the following message:

```
New parameters invalidate partition table.
```

Next you are asked if you wish to reinitialize the partition table. You *must* answer "yes."

```
Reinitialize partition table? y
```

The format program next asks if you wish to dump or modify the partition table. It is not necessary to do so. (If you need to partition the disk, do so with the *dvhtool*(1M) command.)

```
dump partition table (y/n)? n
modify partition table (y/n)? n
```

The program next asks if you wish to reassign any bad blocks. Answer no. (If new back blocks are found during formatting they are automatically added to the bad block list.)

```
Any bad block to reassign (y/n)? n
```

Table 13-2. SCSI Device Parameters for Format Program

Default Parameters	Priam 738	MAXTOR XT 8760S
Error correction	Enabled	Enabled
Recovered Errors	Report	Report
Transfer on error	Enabled	Enabled
Automatic Write Forwarding	n	n
Automatic Read Forwarding	n	n
Retry Count	16	16
Correction Span	0	12
Head Offset Count	0	0
Data Strobe Offset Count	0	0
Recovery Time Limit	No retries	No retries
Buffer Full Ratio	NA	0
Buffer Empty Ratio	NA	0
Bus Inactivity Limit	NA	1000usec
Disconnect Time LImit	NA	0
Connect Time Limit	NA	0
Disk Geometry		
Cylinders	1222	1629
Tracks/Cyl	15	15
Track Configuration		
Blks/Track	19	28
Bytes/Blk	1024	1024
Tracks per Defect Zone	15	15
Alternate Sectors per Defect Zone	15	4
Alternate Tracks per Defect Zone	0	0
Alternate Tracks per Volume	30	45
Interleave	1	1
Track (Spiral) Skew	0	0
Cylinder Skew	0	65535

Finally, the program asks for a go-ahead to format the disk. **Remember that formatting destroys all data on the disk.**

```
Formatting destroys all data: format (y/n)? y
Use current drive defect list (y/n)? y
Scan disk for defects (y/n)? y
```

The format program now formats the disk. **Disk formatting may take up to 30 minutes.**

Once the disk is formatted you are asked a few additional questions. Answer yes to each one.

The program reports new bad blocks to be added to the bad block list, and exits, returning control to the PROM.

Formatting SMD Disk Drives

Formatting SMD disk drives differs from formatting SCSI drives in two respects:

- Each SMD drive is controlled by a VME Controller, each of which can control up to two SMD drives. (Stardent has qualified the Interphase 4200 controller for this purpose.) The controller number you supply to the format program must correspond to the VME address coded on the appropriate Interphase Controller. It is not the same as the UNIX controller number given in *System Overview*, Table 11-5.
- SMD drives use a different set of device parameters.

Figure 13-1 gives controller numbers to supply to the format program as a function of VME address and UNIX device name. For information on the Interphase controller and its switch settings, see the *Field Service Manual* or your Stardent field service representative.

As an example, suppose you wish to format the SMD disk with UNIX device name */dev/dsk/c3d0sx*. Answer as follows:

```
Format Utility

Known Disk Interface Types:
        (1)    SCSI bus
        (2)    Interphase (SMD)

Select Interface Type (1-2, 0 to quit): 2
controller number? 1
unit number? 0
```

When an SMD drive is first shipped from the manufacturer, the manufacturer records the list of defective sectors on the disk medium itself. If the drive you are formatting is new, you need to read the defects list off the drive before the format program erases the list. Once the disk is formatted successfully the defects list is written in the disk volume header.

If the disk is new, answer yes to the question:

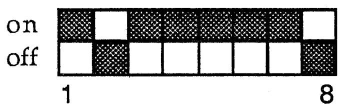
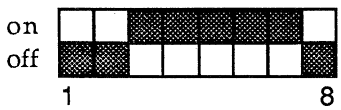
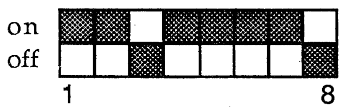
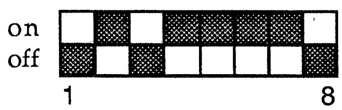
VME Controller Switch Settings	VME Address	UNIX Name	Format Controller
	0x400	c2dXsX	0
	0x600	c3dXsX	1
	0x800	c4dXsX	2
	0xa00	c5dXsX	3

Figure 13-1. Format Program Controller Numbers for SMD Drives

`read defects off the drive (y/n)?` **y** *only if new drive*

If the drive has previously been formatted, you *must* answer "n" to the question.

The program now asks a series of questions similar to those for SCSI disk formatting (see Step 5 above). Note that the device parameters differ from the SCSI parameters. Correct SMD device parameter for the Fujitsu 2361 and 2382 drives are given in Table 13-3. As in the SCSI case, **You do not need to dump or modify the parameters.**

Finally, the program asks if you wish to format the disk. **Remember that formatting destroys all data on the disk.**

`Formatting destroys all data: format (y/n)?` **y**

The format program now formats the disk. **Note that disk formatting may take up to 10 minutes.**

Once the disk is formatted you are asked if you wish to scan the disk. We recommend that you answer yes.

```
perform scan (y/n)? y
```

The default scan of 3 times is fine. **Note that scanning takes about an hour.**

After scanning the program asks about media defect list manipulation.

```
media defect list manipulation when prompted
choose one of (list,add,delete,clear,translate,quit)

Command?
```

This is the point at which you may enter the manufacturer's defect list manually, if necessary. The normal response is "quit."

```
Command? quit
```

(If you think you may need to enter the defects list manually, contact Stardent Customer Support for assistance.)

NOTE
If your disk has a bad sector that was not added to the list during formatting, but is identified through UNIX error messages, you may need to use the format program to manually enter add that sector to the defect list. If so, call Stardent Customer Support for assistance.

The program next asks if you wish to perform a map. Answer yes to map out the defective sectors on the disk.

```
perform map (y/n)? y
```

Finally, the program asks if you wish to dump the bad sector table. It is not necessary to do so. The format program then writes the volume header and exits, returning you to the PROM prompt.

```
dump the bad sector table (y/n)? n

prom #>
```

Table 13-3. SMD Device Parameter Selections for Format Program

Parameter	Fujitsu K2382M
spiral skew	0
number words in gap1	14
number word in gap2	14
number cylinders	745
vol 0 starting head	0
number heads in vol 0	27
vol 1 starting head	0
number heads in vol 1	0
sector interleave	1
number sectors per track	22
number bytes per sector	2048
number retries on error	3
milliseconds per word	132
sector slipping	y
sector forwarding	n
track forwarding	y
multiple volumes	n
transferring data regardless of error	n
recalibrate as last resort	y
runt sector enable	y

Repartitioning the Disk

The default disk partitions are suitable for most Stardent 1500/3000 applications. If you do need to repartition the disk, follow the procedure given here.

WARNING
Don't repartition the root partition on the primary boot disk.

Make sure that the file system on which the partition resides is unmounted.

Data on any partition affected by repartitioning are destroyed.

To repartition the disk use the program *dvhtool*(1M). Login as super-user and run the program with the command

```
# dvhtool
```

The program prompts you for commands. Assume that you wish to repartition the first optional SCSI disk, controller number 0, device number 5. Answer as follows.

```
Command? (read, vd, pt, dp, write, bootfile, or quit): r
Volume? /dev/dsk/c0d5vh

Command? (read, vd, pt, dp, write, bootfile, or quit): pt

current contents:
        part    n_blks  1st_blk type    size
        0:       82080    1080  sysv     80 MB
```

```
         1:       108000    83160   sysv      105 MB
         2:       137970   191160   sysv      134 MB
         3:       164160     1080   sysv      160 MB
         4:       163890   165240   sysv      160 MB
         5:       328050     1080   sysv      320 MB
         6:        82080   247050   sysv       80 MB
         7:           -1       -1   sysv        0 MB
         8:         1080        0   volhdr      1 MB
         9:          540   329400   trkrepl     0 MB
        10:       329400        0   volume    321 MB
        11:          270   329130   diag        0 MB

Disk partitioning strategies:
 1:  0   (80 MB)  1 (105 MB)  2 (134 MB)
 2:  0   (80 MB)  1 (105 MB)  6  (80 MB)
 3:  0   (80 MB)  4 (160 MB)
 4:  3 (160 MB)  2 (134 MB)
 5:  3 (160 MB)  4 (160 MB)
 6:  3 (160 MB)  6  (80 MB)
 7:  5 (320 MB)
Command? (part nblks 1stblk type, or l)
```

The first command reads in the disk volume header for the disk (controller number 0, device number 5). The second command lists the disk partition table. The table reprinted here is for a 128 MB memory system; partition tables for smaller sized systems or for SMD disks are similar. After the partition table is listed you are shown disk partitioning strategies consistent with the partition table. Each partitioning strategy represents a combination of disk partitions that jointly cover the entire disk but do not overlap.

Note a couple of things about this table. First, the entire disk is 321 MB, as you can see by looking at partition 10 (defined as the entire disk). Second, no SCSI disk partition should begin before the end of block 8 and the beginning of block 11. No SMD disk partition should begin before the end of block 9 and the beginning of block 11. This is because the first four cylinders (1024 bytes) on the disk are reserved for the disk volume header.

If you need to repartition the disk we recommend that you modify the partitions listed under one of the partitioning strategies, leaving all other partitions intact. As an example, suppose you want to repartition the disk to obtain a 256 MB partition for data storage. Change the partitions listed under partitioning strategy 1 (partitions 3 and 4) as follows:

```
Command? (part nblks 1stblk type, or l) 3 262144 1080 sysv
Command? (part nblks 1stblk type, or l) 4 65536 263224 sysv
Command? (part nblks 1stblk type, or l) l

current contents:
        part     n_blks  1st_blk type     size
        0:        82080     1080 sysv      80 MB
        1:       108000    83160 sysv     105 MB
        2:       137970   191160 sysv     134 MB
        3:       262144     1080 sysv     160 MB
        4:        65536   263224 sysv     160 MB
        5:       328050     1080 sysv     320 MB
        6:        82080   247050 sysv      80 MB
        7:           -1       -1 sysv       0 MB
        8:         1080        0 volhdr     1 MB
        9:          540   329400 trkrepl    0 MB
        10:      329400        0 volume   321 MB
        11:         270   329130 diag       0 MB

Disk partitioning strategies:
 1:  0  (80 MB)  1 (105 MB)  2 (134 MB)
 2:  0  (80 MB)  1 (105 MB)  6  (80 MB)
 3:  0  (80 MB)  4 (160 MB)
 4:  3 (160 MB)  4 (160 MB)
 5:  5 (320 MB)
```

In the example above we have changed disk partitions 3 and 4 to create one 256 MB partition (partition 3) and 64 MB partition (partition 4). Partition 3 begins at block 1080 and goes for 262144 blocks (256 MB); partition 4 begins at block 1080 + 262144 = 263224 and goes for 65536 blocks (64 MB) to end at block 329400. Together they take up the 320 MB of available space.

NOTE
When you modify partitions you may get a warning about partitions not being cylinder-aligned. The warnings can be ignored.

Now that the partition table has been changed you can write the results and quit *dvhtool.*

```
Command? (read, vd, pt, dp, write, bootfile, or quit): w
Command? (read, vd, pt, dp, write, bootfile, or quit): q
WARNING The new partitioning does not take effect until the next
reboot of the system.
#
```

Make sure to reboot the system so that the new partition takes effect.

Disk Striping

Disk striping allows two (or more) disk partitions on separate disk drives to be combined into one logical disk partition. Disk striping is optional; it can, however, increase disk access rates by more than 50 percent. Disk striping is required if you wish to create logical disk partitions that are larger in size than your physical disks.

For best performance, striped disks should operate across separate disk controllers. The partitions you stripe across need not be the same size, but using partitions of the same size does avoid wasted space.

Use the command *mkfarm* to create a striped disk partition. The command takes all the special device files listed on the command line and combines them into one logical disk partition. For example, the command

```
# mkfarm 0 /dev/dsk/c2d0s5 /dev/dsk/c3d1s5
```

creates a disk striped configuration that spans the SMD disk drive partitions */dev/dsk/c2d0s5* and */dev/dsk/c3d1s5.*

The "0" in the command line refers to the name of the disk farm: */dev/farm/farm0*. ("1" creates */dev/farm/farm1* and so on.)

The disk farm must be configured each time the UNIX system is booted. To ensure this is done, add the *mkfarm* command (as above) to the file */etc/init.d/dfconfig*, which is executed upon system bootup.

The *–m* option to *mkfs* lets you "make" a file system for the disk striped partition. For example, the command

```
# mkfarm -m 1 /dev/dsk/c0d5s5 /dev/dsk/c1d4s5
```

creates a disk-striped file system across disk partitions 5 on Stardent 1500/3000's two optional internal SCSI disk drives. The special device name for the striped partition is */dev/farm/farm1*.

Use the *mkfarm* command with the *–m* option the when you first make the file system. This command need not be reissued every time the system is booted.

Once the disk farm is configured and the file system is made you can mount and use the file system. See *File System Administration* in this guide for information.

TERMINALS

CHAPTER FOURTEEN

Follow the steps in this chapter to attach a terminal to one of the Stardent 1500/3000 RS-232 ports and to configure the UNIX system for RS-232 terminal communications.

Physical Connection

Connect a terminal to one of the available RS-232 ports on one of the Stardent 1500/3000 System Module I/O boards (typically port A or B). Use an RS-232 "twisted" or "null modem" cable with minimum pin support as given in Table 14-1.

Table 14-1. Serial Connections

Stardent 1500/3000 Serial Pin	Terminal Pin	Definition
1	1	frame ground
2	3	transm data (rec'd data)
3	2	rec'data (transm data)
7	7	signal ground

Cables are available from the Inmac catalog. Although the Inmac catalog does not list a standard cable assembly that fits the above requirements, the cables listed in Table 14-2 can be used in conjunction with a shielded null-modem (Male to Female, Inmac part number 335-3) to complete the connection. Note that the Stardent 1500/3000 end requires a male connector.

Table 14-2. Serial Cables

Part Number	Type	Length
394-1†	Male-to-Female	10 feet
394-2†	Male-to-Male	10 feet
340-1†	Male-to-Female	15 feet
340-2†	Male-to-Male	15 feet
395-1†	Male-to-Female	25 feet
395-2†	Male-to-Male	25 feet
340-1†	Male-to-Female	35 feet
340-2†	Male-to-Male	35 feet

Terminal Configuration

Table 14-3 summarizes files and variables needed for terminal configuration. Tasks include assigning the correct special device

Table 14-3. Terminal Configuration [1]

Item	Function	Mod. Status
Commands:		
/usr/bin/tic	translate terminfo file to compiled format	
/usr/bin/captoinfo	translate termcap entry to terminfo descrptn	
/usr/bin/infocmp	compare two terminfo entries	
/usr/bin/tput	initialize terminal with terminfo database	
/usr/ucb/tset	terminal init-reset using /etc/termcap	
/usr/bin/vi	visual editor	
Files:		
/etc/termcap	terminal database	*
/usr/lib/terminfo/termcap	linked to /etc/termcap	
/usr/lib/terminfo	directory, compiled terminal database	
/etc/ioctl.syscon	tty settings for console	
~/.exrc[1]	default vi settings	*
/etc/inittab	actions for init process	**
/etc/gettydefs	actions for getty process	**
/dev/ttyXX	tty special device files	
Environment Variables:		
TERM	set to name of terminal in /etc/termcap	**
TERMCAP	contents of termcap entry, set by tset	*
EXINIT	default vi settings	*
EDITOR	choice of visual editor	**

file, checking to see if the terminal type is recognized, (and if not, creating a new terminal type), enabling the login port, and setting terminal environment variables and options.

Terminal Configuration Tasks

1. Check special device files. Make sure that a UNIX special device file has been created for the port. Remember that RS-232 Port A corresponds to the device tty0; RS-232 Port B corresponds to tty1, and so on. (See *File System Administration* in this guide for general information about special device files.) You can confirm the existence of the special device files by checking the */dev* directory as shown here for tty ports 0 and 1.

```
Stardent 1500/3000: ls -l /dev/tty0 /dev/tty1
crw--w--w-  1 root    root    25, 0 Apr 18 13:45 tty0
crw--w--w-  1 root    root    25, 1 Apr  8 15:31 tty1
```

2. Check the terminal type. A recognized terminal type is required to be able to use *vi*(1), *more*(1), *pg*(1), and *ls*(1) correctly. Check the file */etc/termcap* to be sure your terminal type is included. (Note that the file */etc/termcap* is hard linked to the file */usr/lib/terminfo/termcap;* the files are identical and can be treated as such. */etc/termcap* is the Berkeley UNIX version of the file; */usr/lib/terminfo* is the AT&T UNIX System V version.)

Here is an abridged version of */etc/termcap* which lists currently recognized terminal types (refer to */etc/termcap* itself for terminal parameters and comments). As you can see, many commonly used terminal types already have entries in */etc/termcap*.

```
xtls|xtermls|xterm without alternate screen
xtn|xterm|nvs|nvs100|new xterm terminal emulator (X window system)
tigr|tigr24|Titan console terminal
t4|tigr40|Titan console terminal
t6|tigr60|Titan console terminal
CT|ctrm
Ss|ss|SCALDstation|Valid SCALDstation
h2|2621|hp2621|hp2621a|hp2621p|2621|2621a|2621p|hp2621-fl|hp 2621
h6|hp2626|hp2626a|hp2626p|2626|2626a|2626p|hp 2626
```

† Must be used in conjunction with a null modem adapter if used on Stardent 1500/3000.

[2] The "modification status" column in the table shows whether or not the file requires modification. If the modification status box is empty, no modification is recommended. (Commands fall in this category.) If the box has a single asterisk (*), modification is optional. If the box has a double asterisk (**), the file or variable must be examined (and modified if necessary).

[3] ~ is C-shell terminology for the user's home directory.

```
la|adm3a|3a|lsi adm3a
h4|hp2645|2645|hp45|hp 264x series
Xa|tek|tek4012|4012|tektronix 4012
Xb|tek4013|4013|tektronix 4013
Xd|tek4015|4015|tektronix 4015
Xe|tek4014-sm|4014-sm|tektronix 4014 in small font
Xf|tek4015-sm|4015-sm|tektronix 4015 in small font
X5|4025|4027|4024|tek4025|tek4027|tek4024|4025cu|4027cu|tektronix 4024/4025/4027
X7|4025-17|4027-17|tek 4025 17 line window
X8|4025-17ws|4027-17ws|tek 4025 17 line window in workspace
X9|4025ex|4027ex|tek 4025 w/!
Xr|4025-cr|tek 4025 for curses and rogue
Xs|4112|4113|4114|tek4112|tektronix 4110 series
Xt|4112-nd|4112 not in dialog area:up=^K:ns:tc=4112:
Xu|4112-5|4112 in 5 line dialog area:li#5:tc=4112:
d0|vt100|vt100-am|vt100|dec vt100
d1|vt100|vt100-nam|vt100 w/no am
d2|gt42|dec gt42
d3|vt132|vt132
d4|gt40|dec gt40
d5|vt50|dec vt50
dI|dw1|decwriter I
dh|vt50h|dec vt50h
di|vt100-23|vt100 for use with vt100sys
ds|vt100-s|dec vt100 132 cols 14 lines (w/o advanced video option)
dt|vt100-w|dec vt100 132 cols (w/advanced video)
dt|vt100-w-nam|dec vt100 132 cols (w/advanced video)
df|dw4|decwriter IV
h1|2621-ba|2621 w/new rom, strap A set
h3|2621k45|hp2621k45|k45|2622|hp2622|hp 2621 with 45 keyboard
h5|hp|hewlett-packard
2626-23:li#23:tc=2626:
2626-12:li#12:tc=2626:
2626-12x40:co#40:li#12:tc=2626:
2626-x40:co#40:tc=2626:
2626-11:li#11:tc=2626:
h8|hp2648|hp2648a|2648a|2648|HP 2648a graphics terminal
hb|2640b|hp2640b|2644a|hp2644a|hp 264x series
hl|2621-48|48 line 2621
hn|2621-nl|hp2621-nl|2621|hp2621|hp 2621 with no labels
ht|2621-nt|hp 2621 w/no tabs
hw|2621-wl|hp2621-wl|2621|hp2621|hp 2621 w/labels
Mu|sun|Sun Microsystems Workstation console
su|dumb|un|unknown
pc|vt100pc|dec vt100
```

If you do need to make a new termcap entry or import a termcap from another system, proceed as follows.

(1) Create a "cap" file with parameters for the terminal. For a description of the required format, see *terminfo*(4) in the *Programmer's Reference Manual;* for examples, examine the */etc/termcap* file.

(2) Convert the file to an "info" file with the command

    # **captoinfo** *file.cap* > *file.info*

where *file.cap* is the name of the "cap" file and *file.info* is an output file.

(3) Compile the "info" file with the command

    # **tic** *file.info*

The compiled *info* file is automatically placed in a directory in */usr/lib/terminfo*. The name of the directory is the first letter of the terminal name. If you have created a new terminal type *unicorn*, the compiled *info* file for *unicorn* is automatically placed in the directory */usr/lib/terminfo/u*. The compiled *info* file is necessary for the system to recognize your terminal type. The *cap* file is needed for some Berkeley UNIX commands; you should append it to */etc/termcap*. The uncompiled *info* file need not be kept.

3. Enable the login port: the /etc/inittab file. Check to see if the file */etc/inittab* is configured correctly for the terminal port. The default */etc/inittab* file shipped with Stardent 1500/3000 contains lines similar to the following.

```
t0:23:off:/etc/getty tty0 9600
t1:23:off:/etc/getty tty1 9600
```

Find the line corresponding to the tty port you are using for the terminal, change the ACTION field from "off" to "respawn" to make the port active, and make the baud rate flag (9600 in the example) consistent with the terminal you are using. For instance, for tty port 1 and a 2400 baud terminal, change the "t1" entry to

```
t1:23:respawn:/etc/getty tty1 2400
```

Make sure that you have a */etc/inittab* entry specified for each terminal port. (*System Processes* in this guide contains a full description of the */etc/inittab* file.)

4. Check the /etc/gettydefs file. Stardent 1500/3000 is shipped with a default */etc/gettydefs* file such as the one given below.

```
console#B9600 CLOCAL#B9600 SANE TAB3 CLOCAL#titan login: #console

9600#B9600 CLOCAL#B9600 SANE TAB3 CLOCAL#titan login: #2400

2400#B2400 CLOCAL#B2400 SANE TAB3 CLOCAL#titan login: #1200
```

Check three things in this file.

(1) Make sure that an entry exists for the desired baud rate.

(2) For a hardwired terminal, make sure that parameter CLOCAL is shown in both the second and third pound-sign separated fields. For a modem connection (see *Modems* in this guide), make sure that *CLOCAL* does not appear, and that instead, the third field shows *HUPCL*.

(3) Change the fourth pound sign separated field, if you wish, to select the login prompt that users will see when they log in through the terminal port. Any prompt you select should end with

```
login:
```

5. Set terminal options. Terminal options are set with the *stty*(1) (set tty) command. A common way to do this to include *stty* commands in the *system profile* or in individual users' *.profile* and *.login* files. (See *User Services* in this guide for more on these files.)

Here are some common *stty* options.

echoe	instructs that characters erase when the backspace key is pressed.
erase ^h	defines the BACKSPACE key as erase.
kill ^u	defines CTRL-u as kill.

If you choose to define erase and kill, be aware that the operating system interprets # as erase and @ as kill, and that in certain rare instances you may need to revert to those keys to effect an erase or kill.

For a description of all the *stty* options see the *Commands Reference Manual.*

6. Set terminal environment variables. Make sure that the environment variable TERM is set correctly; otherwise *vi* and other visual editors may not work correctly.

If you are using the C-shell issue this command (or include the command in the user's *.login).*

```
setenv TERM terminal_type
```

where *terminal_type* corresponds to an entry in the */etc/termcap*. If you are using the standard shell issue these commands (or include the commands in the user's *.profile).*

```
TERM=terminal_type
export TERM
```

7. Logging in. The system is now ready to communicate with the terminal over its RS-232 port. Check your terminal manual for terminal-specific configuration. Then when you power-on the terminal you should see the familiar

```
login:
```

prompt. You may now log in to the system.

8. Problems. If you see messages such as

```
unrecognized terminal type
```

or have problems using *vi, more, pg, ls,* or other commands that require knowledge of terminal type, make sure that the terminal type is present in */etc/termcap* and check the values of environment variables EDITOR, and EXINIT. (See *sh*(1) in the *Commands Reference Manual* for more on these variables.)

MODEMS

CHAPTER FIFTEEN

This chapter describes how to connect a Stardent-supplied modem (Part Number 130-0032-02) to the I/O board. The modem can be used either for remote user login or for remote diagnostics. For remote user login purposes the modem should be connected to one of the available RS-232 ports on the System Module I/O board. For remote diagnostic purposes the modem should be connected to the Diagnostic Communications Port (DCP) on the boot master CPU board. For instructions on connecting the modem to the DCP see *Diagnostic Communications Port* in this guide. For information on using the modem in remote diagnostics mode refer to the *Field Service Manual.*

The Stardent-supplied modem uses internal NVRAM settings to handle its configuration. There are no *DIP* switches to manipulate. When you receive the modem, two pre-set NVRAM configurations are included: one for dialing out and one for dialing in. When you first install and power up the modem, it is set to dial-in mode. Details of the configuration settings are given in *Connecting the Modem,* later in this chapter.

RS–232 Cable Description

An RS-232 cable assembly is required to connect a modem to the DCP or I/O board. (The cable must have a male connector at the Stardent 1500/3000 end, and most modems require a male connector at the modem end.) **To avoid problems with cable compatibility use Stardent part number 100-0095-01.** The Stardent cable carries 9 wires with pins 1 through 8 and pin 20 connected straight through. Table 15-1 lists the exact number of pins required.

CAUTION
Do not use a cable with all 25 pins wired straight through (pin-1 to pin-1, pin-2 to pin-2 ... pin-25 to pin-25).

Table 15-1. RS-232 Modem Cable Specification

Pin	Name	Usage
1	Frame Gnd	Protective Ground
2	TD	Transmitted Data
3	RD	Received Data
5	CTS	Clear To Send
7	Common	Signal Ground
8	DCD	Carrier Detect
20	DTR	Data Terminal Ready

Connecting The Modem

CAUTION
When connecting the modem, make sure its power switch is **OFF**. For certain modem switch settings, if Stardent 1500/3000 has a login process (a "getty" program) running on the Diagnostic Control Port (DCP), the modem confuses the operating system and causes it to crash.

It is not necessary to power down your Stardent 1500/3000 system to make the modem connection.

Follow these steps to connect the modem to your Stardent 1500/3000 system:

Step 1. Place the modem power switch in the **OFF** position.

Step 2. Plug the modem power supply into the wall and connect the power supply cable to the back of the modem.

Step 3. Connect the 4-wire telephone cable from your local telephone line to the "Wall" receptacle on the back of the modem. The receptacle labeled "Phone" is for a telephone that can be used on the same line when the modem is not being used.

Step 4. Connect the Stardent–supplied RS-232 cable to the 25-pin connector on the back of the modem. If you plan to use the modem for remote user login, attach the other end to one of the available RS-232 ports on the System Module I/O board (port A or B). If you plan to use the modem for diagnostic purposes attach the cable to the DCP on the boot CPU board. (If you have multiple CPU boards you can identify the boot CPU by examining the LEDs near the top of each board. The CPU whose light bar continuously scans up and down is the boot CPU.)

Step 5. Use the screw attachments on the RS-232 cable connectors to secure the cables.

Step 6. **Do not turn on the modem power switch** until you complete the instructions in "Configuring the Modem Port" below. You should also complete the instructions in "Testing the Modem" below before you power-on the modem for normal use.

Configuring the Modem Port

Before the modem can be used you must correctly configure the system for communication with the modem port. The procedure is different for DCP port versus I/O board connections. Follow the procedure here for I/O board connections; see *Diagnostic Communications Port* in this guide for DCP configuration instructions.

I/O Board Port Configuration

Assume for this section that you have connected the modem to RS-232 port B on the I/O board. The associated UNIX device name is *tty1*. (RS-232 port A is associated with *tty0*; RS-232 port C is associated with *tty3* and so on.)

Step 1. First make sure that your system's *Devices* file contains information about the port and modem. Become superuser, edit the file */usr/lib/uucp/Devices* and add the following lines (if they are not already present in the file).

```
ACU tty1 - 2400 Hayes
ACU tty1 - 1200 Hayes
```

(Substitute *tty0* if the modem is connected to RS-232 port A on the I/O board.)

NOTE
When the modem is used for dial-out purposes, the *cu* (call UNIX) program reads the Devices file from the beginning, looking for the first ACU line that has a matching baud rate. If the associated modem is not busy it is the modem that is used. Depending upon your phone connections you may wish to use one modem more often than others. If so place the preferred modem's ACU line ahead of the others in the file.

Step 2. Check to see whether the file */dev/tty1* is owned by *uucp*. Issue this command:

```
# ls -l /dev/tty1
```

(Substitute *tty0* if the modem is connected to RS-232 port A on the I/O board.) If the system responds with a line that shows *uucp* ownership of */dev/tty1*, the node is already properly configured.

```
crw--w--w   2 uucp  unix   25, 1  May 17 08:43 /dev/tty1
```

If not, as superuser change the ownership of */dev/tty1*:

```
# chown uucp /dev/tty1
```

Step 3. Modify the */etc/inittab* file to accommodate the modem. As superuser edit the file and locate the line that begins with "t1." ("t0" if port A on the I/O board is used). First, check the ACTION entry (the third colon-separated field in the line). For remote login purposes the entry should be set to "respawn." To test the modem, which we highly recommend, set the action to "off." Then add the correct modem baud rate flag to the end of the line. The resultant */etc/inittab* entry looks like this for testing mode (2400 baud operations):

```
t1:2:off:/etc/getty tty1 modem2400
```

and looks like this for remote login mode and 1200 baud operations:

```
t1:2:respawn:/etc/getty tty1 modem1200
```

Testing The Modem

Before you use the modem, test it to see if it can respond to commands. Follow this procedure.

Step 1. Make sure that UNIX is up and running on your system and that you have followed the procedures in "Modem Port Configuration" above in this chapter. Make sure that the */etc/inittab* entry for the modem port shows ACTION "off":

```
dcp:234:off:/etc/getty dcp modem2400
```

or

```
t1:2:off:/etc/getty tty1 modem2400
```

Step 2. As superuser type the command

```
# init q
```

The command forces the init process to reread the inittab file, killing the *getty* process that currently owns the port. This allows you to use the port for outgoing communications.

Step 3. Turn on the modem.

Step 4. In one of your X-Windows, connect to the port via the command:

```
# cu -s2400 -ldcp
```

for DCP connections or

```
# cu -s2400 -ltty1
```

for I/O board connections. This command sets the port speed to 2400 baud and connects your terminal process to the modem port through its special device node.

Step 5. Type the letters **AT** and press RETURN. The modem should respond **OK**.

Step 6. Now, display the NVRAM settings by typing

```
AT&V
```

To use the modem for *remote diagnostics* or *remote login,* the NVRAM configuration must be set as shown in Table 15-2. This configuration is set at the factory as the default setting when you turn on the modem.

Table 15-2. Modem Switch Settings for Remote Diagnostics or Login

Switch Description	Setting	AT Command
DTR signal	active	AT&D1
Result Codes	OFF	ATQ1
Command Echo	OFF	ATE0
Auto Answer	ON	ATS0=2*
Carrier Detect	Active	AT&C1

* The 2 specifies the number of rings before the modem answers.

Step 7. Now type the following commands:

```
AT&F
```

```
AT&V
```

The first command sets the modem to dial-out mode; the second displays the configuration settings. To use the modem in dial-out mode (this is also called *test mode*), the NVRAM configuration settings must be set as shown in Table 15-3.

Table 15-3. Modem Switch Settings for Dial-Out

Switch Description	Setting	AT Command
DTR signal	ignored	AT&D0
Result Codes	ON	ATQ0
Command Echo	ON	ATE1
Auto Answer	OFF	ATS0=0
Carrier Detect	True	AT&C0

Step 8. To disconnect from the modem press the RETURN key, then type:

```
~.
```

You have now established that the modem can communicate with the DCP or I/O board port. If you have any problem completing the steps in this section, check that the cables are properly connected and repeat these steps.

Step 9. Testing is now complete.

Setting Up for Remote Diagnostics and Logins

To use the modem for remote diagnostics and logins, edit the */etc/inittab* file and change the port action to "respawn." Stardent Customer Support or other users can now remotely login to the system. Cycle the power to the modem. Unless you alter the default settings, the modem initializes to dial-in mode every time you turn it on.

You can always select the dial-out settings (shown in Table 15-3) irrespective of any changes by issuing the **AT&F** command.

The dial-in parameters (shown in Table 15-2) should not be changed. However, if you do change them and then want to use the modem for remote diagnostics or logins, you need to reset and save them. Use **AT&V** to check the settings. Reset those that don't match Table 15-2. Save the settings in NVRAM as "Profile 0" by issuing the following command:

```
AT&W0
```

To install Profile 0 as the default power-on setting, enter the following command:

```
AT&Y0
```

To use the modem for remote diagnostics, consult the *Field Service Manual*.

need to add new users to the system, consult the *System Administrator's Guide.*

To use the modem for remote diagnostics, consult the *Field Service Manual*.

PRINTERS

CHAPTER SIXTEEN

This chapter describes how to connect a Centronics-compatible parallel printer or an Apple LaserWriter™ to Stardent 1500/3000 and how to configure the *lp spool* program.

Making the Physical Connection

Parallel (Centronics compatible) printer. To connect a parallel printer, connect a parallel cable to the printer port of the I/O board.

The cable for a Centronics compatible printer is a 36-pin shielded cable with all pins wired straight through (pin 1 of one end wired to pin 1 of the opposite end connector and so on). A male cable end is required at the Stardent 1500/3000 I/O board end. A male or female connector may be required at the printer or other accessory end, so be sure to check your printer equipment or documentation before purchasing a cable.

Table 16-1 lists Inmac part numbers and lengths for various types of Centronics compatible cables.

Table 16-1. Centronics Cables

Part Number	Type	Length
323-1	Male to Male	10 feet
323-2	Male to Male	20 feet
323-3	Male to Male	30 feet
324-1	Male to Female	10 feet
324-2	Male to Female	20 feet
324-3	Male to Female	30 feet

Serial Printer (Apple LaserWriter™). To connect a serial printer, use RS-232 port A or B on the I/O board.

To connect an Apple LaserWriter™, use a Male-to-Male RS-232 cable that has only the pins described in Table 16-2.

Table 16-2. Serial Printer Connections

Stardent 1500/3000 Serial Pin	Printer Pin	Definition
1	1	frame ground
2	3	transm data (rec'd data)
3	2	rec'data (transm data)
7	7	signal ground

Although the Inmac catalog does not list a standard cable assembly that fits the above requirements, the cables listed in Table 16-1 can be used in conjunction with a shielded null-modem (Male to Female, Inmac part number 335-3) to complete the connection.

RS-232 pin assignments vary across printers, so be sure to check your printer manual for its unique requirements. For your information, Table 16-3 lists the complete set of signals for the Stardent 1500/3000 serial port to allow you to determine your own cabling needs.

Table 16-3. Pin Assignments For Serial Connectors

Connector Pin	Signal Name	I/O Type
1	protective ground	
2	transmit data	output
3	receive data	input
4	request to send	output
5	clear to send	input
6	data set ready	input
7	signal ground	
8	carrier detect	input
11	receiver reference	
15	transmit clock	input
17	receive clock	input
18	transceiver reference	
20	data terminal ready	output
22	ring indicator	input

Table 16-4 gives commands and files needed for printer and line printer spooling (*lp spool*) configuration.

Table 16-4. Printer Configuration [1]

Item	Function	Mod. Status
Commands:		
/usr/lib/lpadmin	lp spool configuration	
/usr/lib/lpsched	start lp scheduler	
/usr/lib/lpshut	stop lp scheduler	
/usr/lib/lpmove	move lp spool requests	
/usr/bin/lpstat	print lp status information	
/usr/bin/accept	accept lp requests	
/usr/bin/cancel	cancel lp requests	
/usr/bin/enable	enable lp printer	
/usr/bin/disable	disable lp printer	
/usr/bin/reject	prevent lp requests	
Files:		
/etc/inittab	actions for init process	**
/usr/spool/lp/FIFO	special file used by lp commands	
/usr/spool/lp/default	default destination for lp requests	*
/usr/spool/lp/log	record of print activity since last start	*
/usr/spool/lp/oldlog	info from previous log	
/usr/spool/lp/outputq	binary file of lp output requests	
/usr/spool/lp/pstatus	binary file of printer status	
/usr/spool/lp/qstatus	binary file of destination status	
/usr/spool/lp/seqfile	sequence number of last lp request id	
Directories:		
/usr/spool/lp/interface	executable interface programs	*
/usr/spool/lp/member	pathnames to printer devices	*
/usr/spool/lp/model	model printer interface programs	
/usr/spool/lp/request	used during lp requests	
Special Device Files:		
/dev/ttyXX	printer device files	
/dev/lp	pointer to serial printer device driver	
/dev/centronix	pointer to Centronics printer device driver	

[1] The "modification status" column in the table shows whether or not the file requires modification. If the modification status box is empty, no modification is recommended. (Commands fall in this category.) If the box has a single asterisk (*), modification is optional. If the box has a double asterisk (**), the file or variable must be examined (and modified if necessary).

Printer Configuration: Tasks

1. Check /etc/inittab. If you are using a serial printer, make sure that no *getty* process is spawned on the port you plan to use. Check the file */etc/inittab* file for the appropriate tty entry, and make sure that the ACTION is "off". For a serial printer connected to port A on the system I/O board the */etc/inittab* entry should look like this

```
t0:23:off:/etc/getty tty0 9600
```

(Check the line beginning with "t1" if you have connected the printer to port B.)

2. Link printer device to port. Link the printer device file with the port you plan to use for the printer. Issue the command

```
ln /dev/lp /dev/centronix
```

for a parallel printer and

```
ln /dev/lp /dev/tty0
```

for a serial printer (refer to the tty port to which you have physically connected the printer).

3. Check /etc/init.d/lp. Check to see that the following lines in */etc/init.d/lp* are uncommented and that they correspond to the port being used for the printer:

```
cat -u /dev/tty0 >> /usr/adm/PS.log &
rm -f /usr/spool/lp/SCHEDLOCK
/usr/lib/lpsched
echo 'lp spooler' > /dev/console
```

(Substitute "tty1" if appropriate for a serial printer and */dev/centronix* for a parallel printer.)

4. Dumb printer: check printer output. If you have connected a dumb printer, you can test the connection by "catting" a file to the device */dev/lp*. For instance,

```
# cat /etc/inittab >/dev/lp
```

If the output is identical to what you get when you use *cat* to print the file on the screen, you can proceed to setting up the *lp spool* program. If not, find the symptom in the following list:

Double-spaced output	Locate the CR/LF switch on your printer, if present, and flip it to the alternate position. If there is no such switch, try setting different flags to the *stty* command. The flags to try are *onlcr*, *–onlcr*, *nl*, and *–nl* in several variations until the output is correct.
Output all on one line	This is the opposite problem from the one just described. Try the same things.
Garbage output	Check the baud rate setting. Set the baud rate to the correct setting and turn the printer power off and on again.
Missing output	This indicates a handshaking problem. Make sure the *ixon* and *–ixany* flags are set on the *stty* command line.
No output	Check everything, especially the */etc/inittab* entry to be sure you have inhibited a getty on the printer port.

Lp Spool: Tasks

The *lp spool* program is capable of handling a wide variety of printers and printer ports. When you print a file, the file is stored in the */usr/spool/lp* directory until it is passed along to the actual device that does the printing.

1. Install Documenter's Workbench or printer driver. If you plan to use a Postscript™ compatible laser printer we recommend that you purchase the Documenter's Workbench (DWB) software package from Stardent Computer Corporation. In addition to *troff* text formatting, DWB includes the devps package, which translates formatted (*ditroff*) output into Postscript™ form. If DWB is not purchased you may need to install a printer driver and utilities according to the instructions shipped with your printer software.

If you have ordered DWB as a software update, see *Software Installation* in this guide for installation instructions.

2. Stop the lpsched daemon. Use the *lpshut* command to stop the *lpsched* daemon. You must stop the daemon before issuing most lp administration commands. (The daemon is started by the */etc/rc2* script, discussed in *System Processes* in this guide.)

```
# lpshut
```

3. Log in as lp. A special system login, *lp*, exists for *lp spool* administration. Log in as *lp* before proceeding. If you have a problem doing so, check the */etc/passwd* file. Make sure that *lp* is listed as a user. If you are logging in as *lp* for the first time, make sure to choose a password for the account.

4. Configure the lp spool program. Use the *lpadmin* command to configure the *lp spool* program. To attach a Postscript™ printer to */dev/tty0*, type

```
# lpadmin -pPS -v/dev/tty0 -mPS
```

The *–p* option in the *lpadmin* command names the printer. The *–v* option names the logical device attached to the printer (tty0 in the example. The *–m* option specifies a printer interface program to use with the printer.

Attach Centronix-compatible parallel printers to */dev/centronix* and Tektronix 4693D thermal transfer printers to */dev/tek*. For instance, to configure a Tektronix 4693D thermal transfer printer, type

```
# lpadmin -ptek -v/dev/tek -mtek
```

Next use the *lpadmin* command with the *–d{printername}* option to define a default printer for the system. For example, to establish the Postscript™ printer as the default printer for the system, type

```
# lpadmin -dPS
```

5. Printer Interface Programs. Every printer used with the *lp* spool program must have a *printer interface* program (shell script). The script may be simple or complex, depending upon the printer and the functions you desire. Every print request is routed through the appropriate printer interface program before the request is printed.

The file */usr/spool/lp/interface/PS* (part of Documenter's Workbench) contains the printer interface program for the Postscript™ family of printers and typesetters. Model printer interface programs for other printers are contained in the */usr/spool/lp/model* directory.

Here is a listing of currently available model printer interface programs.

Program Name	Printer
PS	Postscript™ printers and typesetters
tek	Tektronix 4693D color plotter
1640	DASI 1640
5310	AT&T 5310/20 Matrix Printer
dqp10	DQP-10 Matrix Printer
dumb	line printer (dumb)
f450	DASI 450
hp	HP 2631a line printer
lqp40	LQP-40 Letter Quality Printer
ph.daps	Autologic APS-5 phototypesetter
pprx	Printronix line printer (parallel interface)
prx	Printronix line printer

The printer interface programs are shell scripts that can be modified to accommodate other printers. For instance, the following printer interface script allows Stardent 1500/3000 users to use a printer attached to another machine on the network. The script is written for *lpr* output; minor modification can make it work for *troff* output.

```
# lp interface for remote (rsh to stdin) line printer
#
# You will need to setup an account for lp on the remote system
# as this script runs as the lp user on the titan.
#
MACHINE_WITH_PRINTER=remote machine
PRINTER_COMMAND="/usr/ucb/lpr"
#

copies=$4
shift; shift; shift; shift; shift
files="$*"
i=1
while [ $i -le $copies ]
do
        for file in $files
        do
                cat "$file" | /usr/ucb/rsh $MACHINE_WITH_PRINTER $PRINTER_COMMAND
        done
        i=`expr $i + 1`
done
exit 0
```

If you wish to use a printer not on the list and do not wish to write a custom interface script, just use the dumb printer interface:

```
# lpadmin -pdumb -v/dev/tty0 -mdumb
# lpadmin -ddumb
```

6. Enable the printer. To enable the printer use the *accept*(1M) and *enable*(1M) commands with the printer name as the argument:

```
# accept PS
# enable PS
```

The printer should now be ready to accept jobs. If you have problems, see the next section in this chapter.

7. Clean log files. The *lp spool* log files, */usr/spool/lp/log* and */usr/spool/lp/oldlog*, tend to grow rapidly. Schedule cleanup by putting the following commands in a *crontab* file:

```
0 23 * * 5 /bin/su lp -c "cp /usr/spool/lp/log /usr/spool/lp/oldlog"
1 23 * * 5 /bin/su lp -c "cp /dev/null /usr/spool/lp/log"
```

Printers, Lp Spool: Troubleshooting

If you have trouble generating printer output, check this list.

(1) Program rejects additional print requests: Stop the spooler from receiving additional requests with the *reject*(1M) command:

```
# reject PS
destination "PS" is no longer accepting requests"
```

Start the printer again when ready with the *accept* command.

```
# accept PS
destination "PS" now accepting requests"
```

(2) Need to disable output: To disable output, yet leave the spool open to receive additional requests, use the *disable*(1M) command as follows:

```
# disable PS
printer "PS" now disabled
```

To permit output once again, use the *enable* command.

```
# enable PS
printer "PS" now enabled
```

(3) No output: If everything works up to this point but no output is generated, check this list.

(i) outputq: Ensure that the file */usr/spool/lp/outputq* exists.

*(ii)*permissions: Ensure that */usr/spool/lp* and all its subdirectories and files are owned by *lp*.

*(iii)*SUID Ensure that all spool programs have the Set-UID bit on and are owned by lp:

```
# ls -l /usr/lib/lp
-rwsr-xr-x  1 root      81920 Sep  2 10:30 /usr/lib/lp
```

If the set-uid mode is not enabled, see *chmod*(1) in the *Commands Reference Manual.*

(iv) lpsched,lpadmin: Ensure that the *lpsched* and *lpadmin* programs (located in */usr/lib*) are owned by root.

(v) No Daemon?: Make sure the daemon, */usr/lib/lpsched,* is running with the following command:

```
# lpstat -t
```

If it is not running, start it:

```
# /usr/lib/lpsched
```

DIAGNOSTIC COMMUNICATIONS PORT

CHAPTER SEVENTEEN

The Diagnostics Communications Port (DCP) is a special RS-232 port that is used for hardware diagnostics and as the console for the Stardent 1500/3000 Server. The DCP is located on the CPU board and may be configured with a terminal or modem. For multiple CPU board configurations attach the modem or terminal to the DCP on the *boot master* CPU board. (See *System Overview* in this guide for identification of the boot master CPU board.)

DCP Terminal Connections

Follow these steps to connect a terminal to the DCP.

1. Cabling. Cable the terminal to the DCP according to the instructions in the *Terminals* chapter of this guide.

2. Special Device File. The UNIX special device file for the DCP, */dev/dcp,* has already been created. If you wish you can confirm its existence as shown here:

```
TITAN: ls -l /dev/dcp
crw--w--w   1  root     root       22,    0  Apr 18 13:45 dcp
TITAN: ls -l /dev/dcp
```

3. Terminal Type. Follow the instructions in *Terminals* to check the terminal type. We recommend that you use a vt100-compatible terminal.

4. /etc/inittab. Edit */etc/inittab* and change the action in the DCP entry to "respawn":

```
dcp:234:respawn:/etc/getty dcp 9600
```

Also check to see that the baud rate flag (9600 here) corresponds to the baud rate of the terminal or modem being used (as specified in */etc/gettydefs).*

Terminal Options. Set terminal options and environment variables according to the instructions in *Terminals*.

DCP Modem Connections

Follow these steps to configure the DCP for communications with a Hayes-compatible modem.

1. Physical Connection. Follow the instructions in *Modems* to cable and set the modem switches.

2. Become superuser, edit the file */usr/lib/uucp/Devices* and add the following lines (if they are not already present in the file).

```
Direct dcp - 2400 Hayes
Direct dcp - 1200 Hayes
ACU tty1 - 2400 Hayes
ACU tty1 - 1200 Hayes
```

NOTE
When the modem is used for dial-out purposes, the *cu* (call UNIX) program reads the Devices file from the beginning, looking for the first ACU line that has a matching baud rate. If the associated modem is not busy it is the modem that is used. Depending upon your phone connections you may wish to use one modem more often than others. If so place the preferred modem's ACU line ahead of the others in the file.

3. Check to see if the */dev* directory has a special device node for the dcp port that is owned by *uucp*. Issue this command:

```
# ls -l /dev/dcp
```

If the system responds with a line such as the following one that shows "dcp" with ownership "uucp," the node is already configured.

```
crw------   2 uucp  unix  22, 0  May 17 08:43 /dev/dcp
```

If not, as superuser issue this command:

```
# chown uucp /dev/dcp
```

You can now confirm that the correct special device node has been created:

```
# ls -l /dev/dcp
crw------   2 uucp  unix  22, 0  May 17 08:43 /dev/dcp
#
```

4. Modify the */etc/inittab* file to accommodate the modem. As superuser edit the file and locate the line that begins with "dcp." Change the ACTION to "respawn" and add the correct modem baud rate flag to the end of the line. For a 2400 baud modem the line should read

```
dcp:23:respawn:/etc/getty dcp modem2400
```

You are now ready to test the modem connection. Follow the instructions in the chapter entitled *Modems*.

Stardent 1500/3000 Server: Using a DCP Terminal as System Console

If you have a Stardent 1500/3000 Server, you must configure a DCP terminal to be used as the system console. Do the tasks here as part of setting up your DCP configuration.

1. Set NVRAM Variables.

- Check to see that the variable *console* is defined and has the value

```
console=dcp
```

- Check to see that the variable *baud* is defined as

```
baud=9600
```

2. The /etc/inittab File. Check the */etc/inittab* file. The console must be set to dcp, and if a dcp entry exists, the action should be set to "off:"

```
co:23:respawn:/etc/getty dcp 9600
dcp:23:off:/etc/getty dcp 9600
```

STEREO MONITOR

CHAPTER EIGHTEEN

This chapter describes how to install the Stardent 1500/3000 stereo monitor which can be used in place of the standard monitor. Topics include

NOTE
You must have software release 2.0 or later to use the stereo monitor in stereo mode.

- The hardware components which are part of the Stereo Viewer.
- How the Stereo Viewer hardware works.
- How to install the Stereo Viewer hardware.
- How to run software which uses the Stereo Viewer.
- How to write software which uses the Stereo Viewer.

Unpacking

If you ordered a stereo monitor as part of your original Stardent 1500/3000 System, the monitor is shipped on a large cushioned pallet along with the System Module. Cables for the monitor are contained in the accessory box shipped on the same pallet. To unpack the monitor, remove the box from the top. Two people can then lift the monitor off the pallet.

Stereo Monitor Components

Locate the following stereo monitor components in the monitor and accessory boxes.

- Stereo monitor
- LCD shutter
- Stereo viewing glasses
- 50 or 200 foot Stereo Interface Cable
- 36-inch coaxial cable
- Short AC cord

Installing the Stereo Monitor

Follow these steps to install the stereo monitor.

CAUTION
Follow this step carefully to avoid damage to the LCD shutter assembly or to the stereo monitor.

Step 1: Install the LCD shutter. Unpack the LCD shutter assembly and grasp it with both hands at opposite sides of the shutter assembly. The assembly should be held so that the lettering on the front can be read. See Figure 18-1.

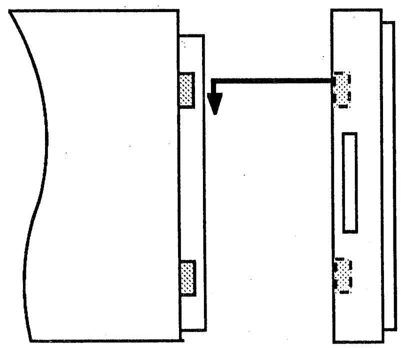


Figure 18-1. Stereo Shutter Installation

NOTE
If you are installing the stereo monitor at the same time you install your Stardent 1500/3000 system, we recommend that you tape the Stereo Interface Cable to the standard Stardent 1500/3000 User Interface Cable with electrician's tape so the two cables can be routed together through ceilings, floors or ductwork. See the chapter entitled *Installing the Stardent 1500/3000 Graphics Supercomputer* for details.

Carefully align the shutter assembly so that the pin and latch assemblies (two on each side) align with each other. Notice the 4-finger circuit board in the upper lefthand corner of the shutter assembly. If your alignment is proper, this circuit board is seated in its corresponding connector upon installation.

Step 2: Connect Stereo Interface Cable. Figure 18-2 shows stereo cable connections between the Stardent 1500/3000 System Module and the User Interface Module (junction box and stereo monitor). Connect the 50 or 200-foot Stereo Interface Cable, to the "Aux Video" port on the graphics board. Attach the other end of the cable to the "Stereo In" port on the junction box.

Step 3: Connect coaxial cables. Four 36-inch coaxial cables with BNC connectors are needed to connect the junction box to the stereo monitor. Three of the cables supply RGB signals. Connect the "red out", "green out", and "blue out" connectors on the junction box to the corresponding "red", "green", and "blue" inputs on the Stereo Monitor. Connect the remaining cable to the "stereo out" port on the junction box and to the unmarked port on the Stereo Monitor. Refer to Figure 18-2.

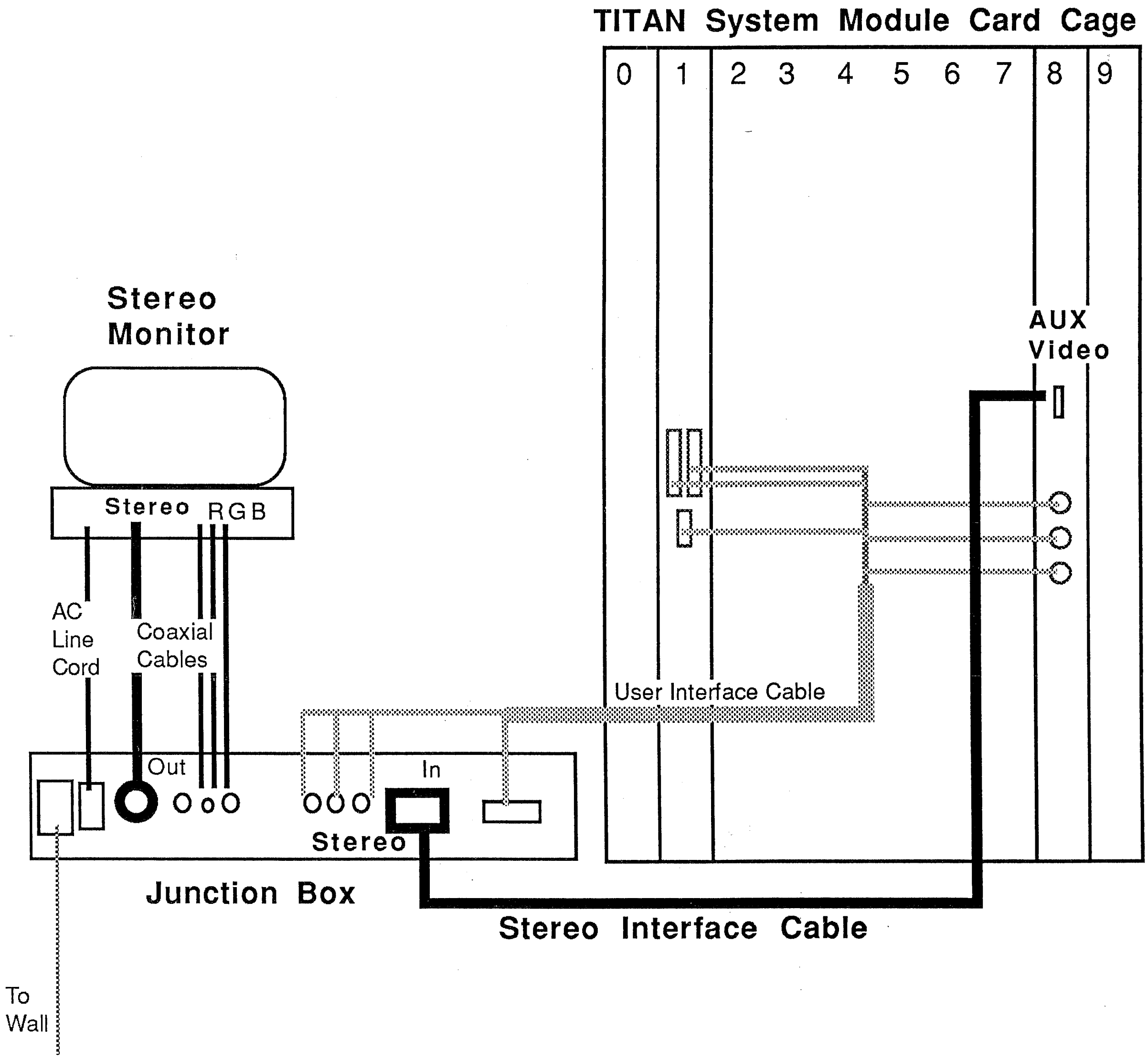


Figure 18-2. Stereo Cable Installation

Step 4: Connect AC cord. Connect the short AC line cord to the receptacles on the back of the stereo monitor and the junction box.

The stereo monitor is now ready to be used as a conventional (non-stereo) monitor. The following sections describe how the stereo monitor in produces a stereo effect and how to write and run software that uses the stereo monitor.

How The Stereo Hardware Creates a Stereo Effect

The shutter assembly that is installed on the faceplate of the stereo monitor contains two high-speed LCD shutter glass panes. In the absense of a stereo switching signal, both of these glass panes are effectively transparent.

Each of the LCD shutter panes can create a window-blind effect on the screen. One shutter creating vertically oriented "blinds" and the other shutter creating horizontally oriented blinds, letting through the light from the monitor in only that specific direction. Figure 18-3 shows the LCD shutter and the glasses as though each forms a fence through which the light must pass. This effect is called vertical or horizontal polarization.

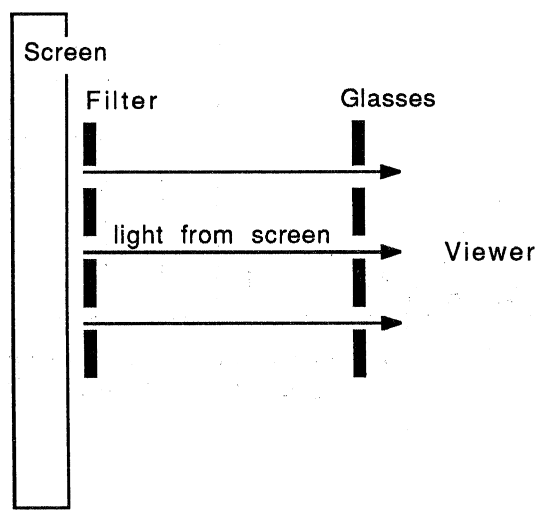


Figure 18-3. Light Transmission Through Polarizing Filters

NOTE
Because this filter cannot block 100% of the light, there will be a slight ghost of the alternate frame reaching both eyes. The stereo effect normally overwhelms the much dimmer "flat" ghost alternate image, making it easy to ignore it.

The stereo viewing glasses have this same polarizing characteristic, wherein the lens of one eye is positioned to view light that is polarized in a vertical direction, and the lens of the other eye is positioned to view light that is polarized in a horizontal direction.

In effect, this is a second set of blinds, installed on the people viewing the screen, that effectively block light polarized horizontally from being seen by the eye that is covered with the vertically polarized lens. The combination of a horizontal–blind and a vertical–blind offers the effect of a cross-hatch through which little light passes. Figure 18-4 illustrates the effect of combining the LCD shutter with the glasses, and shows how the light transmission is affected.

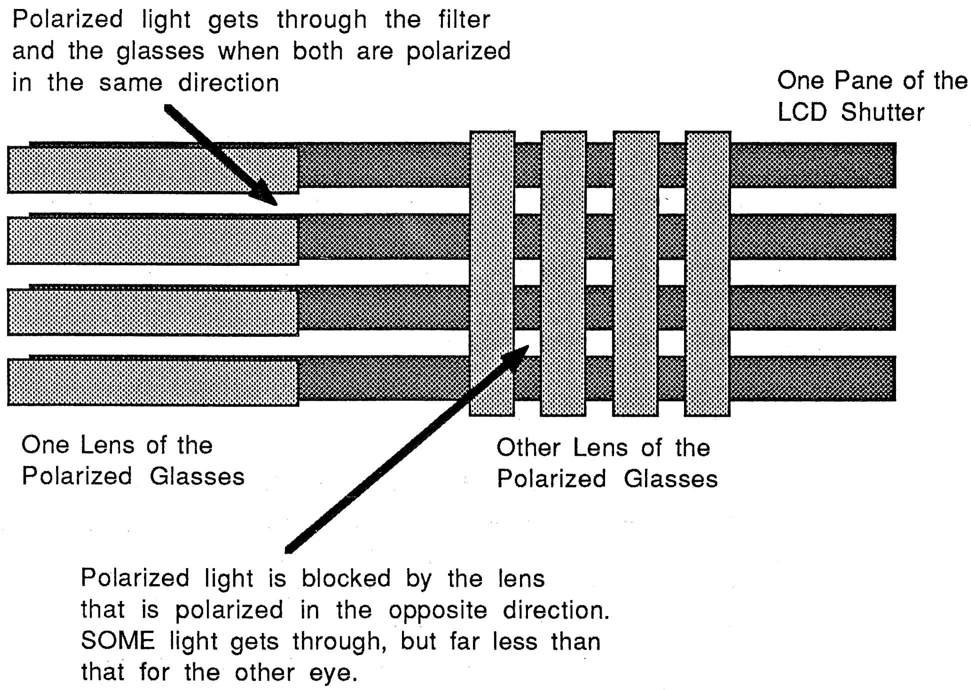


Figure 18-4. Effects On Polarized Light Viewed Through Polarizing Filters

The system presents the two images, one to each eye, in alternating display frames, images created as though viewed from two different angles according to eye separation and distance from the viewer.

For one frame time, the horizontal polarizing shutter is energized on the screen; the eye that has the matching horizontally polarized lens sees the screen image it is supposed to see while the other eye sees nothing.

For the subsequent frame time, the vertical polarizing shutter is energized, the image for the other eye is presented, and the eye that is covered by the vertically polarized lens sees the image it is supposed to see while the other eye sees nothing. The viewer's brain combines the images, completing the stereo effect.

Running Stereo Viewer Software

If you already have software that was designed for Stereo Viewer operation, or you have written your own software according to the instructions in the next section, this section will help show you how to run the software.

Vendor Supplied Application Stereo Viewer Software

If you are running Stereo Viewer software supplied by another vendor, be sure to follow the instructions supplied with the software you received. They should include detailed instructions for running the application software with the Stereo Viewer. You may also wish, however, to read the following sections to familiarize yourself with standard procedures for running software using the Stereo Viewer.

Starting up X11+ in Stereo Mode

X-Windows must be started in stereo mode, as it is not possible to operate the Stereo Viewer with X-Windows in non-stereo mode. To start up X-Windows in stereo one add the *-stereo* option to the *xinit* line, as shown below. For example, many people use the "xstart" script to start X-Windows. For stereo operation the "xstart" script must have an additional parameter "-stereo" added just after the word *Xtitan* in the last line of the script. A sample "xstart" script is shown below, with the parameter added as specified.

```
EXEC="awm -e"
DISPLAY=unix:0
export DISPLAY
xinit -keepserver $EXEC $HOME/.xdesktop -- Xtitan -stereo
```

After xstart is executed, there will actually be two screens created: a normal screen on which your normal window manager is opened, and a stereo screen, currently not displayed.

Opening a Window on the Stereo Screen

To start an Stardent Window Manager on the stereo screen, use the following command from any one of your xterm windows on the normal screen:

```
(setenv DISPLAY unix:0.1; awm &)
```

To start an xterm window on the stereo screen, you must use the command line:

```
xterm -display unix:0.1 &
```

(with any other parameters for xterm).

Switching Between Normal and Stereo Screens

To bring the stereo screen into view, press and hold the ALT key on the Stardent 1500/3000 keyboard, and then press and release the F2 key. Finally release the ALT key. This causes the stereo screen to be displayed. Pressing ALT-F2 again reverses the process, and the normal screen is then visible again.

Viewing in Stereo

Once you have switched to the stereo view, be sure to wear the stereo glasses supplied with the monitor to see the stereo effect. Wearing the glasses while the normal screen is being displayed is not recommended, as screen flicker will be seen.

Exiting from Stereo X-Windows

Exit from X-Windows from the normal screen. Do not exit from X-Windows while the stereo screen is being displayed, or the screen will not be left in the high resolution mode. If you do this by mistake, restart the X server (using xstart or any other equivalent technique) to restore the proper screen display parameters.

Writing Stereo Viewer Software

You must have Release 2.0 software or later to use the Stereo Viewer.

Software written to take advantage of Stereo Viewer operation must use both the X+ window system and the Doré three-dimensional graphics subroutine package.

Shown below is a complete example of a program which uses the stereo viewer.

The code utilizing calls specific to operation of the Stereo Viewer are in the X window setup calls:

- *XBCreateStereoWindow* in the example routine *create_ardentx11_stereo_device.*

The Doré calls specifically involving operation of the Stereo Viewer are:

- *DoDevice* at the end of the example routine *create_ardentx11_stereo_device* (note the argument string input parameter).
- *DoStereoSwitch* in the example routine *create_camera_group.*
- *DoStereo* in the example routine *create_camera_group.*

Instructions for compilation and use are also contained as comments in the code below.

```
/* begin stereoex.c */
/*======================================================================*/
/*
 * Example program of using the TITAN Stereo Viewer
 * with X+ and Dore'.
 *
 * The program will at first show a static stereo image in a window on the
 * screen. A carriage return will cause the objects to rotate for a while.
 * A second carriage return will exit the program.
 *
 * To compile:
 *    cc -O stereoex.c /usr/lib/dore.o -lXd -lXtitan -lXB -lX11 -lm -o stereoex
 *
 * To run:
 *    make sure X+ is running in stereo mode
 *    create an xterm on the stereo screen
 *    go to Stereo screen (<ALT><F2>)
 *    type:
 *       stereoex
 */
/*======================================================================*/

#include "dore.h"
#include "dore_proto.h"

#include <stdio.h>
#include <X11/Xlib.h>
#include <X11/Xatom.h>
#include <X11/Xutil.h>
#include <X11/XTitan.h>


DtColorRGB      backgroundcolor = { .3, .3, .3};
DtReal          rotatevalue     = 0.0;

DtObject        create_ardentx11_stereo_device(DtInt, DtInt, DtInt, DtInt);
DtObject        create_camera_group();
DtObject        create_display_group();
int             rotate_callback();
```

```
/*==================================================================*/
main()
{
   DtVolume      volume;
   DtObject      device;
   DtObject      frame;
   DtObject      view;
   DtInt         i;
                        /* initialize Dore' */
   DsInitializeSystem(0);
                        /* create X11+ stereo window */
   if (!(device = create_ardentx11_stereo_device(0, 0, 512, 256))) {
      printf("can't create ardentx11 stereo device0);
      exit(1);
   }
                        /* set up Dore' device, frame, and view */
   DdInqExtent(device,&volume);
   DdSetViewport(device,&volume);
   if (!(frame = DoFrame())) {
      printf("can't create frame0);
      exit(1);
   }
   DdSetFrame(device,frame);
   DfSetBoundary(frame,&volume);
   if (!(view = DoView())) {
      printf("can't create view0);
      exit(1);
   }
   DgAddObjToGroup(DfInqViewGroup(frame),view);
   DvSetClearFlag(view,DcTrue);
   DvSetBackgroundColor(view,DcRGB,backgroundcolor);
   DvSetRendStyle(view,DcRealTime);
   DvSetBoundary(view,&volume);
                        /* set Dore' display group */
   DgAddObjToGroup(DvInqDisplayGroup(view),    create_display_group());
                        /* set Dore' camera group */
                        /* all Dore' stereo calls in here */
   DgAddObjToGroup(DvInqDefinitionGroup(view), create_camera_group());
                        /* show the objects on the screen */
   DdUpdate(device);
                        /* rotate the objects on the screen */
   printf("Hit return to rotate objects0);
   getchar();
   DvSetUpdateType(view,DcUpdateDisplay);
   for (i=0; i<100; i++) {
      rotatevalue += .1;
      DdUpdate(device);
   }
                        /* wait for user to indicate program exit */
   printf("Hit return to exit0);
   getchar();
                        /* Dore' cleanup */
   DsReleaseObj(device);
   DsTerminateSystem();
}
```

```
/*==========================================================================*/
/*

  Create an X11+ stereo window at given location and size.

*/
DtObject
create_ardentx11_stereo_device(topx, topy, width, height)
     DtInt topx;
     DtInt topy;
     DtInt width;
     DtInt height;
{
   static char argstring[200];
   static XWMHints xwmh = {
      (InputHint|StateHint),     /* flags */
      True,                      /* input */
      NormalState,               /* initial_state */
      0,                         /* icon pixmap */
      0,                         /* icon window */
      0, 0,                      /* icon location */
      0,                         /* icon mask */
      0,                         /* Window group */
   };
   Display      *displayPtr;
   Window       window;
   XVisualInfo  *visual_info;
   XVisualInfo  vinfo_template;
   int          nvisuals;
   Colormap     cmap;
   XSetWindowAttributes attributes;
   XEvent       event;
   XSizeHints   xsh;
   XClassHint   class_hints;
   int nplanes          = 24;
   int buffercount      = 2;  /* set to 1 if no graphics expansion board */
   int attrMask         = CWBackPixel|CWBorderPixel|CWColormap;
   int screen           = 1;

   xsh.flags     = (USPosition|USSize);
   xsh.width     = width;
   xsh.height    = height;
   xsh.x         = topx;
   xsh.y         = topy;

   class_hints.res_name = "dore";
   class_hints.res_class = "Dore";

   /* create an X11 window,and Dore device (window) */

   if ((displayPtr = XOpenDisplay("")) == NULL ) {
      printf("couldn't open display0);
      return (DcNullObject);
   }

   vinfo_template.screen        = screen;
```

```
vinfo_template.class        = DirectColor;
vinfo_template.depth        = nplanes;

visual_info = XGetVisualInfo(displayPtr,
                    VisualScreenMask|VisualClassMask|VisualDepthMask,
                    &vinfo_template,&nvisuals);
if (nvisuals == 0) {
   printf("screen doesn't support specified visual type0);
   return (DcNullObject);
}

cmap = XTitanDefaultDirectColormap(displayPtr,screen);
if (cmap == NULL) {
   printf ("could not allocate color map0);
   return (DcNullObject);
}

attributes.background_pixel  = 0x000000;
attributes.border_pixel      = 0xffffff;
attributes.colormap          = cmap;

                /* create X11+ stereo window */
window = XBCreateStereoWindow(displayPtr,
                        RootWindow(displayPtr,screen),
                        xsh.x,xsh.y,xsh.width,xsh.height,0,
                        nplanes,InputOutput,visual_info->visual,
                        buffercount,1,
                        attrMask,&attributes);
if (window == NULL) {
   printf("couldn't open window0);
   return (DcNullObject);
}

XStoreName(displayPtr,window,"Dore Window");
XSetIconName(displayPtr,window,"Dore Window");
XSetWMHints(displayPtr,window,&xwmh);
XSetSizeHints(displayPtr,window,&xsh,XA_WM_NORMAL_HINTS);
XSetClassHint(displayPtr,window,&class_hints);

/* Catch our own exposure event */
XSelectInput(displayPtr,window,ExposureMask);
XMapWindow(displayPtr,window);
while(XNextEvent(displayPtr,&event)) {
   if (event.type == Expose)
         break;
}
XFlush(displayPtr);

sprintf (argstring, "-display %d -window %d -stereo",
         displayPtr,window);

return ( DoDevice("ardentx11",argstring) );
}
```

```
/*========================================================================*/
/*

  Create a Dore' camera group.
  Contains all Dore' stereo calls.

*/
DtObject
create_camera_group()
{
   static DtPoint3 at          = {0., 0., 0.};
   static DtPoint3 camerafrom  = {0., 5., 10.};
   static DtPoint3 lightfrom   = { 1., 1., 1.};
   static DtVector3 up         = {0., 1., 0.};
   static DtReal color[3]      = {1., 1., 1.};

   DoGroup(DcTrue);
                        /* turn on stereo */
   DgAddObj( DoStereoSwitch(DcTrue) );
                        /* set up stereo attributes for camera */
   DgAddObj( DoStereo(0.05, 10.0) );
   DgAddObj( DoPerspective (30., -1., -100.) );

   DgAddObj( DoPushMatrix() );
        DgAddObj( DoLookAtFrom(at, camerafrom, up) );
        DgAddObj( DoCamera() );
   DgAddObj( DoPopMatrix() );

   DgAddObj( DoPushMatrix() );
        DgAddObj( DoLookAtFrom(at, lightfrom, up));
        DgAddObj( DoLightColor (DcRGB, color) );
        DgAddObj( DoLightIntens(1.0) );
        DgAddObj( DoLightType (DcLightInfinite) );
        DgAddObj( DoLight() );
   DgAddObj( DoPopMatrix() );

   return ( DgClose() );
}
```

```
/*=====================================================================*/
/*

  Create a Dore' display group.
  Contains all objects to be displayed.

*/
DtObject
create_display_group()
{
   static DtReal color1[3] = {1., 0., 0.};
   static DtReal color2[3] = {1., 1., 0.};

   DoGroup(DcTrue);
        DgAddObj( DoBackfaceCullSwitch(DcOn) );
        DgAddObj( DoBackfaceCullable(DcOn) );

        DgAddObj( DoInterpType(DcVertexShade) );

        DgAddObj( DoCallback(rotate_callback, DcNullObject) );

        DgAddObj( DoDiffuseColor(DcRGB, color1) );
        DgAddObj( DoPushMatrix() );
                DgAddObj( DoRepType(DcSurface) );
                DgAddObj( DoTranslate (2.,0.,0.) );
                DgAddObj( DoPrimSurf(DcSphere) );
        DgAddObj( DoPopMatrix() );

        DgAddObj( DoDiffuseColor(DcRGB, color2) );
        DgAddObj( DoPushMatrix() );
                DgAddObj( DoRepType(DcWireframe) );
                DgAddObj( DoTranslate (-2.,0.,0.) );
                DgAddObj( DoPrimSurf(DcBox) );
        DgAddObj( DoPopMatrix() );

    return ( DgClose() );
}
```

```
/*==================================================================*/
/*

  Dore' callback to rotate object.

 */
rotate_callback()
{
   DsExecuteObj( DoRotate(DcYAxis,rotatevalue) );
}

/*==================================================================*/
/* end stereoex.c */
```

KNOB BOX AND TABLET

CHAPTER NINETEEN

This chapter describes how to install and configure the knob box (Stardent Model Number N-KNN/100) and digitizing tablet (Stardent Model Number N-TBL/100) interface devices.

Knob Box

After unpacking the knob box, plug the knob box cable into the receptacle on the bottom of the knob box and into the "Knob Box" port (RS-232 port "D") on the junction box. Attach the metal knob box stand to the back of the knob box in the vertical or horizontal position.

The knob box is now installed and should work with the Stardent 1500/3000 Demo suite with no further configuration. See the *Start Here* guide for information on running the Stardent 1500/3000 demo suite.

Programming the Knob Box

This section will be supplied at a later time.

Tablet

To work properly with the Stardent 1500/3000 Graphics Supercomputer, the digitizing tablet must be physically installed and software-configured.

Plug the cable for the digitizing tablet into its receptacle on the back of the tablet and into the "Tablet" port (RS-232 port "C") on the junction box. Plug the cord for the tablet pen into its receptacle on the back of the tablet.

Configuring the Tablet

This section will be supplied at a later date.

X WINDOWS

CHAPTER TWENTY

This chapter describes how to configure the system to use the X Windows and how to customize the X Windows configuration files. Table 20-1 gives X Windows configuration files and variables.

Table 20-1. X Windows Files and Commands [1]

Item	Function	Mod. Status
Commands:		
/usr/X11/bin/xstart	enter X window system	
/usr/X11/bin/xexit	leave X window system	
Files:		
/usr/X11/awm/system.awmrc	awm window manager system file	
/usr/X11/data/lib/XErrorDB	Xlib error messages file	
/usr/lib/Xdefaults	prototype configuration file	
/usr/lib/Xdesktop	prototype configuration file	
/usr/lib/Awmrc	prototype configuration file	
/usr/lib/Uwmrc	prototype configuration file	
~/.Xdefaults[1]	user configuration file	**
~/.xdesktop	user configuration file	**
~/.awmrc	user configuration file	**
Environment Variables:		
TERM	terminal type	**
DISPLAY	X output destination	**

System configuration tasks consist of checking the presence and format of the configuration files.

[1] ~ is C-shell terminology for the user's home directory.

Customizing X

You can customize the X Window System by modifying values in four user-changeable dot files:

- **.Xdefaults** – User Preferences: Application-specific choices of colors, fonts, window positions and sizes, etc.
- **.xdesktop** – Screen Layout: Initial set of applications and screen layout.
- **.xdeskstereo** – Stereo Screen: Screen layout features for Ardent's stereo viewing.
- **.awmrc** – Window Manager: Window and menu management, key and mouse button binding.

Contents

You will find information on the following pages in this order:

(1) *Inventory*: A list of X Window functions (more or less in English), paired with "translations" in X Window terms, and their location in the dot files.

(2) *Customization Background*: Basic information about syntax.

(3) *Dot Files*: Contents of dot files, with command syntax, definitions, and information about mouse control and change options.

(4) *Examples*: The *.awmrc* default startup file provided with the system, also menu examples including explanation of menu items.

Using this Guide

To customize X, first identify and locate the function (*Inventory*), make sure you understand the command syntax (*Customization Background*), change values with the help of information in the *Dot File Examples* section.

Example #1

Suppose you don't like your window background in deep blue (the way it comes). Check the Inventory under "background color" or "color, background." You will see that the function involved here is "Background" located in the *.Xdefaults* file and described in the *User Preferences* section.

Turn to the *User Preferences* section, where "Background: NavyBlue" tells you what the default is.

To change the background color, edit the *.Xdefaults* file to replace "NavyBlue" with the color of your choice. Use the correct term: look in */usr/X11/rgb/rgb.txt* for the exact name of the color.

After you finished editing, run the *xrdb* command (which acts as the csh *source*) command so that the change in the background color is made the next time you open a window. In this example, the command is: *xrdb .Xdefaults*.

Example #2

Or, suppose you want to have the "active window" (where your pointer is) rise to the top of other windows automatically when you move the pointer into it. In the Inventory, look under "raise window" or "window, raise." You will find "awm.autoraise" in *.Xdefaults*.

So now if you want to make "window raising" automatic, turn to the *User Preferences* section and look for *awm.autoraise*.

The default here is "off" so you should edit the *.Xdefaults* file's *awm.autoraise* entry to "on."

Use *xrdb* and restart the application (*awm*). From here on, each window will automatically rise to the top when the pointer is moved into it.

Example #3

Objective: To bind a new menu to the left mouse button.

Look for: "menu, binding."

The item will refer you to the *Window Manager (.awmrc)* section and the *f.menu* function.

This, unfortunately, is as far as this guide will take you, and you will need to figure out details of customization from documentation more complete and ambitious than this, but at least you will be in the right ballpark, knowing what function to modify, in which dot file.

NOTE
There is a great deal of information in the *man* on-screen help facility. *man awm*, can help with Example #3.

(Still, just as a bonus, here is what to add to the *.awmrc* file in order to associate new menu with the left mouse button:

menu = "lock" {
LockScreen: f.lock
}
f.menu = :root:left down: "Lock"

Then restart *awm* either by using the default menu item in the *WindowOps* menu or by killing and restarting *awm*.)

Example #4

Objective: To make the console xterm window appear in reverse video.

Look for: "reverse video."

The item will refer you to the *Screen Layout (.xdesktop)* section and *xterm*. (Alternatively, make the change in *Xdefaults*.

Again, from here on, you should be on your own (you could use *man xterm*), but let's carry this example task to its conclusion:

In the *.xdesktop* file, add **-rv** to the **xterm** line, so that it will look something like this:

xterm -g 80x8+0+0 -C -n -rv console&

Then exit X and restart it. Your console window will now be in reverse video.

Reminders

A few general tips:

- Use the *man* or *xman* on-screen help facilities. Type *man xterm*, for example, to get *xterm* information, *man xclock* for the clock function, etc. Ask for *man man* to check names of *man* entries. User items have *.x1* suffix, administrative entries *.x3*.
- To find out what fonts are available, use the *xlsfonts* command.

- To find out what colors are available, look at the *rgb.txt* file in the */usr/X11/rgb* directory.
- After modifying items in the dot files, do this to have the changes take place, to implement modifications: use *xrdb* and restart the application for *.Xdefaults* items; restart X for *xdesktop* and *xdeskstereo*; restart *.awm* for *.awmrc* changes.

Inventory

Various functions are given in the first column, the equivalent X Window System item in the second column, and the dot-file location in the third.

Remember that *.awmrc* items are described in the *Window Manager* section, *.Xdefaults* in *User Preferences*, and *.xdesktop* in *Screen Layout*.

Function	X Item	Location
Applications menu	f.menu	awmrc
autoraise	awm.autoraise	Xdefaults
autoselect	awm.autoselect	Xdefaults
background color	Background	Xdefaults
bell volume	awm.volume	Xdefaults
bold font	awm.menu.boldFont	Xdefaults
bold font for titlebar	awm.title.boldFont	Xdefaults
border color	borderColor	Xdefaults
border background	awm.border.background	Xdefaults
border foreground	awm.border.foreground	Xdefaults
border width, emacs	xemacs.borderWidth	Xdefaults
border width, window	borderWidth	Xdefaults
change icon into	f.iconify	awmrc

Function	X Item	Location
window		
change window into icon	f.iconify	awmrc
clock	xclock	xdesktop
clock titlebar	xclock.Title	Xdefaults
color, background	Background	Xdefaults
color, border background	awm.border.background	Xdefaults
color, border foreground	awm.border.foreground	Xdefaults
color, cursor	xterm*cursorColor	Xdefaults
color, foreground	Foreground	Xdefaults
color, titlebar background	awm.title.background	Xdefaults
colormap installation	awm.installColormap	Xdefaults
console window	xterm	xdesktop
cursor color	xterm*cursorColor	Xdefaults
cursor, place in first menu item	awm.autoselect	Xdefaults
database utility	xrdb	xdesktop
display grid image of window or icon during moving/resizing	awm.grid	Xdefaults
emacs border width	xemacs.BodyWidth	Xdefaults
emacs font	xemacs.BodyFont	Xdefaults
focus keyboard keys to one window	awm.frameFocus	Xdefaults
font for emacs	xemacs.BodyFont	Xdefaults

Function	X Item	Location
font for gadgets	awm.gadget.font	Xdefaults
font for icon box	awm.gadget.font	Xdefaults
font for icons	awm.icon.font	Xdefaults
font for popup window	awm.popup.font	Xdefaults
font for resize box	awm.gadget.font	Xdefaults
font for titlebar	awm.title.font	Xdefaults
foreground, border	awm.border.foreground	Xdefaults
foreground color	Foreground	Xdefaults
gadget display	awm.gadgets	Xdefaults
gadget font	awm.gadget.font	Xdefaults
grid image	awm.grid	Xdefaults
icon font	awm.icon.font	Xdefaults
icon pattern	awm.icon.pixmap	Xdefaults
iconify window	f.iconify	awmrc
load average	xload	xdesktop
load monitor title	xload.Title	Xdefaults
locating windows	xterm	xdesktop
locking out client applications	awm.freeze	Xdefaults
lower window	f.lower	awmrc
menu, Applications	f.menu	awmrc
menu cursor display	awm.autoselect	Xdefaults
menu, binding	f.menu	awmrc

Function	X Item	Location
menu, WindowOps	f.menu	awmrc
mouse pointer color	xterm*pointerColor	Xdefaults
move window	f.move	awmrc
pattern for titlebar	awm.title.pixmap	Xdefaults
pointer, color	xterm*pointerColor	Xdefaults
pointer, use in border to activate commands	borderContext	Xdefaults
popup window font	awm.popup.font	Xdefaults
raise window	f.raise	awmrc
raise window	awm.autoraise	Xdefaults
resize window	f.resize	awmrc
restrict window movement to edge of screen	awm.wall	Xdefaults
reverse video	xterm	xdesktop
save lines scrolling off window	xterm*saveLines	Xdefaults
scrollbars	xterm*Scrollbar	Xdefaults
show line image	awm.zap	Xdefaults
specify a bitmap file	awm.path	Xdefaults
start windows	xterm	xdesktop
suspend other applications during moving/resizing	awm.freeze	Xdefaults
title, background color	awm.title.background	Xdefaults
title, foreground color	awm.title.foreground	Xdefaults
titlebar background color	awm.title.background	Xdefaults

Function	X Item	Location
titlebar, bold font	awm.title.boldFont	Xdefaults
titlebar for clock	xclock.Title	Xdefaults
titlebar for window	awm.titles	Xdefaults
titlebar pattern	awm.title.pixmap	Xdefaults
visual bell mode	xterm*visualBell	Xdefaults
volume for bell	awm.volume	Xdefaults
white space between icon margin and border	awm.icon.hPad awm.icon.vPad	Xdefaults Xdefaults
white space between menu margin and border	awm.menu.pad	Xdefaults
window, console	xterm	xdesktop
window, iconify	f.iconify	awmrc
window, locating	xterm	xdesktop
window, lower	f.lower	Xdefaults
WindowOps menu	f.menu	awmrc
window, move	f.move	awmrc
window, raise	f.raise	awmrc
window, raise	awm.autoraise	Xdefaults
window, resize	f.resize	awmrc
window, start	xterm	xdesktop
window titlebar	awm.titles	Xdefaults

With the specific button, you must identify the action of that button. Mouse actions can be:

down when the specified button is pressed down.

up when the specified button is released.

delta indicates that the mouse must be moved the number of pixels specified with the delta variable before the specified function is invoked. The mouse can be moved in any direction to satisfy the delta requirement.

Menu Definition

After binding a set of keys and a menu name to *f.menu*, define the menu to be invoked by using the following syntax:

menu = (*string*) " *menu name* " { "*item name*" : "*action*"
.
.
. }

The *string* in parenthesis is an optional argument which names a pixmap file (see also: *path*) to use as the menu title rather than just using the name of the menu. This is generally only useful if you're using pixmaps for the menu panes as well (see below). Though the *menu name* isn't displayed when you specify *string*, you still need to specify one for *awm* to use when looking up the binding to it.

Enter the *menu name* exactly the way it is entered with the *f.menu* function or the window manager will not recognize the link. If the *menu name* contains blank strings, tabs or parentheses, it must be quoted here and in the *f.menu* function entry.

If you haven't chosen to display a pixmap title in *string*, the menu name will be displayed at the top of the menu in whatever font has been chosen for **menu.boldFont** (or its default).

You can enter as many menu items as your screen is long. You cannot scroll within menus.

Any menu entry that contains quotes, special characters, parentheses, tabs, or strings of blanks must be enclosed in double quotes. Follow the item name by a colon (:).

Menu Action

Window manager functions

Any function previously described, e.g., *f.move* or *f.iconify*. Using *f.menu* results in a pull-right pane which you can use to "walk" between menus (see below). "Walk" by moving the cursor onto the pull-right arrow displayed at the right edge of the pane or by clicking another button in the pane while holding the original one down.

Walking menus

Select the function *f.menu* and separate it from the *menu name* with a colon (:) i.e.,

menu = "foo" { Walking Menu: f.menu: "NextMenu" }

Shell commands

Begin with an exclamation point (!) and set to run in background. You cannot include a new line character within a shell command.

Text strings

Text strings are placed in the window server's cut buffer. The strings must be preceeded by one of:

A caret (^), which is stripped off, signifying that the string will automatically be followed by a newline (i.e., using '^' causes one to be added at the end).

A vertical bar (|), also stripped off, signifying that the string should not end with a newline.

Booleans

Any boolean variable, e.g., *reverse* or *autoraise.* The current state of a boolean variable in a menu will be indicated with a check mark (a check mark means the boolean is set to true). Note that the boolean is not preceeded by *awm.* as it is in the resource database.

Note that menus bound to title bars, gadget boxes or borders cause (where logical) the selected menu action to occur automatically on the client window; you don't need to select a window for the action. However, actions requiring mouse tracking (i.e., move, resize) will usually not work well in this context.

Color Defaults

Colors default to the colors of the root window under any of the following conditions:

- If you run out of color map entries, either before or during an invocation of *awm*.
- If you specify a foreground or background color that does not exist in the RGB color database ($LIBDIR/*rgb.txt*).
- If you omit a foreground or background color.
- If you specify no colors in the resource database.

X Defaults

A number of variables that used to be specified in the *.uwmrc* file (of *uwm*, for those familiar with it) are now retrieved from the resource database. When a value cannot be found, a default (compiled into *awm*) is substituted. A much wider range of options can be specified this way.

Special Resources.

name.wm_option.autoRaise (boolean)

name.wm_option.borderContext (boolean)

name.wm_option.gadgets (boolean)

name.wm_option.icon.labels (boolean)

name.wm_option.title (boolean)

name.wm_option.foreground (string) [Only if -DRAINBOW]

name.wm_option.background (string) [Only if -DRAINBOW]

These resources determine whether or not a given application really wants a title, gadgets, border context area, to be auto-raised etc etc.. The application's CLASS and NAME (in the WM_CLASS property) are checked against the string supplied for *name* (for example: Xclock*wm_option.title: off).

Specifying one of these resources overrides any other boolean settings (i.e., *awm.titles* or *awm.gadgets*) and may be used to turn things on and off at the application and/or class level for applications, regardless of **awm**'s settings.

Both class and name resources are checked, and in that order. Thus specific applications may override settings for their class, if desired.

These resources are "special" as they are checked for under the application's name, not *awm*'s.

For example,

xclock.wm_option.autoRaise

is different from

awm.xclock.wm_option.autoRaise.

The resources *wm_option.foreground* and *wm_option.background* are only meaningful if awm has been compiled with -DRAINBOW. They allow the border color to be specified for individual applications/classes.

Dot Files

Window Manager Resources (.awmrc)

f.iconify

Changes a window into an icon or an icon into a window.

:icon: middle up — Place pointer in icon box (located in upper lefthand corner of titlebar), click middle button.

:g[1]: left down — Place pointer in icon box, hold left button down.

:g[1]: middle down — Place pointer in icon box, hold middle button down.

:g[1]: right down — Place pointer in icon box, hold right button down.

meta :w | icon: middle up — Place pointer on a window or icon, press meta key and click middle button at the same time.

NOTE
Meta is a user-definable control key.

:title: middle up — Place pointer on titlebar, click middle button.

f.lower

Lowers windows.

meta :w | icon: right down — Place pointer on a window or icon, press meta key and right button down at the same time.

:title: right down — Place pointer on titlebar, press right button down.

f.menu (Applications)

Brings up the Applications menu.

m | s : : middle down : "Applications" -- Press meta or Shift key, place pointer on root window (shaded background window located below all other windows), hold middle button down.

:root: middle down : "Applications" -- Place pointer on root window, hold middle button down.

f.menu (WindowOps)

Brings up the WindowOps menu.

:border: right down : "WindowOps" — Place pointer on border, hold right button down.

lock :root: right down : "WindowOps" (Caps/Numeric Lock On)" -- With Caps/Numeric Lock key on, place pointer on root window, hold right button down.

mod2 | lock :root: right down : "WindowOps (Caps/Numeric Lock On)" — With Caps/Numeric Lock key on, place pointer on root window, hold right button down.

mod2 :root: right down : "WindowOps (Caps/Numeric Lock On)" — With Caps/Numeric Lock key on, place pointer on root window, hold right button down.

m | s : : right down : "WindowOps" -- Press meta or Shift key, place pointer on root window, hold right button down.

:root: right down : "WindowOps" — Place pointer on root window, hold right button down.

f.move

Moves a window or icon to a new location.

:border: left down — Place mouse pointer on border of window, hold left button down.

:icon: left motion — Place pointer on an icon, hold left button down.

meta :w | icon: left motion — Place pointer on a window or icon, press meta key and hold left button down at the same time.

:title: left motion — Place mouse pointer on titlebar, hold left button down.

f.raise

Raises a window.

meta :w | icon: left up — Place pointer on a window or icon, click left button.

:title: left up — Place pointer on titlebar, click left button.

f.resize

Resizes a window.

:border: middle down — Place pointer on border of window, hold middle button down.

:g[0]: left down — Place pointer on resize gadget (box located on upper righthand corner of titlebar), hold left button down.

:g[0]: middle down — Place pointer on resize gadget, hold middle button down.

:g[0]: right down — Place pointer on resize gadget, hold right button down.

meta :w | icon: middle motion — Place pointer on a window or icon, press meta key and hold middle button down at the same time.

:title: middle motion — Place pointer on titlebar of window, press middle button.

User Preferences (.Xdefaults)

Background: NavyBlue

Sets background color of windows in navy blue. The background color is the backdrop display.

Foreground: White

Sets foreground color of text white. The foreground color is the text color display in windows, menus, and graphics output.

Smalltalk80*Background: NavyBlue

Sets the background color when using Smalltalk80 programming language.

Smalltalk80*Foreground: White

Sets the foreground color when using Smalltalk80 programming language.

awm.autoraise: off

If "on," the xterm window automatically raises when pointer enters it.

awm.autoselect: off

If "off", displays menu cursor in the menu header.

awm.autoselect: on

If "on," places cursor in the first menu item.

awm.border.foreground: green

Sets the border foreground to green.

awm.border.hilite: on

If "on," causes certain actions to occur when a border gains input focus.

awm.borderContext.background: IndianRed

Specifies the background color of the border color pixmap.

awm.borderContext.boldPixmap: hatch.b

The pixmap file name to load and display the border context area with when the focus is in the border.

awm.borderContext.foreground: red

Specifies the foreground color of the border color pixmap.

awm.borderContext.pixmap: dot.b

Displays the border context area background pattern.

awm.borderContext.width: 4

Specifies the number of pixels wide to make the border context.

awm.delta: 5

Indicates 5 pixel spaces need to be moved before the window manager interpretes it as a command.

awm.frameFocus: on

Highlights the window when pointer enters it and focuses keyboard input into this window.

awm.freeze: on

If "on," locks all other client aplications out of the server during certain window manager tasks, such as move and resize.

awm.gadget.font: vtsingle

Specifies font for gadgets.

awm.gadgets:on

If "on," the gadgets display on the titlebars.

awm.grid: on

If "on," displays grid-like image of the window or icon you are resizing or moving.

awm.hilite:on

If "on," causes certain actions to occur when a window gains input focus. If *title.boldPixmap* is defined, the titlebar background is set to it. If *borderContext.boldPixmap* is defined, the border context area is set to it.

awm.icon.font: 9x15

Specifies the icon text font.

awm.icon.hPad: 5

Indicates the number of pixels to pad an icon horizontally.

awm.icon.pixmap: icon.b

Specifies the icon pattern.

awm.icon.vPad: 5

Indicates the number of pixels to pad an icon vertically.

awm.installColormap: off

Installs a given window's colormap when pointer enters it and installs the default colormap when pointer leaves the window.

awm.menu.boldFont: vtbold

Specifies bold font for the menu.

awm.menu.delta: 25

Indicates the number of pixels the cursor is moved before the action is interpreted by the window manager as a command.

awm.menu.font: vtsingle

Specfies menu font as "vtsingle."

awm.menu.itemBorder: 1

Sets border width of menu item to 1 pixel.

awm.menu.pad: 3

Indicates the amount of space in pixels that the menu is padded (number of bytes used for margins within window and menu boundaries).

awm.path: /usr/include/X11/bitmaps

Allows you to specify a pixmap file rather than simply a text string to display.

awm.popup.font: vt-25

Specifies font for the popup window (e.g. menu windows) text.

awm.push:1

Pushes the window down to make room for the titlebar.

awm.title.background: yellow

Sets titlebar background to yellow.

awm.title.boldFont: vtbold

Specifies bold font for the titlebar.

awm.title.boldPixmap: titlebold.b

Specifies the titlebar pattern.

awm.title.font: vtsingle

Sets titlebar font to "vtsingle."

awm.title.foreground: black

Sets titlebar foreground to black.

awm.title.pixmap: titlepix.b

Specifies the titlebar pattern.

awm.title.push: on

Pushes the window down to make room for the titlebar.

awm.titles:off

If "off," the windows have no titlebars.

awm.volume: 4

Specifies the bell volume, set by the xset command, between an integer 0-7, where 7 is the loudest volume.

awm.wall: on

Restricts window movements to edge of a screen (rootwindow).

awm.zap: on

If "on," causes line image to follow windows or icons from its default position to its new position during resizing or moving.

borderColor: Red

Sets border color to red.

borderContext: on

If "on," the cursor must be on the window border for a command to activate.

borderWidth: 1

Sets border width of windows to 1 pixel.

xclock.Title: off

The clock titlebar is named "off."

xemacs.BodyFont: 9x15

Specifies 9x15 bold font for emac editor.

xemacs.BorderWidth: 2

Sets border width of emacs window to 2.

xload.Title: off

The load monitor titlebar is named "off."

xterm*Font: ardents

Sets font to "ardents."

xterm*ScrollBar: on

If "on," creates windows with scrollbars.

xterm*cursorColor: Red

Sets the cursor color to red.

xterm*iconPixmap: /usr/include/X11/bitmaps/xterm.b

Specifies the path for the icon bitmap.

xterm*pointerColor: Yellow

Sets mouse pointer color to yellow.

xterm*saveLines: 1000

Saves the number of lines to 1000 when you scroll to the top of the window.

xterm.wm_option.title: on

If "on," overrides any other settings for the title.

xterm*visualBell: on

If "on," turns on the visual bell mode at startup.

Screen Layout (.xdesktop)

xclock -analog -g 100x100+830+0&

Displays the time.

-analog — Displays the time in an analog clock form.

-g 100x100+830+0 — Specifies the size and location of the clock. 100 by 100 pixels located at 830 pixels from the right edge of the screen and 0 pixels from the top of the screen.

xload -g 200x100+940+0&

Periodically monitors the system for load average and graphically displays the load using a simple histogram.

-g 200x100+940+0 — Specifies the size and location of the load monitor. 200 by 200 pixels located at 940 pixels from the right edge of the screen and 0 pixels from the top of the screen.

xrdb -merge $HOME/.Xdefaults

X server resource database utility.

-merge $HOME/.Xdefaults — Indicates that the input should be merged with, not replacing, the current contents of the RESOURCE_MANAGER property.

xterm

Starts additional terminal and console windows. These are some of the options and their meaning:

-g 80x50+0+0 — Specifies the size and location of the window. 80 characters wide by 50 lines long located 0 pixels from the right and top edge of the screen.

-sb — Indicates that some number of lines that are scrolled off the top of the window should be saved and that the scrollbar should be displayed.

-sl 1000 — Specifies number of lines to save that have scrolled off the screen is 1000.

-n 'hostname' — Specifies the name for xterm's window.

-C — Indicates this window receives console output.

-fn ardents — Specifies font to be used in displaying text is "ardents."

-n console — Specifies the name for xterm's window as "console."

Stereo Screen (.xdeskstereo)

Same as *Screen Layout (.xdesktop)* above, with the addition of a shell script component such as:

DISPLAY=unix:0.1
export DISPLAY
awm &

which will result on switching the display and starting *awm* on the second screen when X starts.

Examples

.awmrc Default File

```
# This is the system startup file for awm that defines the standard
# portion of the Ardent window manager.  The .awmrc file in the users
# home directory defines the user customizable options but should not
# override the specifications made in this file
#

resetbindings
resetmenus

# resize gadget

gagdet[0] = (resize.b)

# iconify gadget

gadget[1] = (close.b)

#gadget[0] = (gl.b)
#gadget[2] = (skull.b)

# FUNCTION       KEYS    CONTEXT          MOUSE BUTTON ACTIONS
f.move=            :border:    left down
f.resize=          :border:    middle down
f.menu=            :border:    right down   : "WindowOps"

f.resize=          :title:     middle motion
f.move=            :title:     left motion
f.raise=           :title:     left up
f.iconify=         :title:     middle up
f.lower=           :title:     right down

f.iconify=                       :g[1]:          left down
f.iconify=                       :g[1]:          middle down
f.iconify=                       :g[1]:          right down

f.resize=                        :g[0]:          left down
```

```
f.resize=                     :g[0]:          middle down
f.resize=                     :g[0]:          right down

f.raise=    meta  :w|icon:    left up
f.move=     meta  :w|icon:    left motion
f.move=                 :icon:          left motion

f.resize=   meta  :w|icon:    middle motion
f.iconify=  meta  :w|icon:    middle up
f.iconify=        :icon:      middle up

f.lower=    meta  :w|icon:    right down
f.menu=           :root:      right down   : "WindowOps"
f.menu=     mod2  :root:      right down   : "WindowOps
                                (Caps/Numeric Lock On)"
f.menu=mod2|lock  :root:              right down   :
                               "WindowOps (Caps/Numeric Lock On)"
f.menu=     lock  :root:      right down   : "WindowOps
                                (Caps/Numeric Lock On)"
f.menu=     m|s   ::          right down   : "WindowOps"

menu = "WindowOps" {
New Window: !"xterm -sb -sl 1000 -n 'hostname'&"
RefreshScreen: !"xrefresh"
Resize:        f.resize
Lower:         f.lower
Raise:         f.raise
Preferences:    f.menu: "Preferences"
CircUp:        f.circleup
CircDown:      f.circledown
MoveOpaque:    f.moveopaque
Iconify:       f.newiconify
Focus:         f.focus
Destroy:       f.destroy
Restart AWM:   f.restart
Exit AWM:       f.exit
Exit X Windows: !"xexit"
}

menu = "WindowOps (Caps/Numeric Lock On)" {
New Window: !"xterm -sb -sl 1000 -n 'hostname'&"
RefreshScreen: !"xrefresh"
Resize:        f.resize
Lower:         f.lower
Raise:         f.raise
Preferences:    f.menu: "Preferences"
CircUp:        f.circleup
CircDown:      f.circledown
MoveOpaque:    f.moveopaque
Iconify:       f.newiconify
Focus:         f.focus
Destroy:       f.destroy
Exit AWM:       f.exit
Restart AWM:   f.restart
Exit X Windows: !"xexit"
}
```

```
menu = "Preferences" {
Bell Loud:  !"xset b 7&"
Bell Normal:   !"xset b 3&"
Bell Off:   !"xset b off&"
Click Loud: !"xset c 8&"
Click Soft: !"xset c on&"
Click Off:  !"xset c off&"
Mouse Fast: !"xset m 4 2&"
Mouse Normal:  !"xset m 2 5&"
Mouse Slow: !"xset m 1 1&"
Screensaver on:   !"xset s blank"
Screensaver off:!"xset s noblank"
Highlight:  hilite
Autoraise:  autoraise
rootbox: rootResizeBox
}

f.menu=           :root:      middle down : "Applications"
f.menu=     m|s   ::          middle down : "Applications"
```

Preferences Menu

Bell Loud: !"xset b 7&"

Sets bell volume to loudest setting.

Bell Normal: !"xset b 3&"

Sets bell volume to normal setting.

Bell Off: !"xset b off&"

Sets bell volume to off.

Click Loud: !"xset c 8&"

Sets mouse click volume to loudest setting.

Click Soft: !"xset c on&"

Sets mouse click volume to a low volume.

Click Off: !"xset c off&"

Sets mouse click volume to off.

Mouse Fast: !"xset m 4 2&"

Sets mouse tracking speed to a fast setting.

Mouse Normal: !"xset m 2 5&"

Sets mouse tracking speed to a normal setting.

Mouse Slow: !"xset m 1 1&"

Sets mouse tracking speed to a slow setting.

Screensaver on: !"xset s blank"

Turns screen saver on to show a blank screen.

Screensaver off:!"xset s noblank"

Turns screen saver off.

Highlight: hilite

Causes certain actions to occur when the window gains input focus. (See **awm.hilite**)

Autoraise: autoraise

Raises a window automatically when pointer enters the window.

rootbox: rootResizeBox

Put resize (popup) window in upper left corner of root window, rather than on the window being resized.

WindowOps Menu

New Window: !"xterm -sb -sl 1000 -n 'hostname'&"

Creates a new window.

RefreshScreen: !"xrefresh"

Redraws the contents of your screen. This option is useful if you get system messages from outside the X Window system that you want to erase.

Resize: f.resize

Resizes a window.

Lower: f.lower

Lowers a window.

Raise: f.raise

Raises a window.

Preferences:f.menu: "Preferences"

Brings up the Preferences menu which allows you to choose the bell volume, keyclick volume, the Caps Lock key settings, and the mouse tracking speed.

CircUp: f.circleup

Moves a window up to the top of a stack of windows.

CircDown: f.circledown

Moves a window down to the botton of a stack of windows.

MoveOpaque: f.moveopaque

Moves a window or icon to a new location.

Iconify: f.newiconify

Changes a window into an icon or an icon into a window.

Focus: f.focus

Directs all keyboard inputs to a specified window.

Destroy: f.destroy

Terminates a window.

Restart AWM: f.restart

Causes the window manager application to restart, retracing the *awm* search path and initializing the variables it finds.

Exit AWM: f.exit

Exits the window manager application.

Exit X Windows: !"xexit"

Quits X Windows.

NETWORK

CHAPTER TWENTY–ONE

This chapter describes how to install an Ethernet or Cheapernet network and how to configure the system to run the Network File System (NFS). The procedures in this chapter are sufficient if you are configuring a small, simple network or you have prior experience configuring Ethernet/Cheapernet networks. If not, you may need background information on networking and the TCP/IP protocols. See the Preface to this guide for suggested readings.

Making the Physical Connection

Ethernet network. Connect the Ethernet transceiver cable to the Ethernet port on the Stardent 1500/3000 System Module I/O board. Place the Ethernet/Cheapernet selection switch in the "Out" position. Note that Ethernet version 1 is not supported.

Cheapernet network. Attach a BNC "TEE" to the Cheapernet port on the I/O board. Place the Ethernet/Cheapernet switch in the "In" position. If Stardent 1500/3000 is the end of the Cheapernet, install a Cheapernet terminator on the "TEE."

Configuring the Network

Table 21-1 shows files and commands needed to configure an Ethernet or Cheapernet network. Network tasks include assigning machine and network names, making sure that the network daemons are running, and creating administrative files.

Table 21-1. Network Configuration [1]

Item	Function	Mod. Status
Commands:		
/etc/rc2	init level 2 command	
/etc/rc2.d/S30tcp	start TCP/IP daemons, symblink to /etc/init.d/tcp	
/etc/rc2.d/K30tcp	stop TCP/IP daemons, symblink to /etc/init.d/tcp	
/etc/init.d/tcp	start/stop TCP/IP daemons	**
/etc/ifconfig	configure network parameters	
/etc/arp	display, modify Internet to Ethernet maps	
/etc/nvram	display modify NVRAM variables	
/etc/ping	determine in remote host is alive	
/etc/route	manually manipulate the routing table	
/bin/hostname	set/print host name	
/bin/hostid	display, modify host's identifier	
/bin/rcp	remote file copy	
/bin/uname	print host name and OS release info	
/usr/ucb/netstat	display network status	
/usr/ucb/rlogin	login to remote machine	
/usr/ucb/rsh	remote shell	
/usr/ucb/ruptime	display status of network machines (/etc/rwhod)	
/usr/ucb/rwho	display current logins (/etc/rwhod)	
/usr/ucb/telnet	remote login	
/usr/ucb/ftp	file transfer to remote machines	
/usr/ucb/tftp	file transfer to remote machines	
Daemons and Helpers:		
/etc/nda	stream module organizer	
/etc/inetd	Internet TCP/IP daemon	
/etc/ftpd	file transfer protocol daemon	
/etc/routed	network routing daemon	
/etc/rlogind	remote login daemon	
/etc/rshd	remote shell daemon	
/etc/rwhod	remote who daemon	
/etc/telnetd	telenet virtual terminal protocol daemon	
/etc/tftpd	trivial file transfer protocol daemon	

[1] The "modification status" column in the table shows whether or not the file requires modification. If the modification status box is empty, no modification is recommended. (Commands fall in this category.) If the box has a single asterisk (*), modification is optional. If the box has a double asterisk (**), the file or variable must be examined (and modified if necessary).

Table 21-1. Network Configuration [1] (continued)

Item	Function	Mod. Status
Files:		
/etc/hosts	internet address and machine names	**
/etc/hosts.equiv	passwd, group compatible systems	**
/.rhosts	remote hosts allowed to be root on local machine	**
~/.rhosts	remote hosts and users	
/etc/networks	available networks	**
/etc/protocols	Internet protocol mapping	
/etc/services	services available over the network	
/etc/inetd.conf	Internet socket to daemon binding table	
/etc/nda.conf	stream module configuration	
NVRAM Variables:		
etheraddr	Ethernet address	
netaddr	Internet address	**
hostname	machine name	**

Network Tasks

1. Assign hostname. Check to see if a hostname has been assigned to the system. The name is maintained in the NVRAM variable *hostname*. Issue the command

nvram

to obtain a list of all the NVRAM variables. If you see the line

`hostname=`*name*

in the output of the *nvram* command, the hostname has already been assigned. If not, choose a short alphanumeric name and set the *nvram* variable with the command

nvram hostname=*the_name_you_have_selected*

2. Obtain Internet address. If you already have an Ethernet network in place, you can configure your machine to match your current Internet numbering scheme. Otherwise, you need to obtain an Internet address for your system, as a proper Internet address is required for communications with networks external to your installation. This section gives some information about the structure and use of the Internet address; to obtain an Internet address contact the Network Information Center, SRI International, Menlo Park, California, 94025.

Each Internet address is a 32 bit quantity, often expressed as four period-separated decimal numbers, each in the range 0-255. The first portion of the address identifies the network; the second portion identifies the particular host machine. In decimal form the address looks something like this:

```
192.9.200.166
```

DARPA (the Defense Advanced Research Projects Agency, which developed the Internet numbering scheme,) defines three classes of Internet network. The generic addressing for each class is shown in Figure 21-1. Class A networks use the first 8 bits of the address for the network and the remaining 24 bits for the host, Class B networks use the first 16 bits for the network and the remaining 16 bits for the host, and Class C networks use the first 24 bits for the network and the remaining 8 bits for the host.

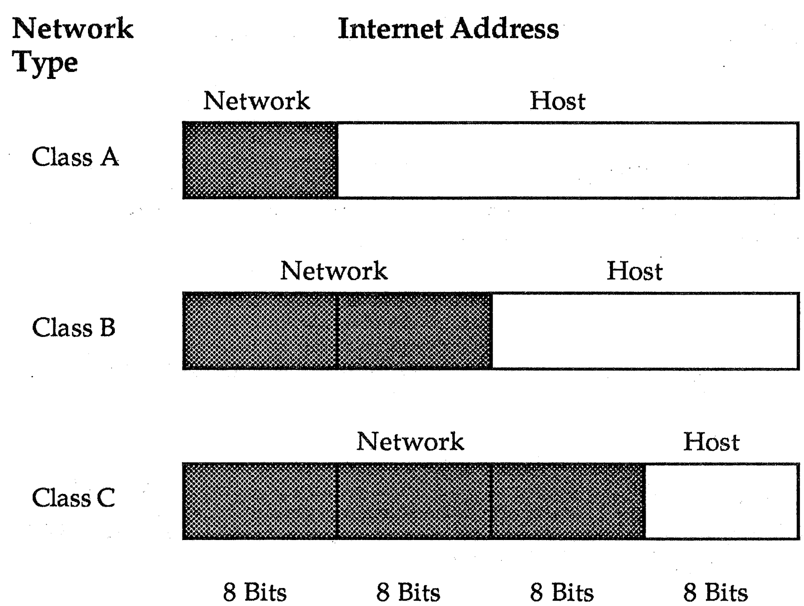


Figure 21-1. Internet Network Classes

The class of network you request from SRI is a function of how many hosts you ultimately plan for your network. (Since only 255 Class A addresses are available, they are reserved for very large, government or corporate institutions. Plan to choose a Class B or C network.) Class C networks can have a maximum of 255 hosts; Class B networks permit you to define subnetworks under your local purview.

4. Define Subnetworks and Subnet Masks. If you are configuring a Class B (or Class A) network you have the option of defining subnetworks for organizational or administrative purposes. For example, subnetworks let you broadcast messages within an organizational subentity (the Chemistry Department) rather than across your entire organization.

You define subnetworks by partitioning the "host" portion of the Internet address, as illustrated in Figure 21-2. You assign some number of leftmost bits for the subnet address, and the remaining bits as the host address within the subnetwork. You are free to partition in any way you choose; as shown in Figure 3-2 we suggest that you select 4 or 8 bits for the subnet address and the remaining bits for the subnet host address.

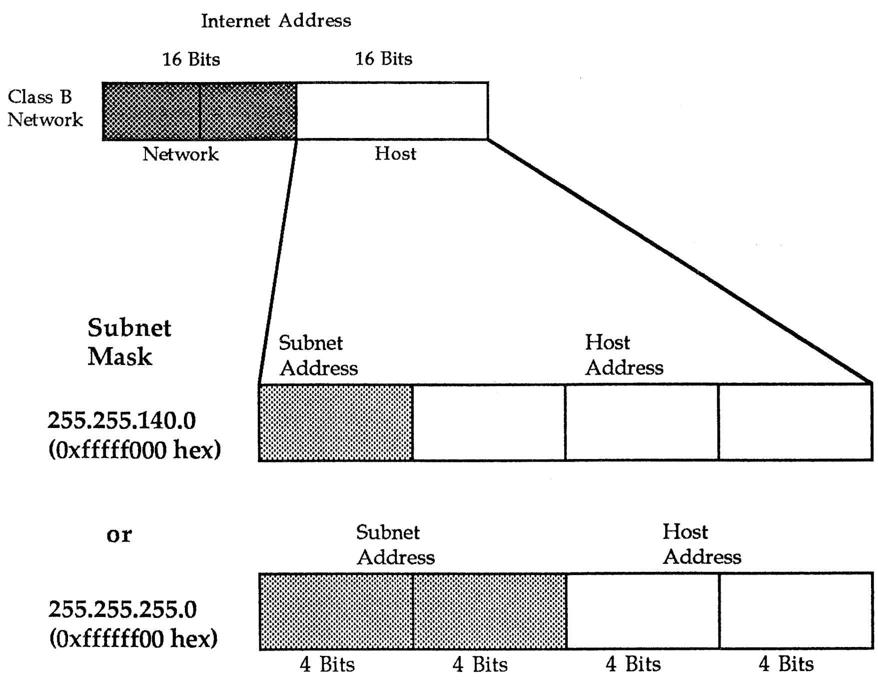


Figure 21-2. Internet Subnetwork Masks

The *subnet mask* is the device the system uses to restrict information flow to a particular subnetwork. To define a subnet mask you "block out" the external fields of the Internet address and leave free space for the subnet portion. The blocking code is all ones in hex. For a Class B network the network mask is

```
255.255.0.0 (decimal) or
0xffff0000 (hex)
```

For an 8-bit subnetwork within the Class B network the subnet mask is

```
255.255.255.0 (decimal) or
0xffffff00 (hex)
```

Once you have defined a subnetwork you can also define a broadcast address, for broadcasting messages to your local subnetwork. The broadcast address consists of the Internet address of the subnetwork, with all '1's (hex) for the subnetwork portion. For instance, the address

```
128.9.200.255
```

is a broadcast address for the subnetwork with Internet address 128.9.200.XXX.

You set the subnet mask and the broadcast address as part of configuring the network with the *ifconfig* command. The *ifconfig* command is executed as part of */etc/init.d/tcp*.

5. Edit /etc/init.d/tcp. The file */etc/init.d/tcp* contains *ifconfig* commands to configure your network. As *su*, edit the file and find lines similar to the following:

```
#
#       configuring network.
#
        if test -x /etc/nda ; then
                if /etc/nda
                then
                        ifconfig et0 $HOSTNAME -trailers up
                        ifconfig lo0 localhost
                        echo "network configured."
                else
                        echo "configure network FAILED"
                        exit 1
                fi
        fi

#
```

If you are configuring a Class C network, you need not change the file. If you are configuring a Class B (or A) network with a subnet mask, you should add a new *ifconfig* command such as the following:

```
ifconfig et0 netmask 0xffffff00 broadcast 128.9.200.255
```

This command defines the subnet mask 255.255.255.0 (0xffffff00 in hex) and a broadcast address of 128.9.200.255. Use addresses consistent with the subnetwork you have defined.

ifconfig is not sensitive to the ordering of options. For instance, either of the following forms is acceptable:

```
then
        ifconfig et0 $HOSTNAME -trailers up netmask 0xffffff00 broadcast 128.9.200.255
                ifconfig lo0 localhost
                echo "network configured."
        else
```

or

```
then
        ifconfig et0 $HOSTNAME -trailers up
        ifconfig et0 netmask 0xffffff00 broadcast 128.9.200.255
        ifconfig lo0 localhost
        echo "network configured."
else
```

6. Administration files. Each machine in the network requires copies of files that describe network machines: */etc/hosts, /etc/hosts.equiv, /etc/networks* and */.rhosts.*

/etc/hosts lists, by Internet address, all the machines in your local domain. The file has this format:

```
Internet-address   machine-name   multi-level-domain-names   aliases   #comment
```

Note the following:

- You can use blanks and/or tabs to separate the fields.
- *Internet-address*, your Internet address, should have the same value as the NVRAM variable *netaddr*. (*netaddr*) is not used by UNIX, but is needed if you wish to be able to boot files from another machine on the network.
- *machine-name* should be the same name as the NVRAM variable *hostname.*
- Multi-level Internet domain names such as

```
moose.stardent.com
```

may be optionally added.

- Aliases may be optionally included, such as *mailhost* as the alias for the network's *uucp* node. (See *Communications* in this guide for more.)
- The default */etc/hosts* file shipped with your system contains one or more example lines.

See *hosts*(4) in the *Programmer's Reference Manual* for details.

/etc/hosts.equiv contains a list of machines known to be password-compatible with your machine for the purpose of accepting. *rcp, rlogin,* and *rsh* commands. Each line in the file has this format:

```
machine-name    user-name(optional)
```

Note the following:

- You can use blanks and/or tabs to separate the fields.
- *machine-name* should be the same as that in the */etc/hosts* file (and the same as the NVRAM variable *hostname*).
- Default files shipped with your system contain one or more example lines.

/.rhosts contains a list of machines authorized to *rlogin* to your machine as "root." It has the same format as */etc/hosts.equiv*. **For security reasons be very careful about selecting machines to list in** */.rhosts*.

See *hosts.equiv*(5) in the *Programmer's Reference Manual* for details.

/etc/networks Identifies networks with which your machine can communicate. Lines in the file have this format:

```
network-name    network-number
```

Note the following:

- You can use blanks and/or tabs to separate the fields.
- *network-name* is an ASCII string that identifies your local network, and network-number is the network portion of the Internet address in the decimal, period-separated notation.
- Default files shipped with your system contain one or more example lines. See *inet*(3N) in the *Programmer's Reference Manual* for details.

VAX
89.0.0.1 hope vax

7. Create administration files: new network. If
new, become super-user (or bring the system dow
mode), create */etc/hosts, /etc/hosts.equiv, /etc/networ*
files with the appropriate machine and network i
rcp copies of the files to the other machines in th
the network is up and running.

89.0.0.13 ~~student titan~~

titan student

8. Create administration files: existing network.
is to be added to an existing network, follow this p

192.~~2~~0.46

(1) Become super-user, or bring the system d user mode.

89.0.0.9 iris silicon x

(2) Make a simple */etc/hosts* file that includ... your machine.

(3) Change to run level 2 so that */etc/rc2.d/S30tcp* is executed:

```
# init 2
```

(1) Have your network administrator add the appropriate information about your machince to the master machine's */etc/hosts* file. (The master machine is the machine that maintains global network information for your network.)

(2) Use *ftp*(1C) (file transfer protocol) to copy the network administration files from the master machine to your machine: */etc/hosts* and */etc/networks*. Issue the command

```
# ftp internet-address-of-master-machine
```

You are then prompted to login to the machine. Once you have done so you can transfer the files. See *ftp*(1C) in the *Commands Reference Manual* for details.

(1) After the files have been transferred you are free to set up your own */etc/hosts.equiv* and */.rhosts* files.

(2) Finally, make sure that only root is allowed to modify the files.

You are now ready to access machines on the network.

Network: Troubleshooting

This section assumes that you have successfully set up an Ethernet or Cheapernet network. If you have trouble completing the original set up, carefully review the steps in the above section. In particular, make sure that the network daemon *nda* is running and that the */etc/hosts* file has been created.

If the network has been running and you experience problems using *rlogin*, *telnet* or *rcp* try the suggestions given here.

1. Use the *ping*(8) utility to try to isolate the problem to a particular machine.[2] First, physically disconnect your Ethernet cable (to avoid interference) and "ping" yourself:

```
# ping 'hostname'
```

(You should be able to "ping" yourself even if the Ethernet is disconnected. If you get no output, the problem is local. Check your administration files and daemons. Most likely *nda* has either been killed or not been started. Make sure that the daemon has died before restarting it; you should never have two versions of *nda* running at the same time.

2. Use the *ifconfig* loopback driver to check your network configuration. Issue the

```
# ifconfig lo0
```

and

```
# ifconfig et0
```

commands and you should see output similar to the following:

```
# ifconfig lo0
lo0: flags=9<UP,LOOPBACK>
        inet 127.0.0.1 netmask ffffff00
# ifconfig et0
et0: flags=9<UP,BROADCAST,NOTRAILERS,RUNNING>
        inet 128.9.200.160 netmask ffffff00 broadcast 128.9.200.255
#
```

[2] *ping* prints a summary of packet transmissions every second, if the machine being "pinged" can be reached over the network. Once you have established that you can communicate with the remote machine, stop the *ping* output by typing <CTRL> C.

The commands check the configuration of your machine's loop-back driver (lo0) and Ethernet driver (et0). If you get an error or output not in the general format above, check the syntax of the *ifconfig* commands in your */etc/init.d/tcp* file. If the syntax is correct, chances are that you will not be able to fix the problem yourself. Call Stardent Customer Support for assistance.

3. Next, "ping" the remote machine:

```
# ping remotemachine
```

If you get no response, these are the likely reasons: (i) the other machine is down or incorrectly configured; (ii) */etc/hosts* has a wrong or missing address; (iii) the network is not hooked up correctly. To check the network you can run a network diagnostic. See the *Field Service Manual* or your Stardent representative for information.

If you can "ping" to a machine on your local network, but not to a machine to the other side of your gateway, either the gateway is not set up correctly or your machine does not have a route to the gateway. Use the command

```
# netstat -r
```

to get more information. *netstat -r* prints routing tables for your machine.

4. Try to *telnet* to yourself:

```
# telnet `hostname`
```

If you can, that means that your administration files are all right; the problem must be elsewhere. If you cannot *telnet* to yourself, check your local administration files and daemons.

5. Try to *telnet* to the remote machine from a third machine. If you are successful, the problem is on the local machine.

6. If you can *telnet* to another machine but cannot *rcp* files, the local machine may not have *inetd* running, may not have correct permissions set (e.g. for */etc/hosts.equiv* or *~/.rhosts*).

7. General suggestion: Your goal is to isolate the problem to a particular machine. Try *ping, telnet* and *rcp* on everyone, including yourself.

Network File System

The Network File System allows sharing of file systems and easy file access across machines. Table 21-2 shows NFS files and commands.

Table 21-2. Network File System Configuration

Item	Function	Mod. Status
Commands:		
/bin/domainname	set/print domainname	
/usr/etc/showmount	export list, remote host mounted	
/usr/etc/nfsstat	NFS statistics	
/usr/etc/rpcinfo	gives registered RPC programs	
/etc/init.d/nfs	start/stop NFS, YP, do NFS mounts	**
/usr/bin/ypwhich	display current server	
Daemons, Helpers:		
/usr/etc/mountd	serves RPC mount requests	
/etc/nfsd	serves NFS file system requests	
/etc/portmap	maps RPC nos. to Internet port numbers	
Files:		
/etc/exports	directory tree exported by local host	**
/etc/mntab	database of locally mounted file systems	
/etc/fstab	mount info	
/etc/rmtab	remote NFS mounts made to local machine	
/etc/rpc	map cmds, RPC nos., cmd aliases	
/config/libnfs.a	nfs library	

Tasks include checking to see if the network daemons are running, restarting daemons if necessary, and checking file system mounting. This section also describes how to configure the system to support Yellow Pages. Note that Stardent 1500/3000 supports Yellow Pages-specific commands, but **does not** support YP versions of standard Unix commands.

NFS: Tasks

1. Check the NFS daemons. Use the command

```
# ps -ef
```

to check that the NFS daemons */usr/etc/mountd, /etc/nfsd,* and */etc/portmap* are running. If not, kill and restart the daemons by issuing the commands:

```
# /etc/rc2.d/K40nfs
# /etc/rc2.d/S40nfs
```

Report any problems (with a core file, if one exists) to Stardent customer support.

2. Check file system mounting. Make sure that you can mount a remote file system using the *mount*(1) command. For instance,

```
# mount -f NFS remotehost:/usr/man /usr/man
```

(See the *Network File System Manual* for details.) You can then run the *mount* command to make sure that the file system has been mounted.

Add any remote file systems you wish to mount regularly to the file */etc/fstab*. File systems in this file are mounted automatically upon entry to run level 2. For the format of the */etc/fstab* file, see *fstab*(4) in the *Programmer's Reference Manual.*

3. Create /etc/exports. Create an */etc/exports* file containing the names of the file systems that you intend to permit remote hosts to mount. Each line of the file contains a file system name. Here is an example:

```
/
/work
/usr/man
/moose
/squirrel
```

NFS: Troubleshooting

(1) If you have trouble mounting a remote file system, try to mount a file system on your own machine:

```
# mount -f NFS localhost:/usr/man /usr/man
```

If you do not succeed, your local NFS daemons */usr/etc/mountd*, */etc/nfsd* and */etc/portmap* may not be running. Check them, as described above.

(1) Use the command

```
# /usr/etc/rpcinfo -p localhost or remotehost
```

You should see at least two entries: One with program number 100005 for mountd and one with program number 100003 for nfs. If you don't your daemons may not be running. Check them, as described above.

(1) Follow the network troubleshooting suggestions given above in this chapter. If the network is running and you can mount local file systems but not remote ones, the problem is probably with the daemons on the remote machine.

Yellow Pages Configuration

This section describes the tasks needed to configure Stardent 1500/3000 as a Yellow Pages client. Note that Stardent 1500/3000 supports Yellow Pages-specific commands, but **does not** support YP versions of standard Unix commands. Table 21-3 summarizes configuration information.

Table 21-3. Yellow Pages Configuration

Item	Function	Mod. Status
Commands:		
/etc/init.d/nfs	start/stop NFS, YP, do NFS mounts	**
/usr/etc/ypstart	set YP domain, start daemons	**
/usr/etc/rpcinfo	list registered RPC programs	
/bin/domainname	set/display YP domain name	
/usr/bin/ypcat	print YP map file	
/usr/bin/ypmatch	print entry in YP map file	
/usr/bin/ypwhich	display current YP server	
Daemon:		
/etc/ypbind	broadcast YP bind request	
Files:		
/etc/hosts	hosts known by local machine	
/etc/yp	directory of YP commands and database	

Note that for Yellow Pages to work the network and NFS must be up and running.

1. Check /etc/init.d/nfs. Make sure that the call to *ypstart* is uncommented.

2. Check /usr/etc/ypstart. Check the file */usr/etc/ypstart* to see that the call to

```
/etc/ypbind
```

is started.

3. Check domainname. Use the command

```
# domainname
```

to see if *domainname* is set to your local YP domain.

4. Try to bind to a YPserver. Issue the command

```
# ypwhich
```

If you can bind to a YP server, the command gives the name of a server as output.

5. See if the YP link has been made. To see if the YP link has been made, issue these commands:

```
# ypcat passwd.byname
# yppoll passwd.byname
```

If the YP link has not been made, try the following:

(1) Make sure that the local daemons *ypbind* and *portmap* have been started.

(2) Make sure that all the daemons on the remote host are running. Run the command

```
# rpcinfo -p host
```

to be sure that the remote host daemons *portmap*, *ypbind*, and *ypserv* are running.

(1) Kill off and restart daemons on the local and remote systems as necessary. The NFS and YP daemons must be killed and restarted as a group. Use */etc/rc2.d/K40nfs* and */etc/rc2.d/S40nfs* to kill and restart the daemons.

COMMUNICATIONS

CHAPTER TWENTY–TWO

This chapter describes the *sendmail* and *uucp* inter-machine communications utilities.

sendmail

The *sendmail* program packages up user mail for delivery from your network (or domain) and includes with the mail a header packet that can be decoded by the receiving domain. *sendmail's* purpose is to route mail to a particular mailer such as Ethernet or *uucp*. Table 22-1 gives *sendmail* configuration commands and files.

The procedure in this section assumes that your Stardent 1500/3000 system is an end node of a network and that another intelligent UNIX mailer on your network handles mailings to external locations. If this is not the case, see the document references in the Preface for background reading and configuration instructions. Note that Stardent 1500/3000's implementation is the standard BSD 4.3 version of *sendmail*.

This section describes the minimal steps necessary to install *sendmail* on your machine:

- Modify the configuration file, */usr/lib/sendmail.cf;*
- Examine and modify (if necessary) the aliases file, */usr/lib/aliases;*
- Start the daemon that controls the *sendmail* program.

If you wish to collect statistical information about mail traffic you can also create the file */usr/lib/sendmail.st*.

Table 22-1. sendmail Configuration[1]

Item	Function	Mod. Status
Commands:		
/usr/ucb/newaliases	implement aliases in aliases file	
/usr/ucb/mailq	lists mail queue	
Files:		
/etc/init.d/bsd	init run commands	
/usr/lib/sendmail	binary sendmail program	
/usr/lib/sendmail.cf	sendmail configuration file (text)	**
/usr/lib/sendmail.fc	frozen sendmail configuration	
/usr/lib/sendmail.st	sendmail statistics file	
/usr/lib/sendmail.hf	help file	
/usr/lib/cf/lanleaf.cf	linked to sendmail.cf	
/usr/lib/cf/lanroot.cf	config file for uucp	*
/usr/lib/sendmail.mail.cf		
/usr/lib/sendmail.subsid..	config file for subsidary machines	*
/usr/lib/aliases	sendmail aliases file	**
/usr/lib/aliases.dir	aliases file (dbm format)	
/usr/lib/aliases.pag	aliases file (dbm format)	
/usr/spool/mqueue/..	directory for mail queue, temp files	
/usr/spool/mqueue/qf..	control (queue) files for messages	
/usr/spool/mqueue/df..	data files	
/usr/spool/mqueue/lf..	lock files	
/usr/spool/mqueue/tf..	temp versions of qf files	
/usr/spool/mqueue/nf..	used when creating a unique id	
/usr/spool/mqueue/xf..	transcript of current session	

sendmail: Tasks

1. Modify the configuration file. Making changes to */usr/lib/sendmail.cf* is full of pitfalls. Be very careful and do not attempt to make any changes other than the ones described here. Always work on a copy of the file, and only copy it back when you are comfortable that all changes have been made correctly.

This task describes how you configure *sendmail* for a local area network. To configure *sendmail* for *uucp* connections, make analogous changes to the file */usr/lib/cf/lanroot.cf*.

[1] The "modification status" column in the table shows whether or not the file requires modification. If the modification status box is empty, no modification is recommended. (Commands fall in this category.) If the box has a single asterisk (*), modification is optional. If the box has a double asterisk (**), the file or variable must be examined (and modified if necessary).

The *sendmail.cf* file is divided into sections, each delimited by a banner comment. (Comment lines have pound sign prefixes.) The assignments in the file all begin with capitalized letter or word prefixes. The first portion of the file (the portion requiring modification) looks something like this:

```
############################################################
############################################################
#####
#####           SENDMAIL CONFIGURATION FILE
#####
#####   Prototype for hosts connecting only to a local ethernet
#####
#####   This file should be used for a leaf node on a local area
#####   network. The UUCP node, which connects to the UUCP network,
#####   should use the file lanroot.cf
#####
############################################################
############################################################

############################################################
####           MODIFICATION OF THIS FILE
####
####    You must tailor this file before it can be used.
####
####    Please find all the locations of "XXX" and replace the
####    "XXX" string with appropriate string as described by
####    the local directions.
####
############################################################


############################################################
###     local info
###
############################################################


#Domain.
#Replace XXX, the domain in user@host.XXX, with your domain name.
DDXXX

#Domain class.
#Replace XXX with list of names for domain, including main name
CDLOCAL XXX XXX

#host on LAN with UUCP (or other) connection
#Replace XXX with hostname of the machine on your net which is
#handles the UUCP connection
DRXXX

#Domain extension.
#XXX may need modification.
#Replace the extension, the extension in user@host.domain.XXX,
```

```
#with the extension in your domain.
DEcom
CEcom ether
```

Change only the (uncommented) items marked with *XXX*. Do not change the capitalized prefixes; they indicate the type of assignment being made.

Domain: Change the *domain name (DD* prefix) to your company or organization name. For instance,

```
DDstardent
```

Domain class: Replace the triple X's following *CDLOCAL* with names for your domain. for instance,

```
CDLOCAL stardent titan Stardent
```

UUCP: Replace the triple X's following the prefix *DR* with the machine on your network that is to be the node for any *uucp* connections. The name you select must also be included in *uucp* node is *moose,* change the entry to DRmoose.

```
DRmoose
```

Alternatively you can use the generic name *mailhost* within the *sendmail.cf* file

```
DRmailhost
```

provided that *mailhost* is then listed as an alias in */etc/hosts.*

```
128.8.12.44     moose   moose.stardent.com   mailhost
```

Domain extension: DARPA specifies values for the Domain extension assignment, given in the example above as *DEcom.* and *CEcom ether*. Accepted values are *com* for commercial, *edu* for education, and *gov* for government. *(ether* refers to an Ethernet network.) If you plan to communicate solely within your own installation, the choice of value is not important. If you plan to communicate outside your installation, see the appropriate DARPA documentation, which can be obtained from the Network Information Center at SRI International, in Menlo Park, California.

2. Examine the aliases file. The */usr/lib/aliases* file lets you define aliases that users can then substitute for actual user mail addresses. Most of the file is optional, in the sense that you can choose to assign aliases or not. The only alias that is *required* to be assigned is *Postmaster,* which must be mapped to an actual login or logins on your system.

Here is an example */usr/lib/aliases* file.

```
##
#  Aliases can have any mix of upper and lower case on the left-hand side,
#       but the right-hand side should be proper case (usually lower)
#
#       >>>>>>>>>>      The program "newaliases" will need to be run after
#       >> NOTE >>      this file is updated for any changes to
#       >>>>>>>>>>      show through to sendmail.
#
#       @(#)aliases 1.1 85/12/19 ACC
##
# Following alias is required by the mail protocol, RFC 822
# Set it to the address of a HUMAN who deals with this system's mail problems.
Postmaster: Bullwinkle

# Alias for mailer daemon; returned messages from our MAILER-DAEMON
# should be routed to our local Postmaster.
MAILER-DAEMON: postmaster

# Aliases to handle mail to programs or files, eg news or vacation
decode: "|/usr/bin/uudecode"
nobody: /dev/null
#
# Aliases for interest lists
#
all: :include:/usr/alias/all
#
# Aliases for a person, so they can receive mail by several names:
#
alan:aab, ast
#
# Aliases for mail delivery to home machine
#

bugs:bugs@flash

aab:aab@flash
al:al@flash
ben:ben@flash
bill:bill@flash
```

The alias for *Postmaster* has been defined as Bullwinkle. Change the value to that of a valid user login on your system.

If you wish to assign other aliases such as those given in the example file, keep in mind these conventions: The general format for alias file entries is *alias:name*

where *alias* is your chosen alias and *name* is the actual name of the user or users. Within this structure the following assignments are legal:

alias:login	Assigns an alias to a user login.
alias:filename	Assigns an alias to *filename*, where the file is identified by an absolute pathname. Mail addressed to the alias is appended to the file.
alias:user1,user2,..	Assigns an alias to a comma-separated list of user logins, given as *user1, user2,* etc. The effect of this assignment is that all users on the list get mail sent to the alias. If more than one line is needed to list the user logins, no newline escape key is needed.
alias:user@somewhere	Assigns an alias to a user login on the machine named *somewhere.*
alias::include:filename	Assigns an alias to the names given in *filename* (absolute pathname). Use this form if you want to use a separate file to list the users assigned to a given alias.
alias:"|filename"	Pipes the mail received by alias to the file given by *filename* (absolute pathname).

You can see examples of each of these assignment types in the sample aliases file given above.

After you have made any changes to the aliases file, you must implement the changes by issuing the *newaliases*command:

```
# newaliases
```

3. Start the sendmail daemon. The following lines, commented out by pound signs, are including in your */etc/init.d/bsd* file.

```
#if [ -f /usr/lib/sendmail ]; then
#       (cd /usr/spool/mqueue; rm -f [lnx]f*)
#       /usr/lib/sendmail -bd -q1h &
#       echo 'Starting sendmail.' >/dev/console
#fi
```

If you want to start the sendmail daemon, remove the pound signs from the beginning of these lines. Now, every time you boot the machine, *sendmail* is started automatically.

4. Create /usr/lib/sendmail.st. The */usr/lib/sendmail.st* file collects statistics about mail traffic. Create it this way, if you want it:

```
# cp /dev/null /usr/lib/sendmail.st
# chmod 666 /usr/lib/sendmail.st
```

This file does not grow by itself.

Notes About the Stardent 1500/3000 Implementation of sendmail

Note 1. The *sendmail* configuration file, *sendmail.cf,* must recognize inbound local mail. It uses the *$W* macro in this process. *$W* is set by *sendmail* by looking up your system's Internet address in */etc/hosts* and taking the first name entry on the line. For instance, if your system's Internet address is 129.8.12.44 and the corresponding line in */etc/hosts* is

```
128.8.12.44     moose   moose.stardent.com
```

sendmail uses *moose* as the local address.

Some networked systems have */etc/hosts* files in which the full multi-level system name is listed before the local version; for instance

```
128.8.12.44     moose.stardent.com      moose
```

If your system's */etc/hosts* file is configured this way, *sendmail* becomes confused about where to send local mail. There are two possible solutions.

(1) Change */etc/hosts* so that the local name is listed first (we recommend this, if possible, to avoid changing the *sendmail.cf* file).

CAUTION
Modifying the *sendmail.cf* file is fraught with danger. Please call Stardent Customer Support at 1-800-537-1104 before attempting

(2) It is possible to modify the *sendmail.cf* file by adding a new macro and substituting for *$W* in the rulesets. Before attempting this, contact Stardent Customer Support for assistance.

Note 2. Stardent 1500/3000 does not current support 4-level or higher domains. If you need to construct 4-level or higher domains, call Stardent Customer Support for assistance.

the chan

uucp

The *uucp* family of commands is used for RS-232 port communications via modem or hardwired connection. Table 22-2 lists *uucp* commands, daemons, and files.

Table 22-2. uucp Configuration

Item	Function	Mod. Status
Commands:		
Files:		
/etc/inittab	actions for the init process	**
/etc/gettydefs	actions for the getty process	**
/usr/lib/uucp/Devices	uucp port/modem descriptions	**
/usr/lib/uucp/Dialers	communications line set-up info	**
/usr/lib/uucp/Systems	communications info for uucico daemon	**
/usr/lib/uucp/Dialcodes	phone abbrevs. for Systems file	*
/usr/lib/uucp/Permissions	permissions for remote machines	*
/usr/lib/uucp/Poll	times to call remote machine	*
/usr/lib/uucp/Sysfiles	assigns substitute uucp files	*
/usr/lib/uucp/Maxuuxqts	max number of uuxqt programs	*
/usr/lib/uucp/Maxscheds	max number of uusched programs	*
/usr/lib/uucp/remote.unknown	for unaccepted remote logins	
/usr/lib/uucp/.Admin/Foreign	accepts msgs from remote.unknown	
/usr/spool/TM..	temporary data files	
/usr/spool/LCK..	lock files	
/usr/spool/C..	work files for queuing of remote jobs	
/usr/spool/D..	data files	
/usr/spool/X..	pre-execution files	
Daemons:		
/usr/lib/uucp/uucico		
/usr/lib/uucp/uusched		
/usr/lib/uucp/uuxqt		

The commands, *cu, ct, uucp* and *uuto* are all documented in the *Commands Reference Manual;* this section describes how you set up the system to use them. Note that Stardent 1500/3000's implementation of *uucp* is the AT&T standard implementation. For additional information refer to standard AT&T or commercially available documentation.

uucp: Tasks

1. Edit /etc/inittab. The *cu* command allows you to use a modem either to answer a remote login attempt to your machine or to login yourself on a remote machine. This bidirectional feature depends on a program named *uugetty*(1M), which spawns a specialized *getty* process.

To execute *uugetty*, include a line such as the following (for tty port 1) in the */etc/inittab* file:

```
t1:23:respawn:/usr/lib/uucp/uugetty -r -t60 tty1 1200
```

A modem should be attached to the terminal line you specify (in the example, *tty1*). The *–r* option is essential to prevent a waiting *uugetty* on the remote machine from colliding with the *uugetty* on your machine and looping forever.

Check also to see that you have an entry in */etc/gettydefs* for your modem.

2. Check Database Files: /usr/lib/uucp/Devices. The */usr/lib/uucp/Devices* file contains descriptions for every port or modem that *cu* or *uucp* uses. A typical *Devices* looks like this:

```
flash tty1 - 9600 direct
ACU tty2 - 2400 hayes
ACU tty2 - 1200 hayes
ACU tty2 - 300  hayes
```

There are five space-separated fields:

```
type line line2 class dialer-token-pairs
```

type
the name of the machine if a direct line; *ACU* (automatic calling unit) if a modem line; *direct* if a direct link for *cu* connections only.

line
the name of the terminal line (port) in the */dev* directory.

line2
a hyphen (–) if *not* an 801-type *ACU;* if an 801-type dialer, the name of the 801-type device.

class
the baud rate of the device; may take a prefix to differentiate between classes of dialers.

dialer-token-pairs
direct if a direct connection; the name of the modem, if the modem is directly connected to the port on your machine (must match a name in the first field of the *Dialers* file described later).

The first line in the sample describes a direct connection (hardwired) to a machine named *flash.* The 3 entries for *tty2* describe just one modem: the different speed arguments permit you to use the modem at any of those speeds.

3. Check Database Files: /usr/lib/uucp/Dialers. The */usr/lib/uucp/Dialers* file contains information that is passed along the communications line *before* you can actually use the line to transfer data. The information that is passed is a sequence of ASCII strings that is transmitted from or expected to be received by your Stardent 1500/3000. A *Dialers* file for the *Devices* file given earlier might look like this:

```
direct
hayes =,-, "" \dAT\r\c OK\r \EATDT\T\r\c CONNECT
```

The entries follow this format:

```
dialer substitutions expect-send ...
```

dialer
This field must match the fifth field of an entry in the *Devices* file; *direct* indicates a direct connection and requires no further entries on the line.

substitutions
This field is a translate string that follows the format expected by the *tr*(1) command; for each pair of characters, the first is mapped to the second. In the example, the = is translated to a comma and the – is also translated to a comma; this dialer requires a comma for "wait to send" and "pause."

expect-send
The remaining fields specify the character strings the dialer waits to receive from the called device (expects) and what the dialer sends when it receives the expected data.

Here are the *expect-send* sequences for the modem in the example.

```
"" \dAT\r\c OK\r \EATDT\T\r\c CONNECT
```

The first *expect* string is " ". This means "Don't wait for any data. Send the next string at once." The following *send* string is *\dAT\r\c*. This means "Send a delay(*\d*) of approximately 2 seconds. Then send the string *AT* and a carriage return (*\r*) and send no new-line (*\c*)."

The next *expect* string is *OK\r*. This means "Wait to receive the string *OK* and a carriage return." The next *send* string means "If you received the string you expected, now enable echo checking (*\E*) and send the string *ATDT* followed by a phone number or a token with *Dialcodes* translation (*\T*) and a carriage return and no new-line.

Finally, the dialer expects to receive the string *CONNECT*. If everything works properly, the connection *between the modems* is now established. To establish the connection between the *machines*, you need to invoke *uucp*.

A full list of the escape sequences (*\x*) you can use is given in the discussion of the *Systems* file in the next section. Once the *Devices* file is modified, you have essentially installed *cu*.

4. Check Database Files: /usr/lib/uucp/Systems. The */usr/lib/uucp/Systems* file contains the information the *uucico* daemon needs to establish the communications link between your machine and a remote machine. Each entry in the file represents a machine that can be called by your machine. More than one entry for each remote machine is permitted. Additional entries represent alternative communications paths that are tried in sequential order.

The following space-separated fields are present in each entry:

```
machine-name time type class phone login
```

machine-name
The node name of the remote machine.

time
The time of day when connections between the machines are permitted. The format of this entry is:

```
time[;retry]
```

The *time* may be specified as a day of the week (*Su, Mo, Tu, We, Th, Fr, Sa*), a weekday (*Wk*), any day (*Any*), or *Never* to prevent your machine from ever trying to establish a connection (your machine may still answer a call initiated by a remote machine).

You may also specify the hours of the day during which a call may be placed. If you don't, calls may be placed any time. To specify a time of day, use a range such as *0800-2300* to mean from 8:00 a.m. through 11:00 p.m. A time span of 0 is permitted.

An optional subfield, *retry*, permits you to specify the amount of time in minutes to wait before a retry. The default is 60 minutes.

Here are a couple of examples:

```
Any2301-0759, SaSu0800-1659
Wk0900-1700;10
```

The first means any day, Sunday through Saturday, from 11:01 p.m. through 7:59 a.m. and Saturday and Sunday, from 8:00 a.m through 4:59 p.m. The second means any weekday between 9:00 a.m. and 5:00 p.m., with a retry interval of 10 minutes.

type
This field is matched against the first field of the *Devices* file to identify the type of device that you are using to establish the connection with the remote machine.

class
The transfer speed of the device making the connection. This field is matched against the *class* field in the *Devices* file.

phone
The phone number is an optional alphabetic abbreviation, as specified in the *Dialcodes* file (see the next section) and a numeric part. In this string, an equals sign (=) tells the ACU to wait for a secondary dial tone before dialing the remaining digits; a hyphen (–) tells the ACU to pause 4 seconds before dialing the next digit.

login
This field contains login information as a series of fields and

subfields in the *expect-send* format described for the *Dialers* file. A number of escape sequences are understood by *uucp*:

\N	a null character (ASCII NUL)
\b	a backspace character
\c	suppress new-line if at end of string; otherwise ignored
\d	delay 2 seconds
\p	pause for about 1/4 to 1/2 second
\E	begin echo checking
\e	end echo checking
\n	a new-line character
\r	a carriage return
\s	a space character
\t	a tab character
\\	a \ character
EOT	two EOT new-lines
BREAK	a break character
\K	same as BREAK
\ddd	collapse octal digit *ddd* to single character

Here's an example of what the *Systems* file might look like:

```
seismo Any2301-0759 ACU Any ORD1234 "" \r ogin:--ogin: aab \
ssword: aab789aab
boss Any flash 9600 - ogin:--ogin: aab ssword: helpout
```

The *ORD1234* entry in the first example is explained in the next section.

In the second example, the *type* field contains the name of your machine as it appears in the *Devices* file. This is how you establish a connection between two machines that are hardwired to each other. You can't use *direct* in this field because even a direct connection must be a machine with a node name.

Also in the second entry, the fifth field contains only a hyphen (–). This is place holder and is not interpreted.

The tricky part of setting up this file is getting the *expect-send* fields written properly. Look at the *expect-send* pairs in the first example:

```
"" \r ogin:--ogin: aab ssword: aab789aab
```

The pair of double quotes (" ") means "Don't wait for anything." The *\r* sends a carriage return to the remote machine.

Here's the trickiest part:

```
ogin:--ogin:
```

Now your machine is going to expect to see the string *ogin:* coming from the remote machine. These are the last 6 characters of the remote machine's login prompt. Why not use *login:*? Because the remote machine may use *Login:* and you would never be able to login. The two dashes next to the first colon mean "Wait 30 seconds and then send the next string." This allows for the possibility that the remote machine prints a long login message and when your machine first tries to read it, your machine does not get the string it expects. By waiting 30 seconds, your machine gives the remote machine enough time to print the entire login message, ending with *ogin:*.

When your machine receives the expected login message, it sends your login account name, in this case *aab*. Then it waits to receive the password prompt (*ssword:*). When your machine gets that, it sends your password, *aab789aab*. If your password is accepted, the job is done.

5. Check Database Files: /usr/lib/uucp/Dialcodes. The */usr/lib/uucp/Dialcodes* file contains abbreviations that can be used in the *phone* field of the *Systems* file. Each entry follows this format:

```
abbr dial-sequence
```

Here's an example:

```
ORD 1=312555
LAX 1=213888
HO 1=801222
```

Now, recall the entry in the *Systems* file:

```
seismo Any2301-0759 ACU Any ORD1234 "" \r ogin:--ogin: aab \
ssword: aab789aab
```

The entry in the *phone* field is expanded to *1=3125551234* and sent to the dialer *if* the token *\T* is present in the appropriate *expect-send* field of the *Dialers* file. Here's the entry from the *Dialers* file:

```
hayes =,-, "" \dAT\r\c OK\r \EATDT\T\r\c CONNECT
```

The *\T* follows the *ATDT* string. At this point, the phone number is sent.

6. Check Database Files: /usr/lib/uucp/Permissions. The */usr/lib/uucp/Permissions* file specifies the permissions available to remote machines with respect to login, file access, and command execution. The file consists of an *entry* made up of several *options*. An *option* takes the following form:

```
NAME=value
```

Each *entry* is a logical line and physical lines are terminated with a \ to indicate continuation. Each *option* pair is separated by a space and no white space is allowed within the *option* list. Colons separate multiple *value*s within an *option*. Blank lines are ignored and comment lines begin with #, continuing until the first unescaped new-line.

An *entry* may be either *MACHINE* or *LOGNAME*.

MACHINE
Permissions that take effect when your machine calls a remote machine. This is the names of remote machines that your machine may call. For example,

```
MACHINE=alpha:beta:gamma
```

LOGNAME
Permissions that take effect when a remote machine calls your machine. This is the login account name for remote machines that call your machine. For example,

```
LOGNAME=okuucp
```

Option names are entered in all uppercase as the left-hand side of a *NAME=value* pair. *Values* may be set or you can accept the default. Here are the available *options:*

REQUEST
Allows the remote machine to initiate a file transfer. Default is no.

```
REQUEST=yes
```

This is a legal *option* for either a *LOGNAME* entry or a *MACHINE* entry.

SENDFILES
Allows the remote machine to take files that your machine has queued up for it. Default is call.

```
SENDFILES=call
```

The *call* specification means that the files queued in your machine will be sent only when your machine calls the remote machine. The *yes* specification permits your machine to send the files as long as the remote machine logged in as one of the names in the *LOGNAME* entry. This *option* is only valid in a *LOGNAME* entry.

READ and *WRITE*
Specify the parts of the file system that the *uucico* process can read from or write to. Default for both is */usr/spool/uucp/uucppublic*.

```
WRITE=/usr/spool/uucp/uucppublic:/usr/news
```

If you change the default in any way, you must specify all pathnames because additional pathnames are not added to the list.

NOREAD and *NOWRITE*
Specify exceptions to the *READ* and *WRITE* default settings. There is no default setting.

```
READ=/ NOREAD=/etc WRITE=/usr/spool/uucp/uucppublic:/usr/news
```

The exception value is recursive.

CALLBACK
Specify that the remote machine cannot execute any transaction until your machine calls back. Default is no.

```
DEFAULT=yes
```

This *option* is used for *LOGNAME* entries.

COMMANDS
Specifies the commands for which *uux* will generate remote execution requests and send to the queue for transfer to the remote machine. Default is *rmail.*

```
MACHINE=alpha:beta:gamma \
COMMANDS=rmail:/usr/local/lp
```

This entry permits remote machines *alpha, beta,* and *gamma* to execute the commands specified. This *option* is not used in the *LOGNAME* entry. The specification *ALL* permits remote machines to execute any command. This is not recommended.

VALIDATE
Specify commands identified with the *COMMANDS option* that require a unique login or password for *uucp* transactions. For example,

```
LOGNAME=okuucp VALIDATE=alpha:beta:gamma
```

specifies that if one of the machines named *alpha, beta,* or *gamma* logs in on your machine, it needs to have used the login *okuucp.*

This *option* links the *MACHINE* entry and its *COMMANDS option* with the *LOGNAME* entry. This link is important because the execution daemon (*uuxqt*) is not running while the remote machine is logged in. Execution files from the remote machine are put in its spool directory (*/usr/spool/uucp/remote_machine_name*) when the files are received by your machine. When the files are executed by the *uuxqt* daemon, it uses the spool directory name as an index into the *Permissions* file and searches that file for a matching *MACHINE* entry. The daemon gets the *COMMANDS option* list from that entry. For example:

```
MACHINE=alpha:beta:gamma REQUEST=yes \
COMMANDS=rmail:/usr/lbin/rnews \
READ=/ WRITE=/

LOGNAME=okuucp VALIDATE=alpha:beta:gamma \
REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

When you call remote machine *alpha,* you are giving *alpha* permission to put files in its spool directory for execution on your machine. But any machine could attempt to login as *alpha* on your machine. Before you permit a remote machine to execute any files on your machine, you want to make sure that it sends you the privileged password, *okuucp.*

You may combine *MACHINE* and *LOGNAME* entries if the *options* are the same. For example:

```
MACHINE=alpha:beta:gamma REQUEST=yes \
LOGNAME=okuucp SENDFILE=yes \
  READ=/ WRITE=/
```

7. Check Database Files: /usr/lib/uucp/Poll. The */usr/lib/uucp/Poll* file contains information specifying when a remote machine should be called. The format of the entries is

```
system<TAB>hour1 hour2 ... hourN
```

You *must* use a tab character to separate the system name from the hour list.

```
alpha    0 4 8 12 16 20
beta     0 5 10 15 21
gamma    2 10 18
```

The *uudemon.poll* script reads the *Poll* file and sets up a polling work file (always named C.*machine*), in the spooling directory for the remote machine. The scheduler *uusched* reads the work file to determine when to execute the files.

8. Check Database Files: /usr/lib/uucp/Sysfiles. The */usr/lib/uucp/Sysfiles* file permits you to assign different files to be used by *uucp* and *cu* as *Systems*, *Devices*, and *Dialers* files. You might do this to split a large *Systems* file in smaller, more manageable files or to use different *Dialers* files to use different handshaking for *cu* and *uucp*.

The format of the entries is:

```
service=name-list        systems=systems-file-list \
                         dialers=dialers-file-list \
                         devices=devices-file-list
```

The *name-list* may contain *uucico* or *cu* or both separated by a colon. The *file-lists* are also colon-separated lists of files that are searched in the order given for the appropriate information. If a full pathname is given, it is used for the search; if only a file name is given, the default directory */usr/lib/uucp* is searched. For example:

```
service=uucico:cu        systems=Systems:Local_Systems
                         dialers=Dialers:Local_Dialers
```

In this example, the default *Systems* and *Dialers* files are searched first and then, if appropriate entries were not found, two additional files are searched, *Local_Systems* and *Local_Dialers*.

9. Check Database Files: /usr/lib/uucp/Maxuuxqts. The */usr/lib/uucp/Maxuuxqts* file is an ASCII file containing the maximum number of *uuxqt* programs that can run at the same time. A reasonable value is 5.

You need to create this file if you want to establish a limit. To create the file:

```
# cat 5 >/usr/lib/uucp/Maxuuxqts
# chown uucp /usr/lib/uucp/Maxuuxqts
```

10. Check Database Files: /usr/lib/uucp/Maxuuscheds. The */usr/lib/uucp/Maxuuscheds* file is an ASCII file containing the maximum number of *uusched* programs that can run at the same time. A reasonable value is 2.

You need to create this file if you want to establish a limit. To create the file:

```
# cat 2 >/usr/lib/uucp/Maxuuscheds
# chown uucp /usr/lib/uucp/Maxuuscheds
```

11. Check /usr/lib/uucp/remote.unknown.
The */usr/lib/uucp/remote.unknown* file is an executable shell script that sends a message to */usr/spool/uucp/.Admin/Foreign* whenever a remote machine not listed in your *Systems* file attempts to login. You can disable this feature by removing execution permission from the file:

```
# chmod 000 /usr/lib/uucp/remote.unknown
```

This does not allow the unknown machine to login, but simply turns off the logging.

12. Check uucp daemons: uucico. The functions of the *uucico* daemon are to select the device used for the communications link, establish the link with the remote machine, execute the login sequence and permission checks, transfer data and execute files, log results, and notify the user by *mail*(1) that transfers are complete.

The *uucico* daemon is executed by *uucp, uuto,* and *uux* to call the remote machine after all the required files have been created. It is also executed by the *uusched* and *Uutry* programs.

Here is an example of the code from an */etc/init.d/uucp* file that starts this daemon:

```
#! /bin/sh
#       Clean-up uucp locks, status, and temporary files

set `who -r`
if [ $9 = "S" ]
then
        /bin/rm -rf /usr/spool/locks/*
fi
```

When you start the *uucico* daemon, you need to remove any old *LOG* or *LCK* files, otherwise the daemon won't start.

13. Check uucp daemons: uuxqt. The *uuxqt* daemon performs remote execution requests. It searches the remote machine's spool directory for files named *X.name*, and, when it finds one, it opens it to get the list of data files (*D.files*) required for execution. Then it checks access and permission settings for the data files and then checks the *Permissions* file to verify execute permission for the requested command.

The *uuxqt* daemon is started by the */usr/lib/uucp/uudemon.hour* shell script:

```
# /usr/lib/uucp/uudemon.hour
/usr/lib/uucp/uusched &
/usr/lib/uucp/uuxqt &
```

This script, in turn, can be executed from a *crontab* file, with the following entry:

```
39,9 * * * * /bin/su uucp -c "/usr/lib/uucp/uudemon.hour" > /dev/null
```

This runs the program at 9 and 39 minutes past the hour, every hour.

14. Check uucp daemons: uusched. The *uusched* daemon schedules the work that is queued in the spool directory. The daemon is also started by the *uudemon.hour* script, like *uuxqt*.

15. For your information: Administrative Files. Several other administrative files provide temporary work and data space and lock files.

TM files

These are temporary data files created under the spool directory for the remote machine when a file is received from that machine. The file names follow this format:

```
TM.pid.ddd
```

The *pid* is the process ID and *ddd* is a sequential 3-digit number. When the entire file is received, the file is moved to the pathname specified in the *C.* file described below. The file may remain in the spool directory if the system terminates abnormally. The *uucleanup* program removes these files.

LCK files

These are lock files that are created in the */usr/spool/locks* directory for each calling device in use. They prevent duplicate conversations and multiple attemps from the same device. The format of the file names is:

```
LCK..str
```

The *str* is either a device or machine name. The file contains the process ID of the process that created the lock. They may remain in the spool directory if the link terminates abnormally. Lock files are removed after the parent process is no longer active.

C. files

These are work files that are created in a spool directory when work has been queued on your machine for a remote machine. The format of the file names is:

```
C.nameNxxxx
```

The *name* is the name of the remote machine; the *N* is an ASCII character representing the job priority; and the *xxxx* is job sequence number assigned by *uucp*. The work files are readable ASCII text and contain the following information:

- Full pathname of the file to be sent or requested.
- Full pathname of the destination or user name.
- User login name.
- List of options.
- Name of associated data (*D.*) file in the spool directory.
- Mode bits of the source file.
- Remote machine login name. Used for notification of job completion.

D. files

These are data files that are created when the source files are copied to the spool directory. The format of the file names is:

```
D.namexxxxyyy
```

The *name* is the first 5 characters of the remote machine name; *xxxx* is a job sequence number assigned by *uucp*; and *yyy* is a sub-sequence number in case several *.D* files were created for a single work (C.) file.

X. files

These files are created in the spool directory prior to execution of remote commands. The format of the file names is:

```
X.nameNxxxx
```

The *name* is the name of the remote machine; the *N* is an ASCII character representing the job priority; and the *xxxx* is job sequence number assigned by *uucp*. The work files are readable ASCII text and contain the following information:

- Login and name for machine making the request.
- Name of files required for execution.
- Standard input to the command string.
- Machine and file name to receive standard output from command execution.
- Command string.
- Option lines for return status requests.

16. Start the uucp daemons. Three daemons perform all the work in the *uucp* programs. They run as background processes and handle file transfers and command execution. If you need to, you can also start a daemon from the shell.

Use the command

```
# ps -ef
```

to see if the uucp daemon *uucico* daemon is running. If not, check the file */etc/init.d/uucp* make sure the appropriate lines are uncommented. If you are restarting the daemons, remove any old LOG and LCK files. If you don't the daemon won't start. You may also choose to start the *uuxqt* and *uusched* daemons, if desired. They are not necessary, however, to get uucp running.

Stardent

ADMINISTRATION

Section 3

SYSTEM PROCESSES

CHAPTER TWENTY–THREE

Four processes connect a user to the UNIX operating system.

- *init*
  is a general process spawner invoked by the UNIX kernel as the last step in the operating system boot procedure.

- *getty*
  is a process invoked by *init* that enables a line on which a user may login from a terminal or via an RS-232 connection.

- *login*
  is a process that issues a prompt for a password, evaluates the user's response, and, assuming the password is correct, starts the user's shell, *(/bin/sh* or */bin/csh)*.

- */bin/sh* or */bin/csh*
  is a process that executes a user's *.profile* or *.login* and *.cshrc* and completes the connection to the system.

This chapter describes these processes in some detail, referring to the various files on the system that influence or control the outcome of the connection.

System State

At any given time during operation, a UNIX system operates in a state. There are several synonyms for state: run level, run state, run mode, init state, and system state. They all mean the same thing: the current operating level of the system. Each state precisely defines the operation of the Stardent 1500/3000. Table 23-1 lists and describes the states.

Table 23-1. Operating System Run Levels

Run Level	Description
0	Power-down state.
1, s, or S	Single-user mode. Used to install or remove software utilities, backup and restore files and file systems, and check file systems. Both *s* or *S* and *1* initialize the system to single-user mode, but *s* only kills processes spawned by *init* and does not unmount file systems. State *1* unmounts everything except the *root* file system and kills all user processes except those related to the console.
2	Multi-user mode. The normal operating mode for the system. By default, the *root* and the */usr* file systems are mounted.

The init Process

Operating system state is determined and set by the *init* process. The *init* process is the first general purpose process created by the system at boot time. Process 0 is the kernel swap process. *init* is process number 1.

The first thing the *init* process does is scan the */etc/inittab* file. This file contains the instructions *init* uses to create a usable system. The entries in the */etc/inittab* take the following format:

id:level:action:process

The four fields are separated by colons and may take the values given below:

- *id*
  is a one or two character name that uniquely describes an entry.
- *level*
  is 0 or more digits (*0,1,2,3,s,* or *S*). If *level* is null, the *action* is valid in all levels.
- *action*
  is one of nine possible entries that indicates what is to happen on entry to or exit from the given *level*. The selections are:
  - *sysinit*
    run the named *process* before the *init* process sends anything to the system console (namely, the console login prompt).

- *bootwait*
  starts the named *process* the first time *init* goes from single-user to multi-user state after the system is booted. *init* starts the process, waits for its termination, and, when it dies, does not restart the process. If run level for *initdefault* (below) is set to 2, the process runs right after the boot.
- *wait*
  starts the named *process* when going to the specified *level* and waits until it is finished.
- *initdefault*
  enters the specified *level* when *init* starts. The *process* field means nothing for this *action*. The *initdefault* entry in */etc/inittab* specifies the run level to be entered upon autoboot. If *initdefault* is the action, the level field should be 1, s, S or 2, indicating single or multi-user mode.
- *once*
  runs the named *process* once and doesn't restart it again if it finishes.
- *powerfail*
  tells *init* to run the named *process* whenever power is lost while UNIX is running.
- *respawn*
  starts a *process* if it does not already exit, waits for the *process* to finish, and then starts another.
- *off*
  kills or ignores the named *process* when the named *level* is active.

- *process*
  is any executable program, including shell procedures.
- #
  adds a comment to the end of a line. Everything following a # on a line is ignored by *init*.

When changing levels, *init* kills all processes not specified for the new level.

The following */etc/inittab* is similar to the default file shipped with Stardent 1500/3000.

```
is:2:initdefault:
fs::bootwait:/etc/bcheckrc </dev/console >/dev/console 2>&1
s0:0:wait:/etc/rc0 >/dev/console 2>&1 </dev/console
s1:1:wait:/etc/shutdown -y -iS -g0 >/dev/console 2>&1 </dev/console
s2:23:wait:/etc/rc2 >/dev/console 2>&1 </dev/console
s3:3:wait:/etc/rc3 >/dev/console 2>&1 </dev/console
of:0:wait:/etc/uadmin 2 0 >/dev/console 2>&1 </dev/console
```

```
co:23:respawn:/etc/getty console console
t0:23:off:/etc/getty tty0 9600
t1:23:off:/etc/getty tty1 9600
t2:23:off:/etc/getty tty2 9600
t3:23:off:/etc/getty tty3 9600
```

The indicated actions are described below.

Actions for the init Process

init – Step 1

The first thing *init* does after its creation is to search the file */etc/inittab* for entries that specify the *sysinit* or *bootwait actions*. If it finds any, *init* executes the named *process*. In the default file above, the second line of the file runs the *bcheckrc*(1M) program to check the status of the *root* file system, repair any damage to it, and inititate step 2 of the *init* process (below).

```
fs::bootwait:/etc/bcheckrc </dev/console >/dev/console 2>&1
```

Note that the console (*/dev/console*) is specified as the source of standard input and the destination of standard output and standard error. This establishes communication between the system and the console as you enter a run level. Also note that the */etc/bcheckrc* program checks the status of file systems upon entry to all run levels because no *level* was specified. (See the *Commands Reference Manual* for more on the *bcheckrc* command.)

init – Step 2

Next, *init* searches the */etc/inittab* file for an entry of *action* type *initdefault*. This entry specifies the intial run level of *init*, specifically, the run level to be entered upon autoboot. Here is the line in the sample:

```
is:2:initdefault:
```

The *2* in the second field specifies the default run level to be the multi-user state. If you want the system to autoboot to the single-user run level, change the level to *s*.

init – Step 3

Next, *init* sets up communication with the console (*/dev/syscon*) and executes the */bin/su* program to produce a single-user shell owned by root. It then updates the files */etc/utmp* and */etc/wtmp*. (See the *Programmer's Reference Manual* for information about */etc/utmp* and */etc/wtmp*.) *init* then displays the run level on the console screen,

```
INIT: New run level: 2
```

enters its main processing loop, and scans the */etc/inittab* file again. It uses the specified run level as a tag to select the next entries to run.

init passes through the */etc/inittab* file, selecting and executing the level 2 *actions* and *processes* in the order that it finds them. Thus, the following lines are executed in this order:

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

Note that these lines are processed upon entry to any state specified in the second field.

/etc/rcN Files

The */etc/rcN* files contain shell commands that are executed as the system enters state *N*. For instance, */etc/rc2* is executed upon entry to run level 2. Its execution is ordered by a command in the */etc/inittab* file:

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

The name *rc* stands for *run commands*. *rc* files also exist for run levels 0 and 3.

The following */etc/rc2* file is similar to the default file shipped with Stardent 1500/3000.

```
#! /bin/sh
#ident "$Header: rc2.sh,v 1.7 87/01/29 14:54:55 mcneal Exp $"

#       "Run Commands" executed when the system is changing to init state 2,
#       traditionally called "multi-user".


#       Pickup start-up packages for mounts, daemons, services, etc.
set `who -r`
if [ $9 = "S" ]
then
        echo 'The system is coming up.  Please wait.'
        BOOT=yes
```

```
		if [ -f /etc/rc.d/PRESERVE ]	# historical segment for vi and ex
		then
			mv /etc/rc.d/PRESERVE /etc/init.d
			ln /etc/init.d/PRESERVE /etc/rc2.d/S02PRESERVE
		fi

	elif [ $7 = "2" ]
	then
		echo 'Changing to state 2.'
		if [ -d /etc/rc2.d ]
		then
			for f in /etc/rc2.d/K*
			{
				if [ -s ${f} ]
				then
					/bin/sh ${f} stop
				fi
			}
		fi
	fi

	if [ -d /etc/rc2.d ]
	then
		for f in /etc/rc2.d/S*
		{
			if [ -s ${f} ]
			then
				/bin/sh ${f} start
			fi
		}
	fi
	if [ "${BOOT}" = "yes" ]
	then
		stty sane tab3 2>/dev/null
	fi

	if [ "${BOOT}" = "yes" -a -d /etc/rc.d ]
	then
		for f in `ls /etc/rc.d`
		{
			if [ ! -s /etc/init.d/${f} ]
			then
				/bin/sh /etc/rc.d/${f}
			fi
		}
	fi

	if [ "${BOOT}" = "yes" -a $7 = "2" ]
	then
		echo 'The system is ready.'
	elif [ $7 = "2" ]
	then
		echo 'Change to state 2 has been completed.'
	fi
```

The getty Process

The last thing *init* executes is the *getty* process specified in the final field of the */etc/inittab* file. For instance,

```
ct:23:respawn:/etc/getty console console
t0:23:respawn:/etc/getty tty0 9600
```

getty is a process that opens the terminal, sets the terminal line to the correct speed, sets the terminal type, and invokes login.

The line option to *getty (/etc/getty console* in the first line of the given above) is required. It specifies the terminal line in the */dev* directory to which the terminal is to attach itself.

The argument following the line argument is the terminal's speed. It doesn't really set the speed, but is used by *getty* as an index to the */etc/gettydefs* file (see below), which contains more information about the terminal and terminal line.

Note that the term *tty*, even though it is derived from the abbreviation for *t*ele*ty*pewriter, covers all types of access to the system, whether from terminals, printers, modems, or the system console.

See the *Configuration* section of this guide for instructions on how to configure terminals, modems, and printers to work with the system. See *getty*(1M) for details on the options for the *getty* command.

The /etc/gettydefs File

Just as the */etc/inittab* file prescribes the actions for the *init* process, so to does the */etc/gettydefs* file specify the actions for the *getty* process. The general format of the file (explained below) is given here:

```
label# initial-flags# final-flags# login-prompt#next-label
```

The default file shipped with Stardent 1500/3000 looks like this:

```
console#B9600 CLOCAL#B9600 SANE TAB3 CLOCAL#titan login: #console

9600#B9600 CLOCAL#B9600 SANE TAB3 CLOCAL#titan login: #2400

2400#B2400 CLOCAL#B2400 SANE TAB3 CLOCAL#titan login: #1200

1200#B1200 CLOCAL#B1200 SANE TAB3 CLOCAL#titan login: #300

300#B300 CLOCAL#B300 SANE TAB3 CLOCAL#titan login: #19200
```

```
19200#B19200 CLOCAL#B19200 SANE TAB3 CLOCAL#titan login: #9600

modem1200#B1200#B1200 SANE TAB3 HUPCL#titan login: #modem300

modem300#B300#B300 SANE TAB3 HUPCL#titan login: #modem1200
```

The first field, *label*, is the index used by the *getty* process specified in the */etc/inittab* file. The second field, *initial-flags*, specifies the characteristics the system uses to initialize the port (port is a synonym for "terminal line." The third field, *final-flags*, completes setting up the port with the parameters that will be used throughout the coming login session.

The fourth field, *login-prompt*, contains the login prompt string as it appears to a user attempting to login. While this string may theoretically contain anything, remember two things:

(1) Most other systems that try to connect with yours through *uucp* or some other network expect to see the string *login:* followed by a space as part of your system's login prompt. If you don't include such a string at the end of the login prompt you choose, the remote system won't be able to login.

(2) Leave a space after the colon. The *login*(1) program gets confused if you don't.

The last field, *next-label*, establishes a sequence for *getty* to hunt through the */etc/gettydefs* file if it can't establish the connection at the specified speed. The *next-label* field is used just as if it were the argument to */etc/getty* in the */etc/inittab* file in the event that the port can't be initialized at the speed specified in *label*. This sequence continues as long as the *next-label* field refers to a *label* field.

Finally, note that the entries in the */etc/gettydefs* file are separated by blank lines. *These are required*, because *getty* tries to parse adjacent lines as one line.

Flag Arguments

There are dozens of flag arguments to */etc/getty*. The arguments are packaged as bit patterns and sent to the *termio(7)* interface for decoding. See the *termio(7)* entry in the *Commands Reference Manual*. Only a few of the flags need to be used to set up the Stardent 1500/3000's communication lines:

- *B xxx*
  Sets the baud rate to the value *xxx*. For example, *B9600* sets the line speed to 9600 baud.

- *CLOCAL*
  Specifies that a terminal is directly attached to the system (hardwired). The system does not wait for carrier to be present to spawn a *getty* process.

- *SANE*
  Invokes standard settings for normal line characteristics.

- *TAB3*
  Expands tabs to spaces when sent to the terminal.

- *HUPCL*
  Used for modem lines. If the connection is prematurely disconnected, all processes associated with the terminal are terminated by the *HUPCL* signal.

The line speed (*Bxxxx*) and *CLOCAL* (if it applies) must be included in both the *initial-flags* and *final-flags* fields.

Checking /etc/gettydefs

The */etc/getty* command supports a *–c* option which reads the */etc/gettydefs* file, or a different file following the same format, and checks it for accuracy. This is quite useful if you need to create a new version of */etc/gettydefs*.

Copy the */etc/gettydefs* file, edit it, and run it with the the following command:

```
#  /etc/getty -c /etc/gettydefs

**** Next Entry ****
console#B9600 CLOCAL#B9600 SANE TAB3 CLOCAL#speedy login: #console

id: console
initial flags:
iflag- 0 oflag- 0 cflag- 6275 lflag- 0
final flags:
iflag- 2446 oflag- 14005 cflag- 4275 lflag- 53
message: titan login:\□
next id: console

**** Next Entry ****
9600#B9600 CLOCAL#B9600 SANE TAB3 CLOCAL#speedy login: #2400

id: 9600
```

```
initial flags:
iflag- 0 oflag- 0 cflag- 6275 lflag- 0
final flags:
iflag- 2446 oflag- 14005 cflag- 4275 lflag- 53
message: titan login:\□
next id: 2400

**** Next Entry ****
2400#B2400 CLOCAL#B2400 SANE TAB3 CLOCAL#speedy login: #1200

id: 2400
initial flags:
iflag- 0 oflag- 0 cflag- 6273 lflag- 0
final flags:
iflag- 2446 oflag- 14005 cflag- 4273 lflag- 53
message: titan login:\□
next id: 1200
```

The \□ following the login prompt indicates the required space.

You can check the flag values listed by referring to *termio*(7) in the *Commands Reference Manual.*

If you make an error in an entry, */etc/getty* points you in the direction of the error:

```
#  /etc/getty -c gettydefs.new

**** Next Entry ****
2400#B2300 CLOCAL#B2400 SAME CLOCAL#speedy login: #1200 #300

Undefined: B2300
Undefined: SAME
Parsing failure in the ''next id'' field
2400#B2300 CLOCAL#B2400 SAME CLOCAL#speedy login: #1200 #<- error detected here
```

Remember to insert a blank line between entries.

The Last Step

The last thing */etc/getty* does is execute the */bin/login* program, using as arguments the information supplied in the */etc/gettydefs* file. Now you see the familiar system prompt on the terminal screen:

```
titan login:
```

The login Process

The *login* process waits for user input to the *login:* prompt and, when it receives some, evaluates it and takes appropriate action. Provided that *login* receives valid input, the action it takes is specified in the */etc/passwd* file.

The login Prompt

The *login* prompt expects to receive a valid account name. It searches the */etc/passwd* file for a line beginning with the name, then checks to see if there is an associated password. If there is, *login* prompts for the password and turns off screen echo.

```
titan login:  rocky
Password:
```

Any errors cause *login* to print an error message and reprompt for valid information.

If the login attempt is not completed within one minute after the user first tries to login, the *login* process terminates and another one is spawned.

After the */etc/passwd* file has been consulted, the system executes login shell files: */etc/profile* and *.profile* (in the user's home directory) for Bourne Shell users, and *.cshrc* and *.login* (both in the user's home directory) for C-shell users. *User Services* in this guide has more information on choice of login shell and on the */etc/passwd* file.

FILE SYSTEM ADMINISTRATION

CHAPTER TWENTY–FOUR

This chapter presents the Stardent 1500/3000 operating system view of file systems. It also shows you how to perform the tasks necessary to administer and maintain file systems: how to create, mount, and unmount file systems; how to back up files and directories; and how to check and repair file systems.

File System Structure

A file system is simply a structure to store and locate files on a disk. Figure 24-1 shows the main components of a typical UNIX file system.

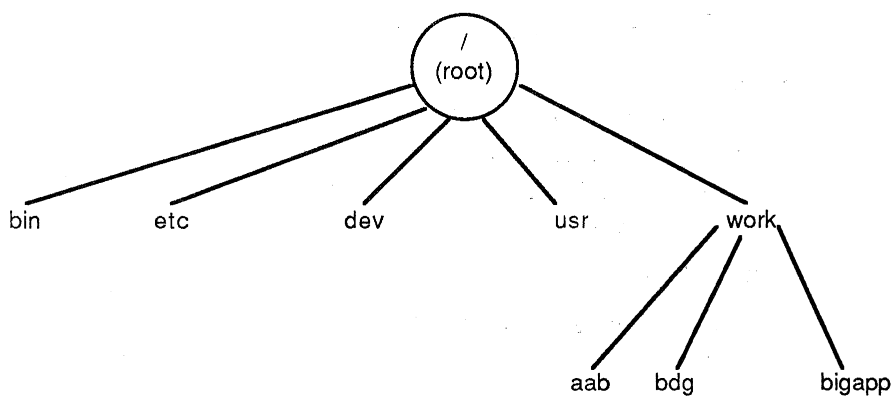


Figure 24-1. Typical File System Structure

This should be quite familiar to you. Yet this is not the way the file system looks to the operating system. The UNIX operating system views a file system as an arrangement of addressable disk space divided into logical blocks of 4096 bytes each. Each file system is associated with its own disk partition.

Stardent 1500/3000 supports two types of the file systems: the System V File System and the Stardent Fast File System (FFS).

System V is the standard UNIX file system type; FFS is an alternative type that offers significantly faster performance. Although each uses logical blocks of 4096 bytes, significant differences exist in the way they store data on disk.

The System V File System

For a System V file system, the disk partition containing the file system has the following kinds of logical blocks.

- the superblock
- a variable number of blocks comprising the i-node-list
- a variable number of storage blocks containing data, the free block list, and indirect addresses.

The Superblock

Much information about a file system is maintained in the superblock:

- file system size and status
  - label, file system name
  - size in physical and logical blocks
  - read-only flag
  - superblock modified flag
  - date and time of last update
- i-nodes
  - total number allocated
  - number of free i-nodes
  - free list (up to 100)
  - index into free list
- storage blocks
  - total number of free blocks
  - free block list (up to 50)
  - index into free block list

Other information about the file system is stored in the superblock as well, but the above list gives you the general picture. The superblock is always stored in block 1 of the file system. (Block 0, the first block on the disk, is used.)

I-nodes

I-nodes (short for information nodes) contain all the information about a file except its name, which is kept in a directory, and the actual data in the file. An i-node is 64 bytes long. You specify the number of i-nodes for the file system at the time you create the file system.

An i-node contains the following information:

- type and mode of a file
- number of links to a file
- owner's user id number
- group id
- size of the file in bytes
- 13 block addresses
- date and time last accessed (read)
- date and time last modified (written)
- date and time created

The 13 block addresses are stored in an array that points to the contents of the file. The first 10 addresses point directly to the first 10 storage blocks of the contents of the file. Each direct block can contain 4096 bytes of file data. If the file is equal to or less than 40960 bytes long, all data in the file are pointed to by the 10 direct blocks.

For files larger than 40960 bytes, the 11th address points to an array of 1024 additional storage blocks, called indirect blocks. The indirect blocks provide an additional 1 Mbyte of storage.

If that is still not enough space, the 12th address points to another array of 1024 more storage blocks, called double indirect blocks. Each of these blocks contains the address of yet another 1024 storage blocks (each containing 1024 bytes). With the double indirect blocks you can address a file of over 1 GByte.

Finally, if your file needs still more space, the 13th disk block address contains the address of 1024 additional double indirect blocks. These are called triple indirect blocks. They let you address up to the largest possible file size: 4GByte - 1. (maximum file size is actually constrained by the operating system, not by physical limitations.)

Storage Blocks

The rest of a System V file system contains storage blocks that either hold the contents of files or hold entries for directories. Directory blocks contain 2 bytes for the i-node number (often called the i-number) and 14 bytes for the name of the file or subdirectory.

Free Blocks

Free blocks are blocks that are not currently used as i-nodes, storage blocks, or indirect address blocks. They are available as storage blocks when new files or directories are created. Free blocks are chained together in a linked list in which each block in the list holds the address of the next block in the list.

The Stardent Fast File System

The Stardent Fast File System (FFS) achieves fast performance by storing the contents of files in contiguous storage blocks whenever possible. The goal is to minimize the number of disk accesses required when a file is read from the disk.

For a Fast File System the disk partition containing the file system is divided into two pieces: an overhead section and a data section. The boundary between the two is determined when the file system is created.

FFS Overhead Section

The overhead section of a FFS contains the following elements:

- the superblock
- the free block bit map
- i-nodes
- the i-node indirect block
- directories
- pipes
- file indirect blocks

The superblock in a FFS is similar to that in a System V file system, except that the FFS superblock contains no free block or free i-node lists. Free blocks are identified by the free block bit map, and free i-nodes are identified by the i-node indirect block.

Pipes are placed in the overhead area because they are temporary; directories are placed there because they are relatively small and static. By including pipes and directories in the overhead section, more contiguous storage space is made available in the data section of the disk partition.

The FFS doesn't preallocate the number of i-nodes available for the file system. If more i-nodes are needed than can fit in the overhead area, a free block can be used for i-nodes. If this happens, the pointer to the new i-nodes is included in the i-node indirect block.

FFS I-nodes are similar to System V i-nodes. They differ mainly in the way they make reference to the contents of files. Instead of listing storage blocks, FFS i-nodes refer to a file's starting block and the size or *extent* of the file (in blocks). The approach is well suited to addressing large contiguous chunks of data.

An FFS i-node has two additional block pointers. The first points to a block filled with data block addresses (indirect block); the second points to a block filled with indirect block addresses (double indirect block). Together, these are sufficient to address the largest possible file (4Gbytes -1).

The Root File System

The root file system contains the UNIX kernel and all administrative and system files. For administrative convenience and safety we recommend that you do not add any user files to the root file system. The only root file system files you need to touch are administrative files such as those in */etc.*

Keeping user files out of the root file system makes routine backups easier and may avoid internal problems.

Creating A File System

The process of creating a file system has several steps.

(1) You identify the disk partition that is to hold the file system, associating a *special device file* with the disk partition.

(2) You "make" the file system specifying the disk partition to be used, the number of disk blocks to be allocated for storage, and the file system type.

(3) You create a directory to serve as the file system mount point.

Once these steps are complete you are ready to mount and use the file system.

Assigning A Disk Partition

System Overview in this guide describes the default disk partitions and their device names. To assign a disk partition, you need simply to select the *special device file* that identifies the disk partition to the operating system.

Special Device Files

When the Stardent 1500/3000 operating system is booted the root file system is mounted. One of the directories contained in the root file system is the */dev* directory. The */dev* directory is populated with special device files. Special device files contain no data. Rather, they represent pointers to device drivers, the software interfaces between the Stardent 1500/3000 operating system and physical devices. (The device drivers themselves are compiled as part of the operating system kernel.)

Subdirectories within the */dev* directory contain special device files of particular types. For instance, disk special device files are contained in */dev/dsk* (disk block special device files) and */dev/rdsk* (disk character or raw special device files).

Here are some entries from the */dev/dsk* and */dev/rdsk* directories:

```
# ls -l /dev/dsk/c0d5*
brw-rw-r--   1 root     root        0, 16 Apr 25 15:55 c0d5s0
brw-rw-r--   1 root     root        0, 17 Apr 25 15:55 c0d5s1
brw-rw-r--   1 root     root        0, 18 Apr 25 15:55 c0d5s2
brw-rw-r--   1 root     root        0, 19 Apr 25 15:55 c0d5s3
brw-rw-r--   1 root     root        0, 20 Apr 25 15:55 c0d5s4
brw-rw-r--   1 root     root        0, 21 Apr 25 15:55 c0d5s5
brw-rw-r--   1 root     root        0, 22 Apr 25 15:55 c0d5s6
brw-rw-r--   1 root     root        0, 23 Apr 25 15:55 c0d5s7
brw-rw-r--   1 root     root        0, 24 Apr 25 15:57 c0d5vh
# ls -l /dev/rdsk/c2d3*
crw-------   1 root     root       24,  0 Apr 25 16:02 c2d3s0
crw-------   1 root     root       24,  1 Apr 25 16:02 c2d3s1
crw-------   1 root     root       24,  2 Apr 25 16:02 c2d3s2
crw-------   1 root     root       24,  3 Apr 25 16:02 c2d3s3
crw-------   1 root     root       24,  4 Apr 25 16:02 c2d3s4
crw-------   1 root     root       24,  5 Apr 25 16:02 c2d3s5
crw-------   1 root     root       24,  6 Apr 25 16:02 c2d3s6
crw-------   1 root     root       24,  7 Apr 25 16:02 c2d3s7
crw-------   1 root     root       24,  8 Apr 25 16:05 c2d3vh
```

The special device files listed here differ from regular files in that the file names are unusual, the initial character in the permissions list is one of *b* (block) or *c* (character), and no amount is shown in the column where the character count is usually shown. Instead two comma-separated numbers are shown, the major and minor device numbers.

The first character in the permissions listing, *b* or *c*, identifies the device as a block or a character device. The terms *block* and *character* device refer to the way in which data are transferred between the device and main memory. (Character devices are also called *raw* devices.) Disks are generally treated as block devices; tapes, terminals, printers and plotters are generally treated as character devices. Block and character assignments for disks or tapes, however, are a function of how you want to transfer data, not of the physical device itself or its contents. The directories */dev/dsk* and */dev/rdsk*, for instance, contain block and character special files that refer, respectively, to the same physical disk drives and partitions.

The major device number identifies a particular device driver; the minor device number is a parameter to the device driver. It often represents the specific device and partition.

Disk and Tape Special Device File Names

UNIX disk and tape special device file names follow standard formats. For disk devices the format is

```
c <controller number> d <device number> s <section or partition number>
```

or

```
k <controller number> d <device number> s <section or partition number>
```

The letter c is used for disks that are controlled via Stardent 1500/3000 I/O board 1 (slot 1 of the Stardent 1500/3000 card cage); the letter k is used for disks that are controlled via Stardent 1500/3000 I/O board 2 (slot 8 of the Stardent 1500/3000 card cage).

Here are a couple of examples:

```
c1d5s0
```

refers to the root partition of the primary boot disk.

```
k2d1s0
```

refers to an SMD disk device connected to a VME controller that communicates via the VME expansion board in slot 9 of the

Stardent 1500/3000 card cage. (The VME expansion board in slot 9 is controlled by the I/O board in slot 8.)

The full pathnames of the example files are

```
/dev/dsk/c1d5s0
```

and

```
/dev/dsk/k2d1s0
```

System Overview in this guide gives guidelines for assigning device names.

NOTE
Although tape devices can be used in block mode, they are generally treated as character devices.

Tape special device file names follow the same format as disks, with the exception that no section or partition number is used. Instead, the special device file name codes whether the device is high or low density and whether or not the tape is automatically rewound after use. For instance, the internal cartridge tape drive is connected to controller 0 on the I/O board in slot 1 and the device number is 6. These are legal special device names for the internal cartridge tape drive:

```
/dev/rmt/c0d6h (or /dev/rmt16)
/dev/rmt/c0d6hn (or /dev/rmt20)
/dev/rmt/c0d6l (or /dev/rmt0)
/dev/rmt/c0d6ln (or /dev/rmt4)
```

h means high density, l means low density and n means no-rewind-on-close. The parenthesized names are hard-linked equivalents and may be used interchangeably with the file names in the directory */dev/rmt*.

Creating Special Device Files

The special device files in the */dev* directories and subdirectories shipped with Stardent 1500/3000 should be sufficient for most installations. It is only if you have written your own specialized device drivers that you need to create new special device files. In that event, refer to the *mknod*(1M) pages in the *Commands Reference Manual*. *mknod* is the command used to create special device files.

Making The File System

When you make a file system you specify the disk partition to be used and the type of file system (AFFS or System V) to be created. To create a file system use the *dvhtool* command with the *–m* option. [1] On the command line you specify the volume header of the disk on which the file system is to reside, the number of the disk partition and the type of file system. For instance, the command

```
# dvhtool /dev/dsk/c1d5vh -m 2 AFFS
```

creates an Stardent Fast File System on partition 2 of the primary boot disk. As another example, the command

```
# dvhtool /dev/dsk/c2d0vh -m 1 SysV
```

creates a System V file system on partition 1 of the SMD disk drive *c2d0*.

Creating The Mount Point

A *mount point* for a file system is a directory name that serves as the root of the file system.

To create a mount point for the file system *work*, create a directory by that name under the root directory:

```
# cd /
# mkdir work
```

We recommend that mount points for file systems always be in the root directory.

You are now ready to make the file system available for use.

Mounting and Unmounting a File System

You have now identified the actual physical disk partition that is going to contain the data for the file system, assigned a special device file to identify that file system to the operating system, and created a mount point by making a new directory named */work*. You must now mount the file system to be able to use it. Use the *mount*(1M) command, which associates the physical disk partition with the directory you have created:

[1] You can also create file systems using the *mkfs* command; *dvhtool* invokes *mkfs; dvhtool* is preferrable in that it does not require you to supply explicit blocks/track and tracks/cylinder information about the disk.

```
# mount /dev/dsk/c0d5s3 /work
Warning <> mounted as /work
```

The file system is now available for use. (Ignore the warning.) Additional options to the *mount*(1M) command are described in the *Commands Reference Manual.*

File systems are usually mounted and made available to users at the time the system is booted. If you have created a new file system that you want to have mounted at system bootup, you need to add the file system to the */etc/fstab* file. See *fstab.4* in the *Programmer's Reference Manual* for a description of the format of */etc/fstab.*

You need to be able to unmount file systems in order to perform maintenance tasks such as checking and repairing the file system with *fsck* (1M). In addition, unmounting is required when you shut the system down and recommended when you are backing up the file system.

CAUTION
The *umount* command may fail if a process is either "cd'd" into a directory under the mount point or a process is accessing a file under the mount point.

To unmount a file system, use the umount command with the appropriate special device file or the mount point:

```
# umount /dev/dsk/c1d5s3
```

or

```
# umount /work
```

Batch Processing

The Unix system allows background job processing but does not have a standard batch facility. You can, however, set up a system to process large jobs one at a time or to process large jobs during low use hours.

The following shell script uses *lp spool* to set up a queue for one-at-a-time processing of large jobs.

```
#! /bin/sh

# lp batch spooler script
#
# This script will read commands and execute one line at a time.
# This is so users may spool up their jobs and only one user
# may run their command files at a time.  NOTE: When making
# a command file to submit to batch, you should not place the
# job in the background (with &) as this will defeat the purpose
# of this spooler.  Output will be mailed to the invoking user.
```

```
#

# use copies as number of times to go through commands file
batchid=$1
user=$2
copies=$4
shift; shift; shift; shift; shift
files="$*"
i=1
while [ $i -le $copies ]
do
        for file in $files
        do
                cat "$file" > /tmp/batch.$$
                chmod 755 /tmp/batch.$$
                /tmp/batch.$$ 2>&1 | /usr/ucb/mail -s "Job done:$batchid" $user
                rm /tmp/batch.$$
        done
        i=`expr $i + 1`
done
exit 0
```

To use the program, users with large jobs should place run commands for the jobs in a file called *batch_progs*. The jobs can then be run one at a time by issuing the command

```
# lp -dbatch batch_progs
```

To run jobs during off hours you can set up a *cron* file to execute at a certain time. (type **man cron** for more on the cron daemon).

File System Maintenance

Routine file system maintenance includes the following tasks:

- Monitoring disk usage.
- Backing up and restoring files and directories.
- Checking for file system consistency.

The first two tasks are described in this section. The last task is described later in this chapter.

Monitoring Disk Usage

File systems always have a tendency to grow. By monitoring disk usage you can control file system growth before performance or storage space problems occur.

Monitoring disk usage involves three tasks:

- Monitoring percent of disk space used.
- Monitoring files and directories that grow.
- Identifying and removing inactive files.

Monitoring percent of disk space used

Use the *df*(1M) command to monitor percent of disk space used. The command

```
# df -b
```

displays disk usage for all mounted file systems. Examining the total tells you how much space is still available.

Here is a sample listing.

```
Stardent 1500/3000: df -b
Filesystem              kbytes    used    avail capacity Mounted-on
/dev/dsk/c0d5s0          82080   76180     5900     92%   /
/dev/null                    0       0        0     -1%   /proc
/dev/dsk/c1d5s3          58320      60    58260      0%   /tmp
/dev/dsk/c2d0s4         189360  186344     3016     98%   /caos
/dev/dsk/c2d0s5         189360  178640    10720     94%   /cala
/dev/dsk/c2d0s6         189360  185072     4288     97%   /cahw
/dev/dsk/c1d5s6          50760   50480      280     99%   /SCCS
/dev/dsk/c1d5s5         189000  164880    24120     87%   /calb
Stardent 1500/3000:
```

Monitoring Growing Files and Directories

Certain files and directories tend to grow through normal use, such as those in the following list.

File	Use
/etc/wtmp	history of system logins
/usr/adm/sulog	history of su commands
/usr/lib/cron/log	history of actions of /etc/cron
/usr/adm/messages	record of all system console activity

The frequency with which you need to check growing files is a function of overall system usage. One way to keep their size manageable is to use the *tail*(1) command to remove all but the last entries in the file. For instance

```
# tail -50 /usr/lib/cron/log > /tmp/log
# mv /tmp/log > /usr/lib/cron/log
```

removes all but the last 50 entries in */usr/lib/cron/log*.

Identifying and Removing Inactive Files

You can use the *find*(1) command to locate files that haven't been touched in a while, identifying possible candidates for removal or backup. Here are a couple of examples.

```
# find . ! -newer file -print
```

This command line lists all files not newer than *file*.

```
# find . ! -mtime -7 -print
```

This command line lists all files not modified within the last 7 days.

Backing Up Files and Directories

One of the most important functions of the system administrator is backing up users' files. Several utilities are available to help you do this. You can even automate part of the job.

Make sure that the root file system contains no user files; then you need not back up the root file system. Within the root file system you need only back up the administrative files you have edited, such as those in */etc*.

Backup Devices

You can make backup copies on the built-in 1/4-inch cartridge (QIC) tape, on an optional 1/2-inch reel-to-reel tape drive, on an optional 2 GB cartridge tape drive or on a disk partition. Let's look at some of the attributes of each.

QIC Tape. The QIC tape is a convenient method for maintaining archive files. Each tape stores about 120 Mbytes of data.

Half-inch Tape. A 1/2-inch streaming tape is faster than QIC tape. The amount of data a streaming tape can hold is highly variable; a rough guideline is that a 2400 foot tape at 6250 bpi can store about 160 MB of data.

2 GB Cartridge Tape. The 2 GB cartridge tape drive allows you to back-up large quantities of data at a time. The key disadvantage is the length of time it takes to recover data stored near the end of the tape.

Disk Partition. You may make backups on a partition of your SCSI or SMD disk, but this is not recommended because it is not as safe as making backups on removable media and storing the backups in a separate location.

Backup Strategy

Probably the most difficult thing about making backups is deciding how often and how much to backup. There are no firm rules, but in general, a full backup every month, a partial backup every week, and an incremental backup every day make it easier to restore files that have been changed or added.

The cpio Command

The *cpio*(1) command permits you to make archive copies of files. It is useful for making incremental backups because it is relatively quick and it permits you to restore a single file easily.

Assume you are going to copy all the files for a user with the login name of *squirrel*. Your user keeps files in the */work/squirrel* directory.

Creating a Backup Copy

Put a cartridge tape in the built-in drive and make sure it is write enabled. (See *System Overview* in this guide for tape specifications and instructions on loading the tape.) Enter the following commands:

```
# cd /work/squirrel
# find . -print | cpio -oB > /dev/rmt/c0d61n
```

The *cpio*(1) command copies all the files and directories found below */work/squirrel*. Because you need to specify the input one file per line, it is common to use *cpio* in conjunction with the *find* (1) command.

If you need more than one QIC cassette to do the backup, the following message is displayed on the screen:

```
If you want to go on, type device/file name when ready. ■
```

Replace the filled tape with a new one and enter the special device file name to continue:

```
If you want to go on, type device/file name when ready.  /dev/rmt16
```

If you just type a carriage return in response to the message, *cpio* exits without continuing.

Recovering a Backup Copy

In the event you need to restore the files from the archived QIC copy, use the following commands:

```
# cd /work/squirrel
# cpio -iBd < /dev/rmt16
```

The *cpio* command extracts all files and directories that were created with the preceding *cpio* command. You may specify only certain files this way:

```
# cpio -iBd '*.c' < /dev/rmt16
```

This copies all files ending with the *.c* suffix to the current directory.

By default, *cpio* only overwrites those files that are older than the archived copy. Also, if the command is executed by the superuser, the original permissions and owner of the files remain the same.

Making Incremental Backups

An easy way to make incremental backups is to include a *find* command in a *crontab*(1) file (see *System Overview)* in this guide. The *find* command supports an option that automatically invokes *cpio*. Here is sample *crontab* entry:

```
30 2 * * 4 find /work/squirrel -depth -mtime -8 -cpio /dev/rmt16
```

The entry means "At 2:30 a.m. every Thursday, find all files in the directory */work/squirrel* that have been modified within the last seven days and archive them with *cpio* on the built-in QIC tape drive." Now all you need to do is put a blank QIC tape in the drive before you leave on Thursday. Friday morning when you arrive, the specified files will have been copied to the tape.

Checking the Backup Tape

Regardless of the command you use to make your backup, you should check that the files really made it onto the backup tape. You can do so by issuing the following *cpio* command:

```
# cpio -itv < /dev/rmt16
```

This gives a listing of the files on the tape.

File System Checking and Repair

Every time you boot Stardent 1500/3000, whether automatically or manually, a file system check is done. File system checking is necessary because of potential differences between the image of the file system in main memory and the image of the file system on disk. For instance, whenever the system is running, copies of mounted file systems' superblocks are kept both in main memory and on disk. The copy in main memory is always current, but the copy on disk may not be. If a system crash or other problem occurs between the time a change to the superblock is made and the disk is written, a file system consistency problem results.

The *sync*(1M) command (no arguments) lets you request a disk update at any time. Issuing the *sync* command when a crash seems imminent but has not yet occurred, may prevent file system problems.

File system checking and repair is necessary when a file system has been corrupted (rendered inconsistent for any of a variety of reasons). It is also recommended when you are mounting a file system that is not usually mounted during system booting.

The *fsck* Program

The *fsck*(1) program provides an interactive way to check and repair file systems. It performs consistency checks based on information carried by the file system itself. It displays messages about inconsistencies it uncovers and gives you the option of requesting possible repairs. You can have repairs made by *fsck,* you can do some repairs manually, or you can return to an uncorrupted, back-up version of the file system. You should *never* ignore file system problems, however. They are certain to get worse over time.

What fsck Checks

fsck checks for the following inconsistencies in a file system:

- Blocks that are claimed by more than more i-node or free list.
- Blocks that are claimed by an i-node or the free list outside the range of the file system.
- Incorrect link counts.
- Incorrect number of blocks.
- Directory size not 16-byte aligned.
- Bad i-node format.
- Directory checks:
  - File pointing to unallocated i-node.
  - I-node number out of range.
- Super block checks:
  - More than 65536 i-nodes
  - More blocks for i-nodes than there are in the system.
- Bad free block list format.
- Total free block and/or free i-node count incorrect.

Invoking fsck

To check the root file system, make sure you are in single user mode. To check any file system other than root, make sure that the file system is unmounted.

You invoke the *fsck* program with a command line such as the following in which the device containing the file system is specified:

```
fsck /dev/dsk/c0d5s3
```

Options for the *fsck* command line are given in the *Commands Reference Manual.*

In the following example *fsck* uncovers no problems:

```
# fsck /dev/dsk/c0d5s3

/dev/dsk/c0d2s2 type S54K
  ** Phase 1 - Check Blocks and Sizes
  ** Phase 2 - Check Pathnames
  ** Phase 3 - Check Connectivity
  ** Phase 4 - Check Reference Counts
  ** Phase 5 - Check Free List
```

```
  289 files 6522 blocks 3220 free
#
```

As the example messages show, *fsck* operates in phases. (Some phases are run only if required or in response to a command line option.) As each phase is completed, a message is displayed. At the end of the program a summary message is displayed showing the number of files (i-nodes), blocks, and free blocks.

fsck Messages

In each phase *fsck* reports errors it detects and, if an error is correctable, it asks permission to make the correction (unless overwritten by a command line argument). This section describes messages that are produced by each phase.

The following abbreviations are used in the *fsck* error messages:

```
BLK     block number
DUP     duplicate block number
DIR     directory name
MTIME   time file was last modified
UNREF   unreferenced
```

The single-letter abbreviations given here are used in the description of *fsck* messages that follows. When the messages appear on your screen, you see the full word or phrase.

```
B       block number
F       file (or directory) name
I       i-node number
M       file mode
O       user-id of a file's owner
S       file size
T       time file was last modified
X       link count,
    or  number of BAD, DUP, or MISSING blocks
    or  number of files (depending on context)
Y       corrected link count number
    or  number of blocks in file system (depending on context)
Z       number of free blocks
```

Initialization Phase

In this phase *fsck* checks command line syntax, initializes tables, and opens files. The program terminates on any initialization phase errors.

General Errors

Three error messages may appear in any phase. They each indicate a serious problem.

CAN NOT SEEK: BLK B (CONTINUE?)
: The request to move to a specified block number *B* in the file system failed. Indicates a serious problem (probably a hardware failure) that may require additional help.

CAN NOT READ: BLK B (CONTINUE?)
: The request for reading a specified block number *B* in the file system failed. Indicates a serious problem (probably a hardware failure) that may require additional help.

CAN NOT WRITE: BLK B (CONTINUE?)
: The request for writing a specified block number *B* in the file system failed. The disk may be write-protected.

If you answer no to the CONTINUE? prompt the program terminates. If you answer yes, the program attempts to continue to run the file system check. Often, however, the problem persists, due to an error condition. A second run of *fsck* should be made to recheck the file system.

Phase 1: Check Blocks and Sizes

This phase checks the i-node list. It reports error conditions resulting from

- checking i-node types
- setting up the zero-link-count table
- examining i-node block numbers for bad or duplicate blocks
- checking i-node size
- checking i-node format

Phase 1 has three types of error messages:

(1) information messages

(2) messages with a CONTINUE? prompt

(3) messages with a CLEAR? prompt

The CONTINUE? prompt generally means that some limit of tolerance has been reached. A no response to the CONTINUE? prompt says to terminate the program. A yes response says to continue with the program. If you say yes, a second run of *fsck* should be made to recheck the file system after you have finished.

A no response to the CLEAR? prompt says to ignore the error condition. This response is is only appropriate if you intend to take other action to fix the problem. A yes response to the CLEAR? prompt says to deallocate i-node *I* by zeroing its contents. This may invoke the UNALLOCATED error condition in Phase 2 for each directory entry pointing to this i-node.

Phase 1 Error Messages.

UNKNOWN FILE TYPE I=I (CLEAR?)
: The mode word of the i-node *I* suggests that the i-node is not a pipe, special character i-node, regular i-node, or directory i-node.

LINK COUNT TABLE OVERFLOW (CONTINUE?)
: An internal table for *fsck* containing allocated i-nodes with a link count of zero has no more room.

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

EXCESSIVE BAD BLOCKS I=I (CONTINUE?)
: There is more than a tolerable number (usually 10) of blocks with a number lower than the number of the first data block in the file system or greater than the number of the last block in the file system associated with i-node *I*.

B DUP I=I
: I-node *I* contains block number *B*, which is already claimed by another i-node. This error condition may invoke the EXCESSIVE DUP BLKS error condition in Phase 1 if i-node *I* has too many block numbers claimed by other i-nodes. This error condition invokes Phase 1B

and the BAD/DUP error condition in Phase 2 and Phase 4.

EXCESSIVE DUP BLKS I=I (CONTINUE?)
: There is more than a tolerable number (usually 10) of blocks claimed by other i-nodes.

DUP TABLE OVERFLOW (CONTINUE?)
: An internal table in *fsck* containing duplicate block numbers has no more room.

POSSIBLE FILE SIZE ERROR I=I
: The i-node *I* size does not match the actual number of blocks used by the i-node. This is only a warning. If the *–q* option is used, this message is not printed.

DIRECTORY MISALIGNED I=I
: The size of a directory i-node is not a multiple of 16. This is only a warning. If the *–q* option is used, this message is not printed.

PARTIALLY ALLOCATED INODE I=I (CLEAR?)
: I-node *I* is neither allocated nor unallocated.

Phase 1B: Rescan for More DUPS

When a duplicate block is found in the file system, the file system is rescanned to find the i-node that previously claimed that block. When the duplicate block is found, the following information message is printed:

B DUP I=I
: I-node *I* contains block number *B*, which is already claimed by another i-node. This error condition invokes the BAD/DUP error condition in Phase 2. I-nodes with overlapping blocks may be determined by examining this error condition and the DUP error condition in Phase 1.

Phase 2: Check Path Names

This phase removes directory entries pointing to bad i-nodes found in Phase 1 and Phase 1B. It reports error conditions resulting from

- root i-node mode and status
- directory i-node pointers out of range
- directory entries pointing to bad i-nodes

Phase 2 has 4 types of error messages:

(1) information messages

(2) messages with a FIX? prompt

(3) messages with a CONTINUE? prompt

(4) messages with a REMOVE? prompt

If you respond no to the FIX? prompt, the program terminates. A yes response to the FIX? prompt says to change the root i-node type to "directory." If the root i-node data blocks are not directory blocks, many error conditions are produced.

If you respond no to the CONTINUE? prompt, the program terminates. A yes response to the CONTINUE? prompt says to ignore the DUPS/BAD error condition in root i-node and attempt to continue to run the file system check. If root i-node is not correct, many error conditions may result.

A no response to the REMOVE? prompt says to ignore the error condition. A NO response is only appropriate if you intend to take other action to fix the problem.

A yes response to the REMOVE? prompt says to remove duplicate or unallocated blocks.

Phase 2 Error Messages.

ROOT INODE UNALLOCATED. TERMINATING

The root i-node (always i-node number 2) has no allocate mode bits. The occurrence of this error condition indicates a serious problem. The program stops.

ROOT INODE NOT DIRECTORY (FIX?)

The root i-node (usually i-node number 2) is not directory i-node type.

DUPS/BAD IN ROOT INODE (CONTINUE?)

Phase 1 or Phase 1B found duplicate blocks or bad blocks in the root i-node (usually i-node number 2) for the file system.

I OUT OF RANGE I=I NAME=F (REMOVE?)

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

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

A directory entry *F* has an i-node *I* without allocate mode bits. The owner *O*, mode *M*, size *S*, modify time *T*, and filename *F* are printed. If the file system is not mounted and the *–n* option was not specified, the entry is removed automatically if the i-node it points to is character size 0.

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

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

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

Phase 1 or Phase 1B found duplicate blocks or bad blocks associated with file entry *F*, i-node *I*. The owner *O*, mode *M*, size *S*, modify time *T*, and filename *F* are printed.

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

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

Phase 3: Check Connectivity

This phase concerns itself with the directory connectivity seen in Phase 2. It reports error conditions resulting from

- unreferenced directories
- missing or full *lost+found* directories

Phase 3 has two types of error messages:

(1) information messages

(2) messages with a RECONNECT? prompt

A no response to the RECONNECT? prompt says to ignore the error condition. This invokes the UNREF error condition in Phase 4. A NO response is only appropriate if you intend to take other action to fix the problem. A yes response to the RECONNECT? prompt says to reconnect directory i-node *I* to the file system in the directory for lost files (usually *lost+found*). This may invoke a *lost+found* error condition if there are problems connecting directory i-node *I* to *lost+found*. This invokes CONNECTED information message if link was successful.

Phase 3 Error Messages.

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

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

SORRY. NO lost+found DIRECTORY

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

SORRY. NO SPACE IN lost+found DIRECTORY

There is no space to add another entry to the *lost+found* directory in the root directory of the file system; *fsck* ignores the request to link a directory in *lost+found*. This invokes the UNREF error condition in Phase 4. Clean out unnecessary entries in *lost+found* or make *lost+found* larger (see Procedure 5.2).

DIR I=I1 CONNECTED. PARENT WAS I=I2

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

Phase 4: Check Reference Counts

This phase checks the link count information seen in Phases 2 and 3. It reports error conditions resulting from

- unreferenced files
- missing or full *lost+found* directory
- incorrect link counts for files, directories, or special files
- unreferenced files and directories
- bad and duplicate blocks in files and directories
- incorrect total free-i-node counts

Phase 4 has 5 types of error messages:

(1) information messages

(2) messages with a RECONNECT? prompt

(3) messages with a CLEAR? prompt

(4) messages with an ADJUST? prompt

(5) messages with a FIX? prompt

A no response to the RECONNECT? prompt says to ignore this error condition. This invokes a CLEAR error condition later in Phase 4. A yes response to the RECONNECT? prompt says to reconnect i-node *I* to file system in the directory for lost files (usually *lost+found*). This can cause a *lost+found* error condition in this phase if there are problems connecting i-node *I* to *lost+found*.

A no response to the CLEAR? prompt says to ignore the error condition. A NO response is only appropriate if you intend to take other action to fix the problem. A yes response to the CLEAR? prompt says to deallocate the i-node by zeroing its contents.

A no response to the ADJUST? prompt says to ignore the error condition. A NO response is only appropriate if you intend to take other action to fix the problem. A yes response to the ADJUST? prompt says to replace the link count of file i-node *I* with *Y*.

A no response to the FIX? prompt says to ignore the error condition. A NO response is only appropriate if you intend to take other measures to fix the problem. A yes response to the FIX?

prompt says to replace the count in the super-block by actual the count.

Phase 4 Error Messages.

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

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

SORRY. NO lost+found DIRECTORY

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

SORRY. NO SPACE IN lost+found DIRECTORY

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

(CLEAR)

The i-node mentioned in the immediately previous UNREF error condition cannot be reconnected.

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

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

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

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

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

The link count for *F* i-node *I* is *X* but should be *Y*. The filename *F*, owner *O*, mode *M*, size *S*, and modify time *T* are printed.

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

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

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

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

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

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

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

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

FREE INODE COUNT WRONG IN SUPERBLK (FIX?)

The actual count of the free i-nodes does not match the count in the super-block of the file system. If the *–q* option is specified, the count will be fixed automatically in the super-block.

Phase 5: Check Free List

This phase checks the free-block list. It reports error conditions resulting from

- bad blocks in the free-block list

- bad free-block count
- duplicate blocks in the free-block list
- unused blocks from the file system not in the free-block list
- total free-block count incorrect

Phase 5 has 4 types of error messages:

(1) information messages

(2) messages that have a CONTINUE? prompt

(3) messages that have a FIX? prompt

(4) messages that have a SALVAGE? prompt

A no response to the CONTINUE? prompt says to terminate the program. A yes response to the CONTINUE? prompt says to ignore the rest of the free-block list and continue execution of *fsck*. This error condition will always invoke the BAD BLKS IN FREE LIST error condition later in Phase 5.

A no response to the FIX? prompt says to ignore the error condition. A NO response is only appropriate if you intend to take other action to fix the problem. A yes response to the FIX? prompt says to replace count in super-block by actual count.

A no response to the SALVAGE? prompt says to ignore the error condition. A NO response is only appropriate if you intend to take other action to fix the problem.

A yes response to the SALVAGE? prompt says to replace the actual free-block list with a new free-block list. The new free-block list will be ordered according to the gap and cylinder specs of the *–s* or *–S* option to reduce time spent waiting for the disk to rotate into position.

Phase 5 Error Messages.

EXCESSIVE BAD BLKS IN FREE LIST (CONTINUE?)

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

EXCESSIVE DUP BLKS IN FREE LIST (CONTINUE?)

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

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

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

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

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

FREE BLK COUNT WRONG IN SUPERBLOCK (FIX?)
The actual count of free blocks does not match the count in the super-block of the file system.

BAD FREE LIST (SALVAGE?)
This message is always preceded by one or more of the Phase 5 information messages. If the *–q* option is specified, the free-block list will be salvaged automatically.

Phase 6: Salvage Free List

This phase reconstructs the free-block list. It has one possible error condition that results from bad blocks-per-cylinder and gap values.

Phase 6 Error Messages.

DEFAULT FREE-BLOCK LIST SPACING ASSUMED This is an advisory message indicating the blocks-to-skip (gap) is greater than the blocks-per-cylinder, the blocks-to-skip is less than 1, the blocks-per-cylinder is less than 1, or the blocks-per-cylinder is greater than 500. The values of 7 blocks-to-skip and 400 blocks-per-cylinder are used.

Care must be taken to specify correct values with the *-sX* option on the command line. See the *fsck*(1M) and *mkfs*(1M) manual pages for further details.

Cleanup Phase

Once a file system has been checked, a few cleanup functions are performed. The cleanup phase displays advisory messages about the file system and status of the file system.

Cleanup Phase Messages.

X files Y blocks Z free

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

***** BOOT UNIX (NO SYNC!) *****

This is an advisory message indicating that a mounted file system or the root file system has been modified by *fsck*. If the UNIX system is not rebooted immediately without *sync*, the work done by *fsck* may be undone by the in-core copies of tables the UNIX system keeps. If the *–b* option of the *fsck* command was specified and the file system is *root*, a reboot is automatically done.

***** FILE SYSTEM WAS MODIFIED *****

This is an advisory message indicating that the current file system was modified by *fsck*.

USER SERVICES

CHAPTER TWENTY–FIVE

This chapter describes how you add and remove users from the system, the user environment, and how you, as the system administrator, communicate with users.

Adding A User

Every user on the system is assigned a unique *user ID* and a unique login name. A unique login directory is usually assigned as well. These items, along with other information, are stored in a file named */etc/passwd*.

You do the following tasks to add a user to the system:

Task 1: Make an entry for the user in */etc/passwd*.

Task 2: Create a home directory for the user.

Task 3: Copy X Window configuration files to the user's home directory.

The next sections describe user administrative files and show how to do the above-mentioned tasks.

The /etc/passwd File

Each entry in the */etc/passwd* file provides information about a user. The general format of an entry is as follows:

```
login-name:encrypted-password:user-id:group-id:account-info:home-directory:login shell
```

Here is a description of each field.

- *login name*
  This name should be unique for each user. It must be specified in lower-case letters only.

- *encrypted password*
  An encrypted version of the user's password is stored in this field. The password may be any combination of digits and upper- and lower-case characters, but it must begin with an alphabetic character, contain at least 7 characters, and contain at least one non-alphabetic character.

- *user id*
  This is the number that identifies the user to the system. It may be any integer smaller than 65535. To avoid conflicts with reserved user ids, do not assign any ids lower than 100. Note that the super-user has user id zero.

- *group id*
  This field may be blank. If it is used, it identifies the group to which this user belongs. The */etc/group* file contains the active group id numbers. The group id generally refers to a specific group of users such as members of a project. To avoid conflicts with reserved group ids, do not assign any group ids lower than 100.

- *account*info
  This field may be blank. If it is used, it contains additional information about the user, such as the user's real name and phone number. The *finger*(1) command uses this field to generate user information that can be listed upon request. See *finger*(1) and *chfn*(1) in the *Commands Reference Manual.*

- *home directory*
  This is the directory the user is placed in after a successful login. It must be specified as a full pathname from the root directory. It is the directory specified by the *HOME* environment variable.

- *login shell*
  This is the shell assigned to the user after a successful login. Typically it is */bin/sh* or */bin/csh,* but it may be any executable program.

All entries for one user must entered on a single line with the fields separated by colons. Here is an example */etc/passwd* similar to the default file shipped with Stardent 1500/3000:

```
root:9Eeqv2PzdaKvE:0:0:Super-usr:/:/bin/sh
daemon:noway:1:2:User for running deamons:/:
uucp:noway:5:5:uucp:/usr/lib/uucp:
lp:segql8zuz8E52:6:3:lp daemon:/usr/lib/lp:
sys:noway:7:3:SYS login:/usr/lib/lp:
```

```
bin:noway:10:1:User for installing software:/etc:
sally:Wee65kp01X:144:10:Alan A. Bush:/work/sally:/bin/csh
```

Note that the first six lines refer to special system functions, while the last line is for a typical system user.

Task 1: Making an Entry in /etc/passwd

To add a new user to the */etc/passwd* file you must assign a user ID and make entries in all the other required fields.

NOTE
To avoid conflicts with reserved user and group ids, don't assign any user or group ids lower than 100.

Choose a default password to give users when they first log in. You can do that by choosing an existing password and copying the encrypted version of the password into the user's password field. Users can then change the password with the *passwd*(1) command when they log in for the first time. See the *End User Interface Guide* or the *Commands Reference Manual* for instructions on using the *passwd* command.

Forgotten Passwords

If a user forgets her or his password, you can remedy the situation by logging in as super-user and issuing the *passwd* command:

```
# passwd user
New password: (type in a temporary password)
Re-enter new password: (re-enter the password)
#
```

As the super-user you are not prompted for the old password. With knowledge of the temporary password the user can now log in. The user can then reset the password as desired.

The /etc/group File

The */etc/group* file is similar to the */etc/passwd* file. It contains entries for groups of users who usually share common job responsibilities.

Like the password file, the group file consists of colon-separated lines, one line for each group:

- *group name*
  The name of the group.

- *group passwd*
  If the group uses a group passwd, the encrypted password is stored in this field. In general, group passwords are not used. The entry *VOID* or a blank indicates that no password has been defined for this group.

- *group id*
  The group id is a small integer that uniquely identifies the group. This is the number that is used in the user's password file entry in the group id field.

- *login name1, login name2, ...*
  This is a comma-separated list of user login names belonging to the group. When a new user is added, you need only add the name of the new user to the end of the list.

A sample entry is shown below:

```
finance:VOID:22:drt,pbw,sally
```

Task 2: Creating a Home Directory

Making a home directory for a new user is very much like making a directory for your own use. The main difference is that you must be logged in as the super-user and you must remember to change the ownership of the new user's directory to the user. The following series of commands makes a home directory for a user with the login name of *sally* and changes ownership of the new directory to that user:

```
# cd /work
# mkdir sally
# chown sally sally
# chgrp sally finance
# cd sally
# su sally
sally@host cp /usr/lib/Profile .profile
sally@host cp /usr/lib/Login .login
sally@host cp /usr/lib/Cshrc .cshrc
sally@host exit
#
```

The *chown* and *chgrp* commands, respectively, change the ownership of the newly-created directory and add user *sally* to the group named *finance.*

The last six commands in the example copy local versions of template *.profile, .login,* and*.cshrc* files to the new user's home directory. It is a convenience to users if you create a local, template version of these files for users to use and modify as they wish. See *User Environment,* below.

The above procedure gives the minimal actions needed to set up the files and directories for a new user. By turning the procedure into a more general shell script, you can simplify the process of creating and assigning ownership for other user files and directories.

Task 3: Copy X Window Configuration Files

Move to the user's home directory and copy the system's default X Window's configuration files (Xdefaults, Xdesktop and Awmrc) as follows:

```
# cp /usr/lib/Xdefaults .Xdefaults
# cp /usr/lib/Xdesktop .xdesktop
# cp /usr/lib/Awmrc .awmrc
```

The new user may now use the *xstart* command to invoke the X Window System after logging in. See the chapter entitled *X Windows* for more on the X configuration files and see the *Start Here*

User Environment

guide for information on how to use X. Stardent 1500/3000 users can use either the standard UNIX (Bourne) shell, */bin/sh,* or the C-shell, */bin/csh* as their interface to the operating system. (Users invoke the X Window System with the *xstart* command. See the *X Window Manual* for details.)

Login Shell Selection

A user's choice of shell interface should be included in the */etc/passwd* file when the user is added to the system. It is the last item in the user's */etc/passwd* entry.

```
root:wr&%{x81:0:1:superuser:/:/bin/sh
sally:Wee65kp01X:144:10:Sally. Philips:/work/sally:/bin/csh
```

If the standard or Bourne shell is specified, the following two files are executed in order whenever a user logs in:

(1) the *system profile, /etc/profile*

(2) the user's individual *.profile* file, located in the user's home directory.

If the C-shell is specified, the following two files are executed in order whenever a user logs in:

(1) the user's *.cshrc* file, in the user's home directory

(2) the user's *.login* file, in the user's home directory

The */etc/profile* and *.profile,* files or the *.login* file are executed each time the user logs in; the *.cshrc* file is executed every time a new shell is spawned.

NOTE
Some shell commands (such as *ls*) have both AT&T System V and BSD versions. To make sure that users get the commands consistent with their login shell choice, check that the environment variable $PATH lists the pathnames */usr/bin* (for Bourne shell users) and */usr/ucb* (for C-shell users) in correct order. If Bourne shell is the choice, */usr/bin* should be listed first; if C-shell is the choice, */usr/ucb* should be listed first.

Template versions of user login files are located in */usr/lib* under the names *Cshrc, Login* and *Profile.* (See the *End-User Interface Guide* and the *Programmer's Reference Manual* for more on the *.profile*, *.login*, and *.cshrc* files.)

The system profile, */etc/profile,* contains overall system variables, commands, and messages. A system profile such as the one below is shipped with your Stardent 1500/3000.

```
PATH=$PATH:
TERM=titan40
TZ=PST8PDT
MAIL=/usr/spool/mail/$LOGNAME
export TERM PATH TZ MAIL
```

You can use the system profile to customize your system's environment and to execute commands for and send messages to any Bourne shell users on the system. There is no analogue to the system profile in the C-shell world, so any commands or messages for C-shell users must be included in their individual *.login* or *.cshrc* files.

Users can change shells during a login session by issuing the command *sh* or *csh.*

In some cases you may wish to give a user access to only the restricted shell, or *rsh.* You might do this if you wish to restrict the user to a single interactive application program. The restricted shell does not allow a user to change directories, change the value of the environment variable $PATH, access files other than those in the user's home directory or specified in $PATH, or redirect output. To choose the *rsh* interface, simply include that information in the */etc/passwd* file:[1]

```
rocky:Wee65kp01X:120:8:Rocket J. Squirrel:/usr/rocky:/lib/rsh
```

Communicating With Users

As system administrator you have various ways available to communicate information to users on the system, including messages received by users at login time and special announcements that are immediately written to all users.

Login Special Messages

The system profile allows you to send messages to Bourne shell users as they log in. You add the line

```
cat /etc/motd
```

to the system profile to list the file */etc/motd* (message-of-the-day). Then place any messages in the */etc/motd* file. C-shell users can list the file by including the above command in their individual *.login* files.

You can also use the *news* and *mail* commands to send messages to users. The *news* command lets you create bulletin board notices that can be read by users if they wish. *mail* lets you send individualized messages to one or more users. See the *End User Interface Guide* for instructions on using *mail* and the *Commands Reference Manual* for a complete description of both commands.

Special Announcements: The wall Command

Special or emergency announcements, such as the imminent shutdown of the system, can be transmitted immediately to all users through the *wall*(1) (or write-all users) command. Just issue the command, followed by the message you want to send, followed by end-of-file (CTRL D). Make sure you are super-user when you issue the command, so you can overwrite any write protections individual users may have invoked. Here is an example:

---

[1] Don't confuse the restricted shell with the Berkeley UNIX remote shell, or *rsh* (*/bin/rsh*). See *sh* in the *Command Reference Manual* for details on the restricted shell and *rsh*(1C) for information about the remote shell.

```
# wall
Please log off immediately, the system must be brought down !!
CTRL D
#
```

Removing a User

The most important thing to remember about removing a user and all attendant files from the system is that once the user's files are gone, they're gone for good. Make a tape copy of the user's files before removing them. An easy way to that is to use the *find*(1) command:

```
# find /work/sally -user sally -cpio /dev/rmt16
```

This command locates all files and directories owned by *sally* and copies them to a tape mounted in the QIC tape drive. Now it is safe to go ahead and remove the user's files from the system.

The command to remove all the user's files is quite simple:

```
# cd /work/sally
# rm -rf sally
```

The *rm* command forces (*–f*) the recursive (*–r*) removal of all files and directories, including the current directory, owned by *sally.*

Now you need to remove the user from the */etc/group* file. Use an editor to do this.

To remove a user from the mail system, use the following commands:

```
# cd /usr/spool/mail
# rm sally
```

Finally, use an editor to delete the user's entry in the */etc/passwd* file.

Be careful about recycling "sally's" user id. When you make a tape backup of *sally's* files, the ownership is specified by user id, not login name. If *sally's* user id is given away to someone new, her files are now owned by the new user.

SYSTEM SECURITY AND TUNING

CHAPTER TWENTY-SIX

This chapter gives some basics on system security and tuning.

System Security

As a system administrator you are responsible for ensuring security on your Stardent 1500/3000 system. We recommend the following:

- Control physical access to the machine. The only way you can prevent security problems associated with physical access is to keep your Stardent 1500/3000 in a secure place, take the key when it is not in use, and do not allow unsupervised access to the system console.

- Require each login to have a password. Passwords should not be obvious; avoid names, numbers, and dates that can easily be guessed by a creative potential intruder. Passwords can be aged if you wish, but if you only have a small number of users on the machine, that shouldn't be necessary. See the *Commands Reference Manual* for details on password aging and the *End User Interface Guide* for general information on passwords.

- Set default permissions to be as restrictive as possible, given your machine's application and users. You can add a *umask*(1) command to your system profile to set the default permissions. For instance, the command line

  ```
  # umask 137
  ```

  sets default permissions to read-write for user, read for group, and none for other. See the *End User Interface Guide* and the *chmod*(1) and *umask*(1) commands in the *Commands Reference Manual* for more on how permissions are set.

NOTE
Note that Stardent 1500/3000 does not support password aging and that passwords must be kept in the file */etc/passwd*.

The *umask*(1) command restricts permissions using the logical inverse of the octal permission codes. In the example, the code 137, meaning 1 for user, 3 for group, and 7 for other, denies execute permission (1) for the user, write and execute permission (2 and 1) for the group, and read, write, and execute permission (4, 2, and 1) for others.

- Be careful about dial-up ports. Don't let sensitive, unencrypted information pass over dial-up lines.

- Guard the super-user password closely. Keep the number of people that know the password small, and change the password frequently. Examine the */usr/adm/sulog* file regularly to see if any unauthorized commands have been issued. Make sure you log off as super-user if you must be away from your terminal.

- Make backups of files and directories on a regular, frequent basis. In this way you can recover easily if a security violation results in corrupted data. See *File System Administration* in this guide for details.

- Use the restricted shell, */lib/rsh,* for users who only need access to a single, interactive program.

In addition to following these guidelines, you should be aware of potential security pitfalls involving special permission bits: the set-user-id (set-UID) bit and the set-group-id (set-GID) bit.

Special Permission Bits

Special permission bits are the UNIX system's way of letting users do necessary administrative functions by executing or writing to files that are normally reserved for super-user use. It is an useful scheme that unfortunately has serious security risks. To avoid problems you need to monitor the status of files that have special permission bits set.

The process of setting user passwords provides a good example of special permission bits. When a user changes a password with the *passwd* command, that change is recorded as a new encrypted password in the */etc/passwd* file. The */etc/passwd* file, however, only allows write permission for the super-user. For the changed password to be implemented, the user must be given temporary super-user privileges. This is done through a set-UID bit in the */bin/passwd* file. (The */bin/passwd* file is the file that contains the *passwd* program.)

Here is a typical listing for the */bin/passwd* file:

```
Stardent 1500/3000: ls -l /bin/passwd
-rwsr-xr-x 1 root     10332 Jan 07 1987  /bin/passwd
```

The *s* that appears in the user-execute position of the permissions summary is the set-UID bit. It gives the person issuing the *passwd* command super-user privileges *while the command is running*.

Set-GID bits appear in a similar fashion, though the meaning is somewhat different. Here is an example.

```
Stardent 1500/3000: ls -l /usr/local/bin/action
-rwsr-sr-x 1 project1     10332 Jan 07 1987 /usr/local/bin/action
```

Here, users executing the program */usr/local/bin/action* are given the same privileges as the members of the group *project1, while the program is executing*.

How do set-UID and set-GID bits cause trouble? A writable set-UID file can be another program copied to it. For example, if the *su* (switch user) command has write permission for others, anyone can copy the shell onto it and get a password-free version of *su*. Set-GID bits are far less dangerous, but in some cases they can cause trouble.

The best way to avoid set-UID problems is to not let any set-UID program be writable by anyone other than root. You can also save trouble by restricting use of the set-UID bit to programs that are shipped that way with your Stardent 1500/3000.

System Tuning

Parameters defined in the */config/master.c* file can affect aspects of the system's performance. We recommend that you touch only the variables listed in Table 26-1 and described below.

Table 26-1. System Tuning Variables[1]

Item	Function	Default	Mod. Status
Files:			
/config/master.c	kernel configuration file		
Variables (in master.c):			
NPROC	maximum number of processes	160	*
NBUF	maximum number of buffers	200	*
MAXQUANTUM	CPU time between successive looks for new jobs	30	*

1. Number of processes. The variable *NPROC* determines the maximum number of system processes. Choosing a smaller number increases available memory, but the overall effect on performance is not significant.

2. Number of buffers. The number of buffers, *NBUF*, can be changed as desired; however, we recommend that the number remain above 80.

3. CPU time between successive looks for new jobs. The variable *MAXQUANTUM* gives the CPU time between successive looks for new jobs. The units are 10ms, so the value 30 means an interval of .3 seconds. Changing *MAXQUANTUM* affects the trade-off between response time and throughput. We recommend that you keep the value 30, unless response time is particularly important; in that case, lower the number. Do not, however, assign a value lower than 10.

[1] The "modification status" column in the table shows whether or not the file requires modification. If the modification status box is empty, no modification is recommended. (Commands fall in this category.) If the box has a single asterisk (*), modification is optional. If the box has a double asterisk (**), the file or variable must be examined (and modified if necessary).

SOFTWARE INSTALLATION

CHAPTER TWENTY–SEVEN

This chapter shows you how to install Stardent 1500/3000 system software from tape. It describes system software installation and installation of add-on software products (Fortran, NFS, DWB, and so on) and patches.

Installing Add-ons and Patches

To install software add-ons or patches from tape, use the shell script */etc/installpkg*. Follow this procedure.

Step 1. Insert the cartridge tape and wait for it to rewind.

Step 2. As root or in single-user mode type

```
# installpkg
```

A message about the contents of the update appears on the screen, together with any special installation instructions. You see file names listed as the files are read in. When the read-in is complete, you get a system prompt. The update is now installed.

Step 3 (for NFS and all add-ons that cause the UNIX kernel to be reconfigured). If you wish to save the old UNIX kernel located in */unix*, copy it to a new location before rebooting the system.

Installing System Software

Stardent 1500/3000 operating system software is shipped on an Stardent distribution boot tape (tape 1 of 2). The boot tape contains files located in consecutive sections of the tape.

- Section 0 contains a copyright notice,
- Section 1 contains an installation shell script,
- Section 2 contains the format program,

- Section 3 contains the standalone shell (SASH),
- Section 4 contains the mini-root,
- Section 5 contains the root file system.

A second tape is also shipped with the system. The tape contains the X+ Window System, system acceptance test, MATLAB programs, benchmarks, man pages, and example and demo programs. You can install these programs if you wish (if they are not already on disk).

If you are installing system software on a new or previously corrupted disk, follow the procedure in the next section, "Installing System Software, New Disk." If you are updating your system, be sure to read any Release Notes shipped with your system software, prior to proceeding. The Release Notes warn about any specific installation requirements for the the release you are installing. Then follow the steps in "Installing System Software Updates" below in this chapter.

Installing System Software, New Disk

Step 1: Format the disk, if necessary. Stardent 1500/3000 is shipped with a formatted primary boot disk. If you need to format the disk, follow the procedure in *Disk Drives* in this guide.

Step 2: Copy the mini-root to a the swap partition. The mini-root contains enough of the UNIX system to let you install the entire root file system. Once you do that, the mini-root is no longer needed. You should copy the mini-root to the default *swap space* partition on the disk (usually partition 1). The mini-root is then erased as the swap space is used.

Issue the following command to copy the mini-root from section 4 on the tape to partition 1 on the primary boot disk. The command assumes that you are using the default swap partition (partition 1) on the default primary boot disk (controller 1, device 5).

```
prom 1 >   copy scsi(0,6,4) scsi(1,5,1)
```

NOTE
You can use the *nvram* command to check your current swap partition.

The mini-root is about 12 MB. As the copy proceeds, the number of bytes copied is flashed and updated.

Step 3: Boot the sash. Once you have formatted the disk and copied the mini-root to the temporary partition, you should boot the SASH.

```
prom 1 > b scsi(0,6,3)
```

When the SASH has been booted you get a SASH prompt:

```
sash 1 >
```

Step 4: Boot UNIX from the mini-root. Boot UNIX from the mini-root as follows:

```
sash> b scsi(1,5,1)unix rootdev=scsi(1,5,1)
```

When booting is complete, you are in single-user mode UNIX and you see the familiar # prompt appear.

Step 5: Populating the root file system.
Type

```
# /etc/mkroot
```

/etc/mkroot is a shell program that makes and mounts the root file system and copies (using *cpio)* the root files from tape to the newly created root file system. It prompts for the file system to be used as the root file system. Here is an example session.

```
# mkroot
Installing the root filesystem (with default partition /dev/dsk/c1d5s0)

Enter alternative partition (e.g. /dev/dsk/c1d4s0),
if desired (cr for default): <RETURN>

Make a new filesystem on /dev/dsk/c1d5s0? (y/n) y

        output of mkfs

Enter 'q' if there are files on /dev/dsk/c1d5s0
that you would like to save, but have not done so yet: <RETURN>

Mounting the root filesystem on /mnt

Warning: <> mounted as /mnt

Copying in unix files

        few minutes wait, then long list of files printed

Setting /etc/hosts and /etc/hosts.equiv:

Do you want to make a file system on /dev/dsk/c1d5s2? (y/n) y

this destroys all data on /dev/dsk/c1d5s2
are you sure you want to proceed? (y/n) y

        output of mkfs
```

```
#
Copying /stand/sash into the volume header of /dev/dsk/c1d5s0

Done.
#
```

Step 6: Boot UNIX from the complete root file system. At this point in the installation procedure, the complete root file system has just been created and installed, and UNIX is running on the mini-root file system. Now boot UNIX from the complete root file system that was just copied from tape.

Type the following commands to leave the mini-root:

```
# sync
# sync
# halt
```

Before booting UNIX you can check to see if the PROM environment variables have the correct default values. Type **n** at the PROM prompt to see if you have the following defaults:

```
bootfile=unix
bootmode=a
path=scsi(1,5,0)
rootdev=scsi(1,5,0)
swapdev=scsi(1,5,1)
secondary=scsi(1,5,8)sash
```

If you need to set the values of the PROM environment variables, use this syntax:

prom 1> **n** *variablename=value*

For example,

prom 1> **n bootfile=unix**

You can now boot UNIX from the default disk partition. Since all defaults are set, you need no arguments on the command line:

prom 1> **b**

The UNIX system boots to the run level specified in the */etc/inittab* file. The default level is 1 (single-user). See *System Processes* in this guide for a general description of the */etc/inittab* file.

Step 7: Create remaining file systems. At this point in the initial software installation procedure, the primary boot disk is formatted, and the primary disk contains the complete root file system. It is now possible to create other file systems to reside in available partitions on the disk. See *File System Administration* in this guide for details.

Step 8: Install optionally installable system software. Optionally installable software such as examples and demos can be installed using */etc/installpkg*. The general procedure is in *Installing Add-ons and Patches* at the beginning of this chapter. Before installing optionally installable software be sure to read any *Software Release Notes* that accompanied the distribution and follow any instructions given there.

Installing System Software Updates

To update the existing root file system, use the *rlsupdt* (release update) shell script. Note that *rlsupdt* is an alternative to *mkroot;* you use one or the other, **not both**.

Step 1. Read the entire section before you begin the installation process.

Step 2. Halt the system using */etc/shutdown*

```
# cd /
# /etc/shutdown -y -g0

      [ system shutdown output ]

The system is down.
# sync
# sync
# halt
```

Step 3. Load the first distribution tape (1 of 2), and copy the miniroot from tape to your swap partition. Note that the following instructions give default values for root and swap partitions; if you have changed them on your system, substitute the appropriate values:

NOTE
Use the "nvram" command at the PROM level to check the location of root and swap on your system.

```
prom #> copy scsi(0,6,4) scsi(1,5,1)
```

The copy takes about 10 minutes to complete; the miniroot is approximately 12 MBytes.

Step 4. Boot the operating system from the miniroot:

```
prom #> b scsi(1,5,1)unix rootdev=scsi(1,5,1)
```

Step 5. Run *rlsupdt* to upgrade the system:

```
# rlsupdt
Stardent software incremental update procedure

The following files will be preserved:
                .cshrc
                .login
                .rhosts
                .profile
                etc/TIMEZONE
                etc/checklist
                    .
                    .
                    .
        (and so on -- list of files from /prsrv)
```

If you want to add to this list, answer 'y' to the following question. Edit the '/prsrv' file and add the files you want to preserve, then invoke 'rlsupdt' again to continue.

```
Do you want to add files to the preserve list? (type 'NO' to start update):
```

You can add to the */prsrv* file and restart, or type "NO" to start the update, as follows:

```
Do you want to add files to the preserve list? (type 'NO' to start update): NO
Enter full path name of root partition (default /dev/dsk/c1d5s0):
 /dev/dsk/c1d5s0
Checking /dev/dsk/c1d5s0
(fsck output)
```

When *fsck* is through, the necessary filesystems used in the upgrade are mounted, and an attempt is made to find any 'preserve' files in the current root file system. You can ignore any "find: stat() failed" messages that appear if 'preserve' files are not found. The script also removes files listed in the /destroy file; these are files that need to be removed from your system for a clean update. Again, don't worry if files are not found.

```
Copying preserve files

Removing /usr/include/bsd/netinet so that new link may be built
Removing /usr/X11 so that new link may be built

Winding tape forward . . .

Doing incremental update
(cpio output)
```

The script copies all designated 'preserve' files to /preserve, and then copies the complete new distribution from tape. It takes about 5 minutes to seek forward to the *cpio* file containing the content of root, and then about 20 minutes to complete the copy. When it finishes:

```
Copy back ALL preserved files? (y/n) y

Copying back preserved files
```

If you wish to copy back only certain preserved files, answer 'n' to the query and you are given the opportunity to copy back file by file.

This completes the update of the root partition.

Step 6. Halt your system, make sure nvram settings for the following are correct, then boot the new root:

```
# sync
# sync
# halt
prom #> nvram
 . . .
bootfile=unix
secondary=scsi(1,5,8)sash
rootdev=scsi(1,5,0)
swapdev=scsi(1,5,1)
path=scsi(1,5,0)
screen_font=2
 . . .
prom #> b
```

Step 7. The second tape cartridge of your distribution contains optional software, including the X+ Window System software, a system acceptance test, standard benchmarks, MATLAB software, online man pages, Doré example programs, and Demonstration programs. Each of these can be optionally loaded to the destination directory you specify.

If you wish to load any or all of these, use a partition other than your root partition. The programs are loaded by default into an */opt* directory, which can be the mount point of an alternate partition. Since utilities in some of the areas are pathname dependent, the install script automatically builds links back to the root as needed ("Link Directory").

If you wish to load all the optional software into */opt*, mount a partition with approximately 95 MBytes of space as */opt* (see your current *Software Release Notes* for exact current sizes), and proceed (the script confirms space is adequate). Note that, since some of the packages (demos, examples, benchmarks) contain sources that require additional space if compiled, you may want to increase the space allotted accordingly.

If you want to install only certain packages, or wish to spread the packages into different partitions you can install them individually. Here is an example that assumes that the X+ Window System software and man pages are being installed into an */opt* directory:

When you are ready to begin the install, load the second tape cartridge (2 of 2) in your drive, **make sure the system is in multiuser mode** ("init 2"), and type:

```
# installpkg

Reading past copyright
Beginning to install the Optional packages. . .

Calculating the available space in /opt. . .

Package              Size in Blocks    Default Directory    Link Directory

X+ Window System          #              /opt/X11             /usr/X11
man pages                 #              /opt/man             /usr/man
system acceptance                   #              /opt/tests             /tests
benchmarks                #              /opt/bench
MATLAB programs           #              /opt/matlab          /usr/lib/matlab
Dore example programs     #              /opt/examples
Demo programs             #              /opt/demo

Total disk space required for all packages: # Blocks

Do you want to install ALL of these
packages in /opt? n

Do you want to install the X+ Window System (need # Blocks)? y

Where would you like the X+ Window System
installed (default is /opt/X11)? <CR>

/opt/X11 does not exist, create? y

Calculating the available space in /opt/X11 . . .

Do you want to install the systests (need # Blocks)? n

The installation of systests will be skipped.

Do you want to install the benchmarks programs (need # Blocks)? n

The installation of benchmarks programs will be skipped.
```

```
Do you want to install the MATLAB programs (need # Blocks)? n

The installation of MATLAB programs will be skipped.

Where would you like the man pages
installed (default is /opt/man)? <CR>

Calculating the available space in /opt/man . . .

Do you want to install the dore example programs (need # Blocks)? n

The installation of dore example programs will be skipped.

Do you want to install the demo programs (need # Blocks)? n

The installation of demo programs will be skipped.

Starting to install

Recalculating the available space in /opt/X11. . .

Copying X+ Window System package into /opt/X11

      [ cpio output ]

Linking /opt/X11 to /usr/X11

Reading past systests
Reading past benchmarks programs
Reading past MATLAB

Recalculating the available space in /opt/man. . .

Copying man pages into /opt/man

      [ cpio output ]

Linking /opt/man to /usr/man

Making indices

Finished with the installation.
Rewinding tape.
Cleaning up.
```

The shell script for running MATLAB is copied to */usr/bin* during installation. If you have */usr/bin* in your path, you can run MATLAB by simply typing

<> **matlab**

For the demos, log out and in again as "demo," and you are prompted to start the window system and the demos.

Step 8. When you are finished installing software you can save approximately 5 MBytes by removing the man page sources installed on your system (in */usr/man/man(x)*). These raw sources are necessary for the recompilation of the indices used by *man*(1), so all pages must be installed and the indices rebuilt before removing the sources. Issue these commands (make sure the install is complete before doing so):

```
# cd /opt/man
# rm -rf man*
```

You can always re-install from tape later, if you purchase an additional package.

Step 9. If your system includes additional software packages (NFS, DWB, Fortran), you can install them by loading the appropriate tape and using *installpkg*. See the *Software Release Notes* pertaining to those packages for specific installation instructions.

SYSTEM CRASH PROCEDURE

CHAPTER TWENTY–EIGHT

It is sometimes important to be able to generate a system dump to aid Stardent engineers in diagnosing system failures.

CAUTION
We recommend that the dump device you specify in the *dumpdev* variable refer to the "swap," or other unused partition on the disk. If the cartridge tape drive is specified as the dump device, be sure an empty or scratch tape is installed in the tape drive before you issue the *sysdump* command.

In a typical scenario you have just experienced a panic or assertion failure. You press the reset switch and see the PROM prompt appear. You can now generate a system dump with the *sysdump* command. (The system dump can later be analyzed using the *crash*(1M) utility.)

The *sysdump* command requires you to specify the device that is to accept the dump. You can do that on the command line or you can specify the dump device in the NVRAM variable *dumpdev*.

To specify the device on the command line, use the syntax

`prom 1>` **sysdump** *devicename*

For instance, to copy the system dump to cartridge tape, insert a 1/4-inch tape in the internal cartridge tape drive (see the chapter entitled *System Overview* for cartridge tape drive instructions) and issue the following command:

`prom 1>` **sysdump scsi(0,6,0)**

Once the system dump has been copied you can send the tape to Stardent Customer Support for analysis.

To analyze the system dump locally, use the swap partition on the primary boot disk as the dump device. Issue this command:

`prom 1>` **sysdump scsi(1,5,1)**

If you select the swap partition (or other unused disk partition) as the dump device and then wish to transfer the crash dump to tape, boot the Unix system and then use the *dd* command as follows:

```
# dd if=/dev/dsk/c1d5s0 of=/dev/rmt16 bs=1024K count=N
```

where

Setting the value of dumpdev

You can set the NVRAM variable *dumpdev* from UNIX (as super-user) or from the PROM by issuing the *nvram* command:

prom 1> **nvram dumpdev=***devicename*

Once the dump device is specified you can take a crash dump simply by issuing the command

prom 1> **sysdump**

Stardent

PROM
MANUAL

Appendix A
Installation/Administration Guide

CONTENTS

List of Figures

List of Tables

PREFACE

The Stardent 1500/3000 system supplies a PROM that is accessed during system power-on or reset and provides the minimal set of functions needed to get a Stardent 1500/3000 sytstem running. These functions include initial system conditioning, console selection, primary boot, primary dump, examine deposit, and confidence testing.

This document describes the features and operation of this PROM. There is also debugging information and a description of system boot and the interface to the standalone programs.

There are two distinct levels of operation of the Boot PROM.

- PROM - the Boot PROM level.
- DPROM - the debug (bringup) PROM level.

In the PROM level, you can obtain a list of the commands that it understands by simply typing **help** or **?** followed by a RETURN.

PROM
: PROM does not know about Unix file systems, but can do everything that **sash** can do. Its prompt, when active, is **promX>**.

DPROM
: DPROM is the debug PROM state. If there is a problem with system memory or I/O, or if you want to execute without using memory (restricting yourself to PROM/MIPS CPU/ and DCP port), you can bring the system into a debugger state with more limited capabilities in what is called the debug PROM. The prompt will then be "dprom>". Chapter 3 describes the debug PROM.

SASH

SASH, the Standalone Shell is a more richly configured version of the PROM Monitor. It provides most of the same commands as the PROM Monitor, but is configured with more device drivers and file system types. SASH exists so that the PROM Monitor need not be dependent on the operating system and file system formats.

SASH knows about file systems and path names. It can perform all of the commands shown here (with one exception - there is no init command in SASH) The SASH prompt is: **sash x>**, where X is the number: 0, 1, 2 or 3 depending on which of the CPU boards is executing boot PROM code at this time. Sash is resident in RAM and so it must be loaded from disk, tape, or over the ethernet. It may be read from the volume header of the boot disk when you give the *boot* command, typically by specifying:

```
b(0,0,8)sash
```

It is possible to boot from the volume header of the disk, or from tape. See the boot command for more information.

About This Manual

- Chapter 1 provides general details about what happens during the boot process.
- Chapter 2 provides details about the commands that SASH and PROM recognize.
- Chapter 3 provides details about the commands that DPROM recognizes. Note that although the commands are often similar, the command syntax may differ.

As the product matures, we expect that the command syntax will become common over all levels of the prom monitor. In particular, some of the DPROM functions will be entirely replaced by functions in the PROM level instead, thereby making the syntax common between the levels of prom.

Additionally, as indicated in *Appendix B*, there are currently two different sequences that the error indicating LEDs on the CPU board cycle through, one sequence for DPROM, then a second sequence for PROM. As DPROM functions are merged, the error indicators will merge as well, so that an LED pattern is used for only a single error indication.

BOOT PROM

CHAPTER ONE

When the Stardent 1500/3000 is first powered on, there may be as many as four CPU boards installed in the system. All of the CPU boards run the exact same boot prom, but only one of the CPU boards actually becomes the master processor (called the boot processor). The CPU board whose CPUID is the lowest becomes the boot processor. The actual boot sequence for all processors is shown in Figure A1-1. The point at which non-boot processor halt their processing is shown as well.

```
Clear MIPS Status Register
Toggle all LEDs
Zero MIPS Cause Register
Check keyswitches for 2400 or 9600 baud diagnostics messages
Serial port self-test (DCP)
Enable the Bus and set the bus watcher
Clear Doorbell Register
Find CPU ID
Determine Who Is BootMaster
If BootMaster, Set BootMaster bit
If Power-on Reset, perform CPU go/nogo test
Test ETLBs as memory
If not the boot processor, go to the slave routine.
If Power-on sequence, do slave CPU go/nogo test.
Test Slave's ETLBs as memory
If Boot Processor, continue.

Clear LED register, set Green (pass)
Perform Diagnostics on all boards
  (for all boards, set FAULT LED while tests
   in progress; if tests pass reset FAULT,
   light PASS LED for each board)
  Memory
  I/O
  Graphics
Read PROM Global area
 Initialize PROM 'world'
 Check the keyswitch to check for autoboot status.
 If autoboot is set in NVRAM, boot SASH and UNIX.
   otherwise output main prompt (SASH or PROM).
```

Figure A1-1. PROM Boot Sequence

Status information is output while the boot process is taking place. The selftest display sequence is sent only to the DCP. Once the selftest is complete, the output is sent to the graphics console. The output is similar to that shown in Figure A1-2.

```
Power-Up [go-no-go testing]
DCP Internal Loopback
Checksum bootprom...Done
Test and Init cache...Done
Test and Init ETLBs...Done
Config and Init memory...Done
Test and init I/O Board...Done
Test and init Graphics Board...Done
Self Test Complete
Titan Monitor, Version 6.5, Fri Jan  8 18:35:41 EST 1988
Board inventory:
slot  status       revid    board type
1     0x01580000    18       I/O
5                   0        CPU, cpuid: 2
6     0x01ff0027    1        Memory, 32 Mbyte (memid: 0)
8     0x01584000    0        Graphics
```

Figure A1-2. Typical Power-On Display Output

If the memory fails to respond, the prompt **dprom>** appears immediately. The system responds only at the **DPROM** level as described in Chapter 3 of this manual.

If an exception occurs and there is no handler linked in to handle the exception, again the **DPROM** level is invoked. Some clues to the location of the problem can be determined from the diagnostic messages that appear on the console before the system enters **DPROM** mode.

Note:. A method to execute extended diagnostics will be added to the DPROM capabilities. Not yet available.

Memory Card Installation

When memory cards are installed in the Stardent 1500/3000, the best performance can be obtained if the cards are installed as shown below. The default memory operation is 8-way interleaved. If memory cards are installed properly, it allows the system to use 16-way interleaving in place of 8-way interleaving. Using 16-way interleave improves the performance of the memory system and thus is highly desireable.

- If the Stardent 1500/3000 has four memory cards, and all have the same capacity, 8 MB per card, 16 MB per card or 32 MB per card, the cards should be installed in the labeled memory card slots, that is, slot numbers 3, 4, 6 and 7.

- If the Stardent 1500/3000 has only two memory cards, and both have the same capacity, (8, 16 or 32 MB per card), the two memory cards should be installed in the memory slots labeled for a two memory card system, namely slots 3 and 6.

- If the Stardent 1500/3000 has four memory cards, and the sizes of the memory on the cards is not the same, there should at least be two sets of pairs of the same memory size. That is, there should be two card pairs, one pair having a higher memory capacity, and one pair having a lower memory capacity:

Higher Capacity Card Pair		Lower Capacity Card Pair
two cards 32 MB each	and	two cards 16 MB each
or		
two cards 32 MB each	and	two cards 8 MB each
or		
two cards 16 MB each	and	two cards 8 MB each

  - Install the higher capacity card pair in slots 3 and 6.
  - Install the lower capacity card pair in slots 4 and 7.

If the user installs memory cards in any combination other than those listed above, the boot PROM will only enable 8-way interleaving. In other words, the following combinations of memory cards automatically select a less efficient operating mode for the memory:

- An odd number of memory cards installed.
- Two memory cards installed, with a different memory capacity on each card.
- Two memory cards of the same capacity installed, but in the wrong slots. (If installed in slots 4 and 6 or slots 3 and 7 as a pair, the memory cards will function, but not as efficiently).
- Four memory cards installed, but with one matched pair (same memory capacity) installed in the even numbered slots and the other matched pair installed in the odd numbered slots.

Once again, the operating system can use the memory most efficiently when the matched pair of memory cards having the largest memory capacity is installed in slots 3 and 6.

Front Panel Indicators

There are two front panel indicators. One is a red LED that lights the Ardent Computer Corporation logo. This LED monitors the state of the power supply. As long as power is applied, this LED is lit.

The other front panel indicator is a red LED that is used for diagnostic purposes. The Diagnostic LED of the Stardent 1500/3000 shows the boot progress as follows:

Preboot	LED is Blinking.
Booting Unix	Diagnostic Front Panel LED flashes slowly while Unix is being loaded
Running Normally	Diagnostic Front Panel LED is OFF
Problem	Diagnostic Front Panel LED is on solidly. User should examine the console for error messages and examine the CPU Board (and other board) LED panels for clues about the failure.

PROM COMMANDS

CHAPTER TWO

Command Summary

Table A2-1 is a summary of the commands that PROM and SASH understand, along with the acceptable syntax for each command.

Table A2-1. Command Summary

<table>
<tr><th colspan="2">Initialization Commands</th></tr>
<tr><td>Help</td><td>h[elp] or ?</td></tr>
<tr><td>Boot</td><td>b[oot] [<file>] [<args>]</td></tr>
<tr><td>Load only(serial)</td><td>l[oad] [<character_device>]</td></tr>
<tr><td>Init PROM</td><td>i[nit]</td></tr>
<tr><td>Reset</td><td>res[et]</td></tr>
<tr><th colspan="2">Breakpoint Commands</th></tr>
<tr><td>Start, restart</td><td>g[o] <PC></td></tr>
<tr><td>Show/Set Break</td><td>br[k] [<address>]</td></tr>
<tr><td>Single Step</td><td>s[ingle] [<instruction_count>]</td></tr>
<tr><td>Mod Single Step</td><td>S[ingle] [<instruction_count>]</td></tr>
<tr><td>Go Till Address</td><td>got[o] <address></td></tr>
<tr><td>Clear Selected BP</td><td>c[lear] [<breakpoint_list>]</td></tr>
<tr><th colspan="2">Memory Commands</th></tr>
<tr><td>Fill Mem.</td><td>f[ill] [-b | -h | -w] [-v <value>] <range></td></tr>
<tr><td>Show Mem.</td><td>d[ump] [-b | -h | -w] [<range>]</td></tr>
<tr><td>Modify Mem.</td><td>m[odify] [-b | -h | -w] [<address>] [= <value>]</td></tr>
<tr><th colspan="2">CPU/Register Commands</th></tr>
<tr><td>Switch CPUs</td><td>x <cpuid></td></tr>
<tr><td>Show/Mod Regs.</td><td>r[egister] [<register_name> [= <value>]]</td></tr>
<tr><td>Flush Cache</td><td>fl[ush] [-d | -i | -s]</td></tr>
<tr><td>Process ID</td><td>p[id] [<decimal_value>]</td></tr>
<tr><td>TLB values</td><td>t[lb] [<table_range> [= <tlbhi> <tlblo>]]</td></tr>
<tr><td>ETLB values</td><td>e[tlb] s0 | s1 | r0 | r1 [<table_range> [= <tlbhi> <tlblo>]]</td></tr>
<tr><td>NVRAM values</td><td>n[vram] [<variable_name>[=<string>]]</td></tr>
<tr><td>Disassemble</td><td>dis[assemble] <range></td></tr>
<tr><td>Debug PROM</td><td>z</td></tr>
<tr><td>Copy Files</td><td>cp <src> <dest></td></tr>
<tr><td>Dump System</td><td>sy[sdump] [<dumpdevice>]</td></tr>
</table>

Conventions

1. All address expressions are treated as hexadecimal unless explicitly encoded by the user as hex, binary, octal or decimal. There is one exception: the shift count in an address expression defaults to decimal.

2. All counts and table indexes are treated as decimal, unless explicitly encoded by the user (because you can specify hex, decimal, octal and binary.)

3. Optional items in a command are shown in square brackets.

4. For **sash** and **prom**, commands are recognized by their first one or two characters. You can use the other characters if you wish, but only those first couple are needed. For **dprom**, you must specify the entire command, not just an abbreviated version.

5. A command is terminated by pressing RETURN. RETURN is indicated in the paragraphs that follow by a notation "<CR>".

6. For **sash**, commands are case-sensitive. That is, if specified in lower case, a command must be entered in lower case and so on. For **dprom**, commands may be entered in either upper or lower case.

7. Command parsing is based on proper substrings of the command names. Look at the command named "boot". To execute this command, you can enter any proper substring of the full command name, at minimum the first characters as specified in the syntax. The syntax says that it would accept "bo", "boo" or "boot", followed by a delimiter character that could be a space or a RETURN. If you entered something like "boot1", it would not be considered a proper substring of the command and would respond "unrecognized command".

8. Note that there are several ways to access memory. You specify the PROM memory access method by using modifiers to memory requests. In the table, commands that accept addresses also accept modifiers that specify how the address is to be interpreted. For example, the m (modify) command can accept three different modifiers, 'c', 'u' and 'v':

```
 m - access memory as cached physical address
mc - same access as 'm' alone
mu - access memory as uncached physical address
mv - access a virtual address
```

(This also applies to the d command, go, brk, goto and fill.) If you simpy use "m" instead of explicitly specifying the access type, you get the default, which is cached physical addressing.

9. Where a command accepts an address *range,* that range can be specified in one of two different formats:

- Format 1: <address>:<address>
  specifies the beginning and ending addresses (inclusive) for the address range.
- Format 2: <address>[[,]<count>]
  specifies the beginning address with a count of the total number of memory locations to be affected by the command.

Alphabetical List Of Commands

The commands are listed in alphabetical order, sequenced by the letters that are used to call the command rather than by the command name. For example, "b" for *boot* is listed prior to "br" for *breakpoint* because the alphabetical sequence for the minimum letter sequence to request the breakpoint command precedes the boot command minimum sequence in the alphabetical sequence.

Display/Set Breakpoints (Command = br)

The command **br** sets a permanent breakpoint. A breakpoint causes a break every time it is encountered. A breakpoint must be explicitly cleared.

If a program stops at a breakpoint and you resume with a go or goto, the breakpoint is removed just long enough to execute the real instruction, then the breakpoint is reinstalled.

```
br[k] [<list>]
```

- Typing br or brk with no address displays all breakpoints. The system numbers the breakpoints and you can later clear them by number. Temporary breakpoints are labeled as such within the breakpoint list.

- Typing br with an address sets a permanent breakpoint at that address.

- 16 concurrent breakpoints may be set.

- Allows modifiers brku, brkc, brkv You specify the PROM memory access method by using modifiers to memory requests. The brk command can accept three different modifiers, 'c', 'u' and 'v':

  ```
   brk - access memory as cached physical address;
  brkc - same access as 'd' alone
  brku - access memory as uncached physical address
  brkv - access a virtual address
  ```

  If you simply use "brk" instead of explicitly specifying the access type, you get the default, which is cached physical addressing.

Boot (Command = b) (Command = bn)

bo or **bn**

boot the system - allows you to start any program from a boot source. This can be a disk, tape or the network.

```
b[oot] [ [<filename>] [<args>] ]
or
bn [ [<filename>] [<args>] ]
```

where

- **b** automatically starts executing at the entry point of the program. **bn** returns to command mode after loading.
- <filename> is the name of the file to load and run. If you boot without a filename, then the default file, is loaded and run. The name of the default file is located in the NVRAM, and is comprised of a pair of strings. One string defines the *path* (which normally defines a physical device or file system) and the other string defines the filename that is to be booted. See the topic titled *Titan Boot Procedures1 in this same section for more information.*

Titan Boot Procedures

The front panel key switches and the PROM monitor environment variables control the Titan boot sequence.

From the front panel one can: power-on and boot the PROM, reset and boot the PROM, power-on autoboot, reset autoboot, and cause UNIX to give to control to PROM monitor. The action taken by "autoboot" is dependent on the settings of the PROM environment variables.

The key actions to cause the various types of boots are given below. The upper key has a off, on, and spring loaded position. The lower key has a left and a right position. Remember that the bottom key controls the semantics of the upper key. The lower key in the right position indicates automatic boot procedures and in the left position indicates manual boot procedures. A summary of the front panel boot options is given below:

Top Key	Bottom Key	Action
Off to Power On	Diagnostic	Boot PROM
Off to Power On	Normal	Autoboot
Power On to Spring Loaded	Diagnostic	UNIX jump to PROM monitor
Power On to Spring Loaded	Normal	Reset, then autoboot
Power On to Spring Loaded	Normal, Diagnostic	Reset, boot PROM

Please note that the last front panel boot procedure requires the lower key to first be placed in the right position and then in the left position. This boot procedure will be discussed below.

From the PROM monitor one can boot an arbitrary standalone program, boot UNIX to single-user run level, or boot UNIX to multi-user run level.

Auto-booting using the Front Panel

Using the key procedure below, one can cause the Titan to autoboot the file specified by the PROM environment variables. If this boot file is UNIX, then UNIX will autoboot to the default run level specified by **/etc/inittab**. The default run level specified by the inittab file can be configured to be single user or multi-user.

To boot UNIX to the default inittab run level follow the procedure below.

Turn the lower key to the right position.

If the power is currently off, turn the upper key to the power-on position. If the power is currently on, turn the upper key to the spring-loaded position to the right.

The PROM monitor will boot and detect the autoboot request by reading the position of the lower key and seeing that it is in the right position. If UNIX is the default boot file, it will be booted. UNIX will look at the lower key switch and since it is in the right position, UNIX will continue the booting process to the default run level specified by the "/etc/inittab" file.

1.1. Booting the PROM Monitor From a Power-Off State Follow the procedure below to boot the PROM monitor using the front panel from a power-off state.

Turn the lower key to the left position.

Turn the upper key from the power-off to the power-on position.

Since the lower key is left, this upper key switch action request a manual boot.

The Titan will power on, the PROM monitor will go through its power on diagnostic sequence, and will prompt,

prom 1>

Please note that the **1** indicates CPU which is running the PROM monitor code. This CPU number can be any number between 0 and 3.

1.2. Booting the PROM Monitor From a Power-On State The procedure to boot the PROM monitor from a power-on state is a bit complicated. This results from the fact that this functionality must be provided by a front panel operation and and all the simple key switch operations have been allocated.

The difference between this procedure and other boot procedures is that the lower key must be moved from the right to the left position within a two second time window.

Follow the procedure below to boot the PROM monitor using the front panel from a power-on state.

Grasp lower key with your left hand. Do not let go of this key until this boot procedure is complete.

Align the lower key, by pushing in against the spring, so that it can be turned easily from the right to the left position. Note: turning only the lower key as no affect on a running system.

Turn the lower key to the right position.

Turn the upper key from the power-on to the spring loaded position.

Now, within two seconds, turn the lower key to the left position.

When the upper key is turned to the spring loaded position, the hardware reads the position of the lower key. Since it is in a right position, the hardware performs a reset. After the PROM monitor completes the machine diagnostics, it reads the lower key to determine if an automatic or manual boot is to be performed. Before the PROM monitor reads the lower key, you have turned the key to the left position, indicating a manual boot. Note, the PROM monitor guarantees the that a minimum of two seconds is provided from the system reset (upper key to the spring loaded position) until the PROM monitor reads the lower switch to determine automatic or manual boot.

When this key sequence is performed, the Titan will reset, the PROM monitor will go through its power on diagnostic sequence, and will prompt.

Manual Boot Procedures from the PROM monitor

To boot the PROM monitor, use the procedure described above. Once you have done this, you will see the PROM monitor prompt "prom 1> " on the console.

As stated above, the PROM monitor offers many booting alternatives. From the PROM monitor one can boot: an arbitrary standalone program, UNIX to single user run level, or UNIX to the default inittab run level.

1.2.1. PROM Boot Environment Variables The default boot actions taken by the PROM monitor are controlled by the PROM monitor environment variables. These variables are kept in non-volatile memory (NVRAM) and specify the default boot device, root device, boot file name, etc. Note that these variables are in NVRAM and persist across Titan power shutdowns.

When at the PROM monitor prompt, typing a **n** will print out the PROM environment variables. To set the PROM environment variables, use the syntax given below:

prom 1> **n**<variable name>=<string>

For example to set boot file to the string unix you would type:

prom 1> **n bootfile=unix**

To clear a PROM environment variable, type n, the variable name and a return, e.g.:

prom 1> **n foo**

The relevant boot related PROM monitor environment variables are given below:

bootfile -- The bootfile variable specifies the file to be loaded into memory and run. This typically is the path name to the UNIX kernel.

path -- This variable describes the device to be used to find the

bootfile. Examples of settings for the path variable are given below:

bfs()	Look for the bootfile using the Ethernet boot file server
scsi(0,0,0)	Look for the bootfile on SCSI controller 0, disk 0, partition 0
scsi(0,6,2)	Boot the file with tape index 2 from the cartridge tape.

Note that when the path specifies cartridge tape, it does not use the name specified by bootfile. It just uses the cartridge tape index to find the file. Note that 0 implies the first file, 1 implies the second file, etc.

rootdev -- The rootdev variable specifies the disk partition which UNIX will use as the root file system.

swapdev -- The swapdev variable specifies the disk partition which UNIX will use for swapping and paging.

bootmode -- The bootmode variable is used only the system in booting and the key switch is in the right position. The **bootmode** variable is set to **a** to indicate an autoboot and is unset to indicate a manual boot.

secondary -- In order to insulate the PROM monitor from changes, some of the device dependent and file system dependent knowledge has been exported to a secondary boot program know as **sash**, the standalone shell. The program sash is a larger, disk based, version of the PROM monitor. It supports the same syntax and commands as the PROM monitor and additionally understands the UNIX System V file system.

The program sash is used by the PROM monitor to complete the booting process. If the boot command specifies a UNIX file, the PROM monitor boots sash, which continues the booting request.

The default settings for the boot related PROM environment variables is given below:

bootfile	=	unix
path	=	bfs()
rootdev	=	scsi(1,5,0)
swapdev	=	scsi(1,5,1)
bootmode	=	a

secondary = scsi(1,5,8)sash

PROM Boot Commands. The PROM monitor offers two boot commands **boot** and **bn**. The syntax for these commands is given below:

b[oot] [<boot device>] [<boot file> <boot arguments>]
bn [<boot device>] [<boot file> <boot arguments>]

The **boot** command is a load and go boot command. The specified boot file is loaded into memory and it is given CPU control.

The **bn** command is a load and do not start boot command. The specified boot file is loaded into memory and is not started. The PROM monitor retains control of the CPU. This boot mode is used so that break points can be set before the program gets CPU control. Note after the break points are set, the program is given control by the **go** command. Please refer to the **go** command listing in the PROM Manual for more information.

For each boot command **b** or **bn**, the arguments <boot device> and <boot file> are optional. If they are not present in the boot command, then they are replaced with the values in the PROM environment variables **path** and **bootfile**.

Examples of valid boot commands are given below:

Typed Command	Command Expansion	Command Meaning
b	b <path><bootfile>	Boot using defaults set in environment variables
bn	bn <path><bootfile>	Boot, no start using environment variables
b unix	b <path>unix	Boot the file "unix" on the device specified by path
b scsi(0,0,1)	b scsi(0,0,1)<boot file>	Boot <boot file> from scsi(0,0,1)
b scsi(0,0,0)unix	b scsi(0,0,0)unix	Boot unix from scsi(0,0,0)
b scsi(0,0,8)sash	b scsi(0,0,8)sash	Boot sash from volume header

Boot Arguments to UNIX. The UNIX kernel takes boot arguments. These arguments are passed by the PROM monitor to the UNIX kernel using the standard UNIX and C argument passing conventions. The syntax and meaning of the UNIX boot arguments are given below:
Syntax:

prom 1> b<boot device>**unix** [**-s** | | **-m**] [**rootdev=**<root device>] [**swapdev=**<swap device>]

Meaning:

Argument	Meaning
-s	Boot unix to single-user run level
-m	Boot unix to multi-user run level
-rootdev=	specify the root device, e.g., scsi(0,0,0)
-swapdev=	specify the swap device, e.g., scsi(0,0,1)

Note only one of the arguments **-s** and **-m** should be specified as a UNIX boot argument.

The UNIX kernel during the startup process attempts to determine the boot mode (auto or manual), the root device, and the swap device. If these arguments are specified, UNIX overrides the default settings of the PROM environment variables, and uses the settings provided by these arguments. If these arguments are not specified, then UNIX determines the boot mode, by looking at the front panel switches and determines the root and swap devices by looking at the PROM environment variables.

boot mode -- If the **-s** or **-m** flag is not specified on the boot line, UNIX determines the boot mode by looking at the front panel switches. If the lower switch is in the right position, UNIX performs an auto boot, which boots UNIX to the default "/etc/inittab" level. If the lower switch is in the left position, then UNIX performs a manual boot, which does not use the "/etc/inittab" file and boots UNIX to the single-user run level.

root device -- If the **-r** argument is not specified, then UNIX reads the contents of the NVRAM variable **rootdev**, and uses this disk partition as the root file system.

swap device -- If the **-s** argument is not specified, then UNIX reads the contents of the NVRAM variable **swapdev**, and uses this device as the swap partition.

Partial List of Common NVRAM Variables..

baud = 9600	variable to set baud rate of DCP console
bootfile = unix	
path = scsi	
	a = autoboot; *m* = manual boot

bootmod = [a/m]

rootdev = scsi(1,5,0)

swapdev = scsi(1,5,1)	
dumpdev = scsi(0,6,0)	
secondary = scsi(1,5,8)sash	
console = dcp	
monitor = std/stereo	
screen_fore = *xxxxx*	monitor screen foreground color
screen_back = *xxxxx*	monitor screen background color
screen_curfore = *xxxxx*	monitor screen cursor foreground color
screen_curback = *xxxxx*	monitor screen cursor background color
netaddr = *xxxxx*	network address
etheraddr = *xxxxx*	Ethernet address
hostname = *xxxxx*	name of host machine

Copying Files (Command = cp)

This command lets you copy files. For **prom** and **sash** (standalone shell) is active. Its syntax is:

```
cp <src> <dest>
```

<src> and <dest> are filenames or SCSI devicenames, and can include a fully qualified path.

The copy command is useful, for example, for copying disk partitions from tape to disk or disk to tape with a rare occurrence of copying from one disk partition to another. For example, assume the distribution tape for the system is organized this way:

```
file 0 - format (standalone format program,
                  for formatting hard disk)
file 1 - unix    (standalone unix, bootable from tape)
file 2 - root partition
```

Use the copy command to install the root partition after formatting the hard disk by using the following command:

```
sash> cp scsi(0,6,2) scsi(0,0,0)
```

where:

> scsi(0,6,2) represents controller 0, device 6, and file number 2 on the tape.
>
> scsi(0,0,0) represents controller 0, device 0 (the boot disk), and partition 0 (the root partition).

Clear Breakpoints (Command = c)

To clear specific breakpoints, use this command.

```
c[lear] [<breakpoint_list>]
```

The breakpoint list is a blank-separated list of breakpoint numbers you would have obtained by observing the result of a **b** <CR>. If you enter **c** with no list, all breakpoints are cleared.

Display Memory (Command = d)

This command allows you to display memory read as bytes, half-words or words.

```
d[ump] [-b | -h | -w ] [<range>]
```

- The default count for dump is 1, with the default display mode as bytes.
- Dump remembers the last address and count displayed. If you type **d**, your start dump address will be one greater than the last address displayed, and the count will be the same as the count you entered previously.

Note that the display command does not apply to the values of internal MIPS chip registers. (Some of the registers external to the MIPS chip, mapped into the system physical memory space, do not like to be read as an incorrectly sized item... you will obtain inaccurate readings if such external registers are not read as their correct sizes.)

- If you explicitly enter the count, such as: **d -b 12345678,1** then only one byte will be displayed.

Note that there are several ways to access memory. You specify the PROM memory access method by using modifiers to memory requests. For example, the d (display) command can accept three different modifiers, 'c', 'u' and 'v':

```
 d - access memory as cached physical address;
dc - same access as 'd' alone
du - access memory as uncached physical address
dv - access a virtual address
```

If you simpy use "d" instead of explicitly specifying the access type, you get the default, which is cached physical addressing.

Disassemble (Command = dis)

To get a disassembly of the code at the next location following the most recent location displayed or modified, use the **di** command.

```
dis[assemble] <range>
```

where range is specified as described in the *Conventions* section of this chapter. The default count is 1 instruction.

Note: disassemble also remembers the last address and count that you have requested.

Note that there are several ways to access memory. You specify the PROM memory access method by using modifiers to memory requests. The dis command can accept three different modifiers, 'c', 'u' and 'v':

```
 dis - access memory as cached physical address
disc - same access as 'd' alone
disu - access memory as uncached physical address
disv - access a virtual address
```

If you simpy use "dis" instead of explicitly specifying the access type, you get the default, which is cached physical addressing.

ETLB (Command = e)

This command lets you view and change the contents of the ETLB on the CPU board. This command is slightly different for each of the two Stardent 1500/3000 Processor boards, P2 and P3. The command syntax for the P2 Processor is:

```
e[tlb] s0|s1|r0|r1 [ <table_range> [ = <tlbhi> <tlblo> ]]
```

For the P2 processor board, there are 1024 ETLB entries in each of four partitions (4K total), called s0, s1, r0 and r1. Table A2-2 shows how the partitions names relate to the parts of the ETLB that each can control.

Table A2-2. ETLB Partition Names (P2)

Name	Function
s0	D-pipe
s1	Semaphore Addresses
r0	B-pipe
r1	A-pipe

The command syntax for the P3 Processor is:

```
e[tlb] s|r [ <table_range> [ = <tlbhi> <tlblo> ]]
```

For the P3 processor board, there are 16384 ETLB entries in each of two partitions (32K total), called s and r. Table A2-3 shows how the partitions names relate to the parts of the ETLB that each can control.

Table A2-3. ETLB Partition Names (P3)

Name	Function
s	D-pipe Semaphore Addresses
r	B-pipe A-pipe

This command works identically to the TLB command, but affects a different set of registers and a partition must be specified.

For reading an ETLB entry, the <table-range> parameter is specified as

```
    <start-location> <number-of-entries-to-display>
or
    <start>:<last>
```

For writing an ETLB entry, the <table-range> parameter is interpreted as:

```
    <location-to-write>
```

Example. This example forms an ETLB entry for a semaphore at virtual address 0x20000, but using physical address 0x10000.

```
e s1 0=00020000 00010700       (with P2 processor)

e s 0=00020000 00010700        (with P3 processor)
```

The **s1** selects the semaphore partition. **0=00020000 00010700** says fill location 0 of that paritition with a tlbhi entry of 0x20000 (the virtual address), and the tlblo entry of 0x10000 (the physical address). The value of 700 in the physical address entry sets the V, D, and G bits of the entry. V is the valid bit, saying this entry is valid. D is the dirty bit (also called an M - for modifiable bit) saying it is ok to write to this location. G is the global bit, saying match all PID entries. The high entry specifies PID 0, but the global bit is set, so it will match all PID entries anyway. See *Appendix A* for more information about TLB and ETLB address translation.

Fill Memory (Command = f)

To fill a block of memory with a specific value (as a byte, half-word, or word), use the **fi** command.

```
f[ill] [-b | -h | -w] [-v <value>] <range>
```

- If you do not specify byte, halfword or word by using the -b, -h or -w flags respectively, the system chooses -b as the default.
- If you do not specify -v <value>, the system chooses zero (0) as a default.
- Range is specified as:
  - An address alone (taken as a count of 1)
  - An address, a space or a comma, then a count. The system begins at the specified address and fills a total of <count> items of size byte, halfword or word.
  - An address, a colon, and another address. The system fills the memory between and including the two address endpoints you provide.

Note that there are several ways to access memory. You specify the PROM memory access method by using modifiers to memory requests. For example, the f command can accept three different modifiers, 'c', 'u' and 'v':

```
 f - access memory as cached physical address;
fc - same access as 'd' alone
fu - access memory as uncached physical address
fv - access a virtual address
```

If you simpy use "f" instead of explicitly specifying the access type, you get the default, which is cached physical addressing.

Flush Cache (Command = fl)

This command causes the CPU to mark the current contents of its instruction or data cache as invalid, or to switch the instruction and data caches (the -s option toggles the SWC bit in the MIPS status register).

```
fl[ush] [-d | -i] [-s]
```

If no option is specified, this command flushes BOTH instruction and data cache of the currently operating CPU. To flush only one cache, specify -d for Data Cache, or -i for Instruction Cache.

Go (Command = g)

NOTE
Breakpoints can only be set in RAM Resident code (breakpoint instructions replace program code with a breakpoint instruction).

The command Go begins execution at the current program counter address or at an address you specify. During a breakpoint, the values of all registers were saved in memory. The saved values are restored to the registers as your program restarts.

```
g[o] <address>
```

where <address> is an address expression to be used as the program counter value. If an address is specified, begin execution at that address. If no address is specified, start at the address currently in the saved-PC location.

Note that there are several ways to access memory. You specify the PROM memory access method by using modifiers to memory requests. The g command can accept three different modifiers, 'c', 'u' and 'v':

```
 g - access memory as cached physical address;
gc - same access as 'd' alone
gu - access memory as uncached physical address
gv - access a virtual address
```

If you simpy use "g" instead of explicitly specifying the access type, you get the default, which is cached physical addressing.

Go Till (Command = got)

This command lets you begin execution at the current PC, and continue execution until either the specified address or a permanent breakpoint address is reached.

```
got[o] <address>
```

Note that there are several ways to access memory. You specify the PROM memory access method by using modifiers to memory requests. The got command can accept three different modifiers, 'c', 'u' and 'v':

```
 got - access memory as cached physical address;
gotc - same access as 'd' alone
gotu - access memory as uncached physical address
gotv - access a virtual address
```

If you simpy use "got" instead of explicitly specifying the access type, you get the default, which is cached physical addressing.

Help (Command = h)

If you cannot remember the syntax for a command, type

```
h[elp] [<command_name>]
```

or type

```
?
```

and press RETURN. The command name is optional. If you enter help without a command name, a complete summary of all available commands is printed for you to examine.

Initialize (Command = i)

i allows you to reinitialize all PROM-related variables from scratch. It reinitializes the CPU, including the TLBs, PROM global area, and graphics console with no reset.

```
i[nit] [-g]
```

the -g option initializes the graphics environment *only* (screen foreground and background settings, cursor foreground and background settings, and font settings).

Loading A Program (Command = l)

This command allows you to load a program, through a serial port on the CPU board, into memory for possible execution.

```
l[oad] [<character_device>]
```

The default for the load, if no character device is specified, is serial port number 0 on the CPU board (port 0 is where the debug terminal is connected). The designation "<character-device>" is actually a file name. The acceptable file name in this case is "tty". However, since there are two DCP ports, you can specify either tty(0) or tty(1). If you specify only "tty", you get the default filename, which is tty(0).

The load command expects a file to be sent to it via a serial packet protocol.

If <character_device> can't be loaded, something like the following error message is displayed:

`Can't load` *filename*.

Modify Memory (Command = m)

This command lets you display and/or modify system memory.

```
m[odify] [ -b | -h | -w ] [<address> [= <value>] ]
```

This command allows you to view and, if you wish, change the contents of sequential memory locations viewed and changed as bytes, halfwords or words, depending on which flag you specify. Byte size is the default if no flag is specified.

When you enter the command, the display shows you the address, and the contents of the location, such as:

m -b 1234 <CR>

00001234: 55 _

The underscore in the example above shows where the cursor is placed when the modify command is active.

- To view the next sequential memory location without changing it, press RETURN. The contents of the next sequential location is displayed and is ready to be changed if you wish.
- To change the value, type the new value (or specify an address as 0d..., 0b..., 0o..., 0x...., or as hexadecimal digits, followed by an equals sign, followed by the new value, then press RETURN. This establishes the new value for that location. Then the next sequential location is displayed for possible change.
- To exit this mode, type a period (.). The last location displayed is not modified.

As with **display, modify** remembers the last location changed. If you type a single "m", modify starts where you left off.

The alternate form of the command is

```
<address>=<value>
```

This alternate form allows you to directly enter a new value for without having to display the previous value in that location.

Note that there are several ways to access memory. You specify the PROM memory access method by using modifiers to memory requests. For example, the m (modify) command can accept three different modifiers, 'c', 'u' and 'v':

```
 m - access memory as cached physical address;
mc - same access as 'm' alone
mu - access memory as uncached physical address
mv - access a virtual address
```

If you simpy use "m" instead of explicitly specifying the access type, you get the default, which is cached physical addressing.

NVRAM (Command = n)

The NVRAM contains a set of strings that are analagous to the environment variables under UNIX. Each NVRAM entry takes the form:

```
SOMEITEM=SOMEVALUE
```

The value can be numeric or alphabetical, depending on the application with which it is used. You can refer to NVRAM variables in creating address or data expressions (see Appendix A). To set an NVRAM value, here is the syntax:

```
n[vram] [ <variable_name>[=<string>]]
```

NOTE
NVRAM parameter strings are not allowed to contain blank spaces.

- To view all NVRAM variables, type **n <CR>**.
- To view only one NVRAM variable value, type **n <variable_name> <CR>**.
- To change or add an NVRAM variable, type **n <variable_name>=<your_string> <CR>**.
- To entirely delete an NVRAM variable, type **n <variable_name>=<CR>**.

The following is a list of the standard NVRAM variables and their default values:

```
baud=9600
bootmode=a
path=scsi(1,5,0)
netaddr=(your-own-net-address)
etheraddr=(your-own-ethernet-address)
hostname=(your-own-host-name)
bootdev=scsia0
bootfile=scsi(,,8)sash
monitor=std
```

The last item in the list is *monitor=std*. The alternative to this default is *monitor=stereo*. If you have installed a stereo monitor on a system, and the system boots with the normal NVRAM default (*monitor=std*), no display will be seen. You will have to press the console keys ALT-CTRL-SYS, all at the same time. This toggles between setting the he screen driver to treat the display as stereo or standard, enables the display, and automatically modifies the NVRAM variable to the alternate state. If a stereo monitor is installed, this NVRAM variable should be set to *stereo*.

The *boot* command description provides more details about some of the NVRAM variables shown above. There are additional NVRAM variables that can be set to control screen colors and other attributes. (The default values are set in the file *kgif.c*)

Display/Set Process ID (Command = p)

This command allows you to display or set the process ID (as contained in the TLB or in the ETLB) of the currently executing process. Note that this a DECIMAL number, not hex.

```
p[id] [<decimal_value>]
```

If you simply type **p <CR>** the CPU will display the current values of the tlb process id (tlbpid) and of the etlb process id (etlbpid). If you type **p <value> <CR>**, then you will set both the tlbpid and etlbpid to a new PID value.

```
p[id] [ -f | -s ]
```

If you use the **-f** option, the contents of the *fpupidhash* register will be displayed. The **-s** option displays the contents of the *sempidhash* register.

Reset (Command = res)

This command allows you to issue a system reset. It has the same effect as writing directly to the physical address 0x1E00,0114 with a value of 0. This is the system reset register on the I/O card and forces a sys_reset.

Show/Change Register Contents (Command = r)

This command lets you display or modify the contents of the CPU registers.

```
r[egister] [<register_name_list> [= <value>]]
```

- As the command syntax indicates, you can enter the **r** command alone and it displays the contents of all CPU registers. If the CPU board is a P3 version, the **r** command also displays the contents of the MIPS floating point chip registers.
- Or you can enter **r <register_name>** (you specify the name from Appendix A) and only that register's contents is displayed; or enter **r<list>** and those register's contents are displayed.
- Or you can enter the command, name, an equals sign and a value to change that register's contents to this new value. In other commands, $<regname> can be used as a value (that is, the value in the designated register). For example,

```
goto  $ra
```

which returns to the function that called the current function.

In addition to the simple numbering of registers from 0 to 31, programmers often refer to the machine registers by certain mnemonic names. Table A2-4 and Table A2-5 show the mapping between those names and the actual machine register to which they refer.

Table A2-4. Machine Register Names

Name	Register	Name	Register	Name	Register	Name	Register
zero	$0	t0	$8	s0	$16	t8	$24
AT	$1	t1	$9	s1	$17	t9	$25
V0	$2	t2	$10	s2	$18	K0	$26
V1	$3	t3	$11	s3	$19	K1	$27
a0	$4	t4	$12	s4	$20	gp	$28
a1	$5	t5	$13	s5	$21	sp	$29
a2	$6	t6	$14	s6	$22	fp	$30
a3	$7	t7	$15	s7	$23	ra	$31

Table A2-5. M3010 MIPS Chip Register Names

Name	Register	Name	Register	Name	Register
fpc_irr	$f0	fpc_eir	$f30	fpc_csr	$f31
fpgr0	$f0	fpgr11	$f11	fpgr22	$f22
fpgr1	$f1	fpgr12	$f12	fpgr23	$f23
fpgr2	$f2	fpgr13	$f13	fpgr24	$f24
fpgr3	$f3	fpgr14	$f14	fpgr25	$f25
fpgr4	$f4	fpgr15	$f15	fpgr26	$f26
fpgr5	$f5	fpgr16	$f16	fpgr27	$f27
fpgr6	$f6	fpgr17	$f17	fpgr28	$f28
fpgr7	$f7	fpgr18	$f18	fpgr29	$f29
fpgr8	$f8	fpgr19	$f19	fpgr30	$f30
fpgr9	$f9	fpgr20	$f20	fpgr31	$f31
fpgr10	$f10	fpgr21	$f21		

In addition to those names referenced in the table, there are also the names:

```
pc       - the program counter
mdhi     - Hi half of the TLB entry
mdlo     - Low half of the TLB entry
inx      - pointer into the TLB array
badvaddr - read only bad virtual address
cause    - the MIPS cause register
sr       - the MIPS status register
ctxt     - pointer to the kernel virtual page entry table
semaddr  - semaphore address
```

Single Step (Command = S or s)

This command lets you execute 1 or more program steps. There are two versions of single stepping. In one version (specified as "s"), each instruction that is executed counts as one step. In the second version, (specified as "S"), the PROM treats each subroutine call as though it were a single instruction.

```
s[ingle] [<instruction_count>]
S[ingle] [<instruction_count>]
```

No instruction count implies 1 instruction.

TLB (Command = t)

This command lets you view and change the contents of the TLB on the CPU. The command syntax is:

```
t[lb] [<table_range> [ = <tlbhi> <tlblo> ]]
```

The TLB has 64 entries.

- Table ranges are specified as any other ranges except that table_ranges are assumed to be decimal numbers. See the fill command for a typical example of range specifications. The address specifications for TLB is from TLB address 0 to TLB address 63.
- If you specify a table range with no = sign nor hi-low entries, then the contents of that range of entries is displayed.
- If you specify a table range, then an equals sign, you can then specify the value for the high part of the TLB entry and for the low part of that TLB entry. But the range values actually have no effect when modifications are specified. Only the FIRST address of a range is used to specify the TLB address to modify.

See *Memory Address Mapping*, above, for the meaning of TLB addresses (high and low).

Examples:. If you specify:

```
t <CR>
```

It displays all 64 entries, starting at location 0. If you specify:

```
t 15 10 <CR>
```

It displays 10 entries, starting at entry number 15. If you specify

```
t 0=00001000 00001F00
```

This would form a virtual to physical address translation for virtual address 0x1000 and tlbpid 0 to access physical address 0x1000 as an uncached address, setting the Valid bit, the Modifiable bit and the Global bit in the TLB entry. See *Appendix A* for more information.

Version (Command = v)

This command displays the version number of this boot PROM.

```
v[ersion]
```

Switch CPUs (Command = x)

Use this command to switch which of 4 CPU boards is actually running the PROM monitor. Each CPU has registers, tlb and etlb that are private to that CPU. By switching CPUs, you can access these local entities.

```
x <cpuid>
```

NOTE
Even though a different CPU is now responding to your commands, your diagnostic terminal still remains connnected to the same CPU (the master, startup CPU).

where cpuid is a value of 0, 1, 2 or 3, depending on which CPU boards you actually have installed. The monitor prompt changes to reflect which CPU is responding to the commands. CPU ID is related to the slot number in which that particular CPU board is installed. If you transpose the binary value of the last two bits of the slot number, you will obtain the ID value for the CPU as shown in Table A2-6.

The system **will not run** if two CPU cards are installed in slots that yield the same CPU ID.

Table A2-6. CPU ID vs Slot Number

Slot	CPU ID
2	1
3	3
4	0
5	2
6	1
7	3

The prompt is:

```
dprom>        if the debug PROM is in control
prom X>       if the BOOT PROM is in control
sash X>       if the StandAlone Shell is in control
```

X is a number 0, 1, 2 or 3, representing the ID of the CPU that is in control. If you determine that the CPU that you are running as the master CPU is not functioning properly, you must manually disable the CPU board, and reset or power down and up. The system automatically assigns the lowest numbered (not disabled) cpuid as the master when the power up or reset occurs and the system boots with that master. Move your diagnostic terminal connection to that new master and reboot the system.

Switch to Debug (Command = z)

NOTE
This command will be removed in the near future when the debug and boot PROM functions have been merged.

The SASH and PROM levels of the system depend on the presence of operational system memory. The debug PROM, on the other hand, uses no RAM, but does use the system instruction and data caches as a form of memory. If it appears that memory is not functional, you can ask a higher level to switch to use only the debug PROM to continue the debugging of the system.

```
z    (switches to bringup PROM on current CPU)
```

If you are in the debug PROM and want to go back to the PROM level, use "z" again. It switches you back to the prior level.

DEBUG PROM SECTION

CHAPTER THREE

This chapter describes the debug PROM. **This is not planned to be present in a final customer ship unit.** You will notice a great deal of overlap of functionality between this PROM and the functionality of the actual boot PROM that is described in the previous chapter.

In the current version of the system, the debug PROM contains the basic go/no-go diagnostics, as well as a primitive debug monitor whose actions and access syntax are described here.

You will notice in Appendix B - LED Error Codes that there are two different patterns that are applied to the LED's, depending on which of the two PROMs is running its tests. When the diagnostics are merged into a single PROM, there will also be only one LED error indicator set, and the boot PROM will run all diagnostics.

Entering the Command Line Interpreter

To use this PROM monitor, there must be a 9600 baud terminal connected to the diagnostic port of the boot processor. If the initial diagnostics pass, the simple Command Line Interpreter (CLI) described below becomes active.

CLI Capabilities

This section details the capabilities of the Command Line Interface (CLI) for the various levels of debug PROM. The command line interface (CLI) prompt (dprom>) appears.

Introduction To The CLI

This chapter describes the capabilities of the command line interface for the debug PROMs. Where a particular feature is only implemented on a particular level of PROM (and higher levels), it is indicated in the description.

Documentation Conventions:

Form	Explanation
<address>	is a hexadecimal value, from 1 to 8 digits (0-9,A-F) that represents an address in the system. Address Values MUST be specified on 4 byte boundaries to be valid (last digit MUST be a 0, 4, 8 or C). NOTE: THE PROM DOES NOT CHECK FOR A VALID ADDRESS. Non-aligned accesses will cause an exception!!
<data>	is a hexadecimal value, from 1 to 8 digits.
(CR)	indicates pressing the RETURN or ENTER key on the external diagnostic terminal.

Entering Data For The Commands

If you enter more than 8 digits, only the last 8 of those entered will be used. If less than 8 hex digits are entered, the value of the address or data will be taken as though you entered leading zeros, making the total entry effectively 8 digits anyway.

Error Correction

If you make an error in the entry of data, you can use the BACKSPACE key to erase the previously entered digits or commands.

> For example, if you have entered an address and an equals sign, then you realize that you do not really want to write into that location, you can backspace across the equals sign and thus cancel the memory-write. Pressing (CR) will simply display the memory location whose address you entered.
>
> Or, if you simply have entered an incorrect address or data, you can backspace across it and re-enter the correct address or data value.

If you have entered the cache-test or memory-test command characters, you can abort this test before it begins by simply hitting BACKSPACE instead of (CR).

Memory Address Mapping

To access various memory areas, you must pay attention to the mapping that the system imposes. To access certain kinds of memory, you must specify the address as <address-plus-offset> instead of just as <address>. The areas, and their associated offsets are:

Type Of Memory	Offset Value	Notes
Main RAM (mapped)	0	3
Cached Unmapped	0x80000000	2
Uncached Unmapped	0xA0000000	1

Notes

1. Uncached, Unmapped range of addresses is from A000 0000 to AFFF FFFF.

   This addressing means allows direct access to all system registers. For example, if a system register has an absolute address of:

```
             1EC0 0000
   you add:  A000 0000
             ---------
             BEC0 0000   is the address you
                         use to access it.
```

2. Cached Unmapped range of addresses is from 8000 0000 to 9FFF FFFF.

   This addressing range lets you get to the cache memory on the CPU board on a write-through basis (marks the entry as VALID). However, unless the main RAM is already set up, and the Bus Enable bit is on in the SBus Status register, although any entry that was written can be read back, if an entry is read and is NOT valid, a bus error will occur. It is not possible to handle cache misses without memory installed!

The size of the cache is 16K bytes, or expressed another way, 4K 32-bit words. No address decoding is performed in the region between 0x80000000 and 0x9FFFFFFC other than that of the 16K cache itself. Thus the cache access repeats multiple times within this region.

3. Main RAM (mapped) range of addresses is from 0000 0000 to 7FFF FFFF.

   Main RAM is ONLY accessed through the ETLB's or TLB's. Do not attempt to access main RAM unless you have already set up the ETLB tables. These tables translate a virtual address that you are producing into a physical address that will be put out onto the bus. See the section titled *What Is A Virtual Address* for information about the registers you will have to modify in order to get to real RAM.

CLI

The Command Line Interpreter treats both uppercase and lowercase commands alike. Additionally, there are certain commands, notably the memory commands, that can be executed without entering a command letter. All of the available CLI commands are shown below, with an indication of which level of PROM allows the command to be issued.

When multiple versions of a command are shown, all forms shown are equivalent. The primary source of multiple forms of a command is that the CLI is case-insensitive. That is, upper case and lower case is treated alike by the command interpreter.

Here is a quick summary that lists one form of each command. It is followed by a detailed explanation of each command.

Table A3-1. Summary Of CLI Commands

Command	Format	Notes
Read Memory	<address>(CR)	
Set Memory Display Mode	m-b<address>(CR) m-h<address>(CR) m-w<address>(CR)	Display as BYTES Display as HALFWORDS Display as WORDS
Write Memory	<address>=<data>(CR)	
Cold Start Init	i(CR)	
Warm Start Init	w(CR)	
Begin Endless Memory Test	t(CR)	Requires machine reset to exit
Download A File in S-record Format	L(CR)	File contains the starting address
Go to beginning address of downloaded file	g(CR)	

Read Memory

The commands that read memory are as follows:

```
<address>(CR)    (preferred format)

M<address>(CR)
m<address>(CR)
```

Displays, as an 8 hex digit value, the contents of <address>

```
M(CR)
(CR)
```

Displays the next WORD (32-bit) address sequentially following the previous address entered. Shows next address following a read or a write (since the most recent address entered is remembered and incremented by 4 following each write or each read-with-no-parameter).

Set Memory Display Mode

Using these commands, you can change the memory display mode to interpret the data read as bytes, halfwords or words. The default for the display mode is word (8 hexadecimal digits) mode.

```
M-b<address>(CR)
m-b<address>(CR)
M-B<address>(CR)
m-B<address>(CR)
```

Displays, as a 2 digit hex value (as a BYTE) the contents of <address>

```
M-h<address>(CR)
m-h<address>(CR)
M-H<address>(CR)
m-H<address>(CR)
```

Displays, as a 4 digit hex value (as a HALF-WORD) the contents of <address>

NOTE
A blank space may be used in place of a hyphen.

```
M-w<address>(CR)
m-w<address>(CR)
M-W<address>(CR)
m-W<address>(CR)
```

Displays, as a 4 digit hex value (as a WORD) the contents of <address>

WARNING
Although B is a legal hexadecimal digit, a B following the m or M in any form is interpreted as a request to change the display mode to byte.

If you wish to read or write memory at address Bxxx xxxx, then do NOT use this form of the command. Simply enter the address directly, without the M or m.

Write Memory

The commands that write to memory are as shown here.

NOTE
The memory write command, when used with the M or m, can also set the memory display mode as described above.

```
<address>=<data>(CR)     preferred format

M<address>=<data>(CR)
m<address>=<data>(CR)
```

Stores the value given by <data> into <address>.

Perform Cold Start

Use this command to perform a cold start initialization of the PROM.

```
I(CR)
i(CR)
```

Cold-start initializes everything, testing both memory and cache. It initializes the diagnostics communications port (DCP) at 9600 baud, sets up a stack, and then calls the debugger to get a CLI prompt.

Enter Memory Test Loop

Use this command to begin an endless cache memory test loop.

```
T(CR)
t(CR)
```

To exit, you must reset the machine.

Perform Warm Start

Use this command to perform a warm start initialization of the PROM.

```
W(CR)
w(CR)
```

The difference between cold and warm start is that cold start reinitializes everything, whereas warm start:

- Does not touch memory or cache
- Reinitializes the Stack
- Calls the debugger to get to the CLI prompt

Download

This command downloads S-Record file through the serial port. Additional details about downloading are shown later in this manual.

```
L(CR)
l(CR)
```

Note that the S-Record file itself determines where in memory the downloaded program will reside.

GO

This command allows you to jump to the starting point of a downloaded program.

```
G(CR)
g(CR)
```

Begin execution at the starting point specified within the downloaded S-Record file.

If the downloaded code exits via the assembly language instruction:

```
j   ra  #jump to the address specified as the
        #return address from this subroutine
```

then the monitor will again regain control through the Warm_Start entry point.

TLB MAPPING

APPENDIX AA

This appendix shows how the TLB and ETLB cause virtual addresses to be translated into physical addresses, and how the PROM treats memory access requests.

Memory Address Mapping

When you create a program on the Titan, you can create memory addresses that are a total of 32-bits wide. The processor, however, partitions its address space such that the per-user maximum virtual address is only 31 bits wide. Your 31-bits of legitimate virtual RAM space is, in turn, mapped into the real, physical RAM space by the use of the TLB (translation lookaside buffers) on the processor chip. How the 32nd bit (the MSBit of the address) is used is explained later in this chapter.

Within the TLB, there are a total of 64 table entries that the processor chip uses to keep track of which user virtual addresses are mapped to real physical addresses. Each one of these 64 table entries contains, among other things, 20-bits of virtual page number and a process identifier (PID). If the PID matches that of the currently running process, and if the upper 20 bits of the virtual address matches the 20 bits of virtual page number, then the TLB can complete the translation, providing (among other things) a physical page number (20 bits) which, along with the lower 12 bits of the virtual address, becomes the real address that the processor chip uses to access the data.

The contents of any single entry in the TLB is illustrated in Table AA-1.

The PID value shown above is the Process ID (6 bits). Each process can address up to 2 GB of virtual memory as mentioned above.

Table AA-1. TLB Contents (per entry)

Virtual Frame Number (20-bits)	PID (6-bits)	zeros (6-bits)	Physical Frame Number (20-bits)	Bits (N M V G)	zeros (12-bits)

The Virtual Frame number is taken as the leftmost 20 bits of the virtual address. The Physical Frame Number is the leftmost 20 bits of the physical address that this entry will translate to if there is a match on both the virtual address and the PID.

When a virtual address is presented, the TLB searches all 64 entries in its table simultaneously (a fully associative memory) and signals a hit or a miss. If there is a match on both the PID and virtual frame number, the translation to physical frame number takes place. The physical frame number becomes the upper 20 bits of the real address and the lower 12 bits of the virtual address becomes the lower 12 bits of the physical address, completing the translation.

The machine insists that there does not exist two valid entries with the same Virtual Page Number (VPN).

Also part of the TLB entries are the following four bits: N, M, V and G.

- N - if equal to 0, accesses to the physical memory should go through the cache. If an addressed item is not currently cached, cache it first, then read it from the cache. If N = 1, then the item is treated as non-cache-able.

- V - the "valid" bit. If a 1, this entry in the TLB is valid. If a zero, accessing this entry, although matching both the PID and Virtual Frame number, will cause an interrupt. This bit can be used to develop statistics about references to a particular location if desired.

- M - the "modify" bit. Answers the question "can this page be modified". This bit is off for all text pages (code) and can be on for data pages in a user's process. A write with M = 0 will cause an interrupt.

- Global Bit - When this bit is on, the address translation hardware ignores the PID field when matching.

Mapping Using The TLB

As noted above, the TLB contains 20 bits for physical page number and uses the lowest 12 bits of the virtual address as part of the physical address if the TLB has the entry. This means that the TLB maps the virtual address into pages of 4k bytes each.

With 64 entries, each representing a page of 4k each, it means that at any one instant, the TLB is able to map 256k bytes of physical address space.

If the virtual address mapping is not in the TLB when translation is requested, a TLB miss occurs. The operating system then determines if the requested virtual page (for this PID) is elsewhere in physical memory (in which case the OS can simply write a corresponding TLB entry and retry the translation) or if the requested virtual address is not currently resident and must be paged in from mass storage before the TLB can be updated.

Of the 64 TLB entries, 8 are maintained directly by the OS for its own use. The remaining 56 entries are used on a random basis. In other words, when an entry is to be placed into the TLB, it is written into a random slot position, replacing whatever entry is currently there.

The 32nd Bit

Your program can generate addresses as much as 32 bits wide, but the processor only allows a 31-bit virtual address. The 32nd bit (bit 31) is used along with bits 30 and 29 to perform special mapping of the memory space as shown in Table AA-2.

Table AA-2. Use of the 32nd Bit

Bits 31	30	29	Meaning
0	x	x	Treat the address as a user virtual address
1	0	0	This is to be treated as a cached physical address
1	0	1	This is to be treated as an uncached physical address
1	1	x	This is to be treated as a kernel virtual address

- A zero bit in bit 31 says this is a virtual address; translate it through the TLB.

- By cached physical address, we mean that this address range is a window into the first 512 Mbytes of physical address space. Binary 100 in these first three bits says treat the remaining bits as a physical address, looking in the cache first. This encompasses the lowest 256 Mbytes of RAM addressing, as well as the entire 256 Mbytes of I/O space (which includes the graphics boards, the I/O and the CPU registers, as well as other miscellaneous registers). When you access these items, the access is performed through the cache.

  Note: Normally a programmer using this low level debugger will access memory as cached or uncached physical addresses. If you should find it necessary to access memory as virtual addresses, note that you will have to set up the TLB first. Information about the TLB is available in Appendix A.

- By uncached physical address, we mean that this address range is a window into the first 512 Mbytes of physical address space, but is accessed directly, that is, without going through the instruction or data cache. That is, binary 101 in the most significant 3 bits of the address says treat the remaining bits as a physical address, but don't look in the cache.

- A kernel virtual address is treated just like a user virtual address in that it is TLB mapped somewhere into the real physical memory.

PROM Monitor Access To Upper Bank Of Memory

To specify direct access to memory, you can use the cached or uncached windows (bit 31 equals 1, bit 30 equals 0, bit 29 = any) into physical memory. The description of those windows, though, indicates that using this window you can only access the lowest 512 Megabytes of real addresses. The system memory is organized as shown in Table AA-3, with 256 Mbytes of memory at the lowest address, a 256 Mbyte space assigned to I/O and system registers, and finally a 256 Mbyte space for the upper bank of real memory.

Table AA-3. Real Memory Mapping

Address Space	Assignment
0-256 Mbyte	RAM
256-512 Mbyte	I/O, sys regs.
512-768 Mbyte	RAM

So it would appear that the PROM Monitor could not provide access to the upper bank of memory. Instead of using the direct mapping as specified above, when the PROM Monitor notices any Monitor references to virtual addresses within the range of 512 M to 768 M, it saves TLB entry 0 and uses it to access the real addresses in this range. After the read or write access has completed, the TLB entry is restored to its original value.

CIRCUIT BOARD LEDs

APPENDIX AB

This appendix lists status indications that the CPU diagnostic PROM outputs during its power-up self-test sequence. It also gives the meanings of the LEDs on each of the System Module circuit boards.

CPU Board LED Indicators

Figure AB-1 shows the physical layout of the LEDs on the CPU board.

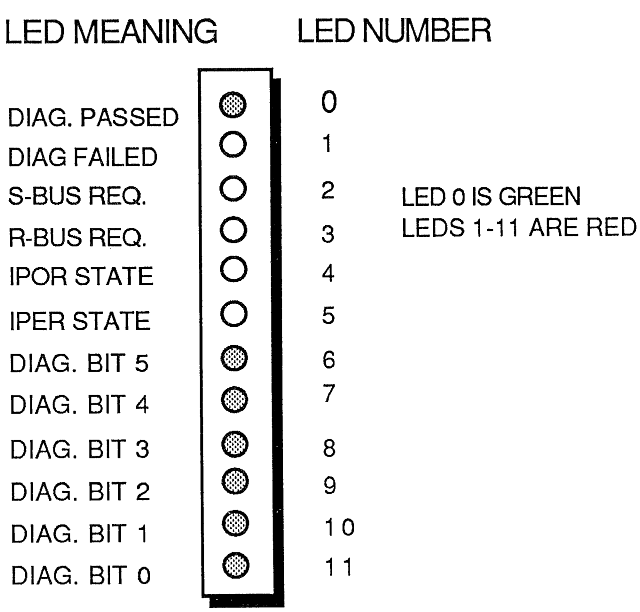


Figure AB-1. CPU Board LED Layout

The top two LEDs are the same on the I/O, memory, graphics, graphics expansion, and VME expansion boards. When the green LED is lit, it means the board passed the diagnostic self-tests; when the red LED is lit, the board failed. The bottom six LEDs are a hex code for steps in the system's initial self-test process.

Reading the LED Self-test Status/Error Indicators

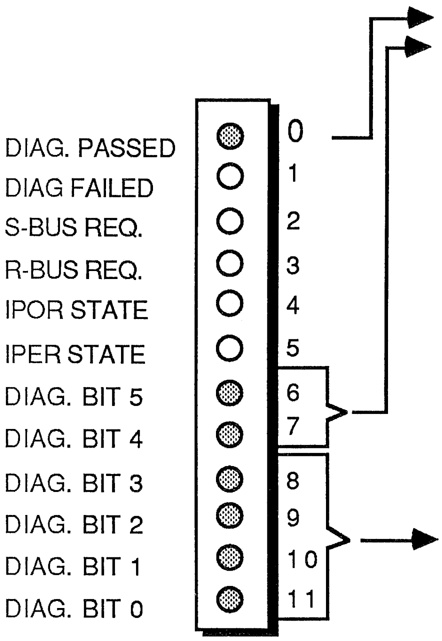
This section lists PROM boot self-test steps and their associated LED codes. The codes are in hex and can be interpreted as shown in Figure AB-2. The CPU board's four lower LEDs (listed as LEDs 8 through 11 in Figure B-2) represent the lower (rightmost) hex digit. The higher (leftmost) hex digit is represented by LEDs 6, 7 and 0 in Figure B-2. The green pass light (LED 0) is the 8's bit in the binary representation of the higher hex digit.[1]) For instance, if LEDs 6, 9 and 10 are lit the corresponding hex number is 26 and the self-test step is

```
CPU_LED_IC_TINV 26 Test I cache for invalidation
```

If LEDs 0, 6 and 11 are lit, the corresponding hex number is a1 and the self-test step is

```
CPU_LED_IOSTO   a1 Test IO board's SCSI stackerout
```

[1] Note that there is no ambiguity in the use of the green pass light (LED 0) as a pass light and as code for a self-test step. The first set of self-test steps listed in the table tests the CPU board itself. While the tests are running the green pass light is off, consistent with the fact that all the self-test codes in this section have zero in the leftmost digit's 8's position. Once the CPU tests have passed the CPU's green pass light is turned on, and consistent with that all the remaining steps in the self-test have a 1 coded in the 8's position of the left-most digit.



LEFT MOST HEX DIGIT

LED 0	N/A -----	LED 6	LED 7	DIGIT
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**2**
0	0	1	1	**3**
1	0	1	0	**A**
1	0	1	1	**B**

RIGHT MOST HEX DIGIT

LED 8	LED 9	LED 10	LED 11	DIGIT
0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**2**
0	0	1	1	**3**
0	1	0	0	**4**
0	1	0	1	**5**
0	1	1	0	**6**
0	1	1	1	**7**
1	0	0	0	**8**
1	0	0	1	**9**
1	0	1	0	**A**
1	0	1	1	**B**
1	1	0	0	**C**
1	1	0	1	**D**
1	1	1	0	**E**
1	1	1	1	**F**

Figure AB-2. Reading CPU Board LED Codes

Here is the list of CPU board LED codes. The LED codes are lit as that step in the CPU boot sequence is performed. If an error occurs during the boot sequence, the LED register will freeze showing the value appropriate to the test that has just begun to run.

CPU_LED_CAUSE	01 zeroing coprocessor cause register
CPU_LED_IC_AD	02 test I cache as mem with addrs as data
CPU_LED_IC_PAT	03 test I cache as mem with mult. pats
CPU_LED_DC_AD	04 test D cache as mem with addrs as data
CPU_LED_DC_PAT	05 test D cache as mem with mult. pats
CPU_LED_C_FILL	06 Fill both caches (I & D)
CPU_LED_FLUSH1	07 flushing i cache
CPU_LED_FLUSH2	08 flushing d cache
CPU_LED_ENABLE	09 enabling bus and bus watcher
CPU_LED_DBCLR	0A clearing doorbell answer register
CPU_LED_CPUID	0B determining cpu id
CPU_LED_BOOT	0C spinning to determine boot processor
CPU_LED_DIAG	0D executing cpu diagnostic
CPU_LED_RING	0E ringing slave processor's doorbells
CPU_LED_BOOTCPU	0F we're the boot processor
CPU_LED_IC_TAGS	10 Test I cache tag bits
CPU_LED_DC_TAGS	11 Test D cache tag bits
CPU_LED_IC_TOFF	12 Test I cache tag offset of 16k
CPU_LED_DC_TOFF	13 Test D cache tag offset of 16k
CPU_LED_IT_CHK	14 Check IC tag offset for misses, par.
CPU_LED_DT_CHK	15 Check DC tag offset for misses, par.
CPU_LED_PXSUM	16 Indicates bootprom checksum section
CPU_LED_a8	17 <unassigned>
CPU_LED_a7	18 <unassigned>
CPU_LED_a6	19 <unassigned>
CPU_LED_a5	1A <unassigned>
CPU_LED_a4	1B <unassigned>
CPU_LED_a3	1C <unassigned>
CPU_LED_a2	1D <unassigned>
CPU_LED_a1	1E <unassigned>
CPU_LED_a0	1F <unassigned>
CPU_LED_TETLB	20 Test the etlbs with address as data
CPU_LED_TETLB1	20 Test the etlbs with multiple patterns
CPU_LED_IC_NAD	22 Test I cache with not addrs as data
CPU_LED_DC_NAD	23 Test D cache with not addrs as data

CPU_LED_IC_INV	24 Invalidate I cache
CPU_LED_DC_INV	25 Invalidate D cache
CPU_LED_IC_TINV	26 Test I cache for invalidation
CPU_LED_DC_TINV	27 Test D cache for invalidation
CPU_LED_IC_PZ	28 Test I cache parity bits on data bits
CPU_LED_DC_PZ	29 Test D cache parity bits on data bits
CPU_LED_IC_TPZ	2A Test I cache parity on tag bits
CPU_LED_DC_TPZ	2B Test D cache parity on tag bits
CPU_LED_I_NFLSH	2C Test I cache not purged doing Dcache
CPU_LED_D_NFLSH	2D Test D cache not purged doing Icache
CPU_LED_IC_CHK	2E Test I cache for misses, parity, etc
CPU_LED_DC_CHK	2F Test D cache for misses, parity, etc
CPU_LED_PORTA_F	30 Serial port A in the SCC8530 failed
CPU_LED_PORTB_F	31 Serial port B in the SCC8530 failed
CPU_LED_ILPAB	32 Internal loopback test on ports A&B
CPU_LED_PORTAB	33 Init ports A&B to 9600 baud
CPU_LED_ETLB_SZ	34 Size the ETLBs small = 2k, big = 8k
CPU_LED_BANKERR	35 wrong memory bank (0=8, 1=16, 3=32)
CPU_LED_EVERR	36 fnd more than 2 mem brds in odd slots
CPU_LED_ODDERR	37 fnd more than 2 mem brds in even slots
CPU_LED_MEMMULT	38 found more than 4 memory cards
CPU_LED_GRPMULT	39 multiple graphics boards in backplane
CPU_LED_GRPNXA	3a expected an NXA looking for graphics
CPU_LED_IOMULT	3b found multiple I/O cards in backplane
CPU_LED_IONXA	3c expected an NXA looking for I/O
CPU_LED_NOMEM	3d didn't find any memory cards
CPU_LED_MEMNXA	3e expected an NXA looking for memory
CPU_LED_SLAVE	3f waiting for boot processor

At this point, the Green LED has been turned on (boot tests "passed", but the testing continues)

CPU_LED_MEMCONF	b0 finding memory boards
CPU_LED_IOCONF	b1 finding I/O board(s)
CPU_LED_GRPCONF	b2 finding graphics board(s)
CPU_LED_MEMDIAG	b3 starting memory config/diagnostic
CPU_LED_MEMC1	b4 reading banks and chip size
CPU_LED_MEMPWR	b5 power fail reset config to mem card
CPU_LED_MEMITLV	b6 configuring interleaves
CPU_LED_MEMEN	b7 enabling ECC
CPU_LED_MEMZ	b8 writing zeroes to all banks
CPU_LED_MEMR	b9 sys reset memory config

CPU_LED_16WAY	ba set memory boards to 16way interleave
CPU_LED_ADMEM	bb test memory with address data
CPU_LED_PATMEM	bc test memory with multiple patterns
CPU_LED_42	bd <unassigned>
CPU_LED_41	be <unassigned>
CPU_LED_40	bf <unassigned>

CPU_LED_IOAMAP	a0 Test IO board's mapper - addrs as data
CPU_LED_IOSTO	a1 Test IO board's SCSI stackerout
CPU_LED_IOSTI	a2 Test IO board's SCSI stackerin
CPU_LED_IODMAC	a3 Test IO board's DMA counter
CPU_LED_IODMAR	a4 Test IO baord's DMA cntrl register
CPU_LED_IOSAD	a5 Test IO board's DAM enable
CPU_LED_IOSCSI	a6 Test IO board's SCSI reset
CPU_LED_IOKB	a7 Do the IO board's keyboard selftest
CPU_LED_57	a8 <unassigned>
CPU_LED_56	a9 <unassigned>
CPU_LED_GRREG	aa Test the GR board's registers (static)
CPU_LED_GRMEM	ab Test the GR board's srams
CPU_LED_GRBT	ac Test the Gr board's brooktrees
CPU_LED_GRINIT	ad Test the GR board's init sequence
CPU_LED_GRR_RW	ae GR board's register r/w test
CPU_LED_50	af <unassigned>

If the any of the above tests fail, the "FAULT" LED is turned on, and the binary value indicated in the table is shown as the error condition.

Circuit Board LED Definitions

The tables below give LED definitions for each circuit board in the Stardent 1500/3000 System Module card cage.

Table AB-1. CPU Board LEDs

LED	Color	Description
0	Green	Diagnostic passed
1	Red	Diagnostic failed
2	Red	S-Bus Request
3	Red	R-Bus Request
4	Red	IPOR state
5	Red	IPER state
6	Red	diagnostic bit 5
7	Red	diagnostic bit 4
8	Red	diagnostic bit 3
9	Red	diagnostic bit 2
10	Red	diagnostic bit 1
11	Red	diagnostic bit 0

Table AB-2. Memory Board LEDs

LED	Color	Description
0	Green	Diagnostic passed
1	Red	Diagnostic failed
2	Red	S-Bus Request
3	Red	R-Bus Request
4	Red	16 way interleave
5	Red	Fatal error
6	Red	ECC
7	Red	Board enabled

Table AB-3. I/O Board LEDs

LED	Color	Description
0	Green	Diagnostic passed
1	Red	Diagnostic failed
2	Red	S-Bus Request
3	Red	I/O busy
4	Red	Memory request
5	Red	Mouse activity
6	Red	LAN activity
7	Red	LAN heart beat
8	Red	SCSI A busy
9	Red	SCSI B busy
10	Red	Interrupt
11	Red	Error lagged

Table AB-4. Graphics Board LEDs

LED	Color	Description
0	Green	Diagnostic passed
1	Red	Diagnostic failed
2	Red	S-Bus Request
3	Red	GR busy
4	Red	Bus error
5	Red	DMA busy
6	Red	RDMA busy
7	Red	WDMA busy

Table AB-5. Graphics Expansion Board LEDs

LED	Color	Description
0	Green	Diagnostic passed
1	Red	Diagnostic failed

Table AB-6. VME Expansion Board LEDs

LED	Color	Description
0	Green	Diagnostic passed
1	Red	Diagnostic failed
2	Red	Busy
3	Red	Strobe
4	Red	Master request
5	Red	Master
6	Red	Slave
7	Red	Interrupt