



AlphaServer DS10

Console Reference

Order Number: EK-DS10S-CR. B01

This manual describes the general operations of the AlphaServer DS10 system. It presents the SRM console (the command-line interface for Tru64 UNIX and OpenVMS operating systems), AlphaBIOS (the graphics interface for Windows NT and Linux), and remote console management.

Notice

The information in this publication is subject to change without notice.

COMPAQ COMPUTER CORPORATION SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL. THIS INFORMATION IS PROVIDED “AS IS” AND COMPAQ COMPUTER CORPORATION DISCLAIMS ANY WARRANTIES, EXPRESS, IMPLIED OR STATUTORY AND EXPRESSLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSE, GOOD TITLE AND AGAINST INFRINGEMENT.

This publication contains information protected by copyright. No part of this publication may be photocopied or reproduced in any form without prior written consent from Compaq Computer Corporation.

© 1999 Compaq Computer Corporation.

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

The software described in this guide is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

Compaq, Deskpro, Fastart, Compaq Insight Manager, Systempro, Systempro/LT, ProLiant, ROMPaq, QVision, SmartStart, NetFlex, QuickFind, PaqFax, ProSignia, registered United States Patent and Trademark Office.

Netelligent, Systempro/XL, SoftPaq, QuickBlank, QuickLock are trademarks and/or service marks of Compaq Computer Corporation.

Linux is a registered trademark of Linus Torvalds in several countries. Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Other product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Warning!

This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

Achtung!

Dieses ist ein Gerät der Funkstörgrenzwertklasse A. In Wohnbereichen können bei Betrieb dieses Gerätes Rundfunkstörungen auftreten, in welchen Fällen der Benutzer für entsprechende Gegenmaßnahmen verantwortlich ist.

Attention!

Ceci est un produit de Classe A. Dans un environnement domestique, ce produit risque de créer des interférences radioélectriques, il appartiendra alors à l'utilisateur de prendre les mesures spécifiques appropriées.

警告使用者:

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策。

に基づくクラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference.

Operation of this equipment in a residential area may cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Any modifications to this device - unless expressly approved by the manufacturer - can void the user's authority to operate this equipment under part 15 of the FCC rules.

Contents

Preface	xiii
Chapter 1 Operations	
1.1 Powering Up the System	1-2
1.2 Power-Up Display	1-4
1.3 Booting Tru64 UNIX	1-6
1.3.1 Booting from a Local Disk	1-6
1.3.2 Booting from a Remote Disk	1-8
1.4 Installing Tru64 UNIX	1-10
1.5 Booting OpenVMS	1-12
1.5.1 Booting OpenVMS from a Local Disk	1-12
1.5.2 Booting OpenVMS from a Disk on a Cluster	1-14
1.5.3 Booting OpenVMS from a Remote Disk	1-16
1.6 Installing OpenVMS	1-18
1.7 Booting Windows NT	1-20
1.8 Installing Windows NT	1-22
1.9 Switching Between Operating Systems	1-24
1.9.1 Switching to Windows NT	1-24
1.9.2 Switching to Tru64 UNIX or OpenVMS	1-25
1.10 Updating Firmware	1-26
1.10.1 Updating Firmware from the CD-ROM	1-28
1.10.2 Updating Firmware from Floppy Disks	1-32
1.10.3 Performing the Update from Floppy Disks	1-34
1.10.4 Updating Firmware from a Network Device	1-38
1.10.5 LFU Commands	1-41
1.11 Hard Disk Partitioning for Windows NT	1-44
1.11.1 Hard Disk Error Conditions	1-44
1.11.2 System Partitions	1-46
1.11.3 How AlphaBIOS Works with System Partitions	1-47
1.12 Using the Halt Button	1-48
1.13 Halt Assertion	1-51

Chapter 2 SRM Console

2.1	Invoking the SRM Console	2-2
2.2	Commands	2-3
2.2.1	Command Summary	2-3
2.2.2	Commands: Syntax	2-5
2.2.3	Commands: Special Keystrokes and Characters	2-6
2.3	Show Commands	2-8
2.3.1	Show Config	2-8
2.3.2	Show Device	2-10
2.3.3	Show Memory	2-12
2.3.4	Show PAL	2-12
2.3.5	Show Power	2-13
2.3.6	Show Version	2-14
2.4	Creating a Power-Up Script	2-14
2.5	Booting the Operating System	2-16
2.6	Configuring a PCI NVRAM Module	2-18
2.7	Testing the System	2-19
2.8	Set Commands	2-23
2.8.1	Set Password	2-23
2.8.2	Set Secure	2-24
2.9	Secure Mode	2-25
2.9.1	Login Command and Secure Mode	2-25
2.9.2	Clear Password	2-26
2.9.3	Resetting the Password	2-27
2.10	Stopping and Starting CPU	2-28
2.11	Updating Firmware	2-28
2.12	Forcing a System Crash Dump	2-30
2.13	Using Environment Variables	2-31
2.13.1	Set envvar	2-31
2.13.2	Show envvar	2-32
2.14	Depositing and Examining Data	2-33
2.15	Reading a File	2-34
2.16	Initializing the System	2-35
2.17	Finding Help	2-36
2.18	Switching from SRM to AlphaBIOS Console	2-37
2.19	Environment Variable Summary	2-38
2.19.1	auto_action	2-40
2.19.2	bootdef_dev	2-40
2.19.3	boot_osflags	2-41
2.19.4	com1_baud	2-43
2.19.5	com1_mode	2-43

2.19.6	console	2-44
2.19.7	ew*0_mode	2-45
2.19.8	ew*0_protocols	2-45
2.19.9	kbd_hardware_type	2-46
2.19.10	language	2-46
2.19.11	os_type	2-47
2.19.12	password	2-47
2.19.13	pci_parity	2-48
2.19.14	pk*0_fast	2-48
2.19.15	pk*0_host_id	2-49
2.19.16	pk*0_soft_term	2-49
2.19.17	tt_allow_login	2-50

Chapter 3 AlphaBIOS Console

3.1	Starting AlphaBIOS	3-2
3.2	Keyboard Conventions and Help	3-4
3.3	System Configuration	3-6
3.3.1	Displaying the System Configuration	3-6
3.3.2	System Board Configuration	3-8
3.3.3	Hard Disk Configuration	3-10
3.3.4	PCI Configuration	3-12
3.3.5	SCSI Configuration	3-15
3.3.6	Memory Configuration	3-17
3.3.7	Integrated Peripherals	3-18
3.4	Updating Firmware	3-19
3.5	Setting Up the Hard Disk	3-21
3.5.1	Creating and Deleting Partitions Manually	3-24
3.5.2	Formatting a FAT Partition	3-26
3.6	Performing Setup Tasks	3-28
3.7	Installing Operating System	3-32
3.8	Selecting the Version of Windows NT	3-34
3.9	Designating a Primary Operating System	3-36
3.9.1	Selecting a Primary Operating System	3-36
3.9.2	Primary Operating System and the Auto Start Option	3-38
3.10	Switching from AlphaBIOS to SRM Console	3-42
3.11	Running Utility Programs	3-44
3.11.1	Running Utilities from a Graphics Monitor	3-45
3.11.2	Running Utilities from a Serial Terminal	3-46

Chapter 4 Remote Console Manager

4.1	RMC Components	4-2
4.2	Terminal Setup	4-3
4.3	Operating Modes	4-4
4.4	Entering the RMC	4-6
4.5	SRM Environment Variables for COM1	4-6
4.6	Status Monitoring	4-7
4.7	Remote Power (On/Off)	4-9
4.8	Remote Halt (In/Out)	4-10
4.9	Configuring Remote Dial-In	4-12
4.10	Configuring Dial-Out Alert	4-14
4.11	Dialing In	4-17
4.12	Resetting the RMC to Factory Defaults	4-18
4.13	Troubleshooting Tips	4-19
4.14	RMC Commands	4-21
4.14.1	clear alert	4-23
4.14.2	clear port	4-23
4.14.3	disable alert	4-23
4.14.4	disable remote	4-24
4.14.5	enable alert	4-24
4.14.6	enable remote	4-25
4.14.7	halt (in/out)	4-26
4.14.8	hangup	4-27
4.14.9	help or ?	4-27
4.14.10	power off	4-27
4.14.11	power on	4-28
4.14.12	quit	4-28
4.14.13	reset	4-29
4.14.14	send alert	4-29
4.14.15	set alert	4-30
4.14.16	set com1_mode	4-31
4.14.17	set dial	4-32
4.14.18	set escape	4-33
4.14.19	set init	4-34
4.14.20	set logout	4-35
4.14.21	set password	4-35
4.14.22	set user	4-36
4.14.23	set wdt	4-36
4.14.24	status	4-36

Appendix A Setting Jumpers

A.1	Warnings and Cautions	A-2
A.2	Remove Power from the System	A-3
A.3	Open the System	A-4
A.4	Remove the Floppy Disk Enclosure	A-5
A.5	Set Jumpers	A-6
A.6	Restore Power	A-8

Examples

1-1	Power-Up Display	1-4
1-2	Booting Tru64 UNIX from a Local Disk	1-6
1-3	Booting Tru64 UNIX from a Remote Disk	1-8
1-4	Installing Tru64 UNIX.....	1-10
1-5	Booting OpenVMS from a Local Disk	1-12
1-6	Booting OpenVMS from a Disk on a Cluster	1-14
1-7	Booting OpenVMS from a Remote Disk.....	1-16
1-8	Installing OpenVMS	1-18
1-9	Starting LFU from the SRM Console.....	1-26
1-10	Booting LFU from the CD-ROM	1-27
1-11	Updating Firmware from the CD-ROM	1-28
1-12	Creating Update Diskettes on an OpenVMS System	1-33
1-13	Updating Firmware from the Floppy Disk	1-34
1-14	Selecting DS10FW to Update Firmware from Internal Floppy Disk	1-37
1-15	Updating Firmware from a Network Device.....	1-38
2-1	Show Config Command	2-8
2-2	Show Device Command.....	2-10
2-3	Show Memory Command	2-12
2-4	Show PAL Command	2-12
2-5	Show Power Command	2-13
2-6	Show Version Command	2-14
2-7	Editing the nvram Script	2-15
2-8	Clearing the nvram Script	2-15
2-9	Boot Command.....	2-17
2-10	Prcache Command	2-18
2-11	Test Command	2-19
2-12	Set Password Command	2-23
2-13	Set Secure Command	2-24
2-14	Secure Mode and Login Command	2-26
2-15	Clear Password Command.....	2-26

2-16	Lfu Command.....	2-28
2-17	Crash Command	2-30
2-18	Setting and Showing Environment Variables.....	2-32
2-19	Creating a User-Defined Environment Variable	2-32
2-20	Deposit Command.....	2-33
2-21	Examine Command.....	2-33
2-22	More Command.....	2-34
2-23	Initialize Command.....	2-35
2-24	Help Command	2-36
2-25	Switching to the AlphaBIOS Console	2-37
2-26	Changing Baud Rate	2-43
4-1	Dial-In Configuration.....	4-12
4-2	Dial-Out Alert Configuration.....	4-14

Figures

1-1	Check Power Setting.....	1-2
1-2	Location of Control Panel and On/Off Button.....	1-3
1-3	Physical Numbering for PCI Slots	1-4
1-4	AlphaBIOS Boot Screen.....	1-20
1-5	Installing Windows NT	1-22
1-6	Starting LFU from the AlphaBIOS Console	1-26
1-7	System Partition Not Defined.....	1-47
1-8	Halt/Reset Button	1-48
1-9	Halt/Reset Jumper	1-49
3-1	Boot Screen	3-2
3-2	AlphaBIOS Setup Screen.....	3-3
3-3	Typical First-Level Help Screen	3-4
3-4	Second-Level Help Screen.....	3-5
3-5	Display System Configuration Screen	3-6
3-6	System Board Configuration.....	3-8
3-7	Hard Disk Configuration	3-10
3-8	PCI Configuration	3-12
3-9	PCI Slots	3-13
3-10	Advanced PCI Information	3-14
3-11	SCSI Configuration.....	3-15
3-12	Memory Configuration	3-17
3-13	Integrated Peripherals.....	3-18
3-14	Updating Firmware	3-19
3-15	Hard Disk Setup Screen.....	3-21
3-16	Create New Partition Dialog Box.....	3-24

3-17	Delete Partition Dialog Box	3-25
3-18	Formatting a FAT Partition.....	3-26
3-19	Standard Formatting	3-27
3-20	Standard CMOS Setup Screen.....	3-28
3-21	Advanced CMOS Setup Screen	3-30
3-22	Installing Windows NT	3-32
3-23	Operating System Selections	3-34
3-24	Primary Operating System	3-36
3-25	Operating System Selection Setup.....	3-38
3-26	Switching to the SRM Console.....	3-42
3-27	Run Maintenance Program Dialog Box	3-44
3-28	AlphaBIOS Utilities Menu.....	3-45
4-1	Location of RMC Components on Motherboard	4-2
4-2	Setups for RMC Mode	4-3
4-3	Bypass Mode	4-4
4-4	Power Button	4-9
4-5	Halt/Reset Jumper	4-10
4-6	Halt/Reset Button	4-11
4-7	RMC Jumpers (Default Positions)	4-18
A-1	Removing Power from the System	A-3
A-2	Opening the System Cabinet	A-4
A-3	Removing the Floppy Disk Enclosure	A-5
A-4	Setting Jumpers on the Motherboard	A-6
A-5	Restoring Power	A-8

Tables

1	Documentation	xiv
1-1	Control Panel Functions	1-3
1-2	File Locations for Creating Update Diskettes on a PC	1-32
1-3	LFU Command Summary	1-41
2-1	Summary of SRM Console Commands	2-3
2-2	Syntax for SRM Console Commands	2-5
2-3	Special Characters for SRM Console	2-6
2-4	Device Naming Convention.....	2-11
2-5	PCI Address Assignments.....	2-11
2-6	Environment Variable Summary.....	2-38
2-7	Settings for boot_osflags Bootflags (OpenVMS).....	2-42
3-1	AlphaBIOS Option Key Mapping.....	3-46
4-1	SRM Environment Variables for COM1	4-6
4-2	Status Command Fields.....	4-7

4-3	Halt/Reset Button Function under Operating Systems.....	4-11
4-4	Elements of Dial String and Alert String	4-16
4-5	RMC Troubleshooting	4-19
A-1	Jumpers and Factory Default Positions.....	A-7

Preface

Intended Audience

This manual is for anyone who manages, operates, or services an AlphaServer DS10 system.

Document Structure

This manual uses a structured documentation design. Topics are organized into small sections for efficient reference. Each topic begins with an abstract. You can quickly gain a comprehensive overview by reading only the abstracts. Next is an illustration, table, or example, also providing quick reference. Last in the structure are descriptive text and syntax definitions.

This manual has four chapters, one appendix, and an index:

- **Chapter 1, Operations**, provides basic operating instructions, including powering up the system, booting, and operating system installation.
- **Chapter 2, SRM Console**, presents the command-line interface that supports the Tru64 UNIX and OpenVMS operating systems. (NT uses the SRM **test** command, only.) The SRM console is used to bootstrap the operating system, configure and test the system hardware, examine system options for errors, and set or change environment variables.
- **Chapter 3, AlphaBIOS Console**, details the graphical interface that supports the Microsoft Windows NT operating system and some utility programs. This chapter explains how to perform common system management tasks with AlphaBIOS.
- **Chapter 4, Remote Management Console**, describes how to manage the system from a remote location.
- **Appendix A, Setting System Jumpers**, describes how to check and reset if necessary the Halt/Reset select jumper and remote management console jumper.

Conventions

In examples of SRM console output, commands the user enters are presented in **boldface type**, while the system's output is in regular type. Comments on the examples are either called out with circled numbers (❶❷❸) or are preceded by a pound sign (#) and are given in ***boldface italics***.

Documentation Titles

This lists the AlphaServer DS10 documentation.

Table 1 AlphaServer DS10 Documentation

Title	Order No.
<i>Console Reference Guide</i>	EK-DS10S-CR
<i>Rackmount Guide</i>	EK-DS10S-RM
<i>Maintenance / Service Guide</i>	AK-DS10S-MS
<i>Safety Booklet</i>	296382-021
<i>User Reference Card</i>	EK-DS10S-UR
<i>Quick Setup Poster</i>	EK-DS10S-CP

Chapter 1

Operations

This chapter provides basic operating instructions, including powering up the system, booting, and operating system installation. Note that your choice of operating system has already been installed at the factory; this information is provided so that should you decide to change operating systems, you may. It also provides information about updating firmware.

Sections in this chapter are:

- Powering Up the System
- Power-Up Display
- Booting Tru64 UNIX
- Installing Tru64 UNIX
- Booting Open VMS
- Installing OpenVMS
- Booting Windows NT
- Installing Windows NT
- Switching Between Operating Systems
- Updating Firmware
- Hard Disk Partitioning for Windows NT
- Using the Halt Button
- Halt Assertion

1.1 Powering Up the System

To power up the system, check your power setting (see Figure 1-1). Then press the On/Off button to the On position (Figure 1-2). The button is in the lower right hand corner of the front of the system. Check the control panel LEDs.

Figure 1-1 Check Power Setting

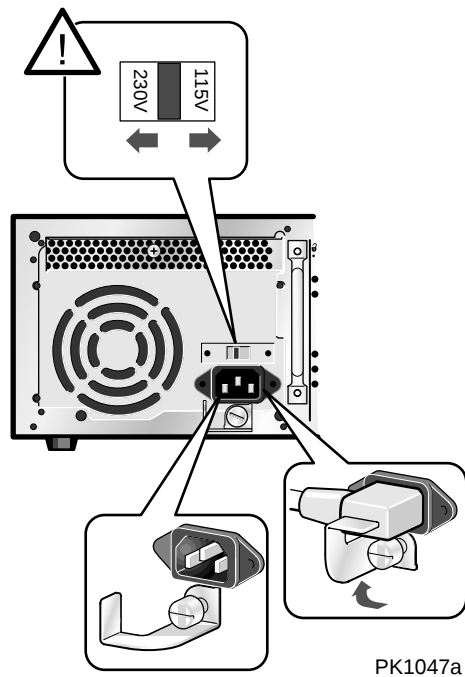
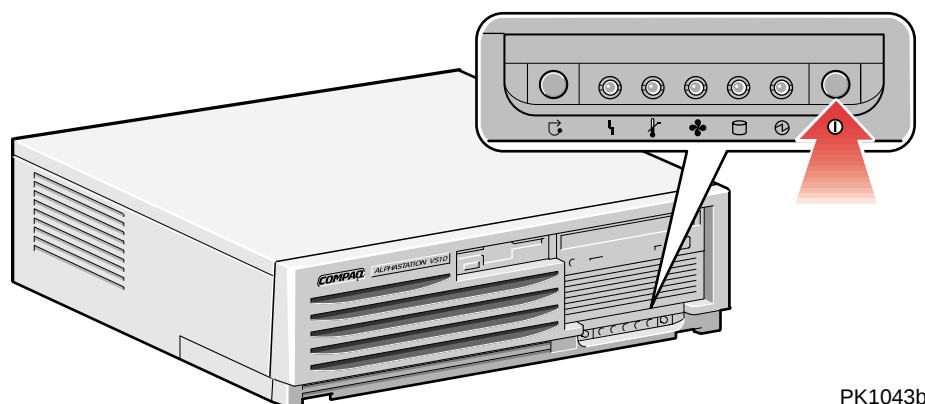


Figure 1-2 Location of Control Panel and On/Off Button



PK1043b

Table 1-1 Control Panel Functions

Symbol	Function
	Halt button. Under OpenVMS and Tru64 UNIX, suspends the operating system and returns control to the SRM console. Under Windows NT, this button initiates a system reset.
	Environmental amber LED. On indicates Temperature or Fan LEDs are on. Flashes when operating system invokes it as an alert.
	Temperature amber LED. On indicates internal temperature exceeds operating conditions. The system shuts down 30 seconds after this LED lights.
	Fan amber LED. On indicates that at least one of the three fans in the system has failed. The system shuts down 30 seconds after this LED lights.
	Disk Activity green LED. Flashes when internal system disks are accessed.
	Power Present green LED. On when power is present in the system.
	Power button. Push in to start the system and connect power. Push again to remove power and stop the system.

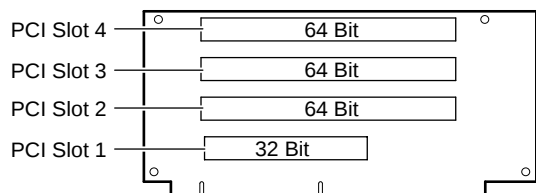
1.2 Power-Up Display

Testing begins after pressing the On/Off button, and screen text similar to that in Example 1-1 displays (if the console terminal is a serial terminal connected to the COM1 port), along with status messages in the control panel display. If the console terminal is a graphics monitor, only the last few lines of the power-up display print.

Example 1-1 Power-Up Display

```
256 Meg of system memory ❶
probing hose 0, PCI
probing PCI-to-ISA bridge, bus 1 ❷
probing PCI-to-PCI bridge, bus 2 ❷
bus 0, slot 9 -- ewa -- DE500-BA Network Controller
bus 0, slot 11 -- ewb -- DE500-BA Network Controller
bus 0, slot 13 -- dqa -- Acer Labs M1543C IDE
bus 0, slot 13 -- dqb -- Acer Labs M1543C IDE
bus 0, slot 14 ❸-- vga -- DEC PowerStorm
bus 2, ❹slot 0 -- pka -- NCR 53C875
bus 2, ❹slot 1 -- pkb -- NCR 53C875
bus 2, ❹slot 2 -- ewc -- DE500-AA Network Controller
bus 0, ❺ slot 16 -- pkc -- QLogic ISP1020
bus 0, ❻ slot 17 -- dra -- Mylex DAC960
Testing the System
Testing the Disks (read only)
>>> ❼
```

Figure 1-3 Physical Numbering for PCI Slots



PK1045-99

- ❶ Memory size is determined.
- ❷ The PCI bridges and attendant buses (indicated as IODn by the console) are probed and the devices are reported. I/O adapters are configured.
- ❸ Power-up slot 14 corresponds to the bottom physical slot, slot 0. See Figure 1–3.

Slot Location	Physical Slot Number	Logical Slot Number
Top	3	17
Second from top	2	16
Second from bottom	1	15
Bottom	0	14

- ❹ These devices are behind bridge of the card in logical slot 15, physical slot 1, second from the bottom.
- ❺ This device in is logical slot 16, physical slot 3.
- ❻ Logical slot 17 is physical slot 4, the top slot.
- ❼ The SRM console banner and prompt (>>>) are printed.
The SRM console is a command-line interface you use to set or read system parameters.

If the **auto_action** environment variable is set to **boot** or **restart** and the **os_type** environment variable is set to **unix** or **openvms**, the Tru64 UNIX or OpenVMS operating system boots.

If the system is running the Windows NT operating system (the **os_type** environment variable is set to **nt**), the SRM console loads and starts the AlphaBIOS console.

See Section 2.19 for information on environment variables. See Chapter 2 for SRM console, and Chapter 3 for AlphaBIOS information.

1.3 Booting Tru64 UNIX

Tru64 UNIX® can be booted from a local disk or a remote disk through an Ethernet connection. Refer to the documentation shipped with the operating system for booting instructions.

1.3.1 Booting from a Local Disk

Example 1-2 Booting Tru64 UNIX from a Local Disk

```
>>> sho device ❶
dka100.1.0.2000.0      DKA100      RZ1CB-CA  LYJ0 ❷
dka300.3.0.2000.0      DKA300      RZ1CB-CA  LYJ0
dka500.5.0.2000.0      DKA500      RZ1EF-AB  0370
dkb0.0.0.2001.0        DKB0        RZ1CB-CA  LYJ0
dkb200.2.0.2001.0      DKB200      RZ1DB-CS  0307
dkb400.4.0.2001.0      DKB400      RZ1CB-CA  LYJ0
dkc0.0.0.16.0          DKC0        RZ1CB-BA  LYG0
dkc200.2.0.16.0        DKC200      RZ1CB-BA  LYG0
dqa1.1.0.13.0          DQA1        CD-532E   1.0A
dra1.0.0.17.0          DRA1        1 Member JBOD
dra2.0.0.17.0          DRA2        1 Member JBOD
dva0.0.0.0.0           DVA0
ewa0.0.0.9.0           EWA0        08-00-2B-86-1B-BA
ewb0.0.0.11.0          EWB0        08-00-2B-86-1B-BB
ewc0.0.0.2002.0        EWC0        00-06-2B-00-26-1C
pka0.7.0.2000.0        PKA0        SCSI Bus ID 7

>>> boot -file vmunix -flags a dkc0 ❸
(boot dkc0.0.0.9.0 -file vmunix -flags a)
block 0 of dkc0.0.0.9.0 is a valid boot block
reading 16 blocks from dkc0.0.0.9.0
bootstrap code read in
base = 1ee000, image_start = 0, image_bytes = 2000
initializing HWRPB at 2000
initializing page table at 1fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

Tru64 UNIX boot - Fri Aug 7 20:30:19 EDT 1999
Loading vmunix ...
. . .
The system is ready.

Tru64 UNIX Version V4.0E (sab128.eng.pko.dec.com) console
login:
```

```
*****
* Starting Desktop Login on display :0...
* Wait for the Desktop Login screen before logging in.
*****
```

4

- ❶ The **show device** command displays device information, including name and type of connection to the system. See Section 2.3.2 for a description of the **show device** command and the device naming convention.
- ❷ The operating system is on the third disk connected to the system through the controller in slot 3 of the PCI. The name of this device, **dkc0**, is used as an argument to the **boot** command.
- ❸ This command loads Tru64 UNIX from the disk **dkc0**, using the boot file **vmunix** and autobooting to multiuser mode. See Section 2.5 for a description of the **boot** command.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 2.19 for information about environment variables.

- ❹ The operating system banner displays.

1.3.2 Booting from a Remote Disk

Example 1-3 Booting Tru64 UNIX from a Remote Disk

```
>>> show device ❶
. . .
ewa0.0.0.8.0          EWA0      08-00-2B-E2-9C-60 ❷

>>>
>>> boot -flags an -protocols bootp ewa ❸
      (boot ewa0.0.0.4.1 -flags an)
      Building FRU table

      Trying BOOTP boot.

      Broadcasting BOOTP Request...
      Received BOOTP Packet File Name: /var/adm/ris/ris0.alpha/hvmunix
      local inet address: 16.122.128.26
      remote inet address: 16.122.128.59
      TFTP Read File Name: /var/adm/ris/ris0.alpha/hvmunix
      .....
      bootstrap code read in
      base = 200000, image_start = 0, image_bytes = 9a0fa0
      initializing HWRPB at 2000
      initializing page table at 1f2000
      initializing machine state
      setting affinity to the primary CPU
      jumping to bootstrap code

      Secondary boot program - Thu Aug 1 22:33:13 EST 1999

      Loading vmunix ...
      .
      .
      .
      The system is ready.

      Tru64 UNIX Version V4.0E (sabl28.eng.pko.dec.com) console ❹
```

- ❶ The **show device** command displays device information, including name and type of connection to the system. See Section 2.3.2 for a description of the **show device** command and the device naming convention.
- ❷ The operating system is on a remote disk accessed through the Ethernet controller in slot 4 of the PCI. The name of this device, **ewa0**, is used as an argument to the **boot** command.
- ❸ This command loads Tru64 UNIX from **ewa0**, autobooting to multiuser mode. See Section 2.5 for a description of the **boot** command.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 2.19 for information about environment variables.

- ❹ The operating system banner displays.

1.4 Installing Tru64 UNIX

Tru64 UNIX is installed from the CD-ROM. Refer to the documentation shipped with the CD-ROM for installation instructions.

Example 1-4 Installing Tru64 UNIX

```
>>> show device
      . . .
dka500.5.0.7.1      DKA500      RRD47  1337 ❶
      . . .
>>>

>>> boot dka500      ❶
(boot dka500.5.0.7.1 -flags A)
block 0 of dka500.5.0.7.1 is a valid boot block
reading 16 blocks from dka500.5.0.7.1
bootstrap code read in
base = 1ee000, image_start = 0, image_bytes = 2000
initializing HWRPB at 2000
initializing page table at 1fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

Tru64 UNIX boot - Thu Jul 16 16:59:31 EDT 1999

Loading vmunix ...
.
.
.
INIT: SINGLE-USER MODE

Initializing system for Tru64 UNIX installation.  Please wait...

*** Performing CDROM Installation

Loading installation process and scanning system hardware.

[The "Welcome to the Tru64 UNIX Installation Procedure" appears.] ❷
```

- ❶ Use the **boot** command to install the operating system from the CD-ROM, which is either dka500 or dqa0.
- ❷ See your operating system documentation for further installation instructions.

1.5 Booting OpenVMS

OpenVMS can be booted from a local disk, a disk connected through a cluster, or a remote disk through an Ethernet connection. Refer to the documentation shipped with the operating system for booting instructions.

1.5.1 Booting OpenVMS from a Local Disk

Example 1-5 Booting OpenVMS from a Local Disk

```
>>> show device ❶
dka200.2.0.7.1      . . . DKA200      RZ1CB-CA  LYJ0
                   . . .
>>>

>>> show boot_reset ❷
boot_reset      ON
>>> show bootdef_dev ❸
bootdef_dev      dka200.2.0.7.1
>>> boot ❹
(boot dka200.2.0.7.1 -flags 0,0)
block 0 of dka200.2.0.7.1 is a valid boot block
reading 893 blocks from dka200.2.0.7.1
bootstrap code read in
base = 1fa000, image_start = 0, image_bytes = 6fa00
initializing HWRPB at 2000
initializing page table at 1fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

OpenVMS (TM) Alpha Operating System, Version 7.1-2
$!Copyright(c) 1999 Digital Equipment Corporation. All rights reserved.
%STDRV-I-STARTUP, OpenVMS startup begun at 30-JUL-1999 11:47:11.04
%MSCPLOAD-I-CONFIGSCAN, enabled automatic disk serving
```

Continued on next page

Example 1-5 Booting OpenVMS from a Local Disk (Continued)

```
. . .  
The OpenVMS system is now executing the site-specific startup commands.  
. . .
```

```
Welcome to OpenVMS (TM) Alpha Operating System, Version V7.1-2 ⑤
```

```
Username:
```

- ① The **show device** command displays device information. See Section 2.3.2 for a description of the **show device** command and the device naming convention.
- ② The **boot_reset** environment variable was previously set to “on,” causing the power-up trace to display when the system initializes (see Section 1.2). See Section 2.19 for commands used with environment variables.
- ③ The **bootdef_dev** environment variable specifies the default boot device. In this example, the default boot device was previously set to dka200.2.0.7.1.
- ④ No boot device is specified in the **boot** command; the default boot device was set with the environment variable. See Section 2.5 for a description of the **boot** command.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 2.19 for information about environment variables.

- ⑤ The operating system banner displays.

1.5.2 Booting OpenVMS from a Disk on a Cluster

Example 1-6 Booting OpenVMS from a Disk on a Cluster

```
>>> show bootdef_dev ❶
bootdef_dev          dual110.0.0.8.0
>>> show device      ❷
. . .
dual110.0.0.8.0      $1$DIA110 (DENVER)      RF74 ❸
. . .

>>> boot              ❹
(boot dual110.0.0.8.0 -flags 0)
Building FRU table
.
.
.
Welcome to OpenVMS Alpha (TM) Operating System, Version V7.1-2 ❺
```

- ❶ The **bootdef_dev** environment variable specifies the default boot device.
- ❷ The **show device** command displays device information, including name and type of connection to the system. See Section 2.3.2 for a description of the **show device** command and the device naming convention.
- ❸ The disk `dua110.0.0.8.0` is on the cluster that includes this system.
- ❹ No boot device is specified in the **boot** command; the default boot device was set with the environment variable. See Section 2.5 for a description of the **boot** command.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 2.19 for information about environment variables.

- ❺ The operating system banner prints.

1.5.3 Booting OpenVMS from a Remote Disk

Example 1-7 Booting OpenVMS from a Remote Disk

```
>>> show device ❶  
    . . .  
ewa0.0.0.8.0      EWA0      08-00-2B-E2-9C-60 ❶  
    . . .  
>>>  
>>> boot ewa0 -flags 0 ❷  
(boot ewa0.0.0.2.0 -flags 0)  
Building FRU table  
  
Trying MOP boot.....  
Network load complete.  
    . . .  
Welcome to OpenVMS Alpha (TM) Operating System, Version V7.1-2 ❸
```

- ❶ The **show device** command displays device information, including name and type of connection to the system. In this example the Ethernet connection is ewa0. See Section 2.3.2 for a description of the **show device** command and the device naming convention.
- ❷ The **boot** command specifies ewa0 as the boot device. See Section 2.5 for a description of the **boot** command.

The **boot** command accepts the name of a boot device, a boot file name through the **-file** option, and boot flags through the **-flags** option. The environment variables **bootdef_dev**, **boot_file**, and **boot_osflags** can also be used to specify the default boot device or device list, the default boot file, and flag information. When an option and the corresponding environment variable are both in a command string, the option overrides the environment variable. The value of the environment variable, however, is not changed. See Section 2.19 for information about environment variables.

- ❸ The operating system banner prints.

1.6 Installing OpenVMS

OpenVMS is installed from the CD-ROM. Refer to the documentation shipped with the OpenVMS kit for complete installation instructions.

Example 1-8 Installing OpenVMS

```
>>> boot -flags 0,0 dka500 ❶
Initializing...
SRROM V3.0 on cpu0
.
. [The initialization display prints. See Section 1.2.]
.
AlphaServer DS20 Console V5.4-x, 2-APR-1999 15:17:48

CPU 0 booting

(boot dka500.5.0.1.1 -flags 0,0)
Building FRU table
block 0 of dka500.5.0.1.1 is a valid boot block
reading 1002 blocks from dka500.5.0.1.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 7d400
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

OpenVMS (TM) Alpha Operating System, Version 7.1x ❷

%SMP-I-SECMSG, CPU #01 message: P01>>>START
%SMP-I-CPUBOOTED, CPU #01 has joined the PRIMARY CPU in multiprocessor operation
Installing required known files...

Configuring devices...

*****

You can install or upgrade the OpenVMS Alpha operating system
or you can install or upgrade layered products that are included
on the OpenVMS Alpha operating system CD-ROM.

You can also execute DCL commands and procedures to perform
"standalone" tasks, such as backing up the system disk.
```

Continued on next page

Example 1-8 Installing OpenVMS (Continued)

Please choose one of the following:

- 1) Install or upgrade OpenVMS Alpha Version 7.1x
- 2) List layered product kits that this procedure can install
- 3) Install or upgrade layered product(s)
- 4) Execute DCL commands and procedures
- 5) Shut down this system

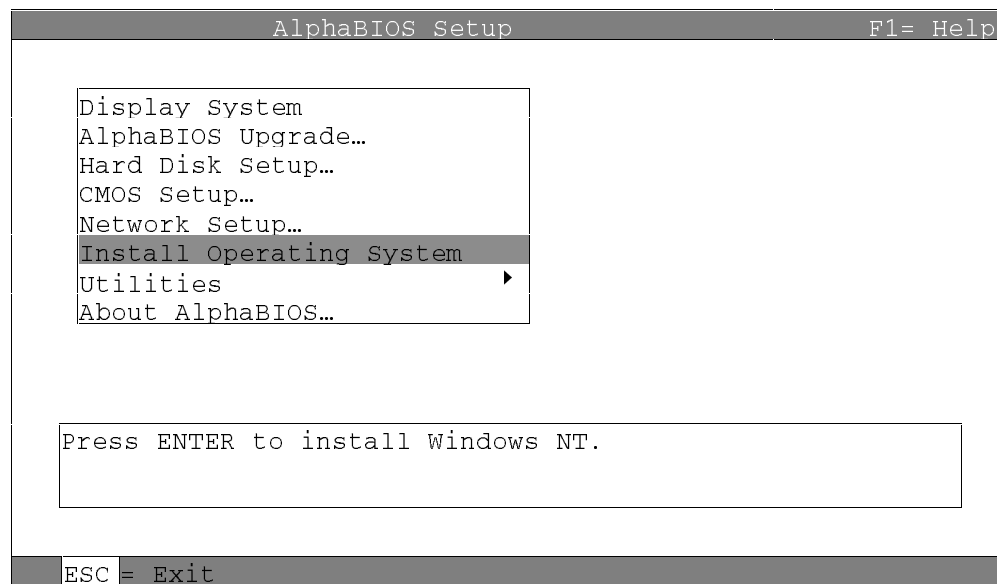
Enter CHOICE or ? to repeat menu: (1/2/3/4/5/?)

- ❶ Use the **boot** command to install the operating system from the CD-ROM, which is either dka500 or dqa0.
- ❷ See your operating system documentation for installation instructions.

1.7 Booting Windows NT

Microsoft Windows NT is started from the AlphaBIOS Boot screen.

Figure 1-4 AlphaBIOS Boot Screen



Two SRM environment variables must be set properly for Windows NT to boot.

The setting of the SRM **os_type** environment variable determines if AlphaBIOS is loaded and started on reset and power-up. If **os_type** is set to **nt**, after the power-up display the SRM console is loaded and started, and it then loads and starts the AlphaBIOS console. AlphaBIOS must be running before Windows NT can be booted. Windows NT requires a graphics monitor as its console. Setting the SRM console environment variable to graphics and having a graphics monitor attached to your system meets this requirement. After setting these two variables, you have to power-down and power-up your system for them to take effect.

The method used for booting Windows NT is determined by the setting of **Auto Start** in the AlphaBIOS **Standard CMOS Setup** screen (see Chapter 3).

- If **Auto Start** is enabled, the primary version of Windows NT starts automatically.
- If **Auto Start** is disabled, use the arrow keys to select the Windows NT version to start. Press Enter to boot Windows NT.

NOTE: *The SRM **console** environment variable must be set to **graphics** before booting Windows NT, though this setting is not necessary to run AlphaBIOS. At the SRM console prompt, >>>:*

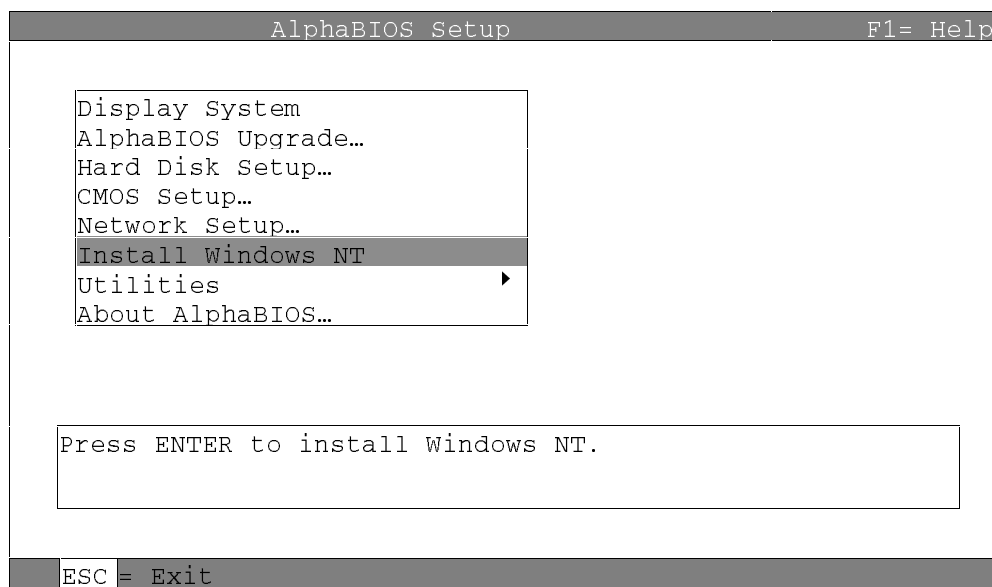
```
>>> set console graphics
```

The Halt/Reset jumper on the motherboard must be set to the Windows NT position for the Halt button to operate (see Section 1.12). If the Halt/Reset jumper is set in the OpenVMS/Tru64 UNIX position, the Halt button will be ignored by the Windows NT operating system and have no effect. Properly set, the Halt button will initiate a system reset.

1.8 Installing Windows NT

Windows NT is installed from the CD-ROM. Insert the CD-ROM into the drive, start AlphaBIOS Setup, select the menu item Install Windows NT, and follow the prompts.

Figure 1-5 Installing Windows NT



The Halt/Reset jumper on the motherboard must be set to the Windows NT position for the Halt button to operate (see Section 1.12). If the Halt/Reset jumper is set in the OpenVMS/Tru64 UNIX position, the Halt button will be ignored by the Windows NT operating system and have no effect. Properly set, the Halt button will initiate a system reset.

Windows NT requires a partitioned and formatted hard disk drive. If your drive is not partitioned or formatted, follow the instructions in Section 3.5 before installing the Windows NT operating system.

Up to three versions of Windows NT can be resident in a system at one time.

If this is a new Windows NT installation, start with this procedure:

1. Use **CMOS Setup** to set the system date and time: start **AlphaBIOS Setup**, select **CMOS Setup**, and press Enter.
2. Perform an express hard disk setup: return to the main **AlphaBIOS Setup** screen, select **Hard Disk Setup**, and press Enter.
3. Perform an express hard disk setup by pressing F7 to enter **Express Setup**.
4. Continue the setup by pressing the F10 key.
5. Go to the procedure below.

This procedure is for all Windows NT installations:

1. Put the Windows NT CD into the CD-ROM drive.
2. Start **AlphaBIOS Setup**, select **Install Windows NT**, and press Enter.

Follow the prompts to complete the installation. For more information on installing Windows NT, refer to the *Installation Guide* in your Windows NT software package.

1.9 Switching Between Operating Systems

The system supports multiple operating systems on different system and data disks not in the machine at the same time. That is, you can have a set of disks for each operating system.

CAUTION: *This operation is not for the faint hearted especially if you have a shadow system disk and shadow arrays. The file structures of each of the three operating systems are incompatible and therefore all disks must be removed from the system and upon reinstallation must be replaced in exactly the same physical locations. It is therefore necessary to keep track of the location of each disk in the system.*

1.9.1 Switching to Windows NT

Use the following procedure:

1. Shut down the operating system and power off the system.
2. Remove and mark the physical location of each disk in the system.
3. Check your Halt/Reset jumper position (see Section 1.12).
4. Either place blank disks or your Windows NT disk set into the system. (If you are placing a Windows NT disk set into the system, be sure that each disk is replaced in the same physical location from which it was removed.)
5. Power on the system.
6. Enter the following commands at the SRM console prompt:

```
>>> set os_type nt
>>> init
```
7. Either install Windows NT, see Section 1.8, or at the AlphaBIOS boot screen, start AlphaBIOS setup (F2), select **CMOS Setup**, and press Enter.
8. Set the system date and time.
9. In CMOS Setup, check that the setup for the floppy and other basic parameters is accurate. Set system-specific parameters, such as the

memory test and password, in Advanced CMOS Setup as needed. Press **F10** to save the changes.

10. From the AlphaBIOS Setup screen select **Utilities**. In the selection box, choose **OS Selection Setup**. Make sure the selections (boot name, boot file, and so on) are what you want. Press **F10** to save the changes.
11. Return to the boot screen and boot Windows NT.

1.9.2 Switching to Tru64 UNIX or OpenVMS

Use the following procedure:

1. Shut down the operating system and power off the system.
2. Remove and mark the physical location of each disk in the system.
3. Either place blank disks or your Tru64 UNIX or OpenVMS disk set into the system. **No matter which disk set you are placing into the system, be sure that each disk is placed in the same physical location from which it was removed.**
4. Power on the system.
5. In AlphaBIOS, access the Advanced CMOS Setup screen and change the Console Selection to Tru64 UNIX or OpenVMS (both use SRM) as appropriate. Press **F10** to save the change. This menu selection changes the **os_type** environment variable to either UNIX or VMS so that the SRM console does not load AlphaBIOS but remains in the system when you reset the system.
6. Press the Halt/Reset button to reset the system.
7. Either install Tru64 UNIX (see Section 1.4) or OpenVMS (see Section 1.6) or boot the operating system.

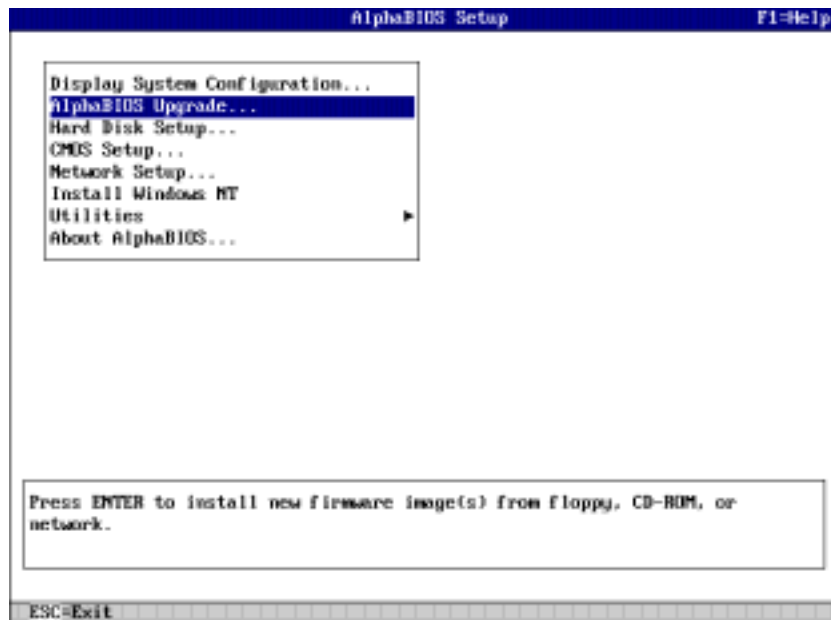
1.10 Updating Firmware

Start the Loadable Firmware Update (LFU) utility by issuing the lfu command at the SRM console prompt, booting it from the CD-ROM while in the SRM console, or selecting Update AlphaBIOS in the AlphaBIOS Setup screen.

Example 1-9 Starting LFU from the SRM Console

```
>>> lfu
      ***** Loadable Firmware Update Utility *****
Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: cda0
. . .
UPD>
```

Figure 1-6 Starting LFU from the AlphaBIOS Console



PK1461-98

Use the Loadable Firmware Update (LFU) utility to update system firmware.

You can start LFU from either the SRM console or the AlphaBIOS console.

- From the SRM console, start LFU by issuing the **lfu** command (see Example 1–9). Also from the SRM console, LFU can be booted from the Alpha CD-ROM (V5.4 or later), as shown in Example 1–10.
- From the AlphaBIOS console, select **Update AlphaBIOS** from the **AlphaBIOS Setup** screen (see Figure 1–6).

A typical update procedure is:

1. Start LFU.
2. Use the LFU **list** command to show the revisions of modules that LFU can update and the revisions of update firmware.
3. Use the LFU **update** command to write the new firmware.
4. Use the LFU **exit** command to go back to the console.

The sections that follow show examples of updating firmware from the local CD-ROM, the local floppy, and a network device.

Example 1–10 Booting LFU from the CD-ROM

```
>>> show device      . . .
dka500.5.0.7.1      DKA500          RRD47  1645
. . .
>>> boot dka500
(boot dka500.5.0.7.1 -flags 0,0)
block 0 of dka500.5.0.7.1 is a valid boot block
. .
jumping to bootstrap code
The default bootfile for this platform is
      [DS10]DS10_LFU.EXE
Hit <RETURN> at the prompt to use the default bootfile.
Bootfile: <CR>
Starting Firmware Update Utility

      ***** Loadable Firmware Update Utility *****
. . .
UPD>
```

1.10.1 Updating Firmware from the CD-ROM

Insert the Alpha CD-ROM, start LFU, and select cda0 as the load device.

Example 1-11 Updating Firmware from the CD-ROM

```
***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: cda0 ❶

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [DS10FW]: DS10CP ❷

Copying DS10CP from DKA500.5.0.1.1 .
Copying [DS10]TCREADME from DKA500.5.0.1.1 .
Copying [DS10]TCSRMMROM from DKA500.5.0.1.1 .....
Copying [DS10]TCARCROM from DKA500.5.0.1.1 .....

-----
Function      Description ❸
-----
Display      Displays the system's configuration table.
Exit         Done exit LFU (reset).
List         Lists the device, revision, firmware name, and
              update revision.
Lfu          Restarts LFU.
Readme       Lists important release information.
Update       Replaces current firmware with loadable data image.
Verify       Compares loadable and hardware images.
? or Help    Scrolls this function table.
-----

UPD>
```

- ❶ Select the device from which firmware will be loaded. The choices are the internal CD-ROM, the internal floppy disk, or a network device. In Example 1–11, the internal CD-ROM is selected.

- ❷ Select the file that has the firmware update, or press Enter to select the default file. The file options are:

DS10FW (default)	SRM console, AlphaBIOS console, and I/O adapter firmware.
------------------	---

DS10CP	SRM console and AlphaBIOS console firmware only.
--------	--

DS10IO	I/O adapter firmware only.
--------	----------------------------

In this example the file for console firmware (AlphaBIOS and SRM) is selected.

- ❸ The LFU function table and prompt (UPD>) display.

Continued on next page

Example 1-11 Updating Firmware from the CD-ROM (Continued)

```
UPD> list ④
```

Device	Current Revision	Filename	Update Revision
Fsb	3.1-x	fsb_fw	3.x
Nt	5.69	nt_fw	5.xx
Plx0	A11	kzpsa_fw	A11
Srm	5.4-x	srm_fw	5.4-xx
Pua	A214	cipca_fw	A2xx

```
UPD> update * ⑤
```

WARNING: updates may take several minutes to complete for each device.

```
Confirm update on: AlphaBIOS      [Y/(N)] y ⑥
```

DO NOT ABORT!

AlphaBIOS Updating to V6.40-1... Verifying V6.40-1... PASSED.

```
Confirm update on: srmflash        [Y/(N)] y
```

DO NOT ABORT!

srmflash Updating to V6.0-3... Verifying V6.0-3... PASSED.

```
UPD> exit ⑦
```

- ④ Use the LFU **list** command to determine the revision of firmware in a device and the most recent revision of that firmware available in the selected file. In this example, the resident firmware for each console (SRM and AlphaBIOS) is at an earlier revision than the firmware in the update file.
- ⑤ The **update** command updates the device specified or all devices. In this example, the wildcard indicates that all devices supported by the selected update file will be updated.
- ⑥ For each device, you are asked to confirm that you want to update the firmware. The default is no. Once the update begins, do not abort the operation. Doing so will corrupt the firmware on the module.
- ⑦ The **exit** command returns you to the console from which you entered LFU (either SRM or AlphaBIOS).

1.10.2 Updating Firmware from Floppy Disks

Create two update diskettes before starting LFU: one for console updates and one for I/O. See Section 1.10.3 for an example of the update procedure.

Table 1-2 File Locations for Creating Update Diskettes on a PC

Console Update Diskette	I/O Update Diskette
DS10FW.TXT	DS10IO.TXT
DS10CP.TXT	TCREADME.SYS
TCREADME.SYS	CIPCA315.SYS
TCSRMRROM.SYS	DFPAA310.SYS
TCARCROM.SYS	KZPAAA11.SYS

1. Download the update files from the Internet.
2. On a PC, copy files onto two FAT-formatted diskettes as shown in Table 1-2.
3. From an OpenVMS system, copy files onto two ODS2-formatted diskettes as shown in Example 1-12.

Example 1-12 Creating Update Diskettes on an OpenVMS System

Console update diskette

```
$ inquire ignore "Insert blank HD floppy in DVA0, then continue"
$ set verify
$ set proc/priv=all
$ init /density=hd/index=begin dva0: tcods2cp
$ mount dva0: tcods2cp
$ create /directory dva0:[ds10]
$ copy tcreadme.sys dva0:[ds10]tcreadme.sys
$ copy DS10fw.txt dva0:[ds10]ds10fw.txt
$ copy DS10cp.txt dva0:[ds10]ds10cp.txt
$ copy tcsrmrom.sys dva0:[ds10]tcsrmrom.sys
$ copy tcarcrom.sys dva0:[ds10]tcarcrom.sys
$ dismount dva0:
$ set noverify
$ exit
```

I/O update diskette

```
$ inquire ignore "Insert blank HD floppy in DVA0, then continue"
$ set verify
$ set proc/priv=all
$ init /density=hd/index=begin dva0: tcods2io
$ mount dva0: tcods2io
$ create /directory dva0:[ds10]
$ create /directory dva0:[options]
$ copy tcreadme.sys dva0:[ds10]tcreadme.sys
$ copy DS10fw.txt dva0:[ds10]ds10fw.txt
$ copy DS10io.txt dva0:[ds10]ds10io.txt
$ copy cipca214.sys dva0:[options]cipca214.sys
$ copy dfpaa246.sys dva0:[options]dfpaa246.sys
$ copy kzpsaA10.sys dva0:[options]kzpsaa10.sys
$ dismount dva0:
$ set noverify
$ exit
```

1.10.3 Performing the Update from Floppy Disks

Insert an update diskette (see Section 1.10.2) into the floppy drive. Start LFU and select dva0 as the load device.

Example 1-13 Updating Firmware from the Floppy Disk

```
***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: dva0 ❶

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [DS10IO,(DS10CP)]: DS10IO ❷

Copying DS10IO from DVA0 .
Copying TCREADME from DVA0 .
Copying CIPCA214 from DVA0 .
Copying DFPAA252 from DVA0 ...
Copying KZPSAA11 from DVA0 ...
.
. [The function table displays, followed by the UPD> prompt, as
.  shown in Example 1-11.]

UPD> list ❸
```

Device	Current Revision	Filename	Update Revision
Fsb	3.1-x	fsb_fw	3.x
Nt	5.69	nt_fw	5.xx
Pkx0	A11	kzpsa_fw	A11
Srm	5.4-x	srm_fw	5.4-xx
Pua	A214	cipca_fw	A2xx

Continued on next page

- ❶ Select the device from which firmware will be loaded. The choices are the internal CD-ROM, the internal floppy disk, or a network device. In this example, the internal floppy disk is selected.
- ❷ Select the file that has the firmware update, or press Enter to select the default file. When the internal floppy disk is the load device, the file options are:

DS10CP (default) SRM console and AlphaBIOS console firmware only.

DS10IO I/O adapter firmware only.

The default option in Example 1–11 (DS10FW) is not available, since the file is too large to fit on a 1.44 MB diskette. This means that when a floppy disk is the load device, you can update either console firmware or I/O adapter firmware, but not both in the same LFU session. If you need to update both, after finishing the first update, restart LFU with the **lfu** command and insert the diskette with the other file.

In this example the file for I/O adapter firmware is selected.

- ❸ Use the LFU **list** command to determine the revision of firmware in a device and the most recent revision of that firmware available in the selected file. In this example, the update revision for console firmware displays as “Missing file” because only the I/O firmware files are available on the floppy disk.

Example 1-13 Updating Firmware from the Floppy Disk (Continued)

```
UPD> update pfi0 ④
WARNING: updates may take several minutes to complete for each device.

Confirm update on: pfi0          [Y/(N)] y ⑤

                                DO NOT ABORT!
pfi0          Updating to 3.10... Verifying to 3.10... PASSED.

UPD> lfu ⑥

***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: dva0

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [DS10IO, (DS10CP)]: ⑦

.
. (The function table displays, followed by the UPD> prompt.
. Console firmware can now be updated.)

UPD> exit ⑧
```

- ④ The **update** command updates the device specified or all devices.
- ⑤ For each device, you are asked to confirm that you want to update the firmware. The default is no. Once the update begins, do not abort the operation. Doing so will corrupt the firmware on the module.
- ⑥ The **lfu** command restarts the utility so that console firmware can be updated. (Another method is shown in Example 1–14, where the user specifies the file DS10FW and is prompted to insert the second diskette.)
- ⑦ The default update file, DS10CP, is selected. The console firmware can now be updated, using the same procedure as for the I/O firmware.
- ⑧ The **exit** command returns you to the console from which you entered LFU (either SRM or AlphaBIOS).

Example 1–14 Selecting DS10FW to Update Firmware from the Internal Floppy Disk

```
>>> lfu
      ***** Loadable Firmware Update Utility *****

Select firmware load device (oda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: dva0

Please enter the name of the firmware files list, or
Press <return> to use the default filename [DS10IO, (DS10CP)]: DS10fw

Copying DS10FW from DVA0 .
Copying TCREADME from DVA0 .
Copying TCSRMMROM from DVA0 .....
Copying TCARCROM from DVA0 .....
Copying CIPCA214 from DVA0
Please insert next floppy containing the firmware,
Press <return> when ready. Or type DONE to abort.
Copying CIPCA214 from DVA0 .
Copying DFPA246 from DVA0 ...
Copying KZPSAA10 from DVA0 ...
.
.
```

1.10.4 Updating Firmware from a Network Device

Copy files to the local MOP server's MOP load area, start LFU, and select ewa0 as the load device.

Example 1-15 Updating Firmware from a Network Device

```
***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: ewa0 ❶

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [DS10FW]: ❷

Copying DS10FW from EWA0 .
Copying TCREADME from EWA0 .
Copying TCSRMMROM from EWA0 .....
Copying TCARCROM from EWA0 .....
Copying CIPCA214 from EWA0 .
Copying DFPAA246 from EWA0 ...
Copying KZPSAA11 from EWA0 ...

.
. [The function table displays, followed by the UPD> prompt, as
.  shown in Example 1-11.]

UPD> list ❸
```

Device	Current Revision	Filename	Update Revision
Fsb	3.1-x	fsb_fw	3.x
Nt	5.69	nt_fw	5.xx
Plxx0	A11	kzpsa_fw	A11
Srm	5.4-x	srm_fw	5.4-xx
Pua	A214	cipca_fw	A2xx

Continued on next page

Before starting LFU, download the update files from the Internet. You will need the files with the extension .SYS. Copy these files to your local MOP server's MOP load area.

- ❶ Select the device from which firmware will be loaded. The choices are the CD-ROM, the internal floppy disk, or a network device. In this example, a network device is selected.
- ❷ Select the file that has the firmware update, or press Enter to select the default file. The file options are:

DS10FW (default)	SRM console, AlphaBIOS console, and I/O adapter firmware.
------------------	---

DS10CP	SRM console and AlphaBIOS console firmware only.
--------	--

DS10IO	I/O adapter firmware only.
--------	----------------------------

In this example the default file, which has both console firmware (AlphaBIOS and SRM) and I/O adapter firmware, is selected.

- ❸ Use the LFU **list** command to determine the revision of firmware in a device and the most recent revision of that firmware available in the selected file. In this example, the resident firmware for each console (SRM and AlphaBIOS) and I/O adapter is at an earlier revision than the firmware in the update file.

Continued on next page

Example 1-15 Updating Firmware from a Network Device (Continued)

```
UPD> update * -all ④
WARNING: updates may take several minutes to complete for each device.

                DO NOT ABORT!
AlphaBIOS      Updating to V6.40-1... Verifying V6.40-1...  PASSED.

                DO NOT ABORT!
kzpsa0         Updating to  A11      ... Verifying A11...  PASSED.

                DO NOT ABORT!
kzpsa1         Updating to  A11      ... Verifying A11...  PASSED.

                DO NOT ABORT!
srmflash       Updating to V6.0-3... Verifying V6.0-3...  PASSED.

UPD> exit ⑤
```

- ④ The **update** command updates the device specified or all devices. In this example, the wildcard indicates that all devices supported by the selected update file will be updated. Typically LFU requests confirmation before updating each console's or device's firmware. The **-all** option eliminates the update confirmation requests.
- ⑤ The **exit** command returns you to the console from which you entered LFU (either SRM or AlphaBIOS).

1.10.5 LFU Commands

The commands summarized in Table 1-3 are used to update system firmware.

Table 1-3 LFU Command Summary

Command	Function
display	Shows the physical configuration of the system.
exit	Terminates the LFU program.
help	Displays the LFU command list.
lfu	Restarts the LFU program.
list	Displays the inventory of update firmware on the selected device.
readme	Lists release notes for the LFU program.
update	Writes new firmware to the module.
verify	Reads the firmware from the module into memory and compares it with the update firmware.

display

The **display** command shows the physical configuration of the system. **Display** is equivalent to issuing the SRM console command **show configuration**. Because it shows the slot for each module, **display** can help you identify the location of a device.

exit

The **exit** command terminates the LFU program, causes system initialization and testing, and returns the system to the console from which LFU was called.

help

The **help** (or **?**) command displays the LFU command list, shown below.

Function	Description
Display	Displays the system's configuration table.
Exit	Done exit LFU (reset).
List	Lists the device, revision, firmware name, and update revision.
Lfu	Restarts LFU.
Readme	Lists important release information.
Update	Replaces current firmware with loadable data image.
Verify	Compares loadable and hardware images.
? or Help	Scrolls this function table.

lfu

The **lfu** command restarts the LFU program. This command is used when the update files are on a floppy disk. The files for updating both console firmware and I/O firmware are too large to fit on a 1.44 MB disk, so only one type of firmware can be updated at a time. Restarting LFU enables you to specify another update file.

list

The **list** command displays the inventory of update firmware on the CD-ROM, network, or floppy. Only the devices listed at your terminal are supported for firmware updates.

The **list** command shows three pieces of information for each device:

- Current Revision — The revision of the device's current firmware
- Filename — The name of the file used to update that firmware
- Update Revision — The revision of the firmware update image

readme

The **readme** command lists release notes for the LFU program.

update

The **update** command writes new firmware to the module. Then LFU automatically verifies the update by reading the new firmware image from the module into memory and comparing it with the source image.

To update more than one device, you may use a wildcard but not a list. For example, **update k*** updates all devices with names beginning with k, and **update *** updates all devices. When you do not specify a device name, LFU tries to update all devices; it lists the selected devices to update and prompts before devices are updated. (The default is no.) The **-all** option eliminates the update confirmation requests, enabling the update to proceed without operator intervention.

CAUTION: *Never abort an update operation.
Aborting corrupts the firmware on the module.*

verify

The **verify** command reads the firmware from the module into memory and compares it with the update firmware. If a module already verified successfully when you updated it, but later failed tests, you can use **verify** to tell whether the firmware has become corrupted.

1.11 Hard Disk Partitioning for Windows NT

The recommended hard disk partition on the first hard disk in your system is:

Partition 1 should be 6 megabytes less than the total size of the drive, but no more than 4095 MB. This large partition holds the operating system and the application and data files.

Partition 2 should be the remaining 6 megabytes. This small partition holds only the few files necessary for your computer to boot.

This arrangement provides two benefits:

- AlphaBIOS requires that a boot partition be formatted with the FAT file system. However, the Windows NT file system (NTFS) provides advantages over FAT, such as additional security and more efficient use of disk space. By keeping the FAT boot partition as small as possible, the maximum amount of space is left available for use as an NTFS partition.
- Most applications install themselves onto drive C by default. Although the program installation drive is usually configurable by the user, many people accept the default of drive C. By making the first partition large, drive C becomes the larger drive. This arrangement makes program installation easier and avoids time-consuming insufficient disk space mistakes.

1.11.1 Hard Disk Error Conditions

Disk Initialization Failed

When you start hard disk setup, if you receive an “Internal error occurred” message, it means that a disk was found, but there was an error in communicating with the disk. The likely conditions that can cause this error are:

- Incompatible or failed disk cables. You may not have the correct cable installed, or the cable might have a broken lead or connector. Try another cable known to be good.
- Disk controller not configured. You may have to run a configuration utility to set up your hard disk controller. Check your controller documentation.

- Disk controller malfunction. Most controllers come with a diagnostic utility to test controller functioning. If a controller error is found, call the manufacturer for a replacement.
- Improper SCSI termination. Many SCSI controllers require that the terminating resistor packs be removed from all drives between the controller and last drive. Only the controller itself, and the last drive connected to the controller (the ends of the “chain”), should have terminating resistor packs left on.

No Hard Disks Found

When you start hard disk setup, if you receive a “No hard drives were found connected to your computer” message, it means that AlphaBIOS could not locate a hard drive. The likely conditions that cause this error are:

- Cable not connected to either the disk or controller. The cable may have worked loose from the connector on the controller or disk drive. Check the cable connections, making sure the cable connectors are fully seated.
- No power to the drive. The power connector may have worked loose from the receptacle on the drive, or the power cable itself may be malfunctioning. Check the cable connections, making sure the cable connectors are fully seated. Try connecting the drive to a different power connector.
- Disk drive malfunction. The disk drive itself may be malfunctioning and not responding to requests from the controller. If this is the problem, then it would appear as though the disk were absent. Replace the drive with a drive known to be good. If the known good drive is correctly detected, then your hard drive is defective. Contact the manufacturer for a replacement.

No Partitions on Disk

If hard disk 0 does not have any partitions defined, then a message will appear when you start hard disk setup, asking if you want to perform an express disk setup. Express disk setup automatically creates the default disk partition arrangement on hard disk 0.

For more information on express setup, see Chapter 3.

1.11.2 System Partitions

To install Windows NT, a system partition must be defined. If no system partition is found, the user is asked to choose from the available FAT partitions.

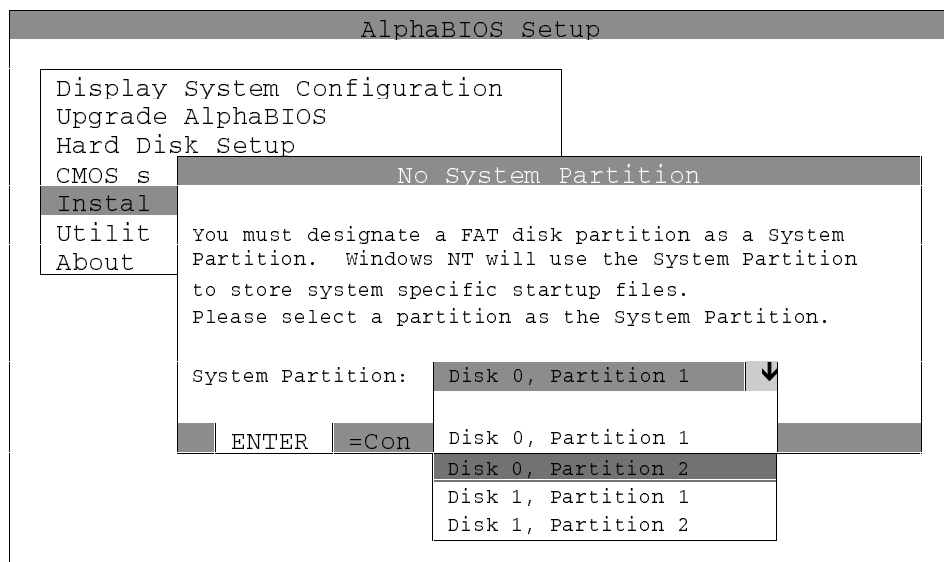
The purpose of the system partition is twofold. First, it tells the Windows NT installation program where to place the OS Loader and hardware support files. Second, upon subsequent restarts of Windows NT, the system partition definition tells AlphaBIOS where the OSLOADER.EXE file is so it can successfully hand off control to the OS Loader and continue the boot process.

The system partition can be the same partition into which Windows NT is installed, or it can be separate. However, the system partition must be formatted with the FAT file system. Because the default recommended partition arrangement on Alpha calls for the partition into which Windows NT is installed to be an NTFS partition, the system partition will almost always be a small FAT partition on the same disk onto which Windows NT is installed.

1.11.3 How AlphaBIOS Works with System Partitions

If you are installing Windows NT for the first time, AlphaBIOS will determine that a system partition has not been defined when you select **Install Windows NT** in the **AlphaBIOS Setup** screen (see Figure 1-7). When this occurs, AlphaBIOS searches for all FAT partitions on the system. If only one FAT partition exists, AlphaBIOS designates that FAT partition as the system partition and continues with the Windows NT installation. If more than one FAT partition exists on your system, AlphaBIOS displays the list of FAT partitions from which you can choose the system partition. After choosing the system partition, the installation process continues.

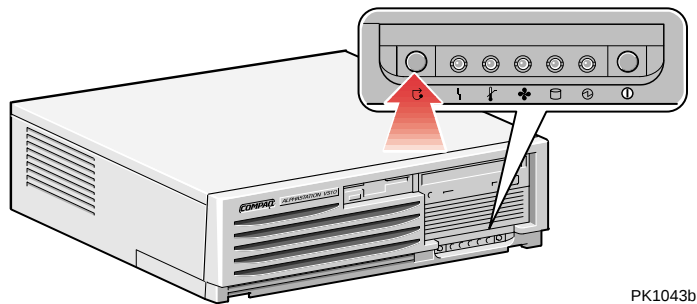
Figure 1-7 System Partition Not Defined



1.12 Using the Halt Button

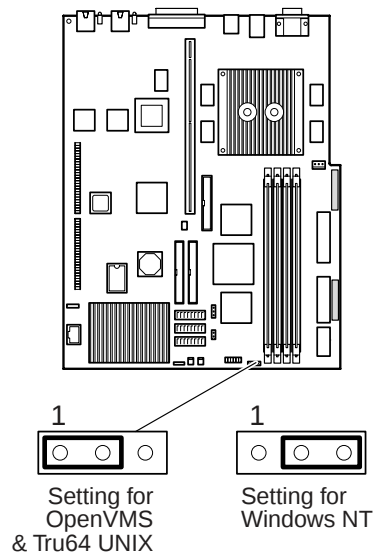
Under OpenVMS and Tru64 UNIX, the halt button pauses the operating system. Under Windows NT, the Halt button resets the system, generating power-up tests and output to the console terminal.

Figure 1-8 Halt/Reset Button



PK1043b

Figure 1-9 Halt/Reset Jumper



PK0246c

Use the Halt button to halt the Tru64 UNIX or OpenVMS operating system when it hangs, clear the SRM console password (see Section 2.9.2), or force a halt assertion (see Section 1.13). The Halt button operates like issuing an SRM **halt** command.

Windows NT does not recognize a HALT, so the jumper on the motherboard is set differently for Windows NT (see Figure 1–9). When the jumper is in the NT setting, pressing the Halt button is like issuing an SRM **initialize** command—the system goes through power-up tests (as shown in Section 1.2).

Your system is shipped with the halt/reset jumper set in the correct position for your operating system. If you are installing Windows NT after you have purchased your system, or if you are switching between Windows NT and other operating systems, you must check your jumper, and possibly have it reset. See Appendix A for resetting the jumper.

Using Halt to Shut Down the Operating System

You can use the Halt button if the Tru64 UNIX or OpenVMS operating system hangs. Pressing the Halt button halts the operating system back to the SRM console firmware. From the console, you can use the **crash** command to force a crash dump at the operating system level. See Section 2.12 for an example.

The Windows NT operating system does not support halts on this system. Pressing the Halt button during a Windows NT session will initiate a system reset.

Using Halt to Clear the Console Password

The SRM console firmware allows you to set a password to prevent unauthorized access to the console. If you forget the password, the Halt button, with the **login** command, lets you clear the password and regain control of the console. Section 2.9.2 describes the procedure.

1.13 Halt Assertion

A halt assertion allows you to disable automatic boots of the operating system so that you can perform tasks from the SRM console.

Under certain conditions, you might want to force a “halt assertion.” A halt assertion differs from a simple halt in that the SRM console “remembers” the halt. The next time you power up, the system ignores the SRM power-up script (nvram) and ignores any environment variables that you have set to cause an automatic boot of the operating system. The SRM console displays this message:

```
Halt assertion detected
NVRAM power-up script not executed
AUTO_ACTION=BOOT/RESTART and OS_TYPE=NT ignored, if applicable
```

Halt assertion is useful for disabling automatic boots of the operating system when you want to perform tasks from the SRM console. It is also useful for disabling the SRM power-up script if you have accidentally inserted a command in the script that will cause a system problem. These conditions are described in the sections “Disabling Autoboot” and “Disabling the SRM Power-Up Script.”

You can force a halt assertion using the Halt button, the RCM **halt** command, or the RMC **haltin** command. Observe the following guidelines for forcing a halt assertion.

Halt Assertion with Halt Button or RMC Halt Command

Press the Halt button on the local system (or enter the RMC **halt** command from a remote system) approximately five seconds after starting power up or when the SRM console is running. The system halts at the SRM console, and the halt status is saved. The next time the system powers up, the saved halt status is checked.

NOTE: *Wait 5 seconds after the system begins powering up before pressing the Halt button or remotely entering the RMC **halt** command. Press the button for several seconds.*

Halt Assertion with RMC Haltin Command

Enter the RMC **haltin** command at any time except during power-up. For example, enter **haltin** during an operating system session or when the AlphaBIOS console is running.

If you enter the RMC **haltin** command during a Tru64 UNIX or OpenVMS session, the system halts back to the SRM console, and the halt status is saved. The next time the system powers up, the saved halt status is checked.

If you enter the RMC **haltin** command when Windows NT or AlphaBIOS is running, the interrupt is ignored. However, you can enter the RMC **haltin** command followed by the RMC **reset** command to force a halt assertion. Upon reset, the system powers up to the SRM console, but the SRM console does not load the AlphaBIOS console.

Clearing a Halt Assertion

Clear a halt assertion as follows:

- If the halt assertion was caused by pressing the Halt button or remotely entering the RMC **halt** command, the console uses the halt assertion once, then clears it.
- If entering the RMC **haltin** command caused the halt assertion, enter the RMC **haltout** command or cycle power on the local system.

Disabling Autoboot

The system automatically boots the selected operating system at power-up or reset if the following environment variables are set:

- For Tru64 UNIX and OpenVMS, the SRM environment variables **os_type**, **auto_action**, **bootdef_dev**, **boot_file**, and **boot_osflags**
- For Windows NT, the SRM **os_type** environment variable and the **Auto Start** selection in the AlphaBIOS **Standard CMOS Setup** screen

You might want to prevent the system from autobooting so you can perform tasks from the SRM console. Use one of the methods described previously to force a halt assertion. When the SRM console prompt is displayed, you can enter commands to configure or test the system. Chapter 2 describes the SRM console commands and environment variables.

Disabling the SRM Power-Up Script

The system has a power-up script (file) named “nvram” that runs every time the system powers up. If you accidentally insert a command in the script that will cause a system problem, disable the script by using one of the methods described previously to force a halt assertion. When the SRM console prompt is displayed, edit the script to delete the offending command. See Section 2.4 for more information on editing the nvram script.

Chapter 2

SRM Console

The SRM console is the command-line interface that supports the Tru64 UNIX and OpenVMS operating systems. (Windows NT uses the SRM **test** command, only.) The SRM console is used to bootstrap the operating system, configure and test the system hardware, examine system options for errors, and set or change environment variables.

This chapter describes the SRM commands and environment variables. Sections in this chapter are:

- Invoking the SRM Console
- Commands
- Show Commands
- Creating a Power-Up Script
- Booting the Operating System
- Configuring a PCI NVRAM Module
- Testing the System
- Set Commands
- Secure Mode
- Stopping and Starting CPU
- Updating Firmware
- Forcing a System Crash Dump
- Using Environment Variables
- Depositing and Examining Data
- Reading a File
- Initializing the System
- Finding Help
- Switching from SRM to AlphaBIOS Console
- Environment Variable Summary

2.1 Invoking the SRM Console

When a system is powered up, the SRM console runs and either remains running or passes control to another console or an operating system. If the system is already running, invoke the SRM console by shutting down the operating system or by pressing the Halt button on the control panel.

If you are running Tru64 UNIX or OpenVMS Alpha operating system

To invoke the SRM console, do one of the following steps:

- Shut down the operating system according to the procedure described in your operating system documentation.
- or--
- Press the Halt button on the control panel, if your system is running either OpenVMS or Tru64 UNIX.

Following one of these steps, the console prompt, >>>, will be displayed. You are now at the SRM console.

If you are running AlphaBIOS

To switch to the SRM console, do one of the following steps:

- Select either **OpenVMS console (SRM)** or **Tru64 UNIX console (SRM)** from the **Console Selection** in the **Advanced CMOS Setup** screen. Save the setting by pressing the F10 key. Push the system Halt/Reset button to start the SRM console.
- or--
- Press the Halt/Reset button, wait 5 seconds, and then press the Halt button. The SRM console will boot.

After you have performed tasks in the console mode, you must boot the operating system with the **boot** command to go back to the operating mode.

2.2 Commands

This section presents a command summary (Table 2-1), Gives the syntax for the console commands (Table 2-2), and explains the special keystrokes and characters available in SRM console mode (Table 2-3).

2.2.1 Command Summary

Table 2-1 Summary of SRM Console Commands

Command	Function
alphabios	Loads and starts the AlphaBIOS console.
boot	Loads and starts the operating system.
clear <i>envar</i>	Resets an environment variable to its default value.
clear password	Sets the password to zero.
continue	Resumes program execution.
crash	Forces a crash dump at the operating system level.
deposit	Writes data to the specified address.
edit	Invokes the console line editor on a RAM file or on the nvram file (power-up script).
examine	Displays the contents of a memory location, register, or device.
halt	Halts the specified processor. (Same as stop .)
help	Displays information about the specified console command.
initialize	Resets the system to a known state.
lfu	Runs the Loadable Firmware Update Utility.
login	Turns off secure mode, enabling access to all SRM console commands during the current session.

Continued on next page.

Table 2-1 Summary of SRM Console Commands (Continued)

Command	Function
more	Displays a file one screen at a time.
prcache	Utility that initializes and displays status of the optional PCI NVRAM device.
set <i>envvar</i>	Sets or modifies the value of an environment variable.
set host	Connects to an MSCP DUP server on a DSSI device.
set password	Sets the console password for the first time or changes an existing password.
set secure	Enables secure mode without requiring a restart of the console.
show <i>envvar</i>	Displays the state of the specified environment variable.
show config	Displays the configuration at the last system initialization.
show cpu	Displays the state of each processor in the system.
show device	Displays a list of controllers and their devices in the system.
show memory	Displays memory module information.
show pal	Displays the version of the privileged architecture library code (PALcode).
show power	Displays information about the power supply, system and PCI fans, CPU fan, and temperature.
show version	Displays the version of the console program.
stop	Halts the processor. (Same as halt .)
test	Runs firmware diagnostics for the system.

2.2.2 Commands: Syntax

Table 2-2 Syntax for SRM Console Commands

Parameter	Attribute or Action
Length	Up to 255 characters, not including the terminating carriage return or any characters deleted as the command is entered. A command longer than 80 characters and without the backslash character (see Table 2-3) causes display of an error message.
Case	Upper- or lowercase characters can be used for input. Characters are displayed in the case in which they are entered.
Abbreviation	Only by dropping characters from the end of words. You must enter the minimum number of characters to identify the keyword unambiguously. Abbreviation of environment variables is allowed with the show command.
Options	You can use command options, to modify the environment, after the command keyword or after any symbol or number in the command. See individual command descriptions for examples.
Numbers	Most numbers in console commands are in decimal notation. Two exceptions, both of which use hexadecimal notation, are addresses and numbers used in the deposit command. The default radix can be overridden by inserting %d before numbers you want to express in decimal, %b before binary, %o before octal, or %x before hexadecimal. Register names (for example, R0) are not considered numbers and use decimal notation.
No characters	A command line with no characters is a null command. The console program takes no action and does not issue an error message; it returns the console prompt. The console supports command line recall and editing.
Spaces or tabs	Multiple adjacent spaces and tabs are compressed and treated as a single space. The console program ignores leading and trailing spaces.

2.2.3 Commands: Special Keystrokes and Characters

Table 2-3 Special Characters for SRM Console

Character	Function
Return or Enter	Terminates a command line. No action is taken on a command until it is terminated. If no characters are entered and this key is pressed, the console just redisplay the prompt.
Backslash \	Continues a command on the next line. Must be the last character on the line to be continued.
←Backspace	Moves the cursor left one position, deleting one character.
Delete	Deletes the previous character.
Help	Entered at the console prompt without arguments, it displays first-level help. When pressed after part of a command, displays options available.
Ctrl/A or F14	Toggles between insert and overstrike modes. The default is overstrike.
Ctrl/B or ↑	Recalls previous command or commands. The last 16 commands are stored in the recall buffer.
Ctrl/C	Terminates the process that is running. Clears Ctrl/S; resumes output suspended by Ctrl/O. When entered as part of a command line, deletes the current line. Ctrl/C has no effect as part of a binary data stream.
Ctrl/E	Moves the cursor to the end of the line.
Ctrl/F or →	Moves the cursor right one position.
Ctrl/H	Moves the cursor to the beginning of the line.
Ctrl/J	Deletes the previous word.

Continued next page.

Character	Function
Ctrl/O	Stops output to the console terminal for the current command. Toggles between enable and disable. The output can be reenabled by other means as well: when the console prompts for a command, issues an error message, or enters program mode, or when Ctrl/P is entered.
Ctrl/P	Halts the machine.
Ctrl/Q	Resumes output to the console terminal that was suspended by Ctrl/S.
Ctrl/R	Redisplays the current line. Deleted characters are omitted. This command is useful for hardcopy terminals.
Ctrl/S	Suspends output to the console terminal until Ctrl/Q is entered. Cleared by Ctrl/C.
Ctrl/U	Deletes the current line.
*	Wildcarding for commands such as show .
" . . . "	Double quotes enable you to denote a string for environment variable assignment.
#	Specifies that all text between it and the end of the line is a comment. Control characters are not considered part of a comment.

2.3 Show Commands

Several commands are used to display the system configuration: **show config**, **show cpu**, **show device**, **show memory**, **show network**, **show pal**, **show power**, and **show version**.

2.3.1 Show Config

The **show config** command displays a list of devices found on the system interconnect and I/O buses. This is the configuration at the most recent initialization.

Syntax: **show config**

Example 2-1 Show Config Command

```
>>> show config
AlphaServer DS10 466 MHz

SRM Console:  V5X-XXX
PALcode:      OpenVMS PALcode V1.56-45, Tru64 UNIX PALcode V1.50-48
Processors
CPU 0         Alpha 21264-4 466 MHz   SROM Revision: V1.8.208
              Bcache size: 2 MB

Core Logic
Cchip         DECchip 21272-CA Rev 2
Dchip         DECchip 21272-DA Rev 2
Pchip 0       DECchip 21272-EA Rev 2

TIG           Rev 2.1
Arbiter       Rev 3.30 (0x7E)

MEMORY
Array #       Size      Base Addr
-----
0             256 MB     000000000

Total Bad Pages = 0
Total Good Memory = 256 MBytes

PCI Hose 00
  Bus 00 Slot 01: Acer Labs M1543C USB
  Bus 00 Slot 03: Acer Labs M1543C PMU
  Bus 00 Slot 07: Acer Labs M1543C
  Bus 00 Slot 09: DE500-BA Network Controller      ewa0.0.0.9.0
                                                    Bridge to Bus 1, ISA
                                                    08-00-2B-86-1B-BA
```

```

Bus 00 Slot 11: DE500-BA Network Controller
                  ewb0.0.0.11.0      08-00-2B-86-1B-BB
Bus 00 Slot 13: Acer Labs M1543C IDE
                  dqa.0.0.13.0
                  dqa1.1.0.13.0      CD-532E
Bus 00 Slot 14: DEC PowerStorm
Bus 00 Slot 15: DECchip 21152-AA
                  Bridge to Bus 2, PCI
Bus 00 Slot 16: QLogic ISP1020
                  pkc0.6.0.16.0      SCSI Bus ID 6
                  dkc0.0.0.16.0      RZ1CB-BA
                  dkc200.2.0.16.0     RZ1CB-BA
Bus 00 Slot 17: Mylex DAC960
                  dra.0.0.17.0
                  dra0.0.0.17.0      1 Member JBOD
                  dra1.0.0.17.0      1 Member JBOD
                  dra2.0.0.17.0      1 Member JBOD
                  dra3.0.0.17.0      1 Member JBOD
Bus 02 Slot 00: NCR 53C875
                  pka0.7.0.2000.0     SCSI Bus ID 7
                  dka100.1.0.2000.0   RZ1CB-CA
                  dka300.3.0.2000.0   RZ1CB-CA
                  dka500.5.0.2000.0   RZ1EF-AB
Bus 02 Slot 01: NCR 53C875
                  pkb0.7.0.2001.0     SCSI Bus ID 7
                  dkb0.0.0.2001.0     RZ1CB-CA
                  dkb200.2.0.2001.0   RZ1DB-CS
                  dkb400.4.0.2001.0   RZ1CB-CA
Bus 02 Slot 02: DE500-AA Network Controller
                  ewc0.0.0.2002.0      00-06-2B-00-26-1C

```

ISA Slot	Device	Name	Type	Enabled	BaseAddr	IRQ	DMA
0	0	MOUSE	Embedded	Yes	60	12	
	1	KBD	Embedded	Yes	60	1	
	2	COM1	Embedded	Yes	3f8	4	
	3	COM2	Embedded	Yes	2f8	3	
	4	LPT1	Embedded	Yes	3bc	7	
	5	FLOPPY	Embedded	Yes	3f0	6	2

>>>

2.3.2 Show Device

The `show device` command displays status for devices and controllers in the system: SCSI and MSCP devices, the internal floppy drive, and the network.

Syntax: `show device (controller_name)`

controller_name The controller name or abbreviation. When abbreviations or wildcards are used, all controllers that match the type are displayed. If no name is given, the display is a list of all devices and controllers in the system.

Example 2-2 Show Device Command

```
>>> sho device
dka0.0.0.14.0          DKA0          RZ2CA-KA  N1H1
dka100.1.0.14.0        DKA100         RZ2CA-KA  N1H1
dqa0.0.0.13.0          DQA0          CD-532E-  A  1.0A
dva0.0.0.0.0           DVA0
ewa0.0.0.9.0           EWA0          08-00-2B-86-1F-71
ewb0.0.0.11.0          EWB0          08-00-2B-86-1F-70
pka0.7.0.14.0          PKA0          SCSI Bus ID 7  5.57
>>>
```

An example of a device name is **dka200.2.0.7.1**. Table 2-4 shows the interpretation of this device name.

Table 2-4 Device Naming Convention

Category	Description												
dk Driver ID	Two-letter designator of port or class driver												
	<table> <tr> <td>dk SCSI disk</td><td>fw FDDI device</td></tr> <tr> <td>dq ATAPI CD-ROM</td><td>mk SCSI tape</td></tr> <tr> <td>dr RAID set device</td><td>mu DSSI tape</td></tr> <tr> <td>du DSSI disk</td><td>pk SCSI port</td></tr> <tr> <td>dv Diskette drive</td><td>pu DSSI port</td></tr> <tr> <td>ew Ethernet port</td><td></td></tr> </table>	dk SCSI disk	fw FDDI device	dq ATAPI CD-ROM	mk SCSI tape	dr RAID set device	mu DSSI tape	du DSSI disk	pk SCSI port	dv Diskette drive	pu DSSI port	ew Ethernet port	
dk SCSI disk	fw FDDI device												
dq ATAPI CD-ROM	mk SCSI tape												
dr RAID set device	mu DSSI tape												
du DSSI disk	pk SCSI port												
dv Diskette drive	pu DSSI port												
ew Ethernet port													
a Storage adapter ID	One-letter designator of storage adapter (a, b, c...).												
200 Device unit number	Unique number (MSCP unit number). SCSI unit numbers are forced to 100 X node ID.												
2 Bus node number	Bus node ID.												
0 Channel number	Used for multi-channel devices.												
7 Logical slot number	Corresponds to PCI slot number, as shown in Table 2-5.												
1 Hose number	0 — PCI 0 1 — PCI 1												

Table 2-5 PCI Address Assignments

Physical Slot #	Logical Slot #	PCI ID_SEL	Description/Position
3	17	AD 28	64-bit slot (top slot)
2	16	AD 27	64-bit slot (second from top)
1	15	AD 26	64-bit slot (second from bottom)
0	14	AD 25	32-bit slot (bottom slot)

2.3.3 Show Memory

The **show memory** command displays information about each memory bank: slot number, size in megabytes, and the starting address.

Syntax: **show memory**

Example 2-3 Show Memory Command

```
>>> show memory
```

Array #	Size	Base Addr
-----	-----	-----
0	128 MB	000000000
1	128 MB	008000000
2	128 MB	010000000
3	128 MB	018000000

```
Total Bad Pages = 0
Total Good Memory = 512 MBytes
>>>
```

2.3.4 Show PAL

The **show pal** command displays the versions of Tru64 UNIX and OpenVMS PALcode. PALcode is the Alpha Privileged Architecture Library code, written to support Alpha processors. It implements architecturally defined processor behavior.

Syntax: **show pal**

Example 2-4 Show PAL Command

```
>>> show pal
pal  OpenVMS PALcode V1.56-45, Tru64 UNIX PALcode V1.50-48
>>>
```

2.3.5 Show Power

The show power command displays status information about the power supply, the system, PCI and CPU fans, and temperature. This command is useful for displaying the error state of a Tru64 UNIX or OpenVMS system that shuts down because of a fan, temperature, or power supply failure.

Syntax: **show power**

If the system can be restarted, use this command. (If it cannot, use the RMC **status** command. See Chapter 4.)

Example 2-5 Show Power Command

```
>>> show power
Power Supply      Status
System Fan       good
PCI Fan          good
CPU Fan          good
Temperature       good

Current ambient temperature is 34 degrees C
System shutdown temperature is set to 55 degrees C

2 Environmental events are logged in nvram
Do you want to view the events? (Y/<N>) y

Total Environmental Events: 2 (2 logged)

1 000 0 0:00 Temperature, Fans, Power Supplies Normal
2 000 0 0:00 Temperature, Fans, Power Supplies Normal

Do you want to clear all events from nvram? (Y/<N>) y

>>>
```

2.3.6 Show Version

The **show version** command displays the version of the SRM console program that is installed on the system.

Syntax: **show version**

Example 2-6 Show Version Command

```
>>> show version
version                V5.4-2 MAY 19 1999 14:24:16
>>>
```

2.4 Creating a Power-Up Script

A special nonvolatile file named “nvram” is stored in EEROM, and is always invoked during the power-up sequence. You can create an nvram script to include any commands you want the system to execute at power-up. You alter, create and edit the nvram script using the SRM **edit** command. With **edit**, lines can be added, overwritten, or deleted.

Syntax: **edit** *file*

where *file* is the name of the file to be edited.

The editing commands are:

Help	Displays the brief help file.
List	Lists the current file prefixed with line numbers.
Renumber	Renumbers the lines of the file in increments of 10.
Exit	Leaves the editor and closes the file, saving all changes.
Quit	Leaves the editor and closes the file without saving changes.
Nn	Deletes line number nn.
Nn text	Adds or overwrites line number nn with text.

NOTE: *It is possible to disable the system by editing the nvram script. For example, if you include the **initialize command** in the script, the system will go into an endless loop. To fix this, press the **Halt** button while the system is powering up. You can then edit the script to delete the offending command.*

Example 2-7 shows how to modify the user-created power-up script, “nvram.” The pound sign (#) indicates explanatory comments. In this example the script is edited to include a command that allows you to boot the Tru64 UNIX operating system over the network.

Example 2-7 Editing the nvram Script

```
>>> edit nvram                                #Modify user power-up script, nvram.
editing 'nvram'
0 bytes read in
*10 set ewa0_protocols bootp
*list                                          #List current file with line numbers.
10 set ewa0_protocols bootp
*exit                                         #Close file and save changes.
27 bytes written out to nvram
>>> nvram                                     #Execute the script.
```

To clear the script, enter line numbers without any text. This deletes the lines. See Example 2-8.

Example 2-8 Clearing the nvram Script

```
>>> edit nvram
editing 'nvram'
20 bytes read in
*10
*exit
0 bytes written out to nvram
>>>
```

2.5 Booting the Operating System

The **boot** command is used to boot the operating system. The **boot** command initializes the processor, loads a program image from the specified boot device, and transfers control to that image.

Syntax: **boot** (-file *filename*) (-flags (*value*)) (-halt)
(-protocols *enet_protocol*) (*boot_dev*)

-file <i>filename</i>	The boot file.
-flags [<i>value</i>]	Specifies additional information to the loaded image or operating system. In Tru64 UNIX, specifies boot flags. In OpenVMS, specifies system root number and boot flags. This qualifier overrides the setting of the boot_osflags environment variable. See the boot_osflags environment variable for a list of settings and their meanings.
-halt	Forces the bootstrap operation to halt and invoke the console program once the bootstrap image is loaded and page tables and other data structures are set up. Console device drivers are not shut down. Transfer control to the image by entering the continue command.
-protocols <i>enet_protocol</i>	Either mop (default) or bootp . This qualifier overrides the setting of the ew*0_protocols environment variable.
<i>boot_dev</i>	A device path or list of devices from which the console program attempts to boot, or a saved boot specification in the form of an environment variable. This qualifier overrides the setting of the bootdef_dev environment variable. Use the bootdef_dev environment variable to define the default boot device string.

Example 2-9 Boot Command

```
>>> b dkc0
/boot dkc0.0.0.16.0 -flags 0)
block 0 of dkc0.0.0.16.0 is a valid boot block
reading 898 blocks from dkc0.0.0.16.0
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 70400
initializing HWRPB at 2000
initializing page table at ffb6000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

      OpenVMS (TM) Alpha Operating System, Version V7.1-2

%SYSINIT-I- waiting to form or join an OpenVMS Cluster
%VMSCluster-I-LOADSECDB, loading the cluster security database
%EWA0, Fast(100baseTX) mode set by console
%EWB0, Fast(100baseTX) mode set by console

      .
      .
      .
%CNXMAN, Completing VMSCluster state transition
$! Copyright (c) 1998 Digital Equipment Corporation. All rights reserved.
%STDRV-I-STARTUP, OpenVMS startup begun at 23-APR-1999 11:31:48.97
%MSCPLOAD-I-CONFIGSCAN, enabled automatic disk serving

      .
      .
      .
The OpenVMS system is now executing the site-specific startup commands.

      .
      .
      .
Welcome to OpenVMS (TM) Alpha Operating System, Version V7.1-2

Username:
```

2.6 Configuring a PCI NVRAM Module

The **prcache** command is used for system configuration to check PCI NVRAM configuration information and battery status, to clear data from the NVRAM module, and to set the date of the next battery replacement. The command is used only with Tru64 UNIX systems.

Syntax: **prcache -{f,z,b}**

- f** Checks configuration information and battery status.
- z** Clears valid data; writes zeros to memory.
- b** Sets the date (month and year) for battery replacement.

Example 2-10 Prcache Command

```
>>> prcache -f
PCI NVRAM Disk Cache: passed
Size: 2MB
PCI Memory Address: 40000000
System ID: 12000000
State: - not valid
Battery Status: good (Charging)
Battery Disconnect Circuit Status: enabled

>>> prcache -z
This command will zero the PCI NVRAM Disk Cache
Do you really want to continue [Y/N] ? : y
clearing disk cache
>>>
```

2.7 Testing the System

The `test` command runs console-based exercisers for devices in the system. Ctrl/C can abort testing.

Syntax: **test** (**-t** *time*) (**-q**) (*option*)

- t** *time* Specifies the run time in seconds. The default for system test is 120 seconds (2 minutes).
 - q** Disables the display of status messages as exerciser processes are started and stopped during testing. Sets the environment variable **d_verbose** to zero.
-

NOTE: *If you are running the Microsoft Windows NT operating system, switch from AlphaBIOS to the SRM console in order to enter the **test** command. Select **Tru64 UNIX Console (SRM)** or **OpenVMS Console (SRM)** from the **Advanced CMOS Setup** of the AlphaBIOS console screen, save the setting by pressing the F10 key, and then reset the system.*

Example 2-11 Test Command

```
>>> test
environment variable t_cnt created
System test, runtime 150 seconds

Type ^C if you wish to abort testing once it has started

Default zone extended at the expense of memzone.
Use INIT before booting

Testing Ethernet device(s)

Testing VGA

Testing Memory

Testing IDE/ATAPI disks (read-only)

Testing SCSI disks (read-only)
Testing DKA* devices (read-only)..
No other SCSI disks to test

Testing floppy drive (dva0, read-only)
No diskette present, skipping floppy test
```

ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
00001ae5 310378496	memtest	memory	4	0 0	310378496	
00001aea 306184192	memtest	memory	4	0 0	306184192	
00001b07 310378496	memtest	memory	3	0 0	310378496	
00001b54 215040	exer_kid	dqa0.0.0.13.	0	0 0	0	
00001b74 8732672	exer_kid	dka0.0.0.14.	0	0 0	0	
00001b7b 8732672	exer_kid	dka100.1.0.1	0	0 0	0	
ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
00001ae5 549453824	memtest	memory	6	0 0	549453824	
00001aea 545259520	memtest	memory	6	0 0	545259520	
00001b07 545259520	memtest	memory	6	0 0	545259520	
00001b54 421888	exer_kid	dqa0.0.0.13.	0	0 0	0	
00001b74 14434304	exer_kid	dka0.0.0.14.	0	0 0	0	
00001b7b 14434304	exer_kid	dka100.1.0.1	0	0 0	0	
ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
00001ae5 780140544	memtest	memory	8	0 0	780140544	
00001aea 780140544	memtest	memory	8	0 0	780140544	
00001b07 780140544	memtest	memory	8	0 0	780140544	
00001b54 631808	exer_kid	dqa0.0.0.13.	0	0 0	0	
00001b74 20168704	exer_kid	dka0.0.0.14.	0	0 0	0	
00001b7b 20168704	exer_kid	dka100.1.0.1	0	0 0	0	
ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
00001ae5 1019215872	memtest	memory	11	0 0	1019215872	
00001aea 1019215872	memtest	memory	10	0 0	1019215872	
00001b07 1019215872	memtest	memory	10	0 0	1019215872	
00001b54 835584	exer_kid	dqa0.0.0.13.	0	0 0	0	
00001b74 27066368	exer_kid	dka0.0.0.14.	0	0 0	0	

```

00001b7b      exer_kid dka100.1.0.1      0    0    0          0
27066368
  ID      Program      Device      Pass  Hard/Soft Bytes Written  Bytes Read
-----
00001ae5      memtest memory      13    0    0      1254096896
1254096896
00001aea      memtest memory      13    0    0      1249902592
1249902592
00001b07      memtest memory      12    0    0      1254096896
1254096896
00001b54      exer_kid dqa0.0.0.13.    0    0    0          0
1043456
00001b74      exer_kid dka0.0.0.14.    0    0    0          0
32800768
00001b7b      exer_kid dka100.1.0.1    0    0    0          0
32784384
  ID      Program      Device      Pass  Hard/Soft Bytes Written  Bytes Read
-----
00001ae5      memtest memory      15    0    0      1488977920
1488977920
00001aea      memtest memory      15    0    0      1484783616
1484783616
00001b07      memtest memory      14    0    0      1488977920
1488977920
00001b54      exer_kid dqa0.0.0.13.    0    0    0          0
1241088
00001b74      exer_kid dka0.0.0.14.    0    0    0          0
39665664
00001b7b      exer_kid dka100.1.0.1    0    0    0          0
39649280
  ID      Program      Device      Pass  Hard/Soft Bytes Written  Bytes Read
-----
00001ae5      memtest memory      17    0    0      1723858944
1723858944
00001aea      memtest memory      17    0    0      1723858944
1723858944
00001b07      memtest memory      16    0    0      1723858944
1723858944
00001b54      exer_kid dqa0.0.0.13.    0    0    0          0
1445888
00001b74      exer_kid dka0.0.0.14.    0    0    0          0
45350912
00001b7b      exer_kid dka100.1.0.1    0    0    0          0
45350912
Test time has expired...

System test complete

>>>

System test, runtime 1200 seconds

Type ^C if you wish to abort testing once it has started

```

```
Default zone extended at the expense of memzone.  
Use INIT before booting  
Testing Ethernet device(s)  
Testing VGA  
Testing Memory  
Testing IDE/ATAPI disks (read-only)  
Testing SCSI disks (read-only)  
Testing floppy drive (dva0, read-only)  
Test time has expired...  
System test complete  
>>>
```

2.8 Set Commands

2.8.1 Set Password

The **set password** command sets the console password for the first time or changes an existing password. It is necessary to set the password only if the system is going to operate in secure mode.

Syntax: **set password**

The password length must be between 15 and 30 alphanumeric characters.

If a password has not been set and the **set password** command is issued, the console prompts for a password and verification.

If a password has been set and the **set password** command is issued, the console prompts for the new password and verification, then prompts for the old password. The password is unchanged if the validation password entered does not match the existing password in the NVRAM.

Example 2-12 Set Password Command

```
>>> set password
Please enter the password:      # Password is not echoed.
Please enter the password again: # Validation is not echoed.
>>>

>>> set password              # Changing a password.
Please enter the password:
Please enter the password again:
Now enter the old password:
>>>

>>> set password              # Password entered is
Please enter the password:      # too short.
Password length must be between 15 and 30 characters
>>>
```

2.8.2 Set Secure

The set secure command enables secure mode without requiring a restart of the console. If the password has been set, the console will be secured and only a small subset of commands can be performed. If a password has not been set, you are prompted to do so.

Syntax: **set secure**

Example 2-13 Set Secure Command

```
>>> set secure                            # In this example a password
                                         # has been set.
Console is secure. Please login.
>>> b dkb0
Console is secure - parameters are not allowed.
>>> login
Please enter the password:                # Password is not echoed.
>>> b dkb0
(boot dkb0.0.0.3.1)
.
.

>>> set secure                            # Password has not been set.
Secure not set. Please set the password.
>>>
```

2.9 Secure Mode

When the console is in secure mode, the only commands recognized are boot, login, continue, and start. Placing the console in secure mode ensures that unauthorized persons cannot gain access to the system. The commands for console security are set password, clear password, and set secure. The login command turns off security features during the current console session.

The **boot** command does not accept command line parameters in secure mode. The console boots using the environment variables stored in NVRAM (**boot_file**, **bootdef_dev**, **boot_flags**). After a successful boot, the console is secured if there is a valid password.

The **start** and **continue** commands are valid on a secure console. After either command is executed, the console is secured if there is a valid password. This prevents an intruder from accessing the system.

2.9.1 Login Command and Secure Mode

The login command turns off the security features, enabling access to all SRM console commands during the current session. The system automatically returns to secure mode as soon as the boot, continue, or start command is entered or when the system is initialized.

When the **login** command is entered, the user is prompted for the current system password. If a password has not been set, a message is displayed indicating that there is no password in NVRAM. If a password has been set, this prompt is displayed:

Please enter the password:

If the password entered matches the password in NVRAM when the prompt is redisplayed, the console is no longer in secure mode and all console commands can be performed.

NOTE: *If you enter the **login** command when a halt assertion exists, the command fails, even if you enter the correct password. See Section 1.13 for information on halt assertion.*

Example 2-14 Secure Mode and Login Command

```
>>> login                                # System is not in secure mode.
>>>
Secure not set. Please set the password.

>>> login                                # System is in secure mode.
Please enter the password:                # Password is not echoed.
>>>

>>> login                                # System is in secure mode.
Please enter the password:                # Incorrect password is
                                         # entered.
Invalid password
>>>
```

2.9.2 Clear Password

The clear password command clears the password environment variable, setting it to zero. This command is used when you want access to all the SRM console commands, but the system is in secure mode. In order to use clear password, you must know the current password.

To clear the password without knowing the current password, use the **login** command, with the Halt button or RMC **halt** command, as described in the section on the **login** command.

Example 2-15 Clear Password Command

```
>>> clear password
Please enter the password:                # Password is not echoed.
Password successfully cleared.
>>>

>>> clear password
Please enter the password:                # Invalid password entered.
Console is secure
>>>
```

2.9.3 Resetting the Password

If you have forgotten the current password, clear the password as follows:

From the Local Console Terminal

1. Enter the **login** command: `>>> login`
2. At the Enter Password: prompt, press the Halt button, then press the Return key.
The password is now cleared and the console cannot be put into secure mode unless a new password is set.

From the RMC

1. Enter the **login** command: `>>> login`
2. At the Enter Password: prompt, enter the RMC escape sequence.
3. At the RMC>>> prompt, enter the **halt** command and then the **quit** command:

```
RMC>>> halt  
RMC>>> quit
```

4. At the SRM console, clear the password:
`>>> clear password`

2.10 Stopping and Starting CPU

The halt (or stop) command stops program execution on a CPU that is still running a booted program.

Syntax: **halt (or stop) 0**

where 0 is the number of the CPU to halt.

For the DS10 system, this command has no effect because the single CPU is halted when the system is at the console prompt.

2.11 Updating Firmware

The lfu command is used to update firmware from the SRM console prompt. The lfu command starts the Loadable Firmware Update (LFU) Utility.

The syntax is: **lfu**

Example 2-16 lfu Command

```
>>> lfu
```

```
Checking dqa0.0.0.13.0 for the option firmware files. . .
Checking dva0 for the option firmware files. . .
```

```
Option firmware files were not found on CD or floppy.
If you want to load the options firmware,
please enter the device on which the files are located(ewa0),
or just hit <return> to proceed with a standard console update: dva0
```

```
Please enter the name of the options firmware files list, or
Hit <return> to use the default filename (ds10fw.txt) :
```

```
Copying ds10fw.txt from dva0. . .
Copying PC264NT.ROM from dva0. . .
Copying DS10SRM.ROM from dva0. . .
```

```

***** Loadable Firmware Update Utility *****
-----
Function      Description
-----
Display       Displays the system's configuration table.
Exit          Done exit LFU (reset).
List          Lists the device, revision, firmware name, and update
revision.
Readme        Lists important release information.
Update        Replaces current firmware with loadable data image.
Verify        Compares loadable and hardware images.
? or Help     Scrolls this function table.
-----
UPD> update *

Confirm update on:
nt
srm
[Y/(N)]y

WARNING: updates may take several minutes to complete for each device.

                DO NOT ABORT!

nt              Updating to 5.69...  Verifying 5.69...  PASSED.
srm             Updating to 5.4-2...  Verifying 5.4-2...  PASSED.
UPD> exit

```

NOTE: *If the system has been shut down from a booted program (most commonly, the operating system) or in some other way halted back to the SRM console, the system must be reset before running LFU.*

See Chapter 1 for more information about LFU.

2.12 Forcing a System Crash Dump

The crash command forces a crash dump at the operating system level (Tru64 UNIX or OpenVMS; has no effect on Windows NT). This command is used when an error has caused the system to hang and can be halted with the Halt button or the RMC halt command. This command restarts the operating system and forces a crash dump to the selected device.

Syntax: **crash** (*device*)

where *device* is the name of the device to which the crash dump is written.

Example 2-17 Crash Command

>>> **crash**

CPU 0 restarting

DUMP: 401408 blocks available for dumping.

DUMP: 38535 required for a partial dump.

DUMP: 0x805001 is the primary swap with 401407, start our last 38534
: of dump at 362873, going to end (real end is one more, for
header)

DUMP.prom: dev SCSI 1 3 0 4 400 0 0, block 131072

DUMP: Header to 0x805001 at 401407 (0x61fff)

DUMP.prom: dev SCSI 1 3 0 4 400 0 0, block 131072

DUMP: Dump to 0x805001:: End 0x805001

DUMP.prom: dev SCSI 1 3 0 4 400 0 0, block 131072

DUMP: Header to 0x805001 at 401407 (0x61fff)

succeeded

halted CPU 0

halt code = 5

HALT instruction executed

PC = ffffffc00004e2d64

>>>

2.13 Using Environment Variables

Environment variables pass configuration information between the console and the operating system. Their settings determine how the system powers up, boots the operating system, and operates. You issue an `init` command to activate a new environment variable.

Environment variables are set or changed with the **`set envvar`** command (where *envvar* is a placeholder for the environment to be changed) and set to default values with the **`set -default envvar`** command. Their values are viewed with the **`show envvar`** command. User-defined nonvolatile environment variables are created with the **`edit`** command described in Section 2.4. Section 2.19 describes the environment variables in detail.

2.13.1 `set envvar`

The `set` command sets or modifies the value of an environment variable. It can also be used to create a new environment variable if the name used is unique. Environment variables are used to pass configuration information between the console and the operating system. The setting of these variables determines how the system powers up, boots the operating system, and operates.

Syntax: **`set (-default) envvar value`**

-
- | | |
|-----------------------------------|--|
| <code>-default</code> | Restores an environment variable to its default setting. |
| <code><i>envvar</i></code> | The name of the environment variable to be displayed. The wildcard <code>*</code> displays all environment variables, which are:
<code>console</code>, <code>kbd_hardware_type</code>, <code>language</code>, <code>ocp_text</code>
and <code>os_type</code> |
| <code><i>value</i></code> | The new value of the environment variable. |

Whenever you modify the value of any environment variables, the new value takes effect only after you reset the system by pressing the Halt/Reset button or issuing the **`initialize`** command.

2.13.2 show envvar

The `show envvar` command displays the current value (or setting) of an environment variable.

Syntax: `show envvar`

<code>envvar</code>	The name of the environment variable to be displayed. The wildcard <code>*</code> displays all environment variables, which are: console, kbd_hardware_type, language, ocp_text and os_type .
---------------------	---

Example 2–18 Setting and Showing Environment Variables

```
>>> show console
console          graphics
>>> set console serial
>>> show console
console          serial
>>> init          # The system must be init'ed for
                  the new parameters to take effect
```

Example 2–19 Creating a User-Defined Environment Variable

```
>>> edit nvram
editing 'nvram'
0 bytes read in
*10 set mopv3_boot 1
*exit
17 bytes written out to nvram
>>>
```

In Example 2–19 the nvram script is edited so that an environment variable called “mop3_boot” is created and set to 1 on each power-up. By default, MOP boot sends four MOP V4 requests before defaulting to MOP V3. This user-created environment variable forces the SRM console to bypass MOP V4 requests. This speeds up MOP booting on networks with MOP V3 software.

2.14 Depositing and Examining Data

Example 2-20 Deposit Command

```
>>> dep -b -n 1ff pmem:0 0      # Clear 1st 512 bytes of physical mem.

>>> d -l -n 3 vmem:1234 5      # Deposit 5 into four longwords starting
                                # at virtual memory address 1234.

>>> d -n 8 r0 ffffffff          # Load GPRs R0 through R8 with -1.

>>> d -l -n 10 -s 200 pmem:0 8  # Deposit 8 in the 1st longword of the
                                # first 17 pages in physical memory

>>> d -l pmem:0 0              # Deposit 0 to physical mem address 0.

>>> d + ff                     # Deposit FF to physical mem address 4.

>>> d scbb 820000              # Deposit 820000 to SCBB.
```

Example 2-21 Examine Command

```
>>> examine pc                  # Examine the program counter.
PC psr: 0 ( PC) 0000000000001170

>>> examine sp                  # Examine the stack pointer.
gpr: F0 ( R30) 0000000000072A60

>>> e -n 6 r4                  # Examine register R4 & the next 6 registers.
grp: 20 ( R4) 0000000000005000
grp: 28 ( R5) 00000000FFFE000
grp: 30 ( R6) 00000003F8000C00
grp: 38 ( R7) 0000000053F761AE
grp: 40 ( R8) 0000010000000000
grp: 48 ( R9) 00000003F7800100
grp: 50 ( R10) 00000000000C7FFC

>>> examine pmem:400EC          # Examine physical memory.
pmem: 400EC A49D0078A47D0070
```

2.15 Reading a File

The `more` command displays a file one screen at a time.

Syntax: **more** (*file...*)

where *file* is the name of the file to be displayed.

The **more** command is similar to that used in the MS-DOS and UNIX operating systems. It is useful for displaying output that scrolls too quickly to be viewed. For example, when you power up the system, the system startup messages scroll, and the messages are logged to an event log

When the >>> prompt displays, you can use the **more** command to display the contents of the event log file.

Example 2-22 More Command

```
>>> more e1                    # Display SRM console's event log one screen at a time.
```

```
. . .  
--More-- (SPACE - next page, ENTER - next line, Q - quit)
```

```
>>> help * | more            # Display online help one screen at a time.
```

```
. . .  
--More-- (SPACE - next page, ENTER - next line, Q - quit)
```

2.16 Initializing the System

The initialize command resets the system and executes power-up tests.

Syntax: **initialize**

After self-tests are executed, the system autoboots unless:

- A halt assertion condition exists (see Section 1.13) –or–
- The **auto_action** environment variable is set to **halt** (see Section 2.19.1).

If the **auto_action** environment variable is set to **boot** or **restart** and no halt assertion condition exists, the system autoboots. In all other cases, the system stops in console mode and does not attempt to boot.

Example 2-23 Initialize Command

```
>>> init
Initializing...
256 Meg of system memory
probing hose 0, PCI
probing PCI-to-ISA bridge, bus 1
probing PCI-to-PCI bridge, bus 2
bus 0, slot 9 -- ewa -- DE500-BA Network Controller
bus 0, slot 11 -- ewb -- DE500-BA Network Controller
bus 0, slot 13 -- dqa -- Acer Labs M1543C IDE
bus 0, slot 13 -- dqb -- Acer Labs M1543C IDE
bus 0, slot 14 -- vga -- DEC PowerStorm
bus 2, slot 0 -- pka -- NCR 53C875
bus 2, slot 1 -- pkb -- NCR 53C875
bus 2, slot 2 -- ewc -- DE500-AA Network Controller
bus 0, slot 16 -- pkc -- QLogic ISP1020
bus 0, slot 17 -- dra -- Mylex DAC960
Testing the System
Testing the Disks (read only)
Testing the Network
System Temperature is 34 degrees C
CPU 0            Alpha 21264-4 466 MHZ    SR0M Revision: V1.8.208
TIG             Rev 2.1
Arbiter         Rev 3.30 (0x7E)

Array #            Size            Base Addr
-----
  0            512 MB            000000000
  1            512 MB            020000000
  2            128 MB            050000000
  3            256 MB            040000000
Total Bad Pages = 0
Total Good Memory = 1408 MBytes
AlphaServer DS10 466 MHZ Console V5.4-2 MAY 19 1999 14:24:16
>>>
```

2.17 Finding Help

The **help** command displays basic information about the use of console commands when the system is in console mode.

Syntax: **help** (*command...*)

Command... Command or topic for which help is requested. The options are:

- | | |
|---|--|
| <i>None</i> | Displays the complete list of commands for which you can receive help. |
| <i>Command_name</i> | Displays information about the console command. |
| <i>Argument_string</i>
(such as “sh”) | Displays information about all commands that begin with that string. |

Example 2-24 Help Command

```
>>> help set
NAME
    set
FUNCTION
    Set an option or modify the value of an environment
    variable.
SYNOPSIS
    set <option> <value> or <envvar> [-] <value>
      where
      <option>={host,mode}
      where
      <envvar>={auto_action,bootdef_dev,boot_osflags,...}
      [-default]
```

2.18 Switching from SRM to AlphaBIOS Console

The **alphabios** command loads and starts the AlphaBIOS console. This is necessary for running AlphaBIOS-based utilities (such as the RAID configuration utility). To switch from SRM to AlphaBIOS, issue the **alphabios** command.

Syntax: **alphabios**

To return to the SRM console:

- If the **os_type** environment variable is set to **unix** or **openvms**, reset the system by pressing the Halt/Reset button,--or--
- If the **os_type** environment variable is set to **nt**, select **Tru64 UNIX console (SRM)** or **OpenVMS console (SRM)** from the **Advanced CMOS Setup** screen, save the setting by pressing key F10 and then reset the system.

To switch permanently to the AlphaBIOS console:

If the permanent operating system is changed to Windows NT, you will want to switch permanently to the AlphaBIOS console. To accomplish this:

1. Set the SRM **os_type** environment variable to **nt**.
2. Set the SRM **console** environment variable to **graphics**.
3. Reset the system with the **initialize** command, the Halt/Reset button, or by power cycling.

Example 2-25 Switching to the AlphaBIOS Console

```
>>> alphabios
```

2.19 Environment Variable Summary

Environment variables pass configuration information between the console and the operating system. Their settings determine how the system powers up, boots the operating system, and operates. Environment variables are set or changed with the `set envvar` command and returned to their default values with the `clear envvar` command. Their values are viewed with the `show envvar` command.

Table 2–6 lists the environment variables. Detailed descriptions follow. The environment variables are specific to the SRM console.

Table 2–6 Environment Variable Summary

Environment Variable	Function
auto_action	Specifies the console's action at power-up, a failure, or a reset.
bootdef_dev	Specifies the default boot device string.
boot_osflags	Specifies the default operating system boot flags.
com*_baud	Changes the default baud rate of the COM1 or COM2 serial port.
console	Specifies the device on which power-up output is displayed (serial terminal or graphics monitor).
cpu_enabled	Enables or disables a specific secondary CPU.
ew*0_mode	Specifies the connection type of the default Ethernet controller.
ew*0_protocols	Specifies network protocols for booting over the Ethernet controller.
kbd_hardware_type	Specifies the default console keyboard type.
language	Specifies the console keyboard layout.

Table 2-6 Environment Variable Summary (Continued)

Environment Variable	Function
ocp_text	Overrides the default OCP display text with specified text.
os_type	Specifies the operating system. Valid entries are: openvms, unix, and nt.
password	A password stored in the NVRAM used to secure the console.
pci_parity	Disables or enables parity checking on the PCI bus.
pk*0_fast	Enables fast SCSI mode.
pk*0_host_id	Specifies the default value for a controller host bus node ID.
pk*0_soft_term	Enables or disables SCSI terminators on systems that use the QLogic ISP1040 SCSI controller.
tt_allow_login	Enables or disables login to the SRM console firmware on other console ports.

2.19.1 auto_action

Specifies the action the console takes any time the system powers up, fails, or resets. When the setting involves autoboot, the system boots from the default boot device specified by the value of the `bootdef_dev` environment variable.

Syntax: **set auto_action** *value*

where *value* can be:

-
- | | |
|----------------|---|
| halt | The system remains in console mode after power-up or a system crash. |
| boot | The system boots automatically when it is turned on and halts after a system failure. |
| restart | The system boots automatically when it is turned on or after it fails. |
-

NOTE: *If a halt assertion exists, the console ignores the auto_action setting and halts at the SRM console. See Section 1.13 for information on halt assertion.*

2.19.2 bootdef_dev

The `bootdef_dev` environment variable specifies one or more devices for booting the operating system. When more than one device is listed, the system searches in the order listed and boots from the first device with operating system software.

Syntax: **set bootdef_dev** *boot_device*

Where *boot_device* is the name of the device on which the system software has been loaded. To specify more than one device, separate the names with commas. Enter the command **show bootdef_dev** to display the current default boot device. Enter the command **show device** for a list of all devices in the system.

2.19.3 boot_osflags

The boot_osflags environment variable passes information to the boot command. That information is dependent on the operating system to be booted.

Syntax for Tru64 UNIX: **set boot_osflags** *flags_value*

where *flags_value* can be:

-
- a** Load operating system software from the specified boot device (autoboot). Boot to multi-user mode.
 - i** Prompt for the name of a file to load and other options (boot interactively). Boot to single-user mode.
 - s** Stop in single-user mode. Boots /vmunix to single-user mode and stops at the # (root) prompt.
 - D** Full dump; implies “s” as well. By default, if Tru64 UNIX crashes, it completes a partial memory dump. Specifying “**D**” forces a full dump if the system crashes.

Syntax for OpenVMS: **set boot_osflags** *root_number, boot_flags*

Where *root_number* is the *directory* number of the system disk on which OpenVMS files are located. For example:

<i>root_number</i>	Root Directory
0 (default)	[SYS0.SYSEXE]
1	[SYS1.SYSEXE]
2	[SYS2.SYSEXE]
3	[SYS3.SYSEXE]

And *boot_flags* are the hexadecimal value of the bit number or numbers set. To specify multiple boot flags, add the flag values (logical OR). See Table 2-7.

Table 2-7 Settings for boot_osflags Bootflags (OpenVMS)

Flags_Value	Bit Number	Meaning
1	0	Bootstrap conversationally (enables you to modify SYSGEN parameters in SYSBOOT).
2	1	Map XDELTA to running system.
4	2	Stop at initial system breakpoint.
8	3	Perform diagnostic bootstrap.
10	4	Stop at the bootstrap breakpoints.
20	5	Omit header from secondary bootstrap image.
80	7	Prompt for the name of the secondary bootstrap file.
100	8	Halt before secondary bootstrap.
10000	16	Display debug messages during booting.
20000	17	Display user messages during booting.

2.19.4 com1_baud

The default baud rate for the system is 9600. With the `com1_baud` environment variable, you can set the baud rate to match that of the device connected to the port.

Syntax: `set com1_baud baud_value`

where *baud_value* is the new baud rate. A list of possible values is displayed by attempting to set this environment variable to an unacceptable value (for example, `set com2_baud xxx`).

NOTE: *Disable the RMC to select a baud rate other than 9600. See Chapter 4.*

You will be asked to confirm the change, as shown in Example 2–26.

Example 2-26 Changing Baud Rate

```
>>> set com1_baud 19200
Embedded Remote Console only supports 9600 baud. Continue? (Y/[N]) n
bad value - com1_baud not modified
>>>
```

2.19.5 com1_mode

The `set com1_mode` command sets the bypass modes of the remote management console (see Chapter 4)

For modem connections, you can set the **`com1_mode`** variable to allow data to partially or completely bypass the RMC. The bypass modes are snoop mode, soft bypass mode, and firm bypass mode.

- In snoop mode, you can type an escape sequence to enter RMC mode. RMC mode provides a command-line interface for issuing commands to monitor and control the system.
- In soft bypass mode, you cannot enter RMC mode. But if an alert condition or loss of carrier occurs, the RMC switches into snoop mode. From snoop mode you can enter RMC mode.

- In firm bypass mode, you cannot enter RMC mode. To enter RMC mode from firm bypass mode, reset the **com1_mode** variable from the SRM console.

Syntax: **set com1_mode** *value*

where *value* can be:

snoop	Data partially bypasses RMC, but RMC taps into the data lines and listens passively for the RMC escape sequence.
soft_bypass	Data bypasses RMC, but RMC switches automatically into snoop mode if an alert condition or loss of carrier occurs.
firm_bypass	Data bypasses RMC. RMC is effectively disabled.

Example:

```
>>> set com1_mode
COM1_MODE (SNOOP, SOFT_BYPASS, FIRM_BYPASS)
```

2.19.6 console

The console terminal can be either a graphics monitor or a serial terminal. The console environment variable specifies which is used. Whenever you change the value of console, you must reset the system by pressing the Halt/Reset button or issuing the initialize command.

Syntax: **set console** *output_device*

where *output_device* can be:

graphics (default)	The console terminal is a graphics monitor or a device connected to the VGA or TGA module.
serial	The console terminal is the device connected to the COM1 port.

2.19.7 ew*0_mode

Sets an Ethernet controller to run an AUI, ThinWire, or twisted-pair Ethernet network. The default is auto-sense. For the fast setting, the device defaults to fast.

Syntax: **set ew*0_mode** *value*

where *value* can be:

au	Device type is AUI.
auto-sense	Device type is sensed by the console.
twisted-pair	Device type is 10BaseT (twisted pair).
fast duplex, twisted-pair	Device type is duplex 10BaseT.
fast	Device type is fast SCSI.
fast FD	Device type is fast full duplex SCSI.
BNC	Device type is BNC.
auto-negotiate	DE500-BA

2.19.8 ew*0_protocols

Enables network protocols for booting and other functions.

Syntax: **set ew*0_protocols** *protocol_value*

where *protocol_value* can be:

mop (default)	Sets the network protocol to mop (Maintenance Operations Protocol), the setting typically used with the OpenVMS operating system.
bootp	Sets the network protocol to bootp, the setting typically used with the Tru64 UNIX operating system.
bootp,mop	When both are listed, the system attempts to use the mop protocol first, regardless of which is listed first. If not successful, it then attempts the bootp protocol.

2.19.9 kbd_hardware_type

Used only on systems with the language variant 3C (Français), this environment variable sets the keyboard hardware type as either PCXAL or LK411 and enables the system to interpret the terminal keyboard layout correctly.

Syntax: **set kbd_hardware_type** *keyboard_type*

where *keyboard_type* can be:

-
- | | |
|---------------------------|--|
| pcxal
(default) | Selects the default keyboard hardware type. |
| lk411 | Selects the LK411 keyboard layout for use with language variant 3C (Français). |

Whenever you change the value of **kbd_hardware_type**, you must reset the system by pressing the Halt/Reset button or issuing the **initialize** command.

2.19.10 language

Specifies the keyboard layout, which is language dependent. The setting of the language environment variable must match the language of the keyboard variant. Whenever you change the value of *language*, you must reset the system by pressing the Halt/Reset button or issuing the **initialize** command.

Syntax: **set language** *language_code*

where *language_code* can be:

-
- | | | | |
|-----------|---------------------------|-----------|-----------------------------|
| 0 | No language (cryptic) | 40 | Français (Suisse Romande) |
| 30 | Dansk (Danish) | 42 | Italiano (Italian) |
| 32 | Deutsch (German) | 44 | Nederlands (Netherlands) |
| 34 | Deutsch (Schweiz) (Swiss) | 46 | Norsk (Norwegian) |
| 36 | English (American) | 48 | Portuguese (Portuguese) |
| 38 | English (British/Irish) | 4A | Suomi (Finnish) |
| 3A | Español (Spanish) | 4C | Svenska (Swedish) |
| 3C | Français (French) | 4E | Belgisch-Nederlands (Dutch) |
| 3E | Français (Canadian) | | |

2.19.11 os_type

The `os_type` environment variable specifies the default operating system. This variable is set at the factory to the setting for the operating system purchased. Use this command to change the factory default setting. Whenever you change the value of `os_type`, you must reset the system by pressing the Halt/Reset button or issuing the initialize command.

Syntax: `set os_type os_type`

where `os_type` can be:

unix	Tru64 UNIX is the default operating system, and the SRM firmware is started during power-up or reset.
openvms	OpenVMS is the default operating system, and the SRM firmware is started during power-up or reset.
nt	Windows NT is the default operating system. The SRM firmware is started, and it loads and starts the AlphaBIOS firmware during power-up or reset. (If a halt assertion condition exists, this environment variable is overridden, and the SRM firmware does not load and start the AlphaBIOS console.)

2.19.12 password

Sets or clears the console password stored in NVRAM.

Syntax: `set password`

where the password is not an argument to the **set password** command; the console prompts the user for the string, which must be between 15 and 30 characters.

2.19.13 pci_parity

Disables or enables parity checking on the PCI bus.

Syntax: **set pci_parity** *value*

where *value* can be:

(default) **on** Enables PCI parity checking.

off Disables PCI parity checking.

Some PCI devices do not implement PCI parity checking, and some have a parity-generating scheme in which the parity is sometimes incorrect or is not fully compliant with the PCI specification. A side effect of this is that superfluous PCI parity errors are reported by the host PCI bridge. In such cases, the device can be used as long as parity is not checked; disabling PCI parity checking prevents false parity errors that can cause system problems.

2.19.14 pk*0_fast

Enables fast SCSI to perform in either standard or fast mode.

Syntax: **set pk*0_fast** *scsi_speed*

where *scsi_speed* can be:

(default) **1** The controller is in fast SCSI mode.

0 The controller is in standard SCSI mode.

If the system has at least one fast SCSI device, set the default controller speed to fast SCSI (1). Devices on a controller that connects to both standard and fast SCSI devices will perform at the appropriate rate for the device. If the system has no fast SCSI devices, set the default controller speed to standard SCSI (0). If a fast SCSI device is on a controller set to standard, it will perform in standard mode.

2.19.15 pk*0_host_id

Sets the controller host bus node ID to a value between 0 and 7.

Syntax: **set pk*_host_id** *scsi_node_id*

where *scsi_node_id* is the bus node ID, a number from 0 to 7.

Each SCSI bus in the system requires a controller. Buses can theoretically support up to eight devices; however, the eighth device must always be a controller. Each device on the bus, including the controller, must have a unique ID, which is a number between 0 and 7. This is the bus node ID number.

On each bus, the default bus node ID for the controller is set to 7. You do not need to change the controller bus node ID unless you place two or more controllers on the same bus.

To list the controllers on your system, enter the command **show device**. SCSI devices begin with the letters “pk” (for example, pka0). The third letter is the adapter ID for the controller. When entering the command **set pk*0_host_id**, replace the asterisk with the adapter ID letter.

2.19.16 pk*0_soft_term

Enables or disables SCSI terminators. This command applies to systems that use the QLogic ISP1040 SCSI controller. The QLogic ISP1040 SCSI controller implements the 16-bit wide SCSI bus. The QLogic module has two terminators, one for the low eight bits and one for the high eight bits.

Syntax: **set pk*0_soft_term** *value*

where *value* can be:

Off	Disables termination of all 16 bits.
(default) low	Enables low eight bits and disables high eight bits.
High	Enables high eight bits and disables low eight bits.
On	Enables all 16 bits.
Diff	Places the bus in differential mode.

2.19.17 tt_allow_login

Enables or disables login to the SRM console firmware on alternate console ports. If the environment variable console is set to serial, the primary console device is the terminal connected through the COM1 port. The command `set tt_allow_login 1` enables logins through either the COM2 port or a graphics monitor.

Syntax: `set tt_allow_login value`

where *value* can be:

-
- | | |
|--------------------|---|
| (default) 1 | Enables login through the COM2 port or a graphics monitor. |
| 0 | Disables login through the COM2 port or a graphics monitor. |

Chapter 3

AlphaBIOS Console

AlphaBIOS is the graphical interface that supports the Microsoft Windows NT operating system and some utility programs. This chapter explains how to perform common system management tasks with AlphaBIOS.

Sections in this chapter are:

- Starting AlphaBIOS
- Keyboard Conventions and Help
- System Configuration
- Updating Firmware
- Setting Up the Hard Disk
- Performing Setup Tasks
- Installing Operating System
- Selecting the Version of Windows NT
- Designating a Primary Operating System
- Switching from AlphaBIOS to SRM Console
- Running Utility Programs

NOTE: To run firmware tests on a Windows NT system, switch to the SRM console and issue the **test** command. See Section 2.7.

3.1 Starting AlphaBIOS

Start AlphaBIOS Setup by pressing F2 from the Boot screen displayed at power-up or reset.

Figure 3-1 Boot Screen

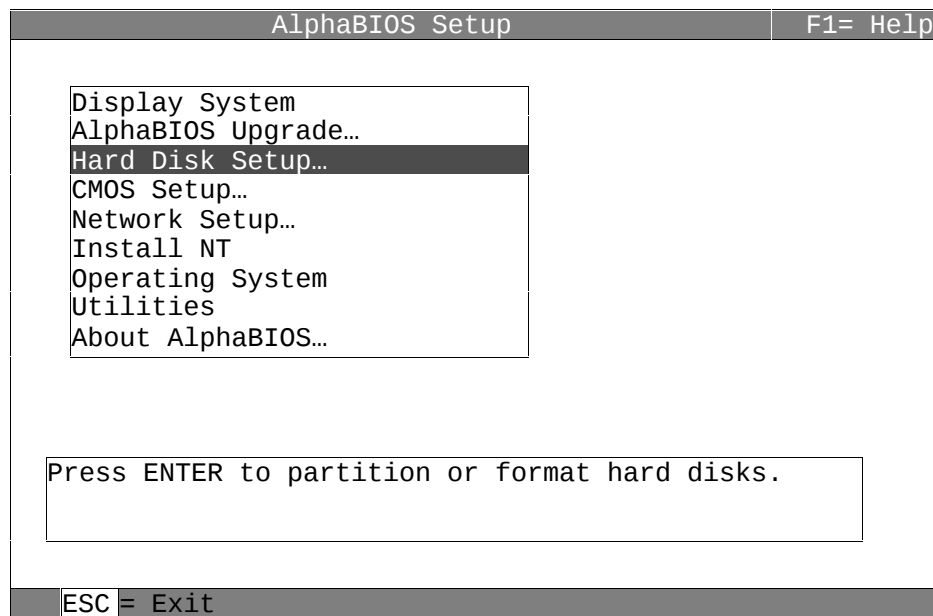


PK1450 - 98

The boot screen shown in Figure 3–1 is displayed at power-up and reset. Press **F2** at this screen to enter the setup program.

The AlphaBIOS Setup screen (Figure 3–2) is displayed. From this screen you can select the tasks to perform. Use the arrow keys to select the menu item you want and press **Enter**. (Refer to Section 3.2 for information on navigating the AlphaBIOS screens.)

Figure 3–2 AlphaBIOS Setup Screen



3.2 Keyboard Conventions and Help

AlphaBIOS uses widely accepted keys and key combinations for navigating the interface and selecting items.

Figure 3–3 Typical First-Level Help Screen

Help: CMOS Setup		F1=Key Help
F3	Change color scheme.	
F6	Enter Advanced CMOS Setup.	
F7	Set factory default CMOS settings.	
ESC	Exit CMOS Setup and discard any changes.	
F10	Exit CMOS Setup and save changes, including changes from Advanced CMOS Setup.	
ENTER=Continue		

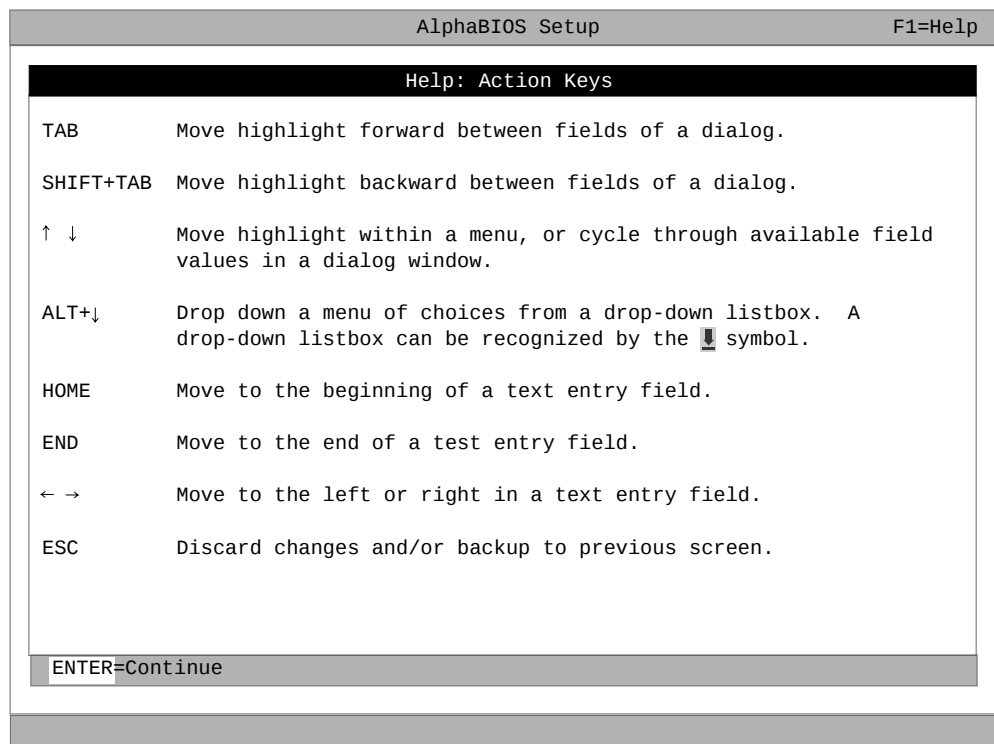
PK-0724

AlphaBIOS uses DOS and Windows keyboard conventions for navigating the interface and selecting items. The valid keystrokes are listed in the keyboard help screens.

Two levels of keyboard help are available. The first level is reached by pressing **F1** once, and shows explanations of the keystrokes available for the specific part of AlphaBIOS currently displayed. An example of the help displayed by pressing **F1** once, in this case from the CMOS Setup screen, is shown in Figure 3–3.

The second level of keyboard help, reached by pressing **F1** from the first help screen, shows explanations of the keystrokes available for navigating the interface throughout AlphaBIOS (see Figure 3–4).

Figure 3-4 Second-Level Help Screen



PK-0725

3.3 System Configuration

3.3.1 Displaying the System Configuration

The Display System Configuration screen gives information about the system's installed processor, memory, attached devices, and option boards. From the AlphaBIOS Setup screen select Display System Configuration, then the category for the information you need.

Figure 3-5 Display System Configuration Screen

Display System Configuration		F1= Help
Systemboard Configuration Hard Disk Configuration PCI Configuration SCSI Configuration ISA Configuration Memory Configuration Integrated Peripherals		
System Type: AlphaServer DS10 Processor: Alpha 21264-4, Revision V1.8.208 (1 Processor) Speed: 466 MHz Cache: 2 MB Memory: 512 MB Floppy Drive A: 3.5" 1.44 MB Floppy Drive B: None Keyboard: U.S. 101-key keyboard AlphaBIOS Version: 6.40-1 990121.1130		
ESC = Exit		

Display the system configuration as follows:

1. Start AlphaBIOS, select **Display System Configuration**, and press Enter.
2. In the **Display System Configuration** screen, use the arrow keys to select the configuration category you want to see.

From this screen, you can view configuration information about:

- System board
- Hard disk
- PCI bus
- SCSI configuration
- ISA configuration (detected ISA devices)
- Memory
- Integrated peripherals

The sections that follow explain the display for each component.

3.3.2 System Board Configuration

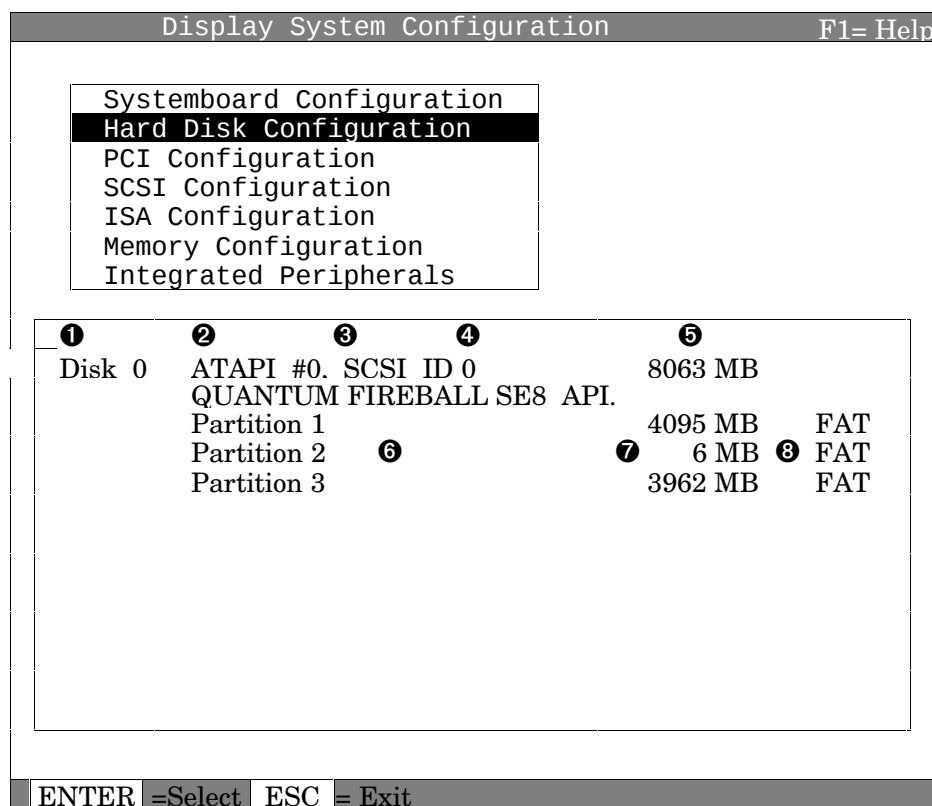
Figure 3-6 System Board Configuration

Display System Configuration		F1= Help
Systemboard Configuration Hard Disk Configuration PCI Configuration SCSI Configuration ISA Configuration Memory Configuration Integrated Peripherals		
❶	System Type: AlphaServer DS10	
❷	Processor: Alpha 21264-4, Revision V1.8.208 (1 Processor)	
❸	Speed: 466 MHz	
❹	Cache: 2 MB	
❺	Memory: 512 MB	
	Floppy Drive A: 3.5" 1.44 MB	
	Floppy Drive B: None	
	Keyboard: U.S. 101-key keyboard	
❻	AlphaBIOS Version: 6.40-1 990121.1130	
ESC = Exit		

- ❶ System type — The model number of the system.
- ❷ Processor — The model and revision of the processor chip. Revision-level information can be useful in troubleshooting problems with technical support personnel.
- ❸ Speed — The speed of the processor.
- ❹ Cache — The amount of static RAM cache memory installed.
- ❺ Memory — The amount of main memory in the system.
- ❻ Firmware version — The version of AlphaBIOS currently running on the system.

3.3.3 Hard Disk Configuration

Figure 3-7 Hard Disk Configuration



*NOTE: This screen is for information only; it cannot be edited. To make changes to the hard disk setup, use the **Hard Disk Setup** screen.*

- ❶ Physical Disk ID – Based on the SCSI ID. The disk with the lowest SCSI ID is disk 0, the disk with the next lowest SCSI ID is disk 1.
- ❷ Controller – The brand and model of the SCSI chip used on the SCSI controller.
- ❸ Controller number – Based on how many SCSI controllers of a particular type are installed in the system. The first controller of a type is always numbered 0.
- ❹ SCSI ID number – A unique number you assign to each SCSI device installed in the system. This is usually done with jumpers or a thumb wheel attached to the drive housing.
- ❺ Size – The raw capacity of the drive. Formatting the drive with different file systems, FAT or NTFS, may result in different usable capacity because of the differences in storage management under those file systems.
- ❻ Partition number – Within a single drive, partition numbers are assigned in sequential order: 1, 2, 3, and so on. The partitions populate the drive from the innermost cylinders to the outermost cylinders. If you have a large hard disk (over 800 MB) and plan to use the FAT file system, break the disk into several smaller partitions because the FAT file system uses disk space more efficiently at smaller partition sizes. This is not a concern for the NTFS file system, however, since it uses disk space very efficiently at all partition sizes.
- ❼ Partition size – The raw (unformatted) storage capacity of the partition. Actual storage space will differ based on the file system with which the partition is formatted.
- ❽ Partition format – The file system (if any) used on a partition. This field displays FAT, NTFS, or unrecognized (if the partition is unformatted).

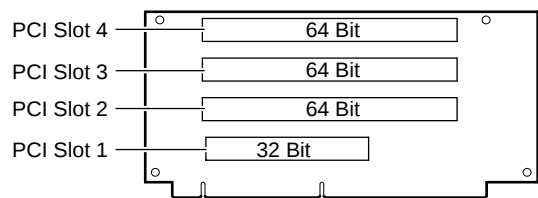
3.3.4 PCI Configuration

Figure 3-8 PCI Configuration

Display System Configuration				F1= Help
Systemboard Configuration Hard Disk Configuration PCI Configuration SCSI Configuration ISA Configuration Memory Configuration Integrated Peripherals				
① Device Name	② Device Type	③ Revision	④ Physical Slot	
AcerLabs USB 5237	USB	3	Embedded	
AcerLabs PCI-ISA	ISA bridge	195	Embedded	
DIGITAL	Ethernet	65	Embedded	
DIGITAL	Ethernet	65	Embedded	
AcerLabs IDE	IDE	193	Embedded	
Matrox Millenium	VGA	0	Slot 3	
ENTER = Select ESC =Exit				

- ❶ Device name — The name and model of the device as recorded in the device's firmware.
- ❷ Device type — Lists the function of the device in the system.
- ❸ Revision — The revision level of the device signifies the number of times it has been updated by the manufacturer.
- ❹ Physical slot — Lists the actual PCI slot number to which the device is attached.

Figure 3-9 PCI Slots



PK1045-99

You can find additional detail about any of the PCI devices listed in the **Advanced PCI Information** screen. Follow this procedure to view this screen:

1. Press Enter to enable selection in the device list.
2. Use the arrow keys to select the device for which you want additional detail.
3. Press Enter, and the detail is displayed.
4. Press Escape to return to the table of PCI devices.

The **Advanced PCI Information** screen is shown in Figure 3-10.

Figure 3-10 Advanced PCI Information

Display System Configuration			F1=Help
Advanced PCI Information			
❶	❷	❸	
Bus Number=0, Device Number=1, Function Number=0			
❹	Configuration Space:		
Register	Hex offset	Hex	
Vendor ID	00	1090	
Device ID	02	5237	
Command	04	0280	
Status	06	0280	
Revision ID	08	03	
Prog. I/F	09	10	
Sub Class Code	0a	03	
Class Code	0b	0c	
Cache Line Size	0c	08	
Latency Timer	0d	20	
Header Type	0e	00	
BIST	0f	00	
Base Address 0	10	03884000	
ENTER =Continue			

- ❶ Bus number — The virtual PCI bus number.
- ❷ Device number — The PCI bus device number.
- ❸ Function number — Represents the number assigned to a particular function on a multifunction device. For example, a combination Ethernet/SCSI controller would be listed twice, with the first function listed as 0 and the other as 1.
- ❹ Configuration header space — Displays the information in the selected device's PCI configuration space.

Press ESC to return to the list of PCI devices.

3.3.5 SCSI Configuration

Figure 3-11 SCSI Configuration

Display System Configuration		F1= Help								
Svstemboard Configuration Hard Disk Configuration PCI Configuration SCSI Configuration ISA Configuration Memory Configuration Integrated Peripherals ATAPI #0 ATAPI #1										
① ATAPI #0, SCSI ID 255, SCSI Bus 0 <table><thead><tr><th>② SCSI</th><th>③ Device</th><th>④ Size</th><th>⑤ Description</th></tr></thead><tbody><tr><td>0</td><td>Disk 0</td><td>8063</td><td>Quantum Fireball SE8 API</td></tr></tbody></table>			② SCSI	③ Device	④ Size	⑤ Description	0	Disk 0	8063	Quantum Fireball SE8 API
② SCSI	③ Device	④ Size	⑤ Description							
0	Disk 0	8063	Quantum Fireball SE8 API							
ENTER = Select ESC =Exit										

- ❶ SCSI controller information — Describes the physical characteristics of the selected SCSI controller. This line includes:

Controller — Brand and model of SCSI chip used on the SCSI controller.

Controller number — Based on the number of SCSI controllers of a particular type in the system. The first controller of a type is always numbered 0.

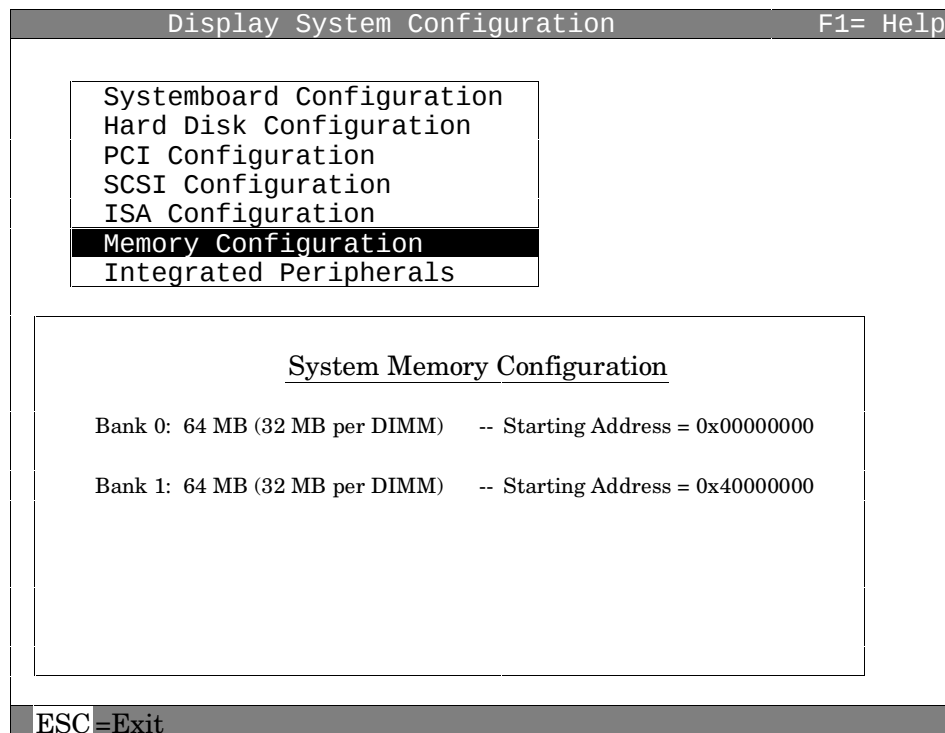
SCSI ID number — A unique number assigned to the SCSI controller. The standard scheme in numbering controllers is for controllers to be SCSI ID 7.

SCSI bus number — Indicates if the controller is first or second in the system. The first controller is tied to SCSI bus 0, and the second to SCSI bus 1.

- ❷ SCSI ID — A unique number you assign to each SCSI device controlled by the controller. This is usually done with jumpers or a thumb wheel attached to the drive housing.
- ❸ Device type — Hard disk, CD-ROM, scanner, or other type of device.
- ❹ Size — The raw capacity of the drive. Formatting the drive with different file systems (for example, FAT and NTFS) may result in different usable sizes because of differences in the way those file systems manage storage. This field is left blank if it is not applicable to a device (for example, a scanner).
- ❺ Description — The name and model of the device as recorded in the device's firmware.

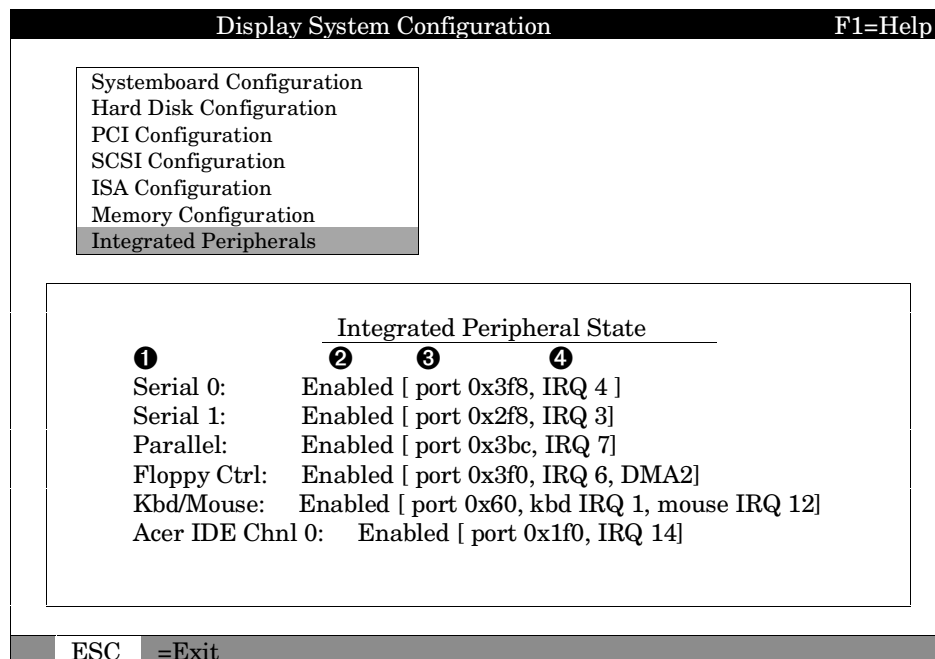
3.3.6 Memory Configuration

Figure 3-12 Memory Configuration



3.3.7 Integrated Peripherals

Figure 3-13 Integrated Peripherals

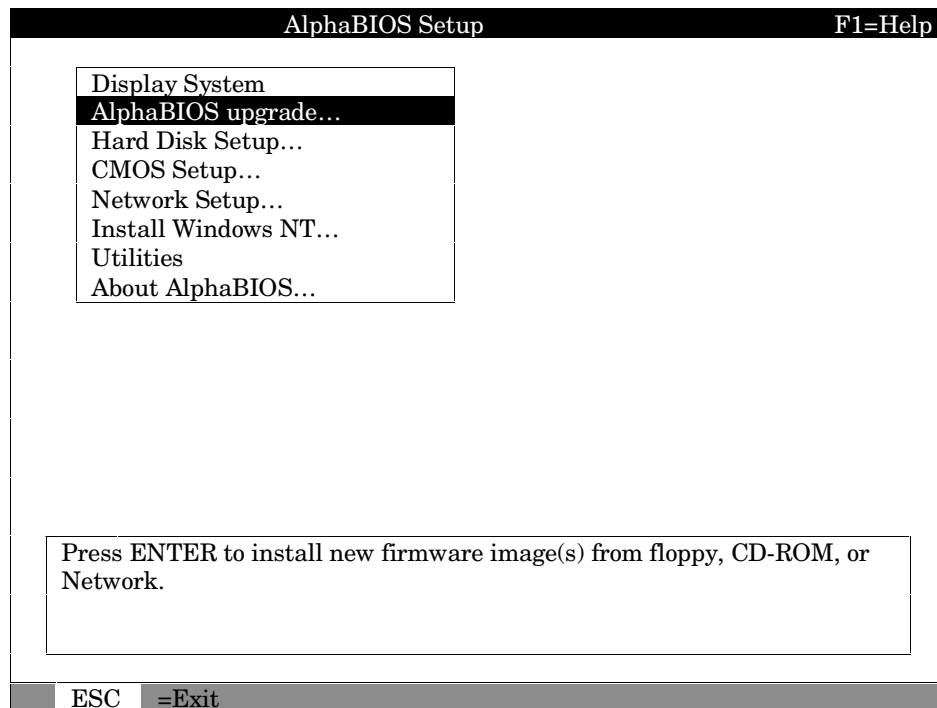


- ① Device type — The physical device as it exists on the system board.
- ② MS-DOS name — Shows if the device is enabled, and if it is enabled, the addressable MS-DOS name for the device.
- ③ Port address — The physical memory location for data received into the device, and sent from the device, respectively.
- ④ Interrupt — The interrupt request line (IRQ) used by the device to get the CPU's attention.

3.4 Updating Firmware

Insert the CD-ROM or diskette with the updated firmware and select **Upgrade AlphaBIOS** from the main AlphaBIOS Setup screen. Use the Loadable Firmware Update Utility (LFU) to perform the update. The LFU exit command causes a system reset.

Figure 3-14 Updating Firmware



As new versions of Windows NT are released, it might be necessary to upgrade AlphaBIOS to the latest version. Also, as improvements are made to AlphaBIOS, you may wish to upgrade to take advantage of new features.

Use this procedure to upgrade from an earlier version of AlphaBIOS:

1. Insert the diskette or CD-ROM containing the AlphaBIOS upgrade.
2. If you are not already running **AlphaBIOS Setup**, restart your system and press F2 when the **Boot** screen is displayed.
3. In the main **AlphaBIOS Setup** screen, select **AlphaBIOS Upgrade** and press Enter.

The system is reset and the Loadable Firmware Update (LFU) utility is started.

4. When the upgrade is complete, issue the LFU **exit** command. The system is reset and you are returned to AlphaBIOS.

3.5 Setting Up the Hard Disk

Either execute an express or a custom hard disk setup. An express setup, described here, creates the recommended partitions on the first hard disk, but it does not map bad sectors. Custom hard disk setup is described in Sections 3.5.1 and 3.5.2.

Figure 3-15 Hard Disk Setup Screen

Hard Disk Setup					F1= Help
①	②	③	④	⑤	
Disk 0	AIC78XX #0, SCSI ID 0			4091 MB	
	DEC RZ1CB-CS (C)				
	Partition 1			4085 MB	FAT
	Partition 2	⑥	⑦	6 MB	⑧ FAT
Disk 1	AIC78XX #0, SCSI ID 1			4091 MB	
	DEC RZ1CB-CS (C)				
	Partition 1			4091 MB	NTFS
⑨					
INSERT	=New	DEL	=Delete	F6	=Format
F7	=Express	ESC	=Exit		

CAUTION: *Hard disk changes are immediate; unintended data loss could occur, use care when changing your hard disk arrangement.*

- ① Physical disk ID — Based on the SCSI ID. The disk with the lowest SCSI ID is disk 0, the disk with the next lowest SCSI ID is disk 1, and so on.
- ② Controller — Brand and model of SCSI chip used on the SCSI controller.
- ③ Controller number — Based on how many SCSI controllers of a particular type are installed in the system. The first controller of a type is always numbered 0.

- ④ SCSI ID number — A unique number you assign each SCSI device in the system. This is usually done by jumpers or a thumb wheel attached to the drive.
- ⑤ Size — The raw capacity of the drive. Formatting the drive with different file systems (for example, FAT and NTFS) may result in different usable sizes because of the differences in how storage is managed under those file systems.
- ⑥ Partition number — Within a single drive, partition numbers are assigned in sequential order: 1, 2, 3, and so on. The partitions populate the drive from the innermost cylinders to the outermost. If you have a large hard disk (over 800 MB) and plan to use the FAT file system, it is a good idea to break the disk into several smaller partitions because the FAT file system uses disk space more efficiently at smaller partition sizes. This is not necessary for the NTFS file system, since it uses disk space very efficiently at all partition sizes.
- ⑦ Partition size — The raw (unformatted) storage capacity of the partition. Actual storage space will differ based on the file system with which the partition is formatted.
- ⑧ Partition format — The file system (if any) used on a partition. This field displays FAT, NTFS, or unrecognized (if the partition is unformatted).
- ⑨ Disk setup options

Insert partition (Insert key) — Use this option to create new partitions. Before creating a new partition, be sure to select an unpartitioned space. Pressing the Insert key while an already partitioned space is selected causes an informational error to be displayed.

Delete partition (Delete key) — Use this option to delete existing partitions. Before deleting a partition, be sure to back up any data you want to save on the partition. Deleting a partition deletes all data on that partition but leaves the rest of the disk unaffected. Pressing the Delete key while an unpartitioned space is selected displays an informational error.

Format partition (F6 key) — Format a partitioned space with the FAT file system.

Express setup (F7 key) — Create the default recommended partition arrangement for Windows NT.

Exit (Escape key) — Return to the **AlphaBIOS Setup** screen.

An express hard disk setup creates the recommended partition arrangement on the first hard disk (disk 0). It does not, however, format the large partition with NTFS, and it does not map bad sectors on the disk. The NTFS formatting can be done once Windows NT has been installed.

To perform an express hard disk setup

1. If you have any needed information on your disk, back it up before proceeding.
2. Start **AlphaBIOS Setup**, select **Hard Disk Setup**, and press Enter.
3. Press F7 to enter **Express Setup**.
4. Press F10 to continue with the setup.

3.5.1 Creating and Deleting Partitions Manually

Use the **Create Partition** and **Delete Partition** options if you need to create a custom hard disk partition arrangement or otherwise manually manage your hard disk partitions.

Figure 3–16 Create New Partition Dialog Box

Hard Disk Setup				
Disk 0	QLISP1040 #0, SCSI ID 3	510 MB		
	Partition 1	200 MB	NTFS	
	Unused	310 MB		
	Create new partition, Disk 0 Partition 2			
Disk 1				NTFS FAT
Available space: 310 MB				
New Partition Size: 200 MB				
ENTER = Continue ESC = Cancel				

To create a partition

1. Start AlphaBIOS and select **Hard Disk Setup**. Press Enter.
2. Select the disk on which to create the partition.
3. If one or more partitions already exist on the disk, select the unpartitioned space.
4. Press Insert. A dialog box is displayed, similar to Figure 3–16.
5. Type the size of the partition to create and press Enter.

To delete a partition

1. Start AlphaBIOS and select **Hard Disk Setup**. Press Enter.
2. Select the partition to be deleted (see ❶ in Figure 3–17).
3. Press Delete. A dialog box is displayed (see ❷).
4. Press F10 to confirm the deletion.

Figure 3–17 Delete Partition Dialog Box

Hard Disk Setup				
Disk 0	QLISP1040 #0, SCSI ID 3	510 MB		
	Partition 1	200 MB	NTFS	
❶	Partition 2	200 MB	FAT	
	Unused	100 MB		
Disk 1	NCRC810 #0, SCSI ID 5	1001 MB		
	Partition 1	995 MB	NTFS	
	Parti	Delete Disk 0, Partition 2		FAT
		The delete operation will destroy all data on partition 2 of Disk 0. ❷ Press F10 to continue. Press ESC to Cancel.		
		F10	=Continue	ES C =Cancel

3.5.2 Formatting a FAT Partition

AlphaBIOS can format partitions with the FAT file system. Use Windows NT to format a partition using NTFS.

Figure 3-18 Formatting a FAT Partition

Hard Disk Setup				
Disk 0	QLISP1040 #0, SCSI ID 3	510 MB		
	Partition 1	200 MB	NTFS	
❶	Partition 2	200 MB	Unrecognized	
	Unused	100 MB		
Disk 1	NCRC810 #0, SCSI ID 5	1001 MB		
	Partition 1	995 MB	NTFS	
	Parti		FAT	
❷	Choose Format Method:			
	Format Method	Standard Format Standard Format Quick Format		
	F10	=Continue	ES C	=Cancel

To format a FAT partition

1. Start AlphaBIOS and select **Hard Disk Setup**. Press Enter.
2. Select the partition to be formatted (see ❶ in Figure 3–18).
3. Press F6. A dialog box is displayed, asking whether to perform a quick or standard format (see ❷). If you select **Quick Format**, the formatting is completed immediately, but no bad sectors are mapped. If you select **Standard Format**, a dialog box similar to that in Figure 3–19 is displayed while the drive is formatted, showing the progress of the formatting. Standard formatting maps bad sectors.
4. Select a format method and press Enter.

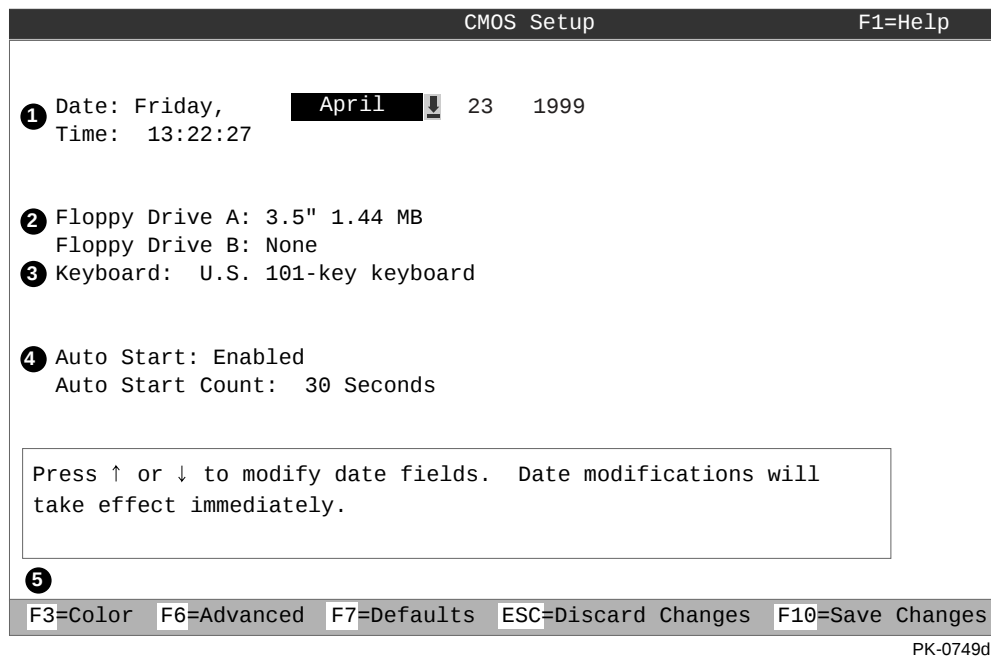
Figure 3–19 Standard Formatting

Hard Disk Setup							
Disk 0	QLISP1040 #0,		SCSI	ID	3		
	510 MB						
	Partition 1				200 MB	NTFS	
	Partition 2				200 MB	Unrecognized	
	Unused				100 MB		
Disk 1	NCRC810 #0,		SCSI	ID	5	1001 MB	
	Partition 1				995 MB	NTFS	
	Partition 2				6 MB	FAT	
Formatting... 25% ESC =Cancel							

3.6 Performing Setup Tasks

CMOS Setup is used to configure several system parameters. CMOS Setup has two modes: Standard CMOS Setup is used to configure basic system parameters; Advanced CMOS Setup is used for system-specific parameters and password protection.

Figure 3-20 Standard CMOS Setup Screen



To enter Standard CMOS Setup

Start AlphaBIOS, select **CMOS Setup**, and press Enter.

- ❶ Date and time — When setting the time, use the 24-hour format. (For example, 10:00 p.m. is 22:00:00.)
- ❷ Floppy drive — The only drive type supported is 3.5 inch, 1.44 MB.
- ❸ Keyboard — The keyboard setting makes it possible to use most language keyboards. To ensure correct character mappings, the language of your keyboard, Windows NT, and the keyboard language selection in **CMOS Setup** should all match.
- ❹ Auto start and auto start count — The **Auto Start** setting determines whether the primary operating system is automatically started after the system is reset or power-cycled. The Auto Start Count setting is the amount of time the **Boot** screen is displayed before the default system is automatically started. This delay gives you the opportunity, after resetting or power-cycling the system, to select another version of Windows NT to start or to enter **AlphaBIOS Setup**.
- ❺ **Standard CMOS Setup** options

Color (F3 key) — Repeatedly cycles through the available AlphaBIOS color schemes.

Advanced (F6 key) — Displays the **Advanced CMOS Setup** screen.

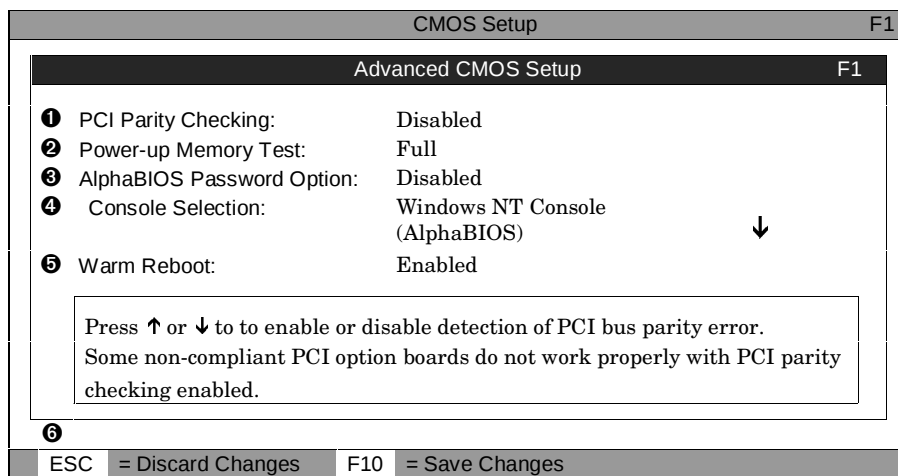
Defaults (F7 key) — Restores the default **Standard CMOS Setup** values without affecting the **Advanced CMOS Setup** values.

Discard changes (Escape key) — Restores the settings in effect when you started **CMOS Setup**. This option also discards changes made in **Advanced CMOS Setup**.

Save changes (F10 key) — Saves changes made in both the standard and advanced modes of **CMOS Setup**.

Continued on next page

Figure 3-21 Advanced CMOS Setup Screen



To Enter Advanced CMOS Setup

1. Start AlphaBIOS, select **CMOS Setup**, and press Enter.
2. In the **Standard CMOS Setup** screen, press F6.

- ❶ PCI parity checking — Enables and disables settings for PCI parity checking, which ensures data integrity across the PCI bus. Because some third-party PCI options do not correctly implement PCI parity generation, the default is Disabled.
- ❷ Power-up memory test — Enables and disables settings for the power-up memory test.
- ❸ Password setup — Enables and disables settings for password setup.

Password protection provides two levels of security for your system: setup, which requires a password to start **AlphaBIOS Setup**, and startup, which requires a password before the system initializes. Startup protection provides more comprehensive protection than setup protection because the system cannot be used at all until the correct password is entered.

- ❹ Console selection — Switches from AlphaBIOS to SRM firmware. If you select OpenVMS (SRM) or Tru64 UNIX (SRM), the next time you reset your system, the SRM firmware will be loaded instead of AlphaBIOS. This menu selection changes the setting of the **os_type** environment variable in the SRM console.
- ❺ Warm Reboot — Enable (the default) results in the system restarting normally on a shutdown and restart from the operating system. Disable results in the necessity to restart the machine by pressing reset or power-cycling.

❻ **Advanced CMOS Setup** options

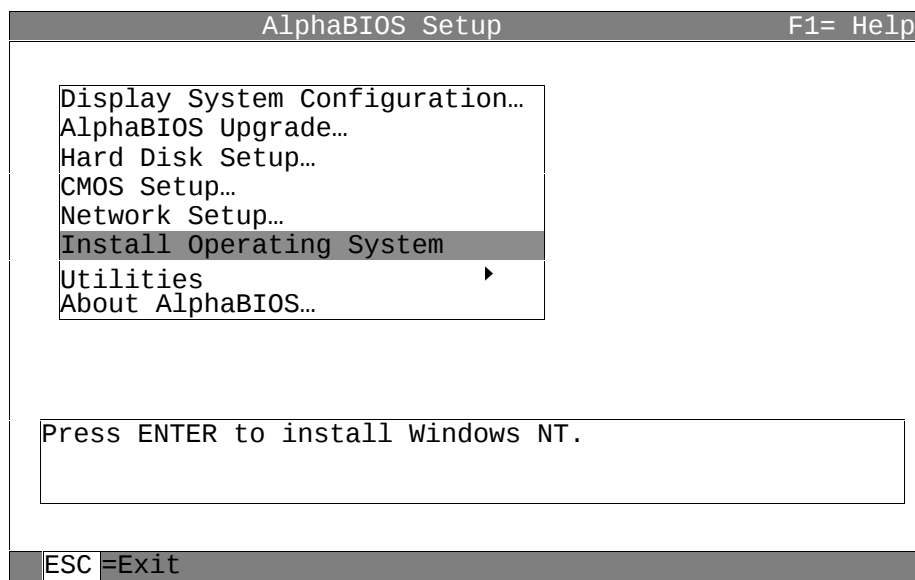
Discard changes (Escape key) — Restores those settings in effect when you started **Advanced CMOS Setup**. This does not discard changes made to **Standard CMOS Setup**.

Save Changes (F10 key) — Saves changes made in **Advanced CMOS Setup**. When exiting **CMOS Setup**, you must also save your changes at the **Standard CMOS Setup** screen for the changes to be stored permanently.

3.7 Installing Operating System

Install your operating system from the CD-ROM. Up to three versions of the operating system can be resident in the system at one time. Read the instructions carefully; it is easy to install the wrong CD-ROM driver.

Figure 3-22 Installing Windows NT



If Windows NT was installed at the factory, Windows NT setup will start automatically the first time the system powers up.

NOTE: *Steps 1 and 2 in the following procedure are necessary only when you are first setting up your system. On subsequent installations and upgrades, begin at step 3.*

1. Use **CMOS Setup** to set the system date and time: start **AlphaBIOS Setup**, select **CMOS Setup**, and press Enter.
2. Perform an express hard disk setup: return to the main **AlphaBIOS Setup** screen, select **Hard Disk Setup**, and press Enter.
3. Put the Windows NT CD into the CD-ROM drive.
4. Start **AlphaBIOS Setup**, select **Install Windows NT**, and press Enter.

Follow the prompts to complete the installation. For more information on installing Windows NT, refer to the *Installation Guide* in your Windows NT software package.

3.8 Selecting the Version of Windows NT

Up to three versions of Windows NT can reside on the system at one time. Select the version that will be started from the Operating System Selection Setup screen.

Figure 3-23 Operating System Selections

Operating System Selection Setup							
Windows NT Server 4.0				Primary Operating System			
Windows NT Server 5.0							
Boot Name:		Windows NT Server 4.0		\os\winnt4\osloader.exe \WINNT4			
Boot File:		Disk 0, partition 2					
OS Path:		Disk 0, partition 1					
OS Options:							
INSERT	= New	F6	= Edit	F8	= Primary	ES	= Discard Changes
DEL	= Delete	F7	= Copy	F9	= Validate	F10	= Save Changes

Operating System Selection Setup							
Windows NT Server 5.0				Primary Operating System			
Windows NT Server 4.0							
Boot Name:		Windows NT Server 5.0		\os\winnt5\osloader.exe \WINNT5			
Boot File:		Disk 0, partition 2					
OS Path:		Disk 0, partition 1					
OS Options:							
INSERT	= New	F6	= Edit	F8	= Primary	ES	= Discard Changes
DEL	= Delete	F7	= Copy	F9	= Validate	F10	= Save Changes

NOTE: *The term “operating system selection,” as it is used in this context, refers to a version of Windows NT. It does not pertain to the Tru64 UNIX and OpenVMS operating systems.*

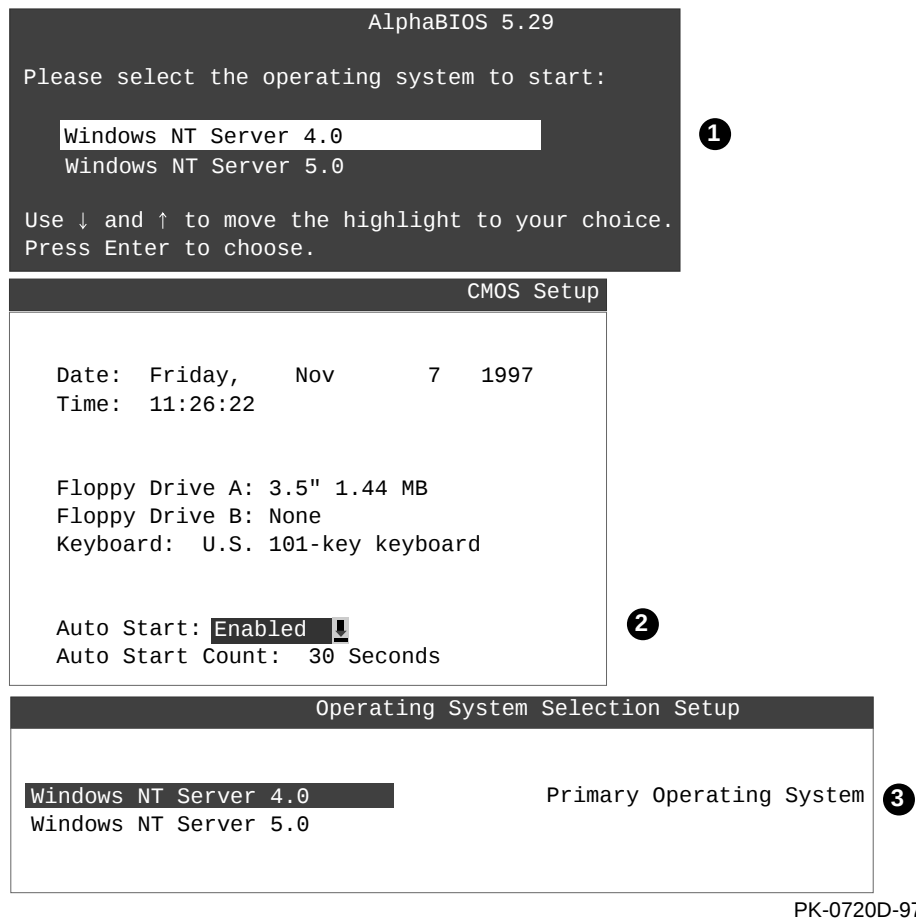
Each operating system selection is a set of information for a version of Windows NT. It describes the disk and partition containing the OSLOADER.EXE file associated with a particular operating system installation, as well as the path to the operating system itself. In addition, the operating system selection contains any options passed to the operating system during boot.

By default, one operating system selection is displayed on the **AlphaBIOS Boot** screen for each operating system installed on the system. It is not normally necessary to modify operating system selections, since the Windows NT installation creates and modifies operating system selections as needed. If you add SCSI controllers after the installation of Windows NT, it may be necessary to modify operating system selections. Depending on what kind of SCSI controller is added, in which slot it is placed, and where its driver is in the startup order relative to other existing controllers, the disk numbering may change and modification of operating system selections may be required.

3.9 Designating a Primary Operating System

3.9.1 Selecting a Primary Operating System

Figure 3–24 Primary Operating System



Three versions of Windows NT can reside on the system at the same time. This is useful in a variety of circumstances—for example, when testing application compatibility across different versions of Windows NT.

Each time you install a separate version of Windows NT, a new operating system selection is created. Although you can start any of the installed versions of Windows NT, one of them must be the primary operating system. The version of Windows NT that you select as the primary operating system is the one that automatically starts if the **Auto Start** option is enabled in AlphaBIOS.

Figure 3–24 illustrates the relationship between multiple installations of the operating-system, **Auto Start**, and the primary operating system.

- ❶ The primary operating system is listed first on the **Boot** screen.
- ❷ The primary operating system starts automatically if the **Auto Start** option is enabled in **CMOS Setup**.
- ❸ The primary operating system can be selected in the **Operating System Selection Setup** screen.

Follow this procedure to designate a new primary operating system:

1. From the **AlphaBIOS Setup** screen select **Utilities**. In the selection box that is displayed, choose **OS Selection Setup**.
2. The **Operating System Selection Setup** screen is displayed. Select the primary operating system from the list displayed.

3.9.2 Primary Operating System and the Auto Start Option

The process of setting up your operating system selections is similar to using an editor. You can make changes to your operating system selections and then either save your changes or exit without saving.

Figure 3-25 Operating System Selection Setup

Operating System Selection Setup			
Windows NT Server 4.0		1	Primary Operating System
2 Boot Name: Windows NT Server 4.0			
3 Boot File:		Disk 0, Partition 2	\os\winnt400\osloader.exe
4 OS Path:		Disk 0, Partition 1	\WINNT400
5 OS Options:			
6			
INSERT=New	F6=Edit	F8=Primary	ESC=Discard Changes
DEL=Delete	F7=Copy	F9=Validate	F10=Save Changes

PK-0737C-97

AlphaBIOS boots Windows NT in two stages. The first stage involves reading an OS Loader program from a disk. The OS Loader program is named OSLOADER.EXE. The selection for the operating system must describe the path where AlphaBIOS will find the OS Loader program. The OS Loader program must be located on a partition formatted with the FAT file system.

Once the OS Loader program executes, it uses the services provided by AlphaBIOS to load the operating system components. After the operating system is loaded, the OS Loader starts execution of the operating system.

- ❶ Primary operating system — The OS that appears first on the **AlphaBIOS Boot** screen. It is also the version of the OS that automatically starts if **Auto Start** is selected. Any of the operating system selections can be the primary operating system.
- ❷ Boot name — Each boot name is associated with an operating system selection. Windows NT setup automatically creates a boot name each time you install the operating system. Because the boot name is only a name, you can modify it at any time without affecting the rest of the operating system selection. The boot name must have at least one character.
- ❸ Boot file — Describes the disk, partition, path, and name of the file that AlphaBIOS passes control to during the process of starting the operating system. This setting is created along with the operating system selection during Windows NT setup, and the user does not usually modify it. However, this setting can be modified if necessary. For example, a developer testing different versions of OSLOADER.EXE can store the different versions in different locations and modify this line to start the operating system with the different versions as needed.

During **Operating System Selection Setup**, you can select the disk and partition for the location of the boot file from a list of choices presented in a list box. To open a list of values for the field, press the Alt and down arrow keys together. When you select a boot file location and name, AlphaBIOS searches for the specified program on the specified partition. If the search fails, a warning is displayed, saying that the file does not exist. You are given the choice to continue with the changes anyway or to cancel the operation and fix the problem.

- ❹ OS path — Describes the disk, partition, and path to the operating system root directory for an operating system selection.

During **Operating System Selection Setup**, you can select the disk and partition for the location of the boot file from a list of choices presented in a list box. To open a list of possible values for the field, press the Alt and down arrow keys at the same time. When you select or enter an OS path, AlphaBIOS searches for the directory. If the search fails, a warning is displayed, saying that the directory does not exist. You are given the choice to continue with the changes anyway or to cancel the operation and fix the problem.

- ⑤ OS options — Lists the startup parameters passed to the operating system for an operating system selection. One example of a startup parameter is whether to start the operating system in debug mode. By default, Windows NT does not add any entries to this field. This field can be modified.
- ⑥ **Operating System Selection Setup** options — You can use the options listed at the bottom of the screen to edit operating system selections. When you edit an OS selection, the fields of the OS selection are validated when you exit the **OS Selection Setup** screen. Depending on the option you choose, one of the following dialog boxes will display.

New OS selection (Insert key) — Displays the **Insert New Operating System Selection** dialog box, with default values for the new OS selection already filled in. You can change these values as necessary.

Delete OS selection (Delete key) — Tags the currently selected OS selection for deletion. Although the OS selection is removed from the screen, it is not actually deleted until you save changes.

You can also delete all of your OS selections at once by pressing the Control and Delete keys at the same time while in the **Operating System Selection Setup** screen.

NOTE: *If you delete all the OS selections, a dialog box is displayed informing you that no OS selections exist and offering three options. You can create a new OS selection, exit without saving changes, or exit and save changes.*

Edit OS selection (F6 key) — Edit all values of an OS selection by selecting the OS selection to edit and pressing F6. A dialog box is displayed with current information. You can then edit the OS selection fields.

Copy OS selection (F7 key) — Create a new OS selection by using an existing OS selection as a template. Select the OS selection you want to copy and press F7. A dialog box is displayed with the values of your OS selection. You can then edit the OS selection fields. If you do not make any changes, a duplicate copy of the OS selection you copied is made.

Primary operating system (F8 key) — Sets the selected OS selection as the primary operating system. When you make an OS selection primary, it is displayed first on the **Operating System Selection Setup** screen with the text “Primary Operating System.” In addition, the primary operating system is displayed first on the **AlphaBIOS Boot** screen and is automatically started if Auto Start is enabled.

Validate OS selection (F9 key) — Validates the fields in the currently selected OS selection. The validation routine checks that the OS loader file and OS directory fields contain valid paths and that the OSLOADER.EXE file exists in the directory specified. At the end of the validation, a dialog box is displayed describing the results of the validation. If there is an error in an OS selection, the validation routine displays a dialog box describing the component of the OS selection that is in error. At this point you can choose to edit the OS selection to correct the error, or delete the OS selection altogether.

You can also validate all OS selections at once by pressing the Control and F9 keys at the same time. All OS selections are validated in the order they are listed on screen.

Discard changes (Escape key) — Returns to the **AlphaBIOS Setup** screen without saving changes.

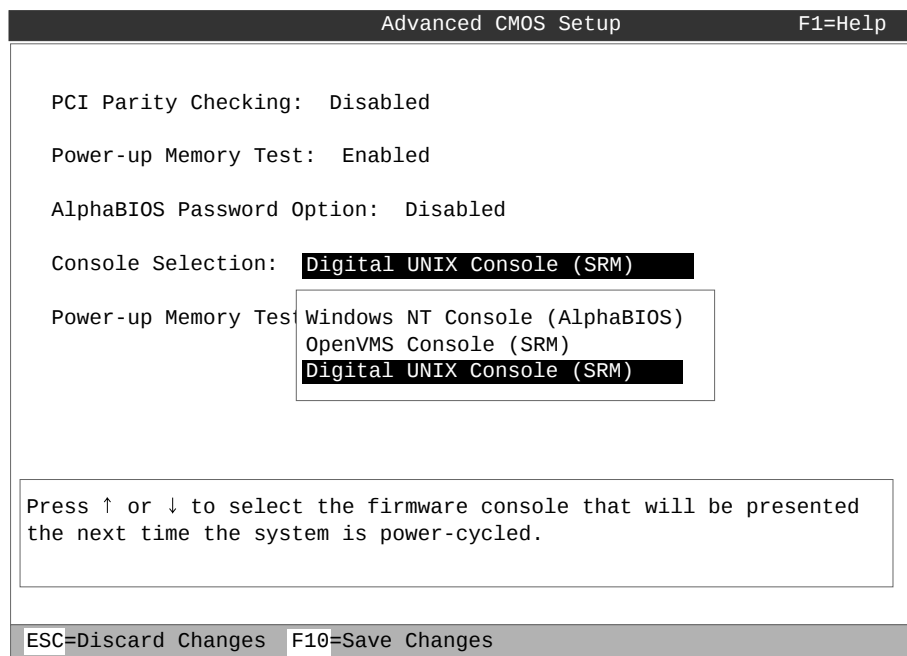
Save changes (F10 key) — Saves changes and returns to the **AlphaBIOS Setup** screen.

3.10 Switching from AlphaBIOS to SRM Console

You can run firmware-based diagnostics from the SRM console. You must be in the SRM to boot OpenVMS or UNIX. To switch from AlphaBIOS to the SRM console, press the Halt/Reset button on the control panel, and the system will power up. Press the Halt/Reset button as the system is going through self-tests and you will be at the SRM console.

Alternatively, select Tru64 UNIX (SRM) or OpenVMS (SRM) from the Advanced CMOS Setup screen and reset the system.

Figure 3-26 Switching to the SRM Console



PK-0750B-96

Firmware-based diagnostics are run from the SRM console. To boot Tru64 UNIX or OpenVMS, you need to change the Halt/Reset jumper on the motherboard (see Appendix A).

Follow this procedure to switch from AlphaBIOS to SRM:

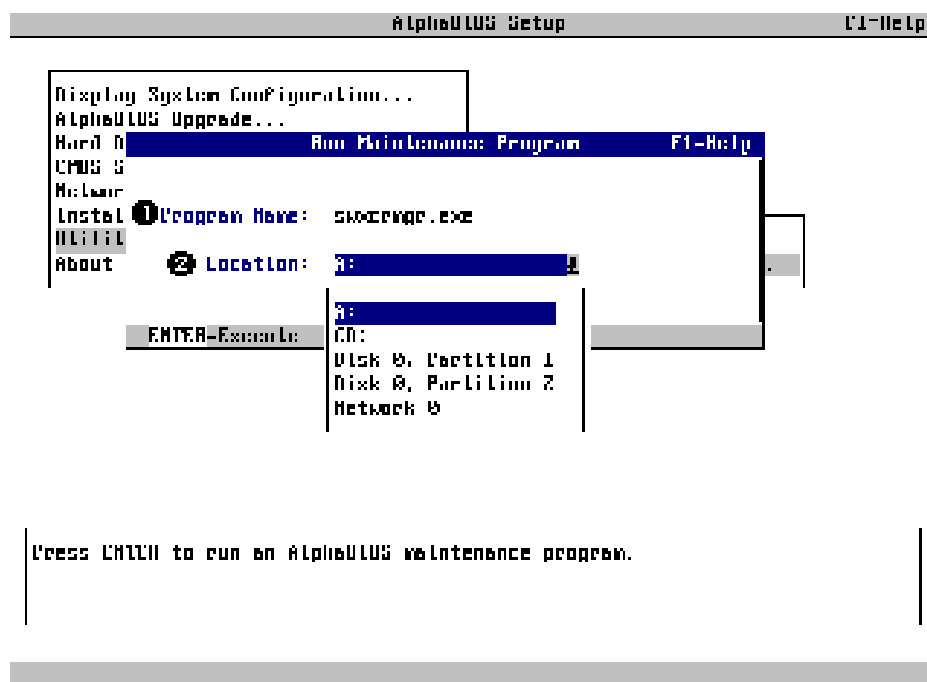
1. In AlphaBIOS, select **CMOS Setup** and press Enter.
2. In the **CMOS Setup** screen press F6. The **Advanced CMOS Setup** screen is displayed.
3. Select **Tru64 UNIX console (SRM)** or **OpenVMS console (SRM)** and press F10.
4. The **CMOS Setup** screen is displayed. Press F10 to save the change.
5. Reset the system by powering off and then on again for the changes to take effect.

To return to the AlphaBIOS console, issue the **alphabios** or **nt** command at the SRM console prompt.

3.11 Running Utility Programs

Maintenance programs such as RAID configuration utilities are run from the AlphaBIOS Utilities menu.

Figure 3-27 Run Maintenance Program Dialog Box



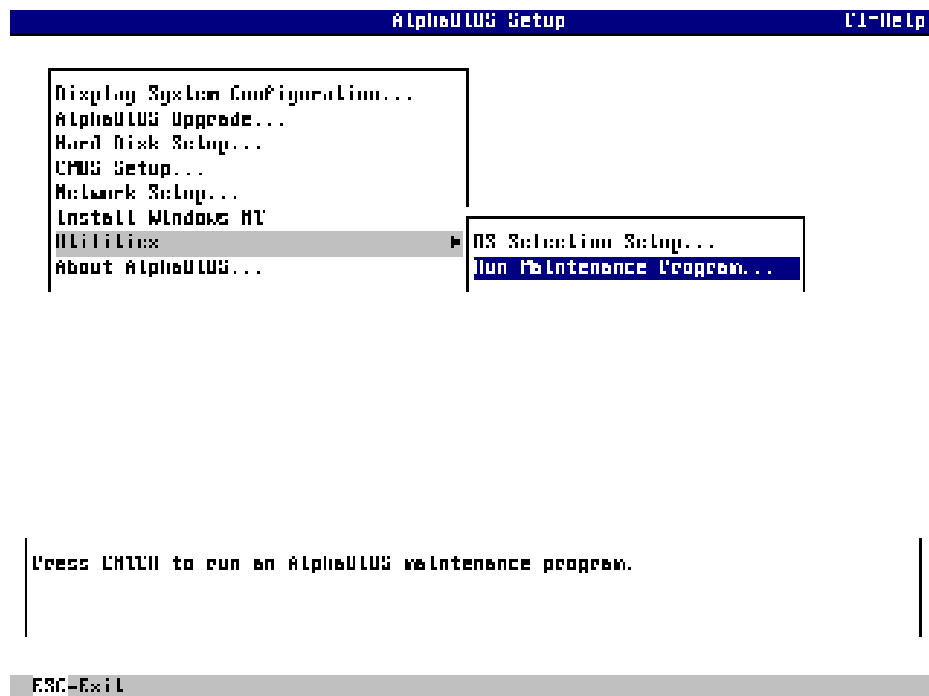
PK14656-98

- ❶ Program name — The program to be run. It must be an executable program with an .EXE extension. It is not necessary to type the extension. Programs run from AlphaBIOS must be written as ARC compatible images.
- ❷ Location — The location from which the program in the Program Name field will be run if no path is entered along with the program name. To display a list of available disks and partitions, press the Alt and down arrow keys at the same time. Once the list is displayed, use the arrow keys to cycle through the selections.

3.11.1 Running Utilities from a Graphics Monitor

1. Start AlphaBIOS Setup. If the system is in the SRM console, set the SRM **console** environment variable to **graphics** and issue the command **alphabios**.
2. From **AlphaBIOS Setup**, select **Utilities**, then select **Run Maintenance Program** from the submenu that is displayed, and press Enter. See Figure 3-28.
3. In the **Run Maintenance Program** dialog box (Figure 3-27), type the name of the program to be run in the Program Name field. Then Tab to the Location list box, and select the hard disk partition, floppy disk, or CD-ROM drive from which to run the program.
4. Press Enter to execute the program.

Figure 3-28 AlphaBIOS Utilities Menu



PK1467-98

3.11.2 Running Utilities from a Serial Terminal

Utilities are run from a serial terminal in the same way as from a monitor with graphics capabilities enabled. The menus are the same, but some keys are different.

Table 3-1 AlphaBIOS Option Key Mapping

AlphaBIOS Key	VTxxx Key
F1	Ctrl/A
F2	Ctrl/B
F3	Ctrl/C
F4	Ctrl/D
F5	Ctrl/E
F6	Ctrl/F
F7	Ctrl/P
F8	Ctrl/R
F9	Ctrl/T
F10	Ctrl/U
Insert	Ctrl/V
Delete	Ctrl/W
Backspace	Ctrl/H
Escape	Ctrl/

1. Start **AlphaBIOS Setup** by issuing the command **alphabios**.
2. Press F2 in the **AlphaBIOS Boot** screen.
3. From **AlphaBIOS Setup**, select **Utilities**, and select **Run Maintenance Program** from the submenu that is displayed. Press Enter.
4. In the **Run Maintenance Program** dialog box, type the name of the program to be run in the Program Name field. Then tab to the Location list box, and select the hard disk partition, floppy disk, or CD-ROM drive from which to run the program.
5. Press Enter to execute the program.

Chapter 4

Remote Management Console

The remote management console (RMC) provides a command-line user interface for entering commands to monitor and control the system. In addition to doing routine monitoring, you can invoke the command-line interface to perform remote power on/off, halt, and reset.

The RMC can detect alert conditions such as overtemperature, fan failure, and power supply failure and automatically dial a pager phone number or another computer system to notify the remote operator of the alert condition.

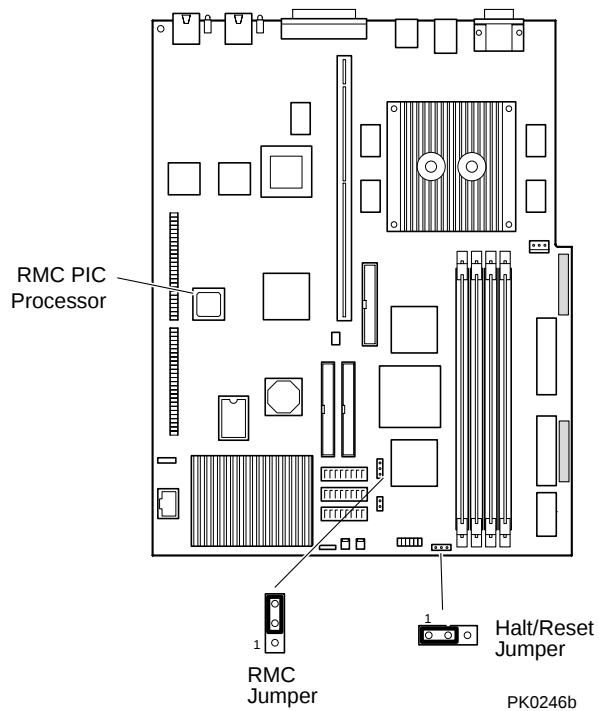
This chapter explains the operation and use of the RMC. Sections are:

- RMC Components
- Terminal Setup
- Operating Modes
- Entering the RMC
- SRM Environment Variables for COM1
- Status Monitoring
- Remote Power (On/Off)
- Remote Halt (In/Out)
- Configuring Remote Dial-In
- Configuring Dial-Out Alert
- Dialing In
- Resetting RMC to Factory Defaults
- Troubleshooting Tips
- RMC Commands

4.1 RMC Components

The RMC resides in an independent microprocessor on the system motherboard and allows a remote operator to connect to the system COM1 port from a modem or from a serial terminal or terminal emulator.

Figure 4-1 Location of RMC Components on Motherboard



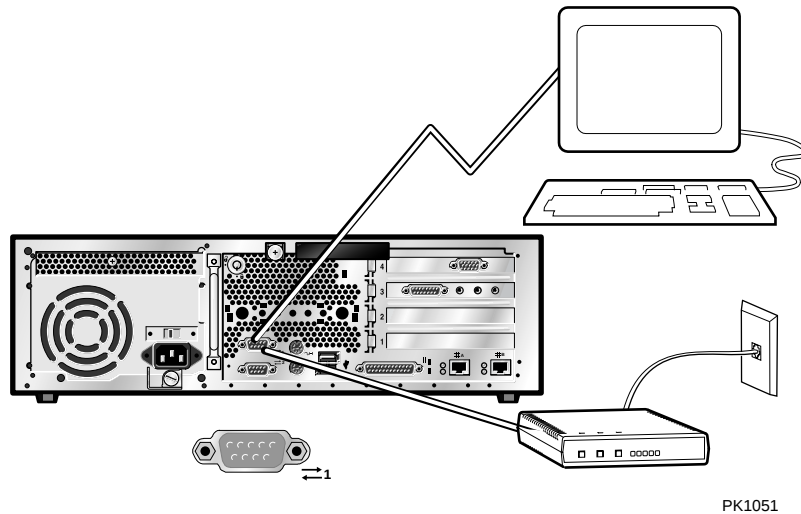
The RMC logic is implemented using an 8-bit microprocessor, PIC17C44, as the primary control device. The RMC PIC is programmed with code to control access to various environmental status bits.

You can gain access to the RMC as long as AC power is available to the system (through the wall outlet). Thus, if the system fails, you can still access the RMC and gather information about the failure. The jumper position shown in Figure 4-1 is the default position set in manufacturing which enables the RMC.

4.2 Terminal Setup

Remote connection is made through a modem connected to the COM1 port. To set up the modem, you first use a local terminal on the COM1 port to set up the parameters of the modem and RMC connection. You then disconnect the terminal and connect the modem.

Figure 4-2 Setups for RMC Mode

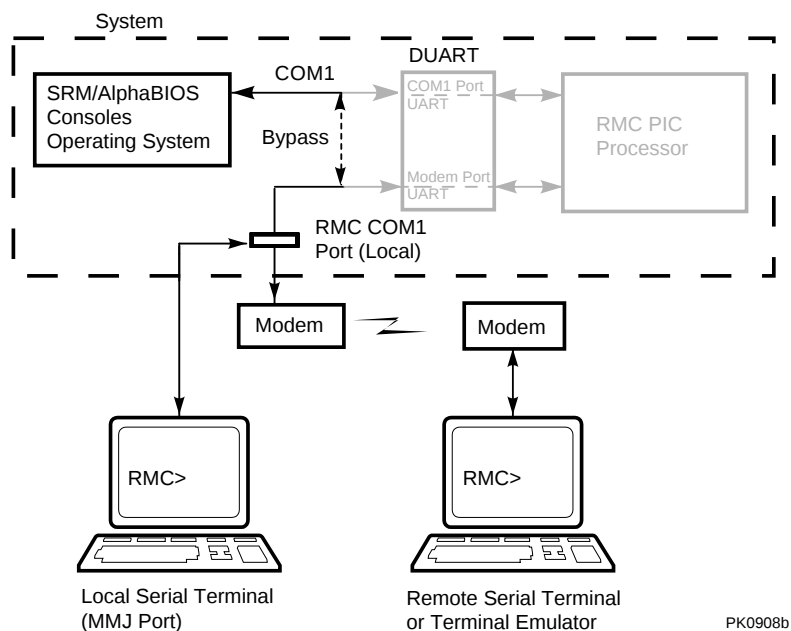


You can connect a terminal or a modem to the COM1 port, but not both at the same time. You use the terminal to set RMC parameters for the connection, then connect the modem. See Section 4.9.

4.3 Operating Modes

RMC runs in three modes on this system: Snoop, soft bypass, and firm bypass. The bypass modes are set with the `set com1_mode` command from the SRM console.

Figure 4-3 Bypass Mode



NOTE: *The internal system COM1 port should not be confused with the external COM1 serial port on the back of the system. The internal COM1 port is used by software to send data either to the COM1 port on the system or to the RMC modem port if a modem is connected.*

Snoop Mode (Default Mode)

In snoop mode data partially bypasses the RMC. The data and control signals are routed from the system COM1 port to the external modem port, but the RMC taps into the data lines and listens passively for the RMC escape sequence. If it detects the escape sequence, it enters RMC mode.

The escape sequence is also delivered to the system on the bypassed data lines. If you decide to change the default escape sequence, be sure to choose a unique sequence so that the system software does not interpret characters intended for the RMC.

Because snoop mode does not filter every piece of data, it is useful when you want to monitor the system but also ensure optimum COM1 performance.

Bypass Modes

In bypass modes, note that the internal system COM1 port is connected directly to the modem port. The RMC passively listens to the escape sequence from the external COM1 port, when the RMC escape sequence is captured.

Soft Bypass Mode

In soft bypass mode all data and control signals are routed directly from the system COM1 port to the external modem port, and the RMC does not listen to traffic on the COM1 data lines. If an alert condition or loss of carrier occurs, however, the RMC detects it and switches automatically into snoop mode. In soft bypass mode, the RMC also initializes the modem for dialing in and dials out when an alert is detected.

Soft bypass mode is useful if management applications need the COM1 channel to perform a binary download, because it ensures that RMC does not accidentally interpret some binary data as the escape sequence.

Firm Bypass Mode

Firm bypass mode effectively disables the RMC. All data and control signals are routed directly from the system COM1 port to the external modem port. In firm bypass mode, the RMC dial-in and call-out features are disabled. Firm bypass mode is useful if you want the system, or an application (such as Insight Manager or ServerWorks)—not the RMC—to fully control the modem port and you want to disable RMC features.

4.4 Entering the RMC

You type an escape sequence to invoke the RMC. You can enter RMC from a modem or the local serial console terminal. You can enter the RMC from the local terminal regardless of the current operating mode. You can set up RMC parameters.

1. Invoke the RMC from a serial terminal by typing the following default escape sequence:

^[^[rmc

This sequence is equivalent to typing Ctrl/left bracket, Ctrl/left bracket, and rmc. On some keyboards, the Esc key functions like the Ctrl/left bracket combination.

2. To exit RMC, enter the **quit** command. This action returns you to whatever you were doing before you invoked the RMC.

4.5 SRM Environment Variables for COM1

Several SRM environment variables allow you to set up the COM1 serial port for use with the RMC.

You may need to set the following environment variables from the SRM console, depending on how you decide to set up the RMC. See Chapter 2 for information on setting SRM environment variables.

Table 4-1 SRM Environment Variables for COM1

Variable	Function
com1_baud	Sets the baud rate of the COM1 serial/modem port. The default is 9600.
com1_flow	Specifies the flow control on the serial port. The default is software .
com1_mode	Specifies the COM1 data flow paths so that data either flows through the RMC or bypasses it. This environment variable can be set from either the SRM or the RMC.
com1_modem	Specifies to the operating system whether or not a modem is present.

4.6 Status Monitoring

Use the RMC status command to check the current state of the system's sensors as well as the current escape sequence and alarm information.

```
PLATFORM STATUS
Firmware Revision: V1.0
Server Power: ON
RMC Halt: Deasserted
RMC Power Control: ON
Power Supply: OK
System Fans: OK          CPU Fan: OK
Temperature: 31.0°C (warnings at 55.0°C, power-off at 60.0°C)
Escape Sequence: ^^[RMC
Remote Access: Disabled
RMC Password: set
Alert Enable: Disabled
Alert Pending: YES
Init String: AT&F0E0V0S0=2
Dial String: ATXDT8,18007598888
Alert String: ,,,,,,1066278#,,,,,,,1#,,,,,30140#;
Com1_mode: SNOOP
Last Alert: AC Loss
Watchdog Timer: 00 seconds
Autoreboot: OFF
Logout Timer: 20 minutes
User String:
```

Table 4-2 Status Command Fields

Field	Meaning
Firmware Revision	Revision of RMC firmware on the microcontroller.
Server Power	On = System is on. Off = System is off.
RMC Halt	Asserted = System has been halted. Deasserted = Halt has been released from halt.

Continued on next page.

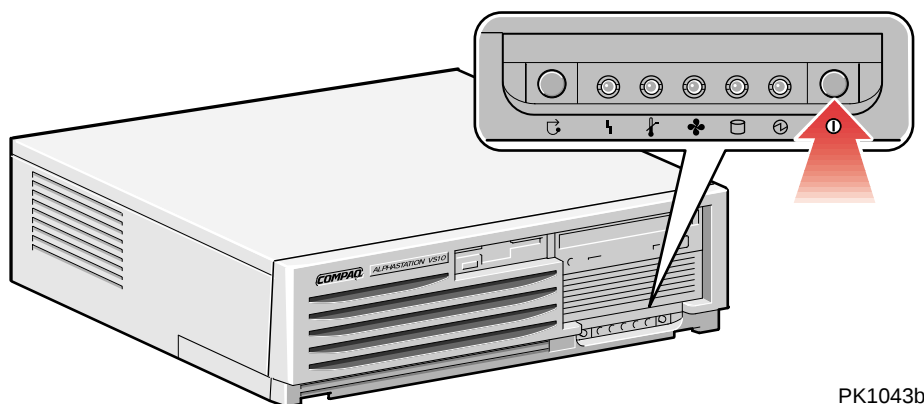
Table 4-2 Status Command Fields (Continued)

Power Control	On = System has powered on from RMC. Off = System has powered off from RMC.
Power Supply	OK = Auxiliary 5V is working.
System Fans	OK = PCI and system fan are operating. FAIL = PCI or system fan failure has been detected.
CPU Fan	OK = Fan on CPU chip is operating. FAIL = CPU fan failure has been detected.
Temperature	Reports current temperature and system limits.
Escape Sequence	Current escape sequence for access to RMC console.
Remote Access	Enable = Modem for remote access is enabled. Disable = Modem for remote access is disabled.
RMC Password	Set = Password set for modem access. Not set = No password set for modem access.
Alert Enable	Enabled = Dial-out enabled for sending alerts. Disabled = Dial-out is disabled for sending alerts.
Alert Pending	Yes = Alert has been triggered. No = No alert has been triggered.
Init String	Initialization string that was set for modem access.
Dial String	Phone number dialed when an alert occurs.
Alert String	Identifies the system that triggered the alert to the paging service. Usually the phone number of the monitored system.
Com1_mode	Can be set to snoop, soft-bypass, or firm-bypass. See Section 4.3.
Last Alert	Type of alert (for example, power supply failed).
Watchdog timer	Reports time set by user using set wdt command; can be up to 60 seconds in increments of 10. When it times out, alert is sent.
Autoreboot	ON = Autoreboot feature enabled. OFF = Autoreboot feature disabled.
Logout Timer	The amount of time before the RMC terminates an inactive modem connection. The default is 20 minutes.
User String	Notes supplied by user.

4.7 Remote Power (On/Off)

You can power a monitored system on or off from the RMC using console monitor commands.

Figure 4-4 Power Button



PK1043b

Remote Power-On

The RMC **power {on, off}** commands can be issued remotely to power the system on or off. They have the same function as the Power button on the control panel. The Power button, however, has precedence.

- If the system has been powered off with the Power button, the RMC cannot power the system on.
- If the system has been powered on with the Power button, and the **power off** command is used to turn the system off, you can toggle the Power button off, then on, to power the system back on.

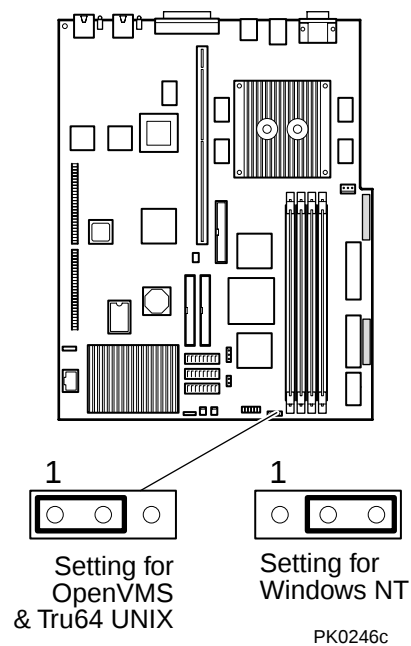
When you issue the **power on** command, the terminal exits the RMC and reconnects to the system's COM1 port.

```
RMC> power on
Returning to COM port
RMC> power off
```

4.8 Remote Halt (In/Out)

You can issue a halt to a monitored system from the RMC using console monitor commands. Under OpenVMS and Tru64 UNIX, the system will halt. Under Windows NT, the command will be ignored.

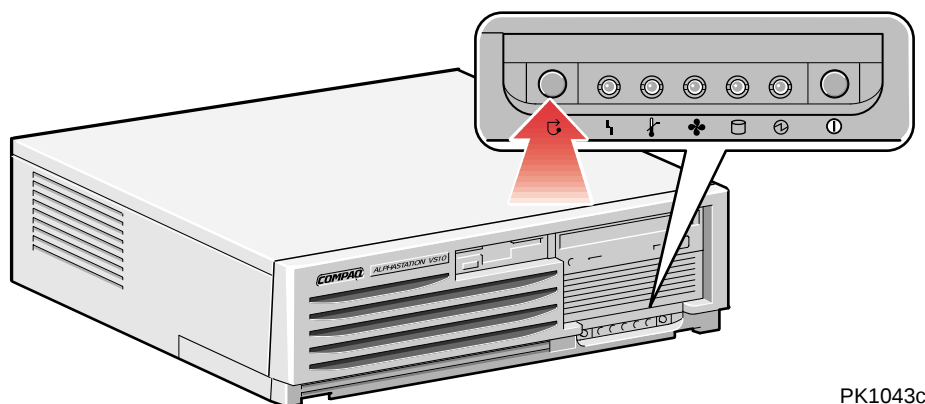
Figure 4-5 Halt/Reset Jumper



The RMC **halt in** command can be issued remotely to halt the system.
The RMC **halt out** command can be issued to deassert the halt.

These commands have the same function as the Halt button on the control panel (see Figure 4-6) for OpenVMS and Tru64 Unix. The Halt button under Windows NT operates differently because the Halt/Reset jumper is set differently (see Figure 4-5).

Figure 4–6 Halt/Reset Button



PK1043c

Table 4–3 Halt/Reset Button Function under Operating Systems

	Operating System	Function
↻	OpenVMS and Tru64 UNIX	Suspends the operating system and returns control to the SRM or RMC console.
	Windows NT	Resets the system.

The Halt button, however, does not override the **halt {in, out}** commands. If you halt the system with **halt in**, you must use **halt out** to deassert the halt state.

When you issue either the **halt in** or **halt out** command under OpenVMS or Tru64 UNIX, the terminal exits the RMC and reconnects to the system's COM1 port.

```
RMC> halt in
Returning to COM port
```

```
RMC> halt out
Returning to COM port
```

4.9 Configuring Remote Dial-In

Before you can dial in through the RMC modem port or enable the system to call out in response to system alerts, you must configure the RMC for remote dial-in.

Connect your serial terminal to the COM1 port and turn it on. Set up the RMC parameters. Then disconnect the serial terminal, connect the modem, and check your configuration.

Example 4-1 Dial-In Configuration

```
RMC> set password ❶
RMC Password: ****
Verification: ****
RMC> set init ❷
Init String: AT&F0E0V0X0S0=2
RMC> enable remote ❸
Modem will be enabled when it is connected ❹
RMC> status ❺
.
.
Remote Access: Enabled
```

NOTE: *The following modems require the initialization strings shown here. For other modems, see your modem documentation.*

Modem	Initialization String
Motorola 3400 Lifestyle 28.8	AT&F0E0V0X0S0=2
AT &T Dataport 14.4/FAX	AT&F0E0V0X0S0=2
Hayes Smartmodem Optima 288 V-34/V.FC + FAX	AT&FE0V0X0S0=2

- ❶ Sets the password that is prompted for at the beginning of a modem session. The string cannot exceed 14 characters and is not case sensitive. For security, the password is not echoed on the screen. When prompted for verification, type the password again.
- ❷ Sets the initialization string. The string is limited to 31 characters and can be modified depending on the type of modem used. Because the modem commands disallow mixed cases, the RMC automatically converts all alphabetic characters entered in the init string to uppercase.

The RMC automatically configures the modem's flow control according to the setting of the SRM **com1_flow** environment variable. The RMC also enables the modem carrier detect feature to monitor the modem connectivity.
- ❸ Enables remote access to the RMC modem port by configuring the modem with the setting stored in the initialization string.
- ❹ Because you have the serial terminal connected and not the modem, the system responds that the modem will be enabled when it is connected.
- ❺ Verify the settings and check that the Remote Access field is set to Enabled, ready for the modem to be connected.
- ❻ Disconnect the serial terminal, connect the modem and check that the modem's TR light is lit. See Section 4.11 for dialing in instructions.

4.10 Configuring Dial-Out Alert

When you are not monitoring the system through a modem connection, you can use the RMC dial-out alert feature to remain informed of system status. If dial-out alert is enabled, and the RMC detects alarm conditions within the managed system, it can call a preset pager number.

You must configure remote dial-in for the dial-out feature to be enabled. See Section 4.9. To set up the dial-out alert feature, enter the RMC from the local serial terminal.

Example 4-2 Dial-Out Alert Configuration

```
RMC> set dial                                ❶
Dial String: ATXDT9,15085553333
RMC> set alert                                ❷
Alert String: ,,,,,,5085554444#;
RMC> enable alert                              ❸
    Modem will be enabled when it is connected
RMC> clear alert                              ❹
RMC> send alert                               ❺
Alert detected!
    # Disconnect the serial terminal and connect the modem at COM1 port.
    # You should receive an alert within 30 minutes of modem connection.
    # Once the alert is received, reconnect the serial terminal and clear the test alert.
RMC> clear alert                              ❻
RMC> status                                  ❼
. . .
Alert Enable: Enabled
```

A typical alert situation might be as follows:

- The RMC detects an alarm condition caused by excessive temperature.
- The RMC dials the operator's pager and sends a message identifying the system.
- The operator dials the system from a remote serial terminal.
- The operator invokes the RMC, checks system status, and powers down the managed system.

- When the problem is resolved, the operator powers up and reboots the system.
- The dial-out alert feature is enabled as part of the dial-in setup. See Section 4.9.

The elements of the dial string and alert string are shown in Table 4–4. Paging services vary, so you need to become familiar with the options provided by the paging service you will be using. The RMC supports only numeric messages.

- ❶ Sets the string to be used by the RMC to dial out when an alert condition occurs. The dial string must include the appropriate modem commands to dial the number. In this example, 1-508-555-3333 is the pager number.
- ❷ Sets the alert string, typically the phone number of the modem connected to the remote system. The alert string is appended after the dial string, and the combined string is sent to the modem when an alert condition is detected. In this example, the system's modem number is 1-508-555-4444.
- ❸ Enables the RMC to page a remote system operator.

Because the modem is not connected at the COM1 port at this time, the RMC notifies you that the alert will be enabled at connection.

- ❹ Clears any alert that may be pending. This ensures that the **send alert** command will generate an alert condition.
- ❺ Forces an alert condition. This command is used to test the setup of the dial-out alert function. It must be issued from the local serial terminal.

The RMC tries to send the alert immediately. Because the modem is not connected at the COM1 port at this time, the RMC resets the paging interval to at most 30 minutes, and will send the alert again. If the pager does not receive the alert within 30 minutes, re-check your setup.

- ❻ Clears the current alert so that the RMC can capture a new alert. The last alert is stored until a new event overwrites it. The Alert Pending field of the **status** command becomes NO after the alert is cleared.
- ❼ Verifies the settings. Check that the Alert Enable field is set to Enabled.

NOTE: *If you do not want dial-out paging enabled at this time, enter the **disable alert** command after you have tested the dial-out alert function. Alerts continue to be logged, but no paging occurs.*

Continued on next page

Table 4-4 Elements of Dial String and Alert String

Dial String	
The dial string is case sensitive. The RMC automatically converts all alphabetic characters to uppercase	
ATXDT	AT = Attention. X = Forces the modem to dial “blindly” (not seek the dial tone). Enter this character if the dial-out line modifies its dial tone when used for services such as voice mail. D = Dial T = Tone (for touch-tone)
9,	The number for an outside line (in this example, 9). Enter the number for an outside line if your system requires it. , = Pause for 2 seconds.
15085553333	Phone number of the paging service.
Alert String	
,,,,,	Each comma (,) provides a 2-second delay. In this example, a delay of 12 seconds is set to allow the paging service to answer.
5085553332#	A call-back number for the paging service. The alert string must be terminated by the pound (#) character.
;	A semicolon (;) must be used to terminate the entire string.

4.11 Dialing In

To start a remote monitoring session, dial in, using the dial string you entered during the setup procedure. Enter the RMC modem password, and then type the escape sequence.

1. After completing the setup procedure, dial in, enter the RMC password at the Modem password: prompt. When a prompt is displayed, type the escape sequence. In the following example, the system is at the SRM console when the modem connection is made.

```
ATDT15085554444
```

```
RINGING
```

```
RINGING
```

```
CONNECT 9600/ARQ/V32/LAPM
```

```
Modem password: *****
```

```
Welcome to RMC V1.0
```

```
>>> ^^[rmc (sequence is not echoed)
```

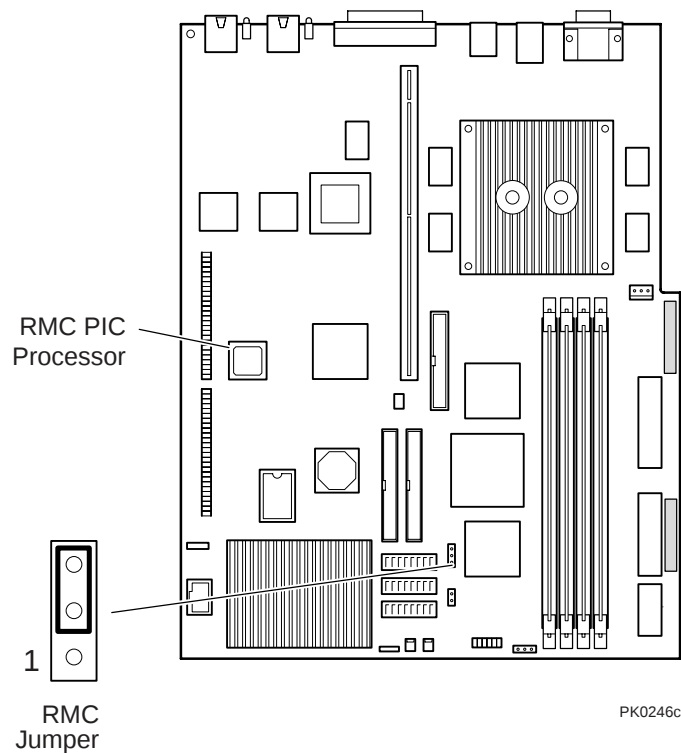
```
RMC>
```

2. At the RMC> prompt, enter commands to monitor and control the remote system.
3. When you have finished your modem session, enter the **hangup** command to cleanly terminate the modem session and disconnect from the system.

4.12 Resetting the RMC to Factory Defaults

If the RMC escape sequence is set to something other than the default, and the sequence has been lost or forgotten, the RMC must be reset to factory settings to restore the default escape sequence. See Appendix A.

Figure 4-7 RMC Jumpers (Default Positions)



WARNING: To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others.

4.13 Troubleshooting Tips

Table 4-5 lists possible causes and suggested solutions for symptoms you might see.

Table 4-5 RMC Troubleshooting

Symptom	Possible Cause	Suggested Solution
If you are not at the RMC prompt, a system reset or initialization disconnects the modem.		
	SRM clears modem connections on power-up.	Reconnect the modem. See Section 4.11.
RMC does not answer when the modem is called.		
	Modem cables may be incorrectly installed.	Check modem phone lines and connections.
	RMC remote access is disabled.	Enter enable remote access command from RMC.
	RMC does not have a valid password set.	Enter set password and enable remote commands from the RMC.
	On power-up, the RMC defers initializing the modem for 30 seconds to allow the modem to complete its internal diagnostics and initializations.	Wait 30 seconds after powering up the system and the RMC before attempting to dial in.
	Modem may have had power cycled since last being initialized or modem is not set up correctly.	Enter enable remote command from the RMC console.

Continued on next page.

Table 4-5 RMC Troubleshooting (Continued)

Symptom	Possible Cause	Suggested Solution
After the system is powered up, the COM1 port seems to hang and then starts working after a few seconds.		
	This delay is normal.	Wait a few seconds for the COM1 port to start working.
New password and escape sequence are forgotten.		
		RMC console must be reset to factory defaults. See Section 4.12.
The remote user sees a “+++” string on the screen.		
	The modem is confirming whether the modem has really lost carrier	This is normal behavior.
The message “unknown command” is displayed when the user enters a carriage return by itself.		
	The terminal or terminal emulator is including a line feed character with the carriage return.	Change the terminal or terminal emulator setting so that “new line” is not selected.
Cannot enable modem or modem will not answer.		
	The modem is not configured correctly to work with the RMC.	Modify the modem initialization string according to your modem documentation.

4.14 RMC Commands

The remote management console supports setup commands and commands for remotely managing the system.

clear {alert, port}
disable (alert, reboot, remote)
enable (alert, reboot, remote)
halt {in, out}
hangup
help or ?
power {on, off}
quit
reset
send alert
set {alert, com1_mode, dial, escape, init, logout, password, user, wdt}
status

Command Conventions

Observe the following conventions for entering RMC commands:

- Enter enough characters to distinguish the command.

NOTE: *The **reset** and **quit** commands are exceptions.
You must enter the entire word for these commands to work.*

- For commands consisting of two words, enter the entire first word and at least one letter of the second word. For example, you can enter **disable a** for **disable alert**.
- For commands that have parameters, you are prompted for the parameter.
- Use the Backspace key to erase input.
- If you enter a nonexistent command or a command that does not follow conventions, the following message is displayed:

```
*** ERROR - unknown command ***
```

- If you enter a string that exceeds 14 characters, the following message is displayed:

```
*** ERROR - overflow ***
```

4.14.1 clear alert

The clear alert command clears the current alert condition and causes the RMC to stop paging the system operator at the remote location.

If the alert is not cleared, the RMC continues to page the remote operator every 30 minutes if both remote access and alerts are enabled.

The **clear alert** command clears the current alert so that the RMC can capture a new alert. The last alert is stored until a new event overwrites it. The Alert Pending field of the **status** command becomes NO after the alert is cleared.

Syntax: **clear alert**

4.14.2 clear port

The clear port command uses the console port modem control signals to clear any “stuck” conditions on the system’s COM1 port.

The **clear port** command attempts to free the port by resetting all UARTs controlled by the RMC if the port is currently locked by an application program, without resetting the entire system.

Syntax: **clear port**

NOTE: *This command also causes the modem to disconnect.*

4.14.3 disable alert

The disable alert command disables the RMC from paging a remote system operator.

Monitoring continues and alerts are still logged in the Last Alert field of the **status** command, but alerts are not sent to the remote operator.

Syntax: **disable alert**

4.14.4 disable remote

The **disable remote** command disables remote access to the RMC modem port and disables automatic dial-out alerts.

Syntax: **disable remote**

4.14.5 enable alert

The **enable alert** command enables the RMC to page a remote system operator.

Before you can enter the **enable alert** command, you must configure remote dial-in and call-out, set an RMC password (**set password** command), and enable remote access (**enable remote** command) to the RMC modem port.

Syntax: **enable alert**

Example:

```
RMC> set dial
Dial String: atxdt9,15085553333
RMC> set alert
Alert String: ,,,,,,5085554444#;
RMC> enable alert
      Modem will be enabled when it is connected
```

The system notifies you that when you connect the modem (after disconnecting the serial terminal at the COM1 port), the alert will be enabled. Before you connect the modem, you can issue the **status** command to see if the Alert Enable field is set to Enabled.

If the **enable alert** command fails, the following error message is displayed:

```
*** ERROR - enable failed ***
```

.

4.14.6 enable remote

The `enable remote` command enables remote access to the RMC modem port by configuring the modem with the setting stored in the initialization string.

This command also allows the RMC to automatically dial the phone number set with the **`set dial`** command upon detection of alert conditions.

Before you can enter the **`enable remote`** command, you must configure remote dial-in by setting an RMC password (**`set password`** command) and initialization string. The **`enable remote`** command remains in effect until you enter the **`disable remote`** command.

Syntax: **`enable remote`**

Example:

```
RMC> set password
RMC Password: ****
Verification: ****
RMC> set init
Init String: AT&F0E0V0X0S0=2
RMC> enable remote
      Modem will be enabled when it is connected
```

The system notifies you that when you connect the modem (after disconnecting the serial terminal at the COM1 port), remote access will be enabled. Before you connect the modem, you can issue the **`status`** command to see if the Remote Access field is set to Enabled.

If the **`enable remote`** command fails, the following error message is displayed:

```
*** ERROR - enable failed ***
```

4.14.7 halt (in/out)

You can issue a halt to a monitored system from RMC using console monitor commands. The effect of this command depends on the setting of the HALT/RESET jumper on the motherboard. See Section 4.8 for a full discussion of this feature.

The **halt in** command is equivalent to pressing the Halt button on the control panel. The effect depends on which operating system you are running, and how the Halt/Reset switch is set. The default settings are:

 Operating System	Function
OpenVMS and Tru64 UNIX	Suspends the operating system and returns control to the SRM or RMC console.
Windows NT	Resets the system.

The operator at the control panel can override the **halt in** command by pressing the Power button.

Syntax: **halt in**
 halt out

Example:

```
RMC> halt in
Returning to COM port
```

NOTE: The **halt in** command does not halt systems running Windows NT. However, if you power up after the **halt in** command has been issued, the system stops in the SRM console. To load AlphaBIOS, issue the **halt out** command before powering up.

You cannot use **halt out** to release a halt if the Halt button on the control panel is latched in. If you issue the command, the following message is displayed:

```
RMC> halt out
Halt button is IN
```

4.14.8 hangup

The hangup command terminates the modem session.

If you do not issue the **hangup** command, the session is disconnected automatically after a period of idle time set by the **set logout** command. The default is 20 minutes

Syntax: **hangup**

4.14.9 help or ?

The help or ? command displays the RMC command set.

Syntax: **help or ?**

Example:

```
RMC> help
clear {alert, port}
disable {alert, reboot, remote}
enable {alert, reboot, remote}
halt {in, out}
hangup
help or ?
power {off, on}
quit
reset
send alert
set {alert, com1_mode, dial, escape, init, logout, password, user, wdt}
status
```

4.14.10 power off

The power off command is equivalent to turning off the system power from the control panel.

If the system is already powered off, this command has no effect. You can override the **power off** command either by issuing a **power on** command or by toggling the Power button on the control panel.

Syntax: **power off**

4.14.11 power on

The power on command is equivalent to turning on the system power from the control panel.

If the system is already powered on, this command has no effect. After the **power on** command is issued, the user's terminal exits the RMC and reconnects to the system's COM1 port.

Syntax: **power on**

Example:

```
RMC> power on
Returning to COM port
```

The **power on** command does not turn on the system if the Power button on the control panel is in the OFF position. Check the Power button if you issue the **power on** command and receive the following message:

```
RMC> power on
Failed to power on.
```

4.14.12 quit

The quit command exits RMC from a serial terminal and returns the user's terminal to the system's COM1 port.

You must enter the entire word for the command to take effect.

Syntax: **quit**

Example:

```
RMC> quit
Returning to COM port
```

4.14.13 reset

The reset command is equivalent to pushing the Reset button on the control panel.

The **reset** command restarts the system. The terminal exits RMC and reconnects to the server's COM1 port. You must enter the entire word for the command to take effect.

Syntax: **reset**

Example:

```
RMC> reset
Returning to COM port
```

4.14.14 send alert

The send alert command forces an alert condition.

This command is used to test the setup of the alert dial-out function, sending an alert condition to the pager or the phone.

However, since your serial terminal is connected while you are entering the command, initially the alert will not be able to be sent. When the RMC cannot page the operator, it may be that the modem or terminal is in use. The RMC waits approximately 30 minutes, and then resends the alert. This allows you to connect the modem, and wait for the alert string.

If the pager does not receive the alert after a half an hour, re-check your setup.

Syntax: **send alert**

Example:

```
RMC> send alert
Alert detected!
```

4.14.15 set alert

The set alert command sets the alert string that is transmitted through the modem when an alert condition is detected.

Set the alert string to the phone number of the modem connected to the remote system. The alert string is appended after the dial string, and the combined string is sent to the modem.

The alert string cannot exceed 63 characters, and consists of the following elements:

“,”,”,”,” Each comma (,) provides a 2-second delay. In this example, a delay of 12 seconds is set to allow the paging service to answer.

5085554444# A call-back number for the paging service. The alert string must be terminated by the pound (#) character.

; A semicolon (;) must be used to terminate the entire string.

The example shown below is generic. Because paging services vary, be sure to listen to the options provided by the paging service to determine the appropriate delay and the menu options.

Syntax: **set alert**

Example:

```
RMC> set alert
alert> , , , , , , 5085551212#;
```

4.14.16 set com1_mode

The **set com1_mode** command specifies the COM1 data flow paths, so that data either passes through the RMC or bypasses it. See Section 4.3 for descriptions of the RMC modes.

Syntax: **set com1_mode** *value*

where *value* can be:

snoop	Data partially bypasses the RMC, but the RMC taps into the data lines and listens passively for the RMC escape sequence.
soft_bypass	Data bypasses RMC, but the RMC switches automatically into snoop mode if an alert condition or loss of carrier occurs.
firm_bypass	Data bypasses the RMC. The RMC is effectively disabled.

Example:

```
>>> set com1_mode
COM1_MODE (SNOOP, SOFT_BYPASS, FIRM_BYPASS)
```

4.14.17 set dial

The set dial command sets the dial string to be used by the RMC to dial out when an alert condition occurs.

The dial string must be in the correct format for the attached modem. If a paging service is to be contacted, the dial string must include the appropriate modem commands to dial the number, wait for the line to connect, and send the appropriate touch tones to leave a pager message. The dial string is limited to 31 characters.

Because the modem commands do not allow mixed cases, the RMC converts all alphabetic characters in the dial string to uppercase.

Syntax: **set dial**

Example:

```
RMC> set dial
Dial String: ATXDT15085553333
```

where:

ATXDT	AT = Attention. X = Forces the modem to dial “blindly” (not seek the dial tone). Enter this character if the dial-out line modifies its dial tone when used for services such as voice mail. D = Dial T = Tone (for touch-tone)
9,	The number for an outside line (in this example, 9). Enter the number for an outside line if your system requires it. , = Pause for 2 seconds.
15085553333	Phone number of the paging service. (In this example it is 1–area code 508 – exchange 555 – number 3333.)

4.14.18 set escape

The set escape command changes the escape sequence used for invoking the RMC.

The escape sequence can be any character string, not to exceed 14 characters. A typical sequence consists of two or more control characters. It is recommended that control characters be used in preference to ASCII characters. Use the **status** command to verify the escape sequence.

Be sure to record the new escape sequence. If you forget the escape sequence, you must reset the RMC to the factory defaults. See Section 4.12 for information on setting the RMC to the factory defaults.

The following example consists of two instances of the Esc key and the letters “FUN.” The “F” is not displayed when you set the sequence because it is preceded by the escape character. Enter the **status** command to see the new escape sequence.

Syntax: **set escape**

Example:

```
RMC> set escape
Escape Sequence: un
RMC> status
.
.
.
Escape Sequence: ^[^[FUN
```

4.14.19 set init

The set init command sets the modem initialization string.

The initialization string is limited to 31 characters and can be modified depending on the type of modem used.

Syntax: **set init**

Some typical initialization strings are:

Initialization String	Modem
at&f0e0v0x0s0=2	Motorola 3400 Lifestyle 28.8
at&f0e0v0x0s0=2	AT &T Dataport 14.4/FAX
at&fe0v0x0s0=2	Hayes Smartmodem Optima 288 V-34/V.FC + FAX

Example:

```
RMC> set init
Init String: at&f0e0v0x0s0=2
RMC>
```

Because the modem commands do not allow mixed cases, the RMC converts all alphabetic characters entered in the init string to uppercase.

The RMC configures the modem's flow control according to the setting of the SRM **com1_flow** environment variable. The RMC also enables the modem carrier detect feature to monitor the modem connectivity.

4.14.20 set logout

The set logout command sets the amount of time before the RMC terminates an inactive modem connection. The default is 20 minutes.

The settings are in tens of minutes, 0–9. The zero (0) setting disables logout. With logout disabled, the RMC never disconnects the idle modem session.

The following example sets the logout timer to 50 minutes.

Syntax: **set logout**

Example:

```
RMC> set logout
Logout Time (0-9 tens of minutes): 5
```

4.14.21 set password

The set password command allows you to set or change the password that is prompted for at the beginning of a modem session.

A password must be set to enable access through a modem. The string cannot exceed 14 characters. For security, the password is not echoed on the screen. When prompted for verification, type the password again. If you mistype, reenter the **set password** command.

Syntax: **set password**

Example:

```
RMC> set pass
RMC Password: ****
Verification: ****
***ERROR - Verification failed, password is not set***
RMC> set pass
RMC Password: ****
Verification: ****
```

4.14.22 set user

The set user command allows you to set a user string to be displayed in the status command.

You may want to make notes regarding the system. The string is limited to 63 characters and is displayed in the User String field when you enter the **status** command. In this example, the operator leaves a reminder that the power supply needs to be replaced.

Syntax: **set user**

Example:

```
RMC> set user
User String: need to replace P/S
```

4.14.23 set wdt

The watchdog timer is a value up to 60 seconds, in increments of 10, set by the user, that can be used by software. This feature is not currently used in any applications or operating systems.

Syntax: **set wdt**

Example:

```
RMC> set wdt
```

4.14.24 status

The status command displays the system status and the current RMC settings.

Table 4–2 describes each field of the **status** command output.

Syntax: **status**

Example: See Section 4.6 for an example of the status command output.

Appendix A

Setting Jumpers

This appendix gives information on the correct settings of the jumpers on the DS10 motherboard. If you are switching to Windows NT from OpenVMS or Tru64 UNIX, you may want to change the HALT jumper.

Sections include:

- Warnings and Cautions
- Remove Power from the System
- Open the System
- Remove the Floppy Disk Enclosure
- Set Jumpers
- Restore Power

A.1 Warnings and Cautions



WARNING: *To prevent injury, access is limited to persons who have appropriate technical training and experience. Such persons are expected to understand the hazards of working within this equipment and take measures to minimize danger to themselves or others.*

The CPU module, RMC PIC processor, and memory modules have parts that operate at high temperatures. Wait 2 minutes after power is removed before touching any module.

Always wear a grounded wrist strap when servicing internal components of the system



CAUTION: *To reduce the risk of electrical shock or damage to the equipment, do not disable the power cord grounding plug. The grounding plug is an important safety feature. Plug the power cord into a grounded (earthed) electrical outlet that is easily accessible at all times. Disconnect power by unplugging the power cord from the electrical outlet or the system*

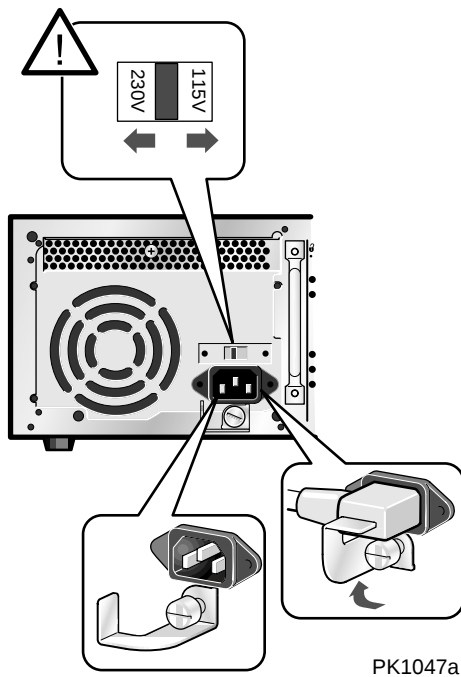
ACHTUNG: *Um Verletzungen zu vermeiden, ist der Zugriff nur Personen gestattet, die eine entsprechende technische Schulung vorweisen können und über die notwendige Erfahrung verfügen. Des weiteren sollten die Gefahren, denen diese Personen bei der Arbeit mit dem Gerät ausgesetzt sein könnten, bekannt sein, und die folgenden notwendigen Schritte eingeleitet werden, um das Gefahrenrisiko für sowohl diese als auch andere Personen zu minimieren:*

- 1. Entfernen Sie leitfähigen Schmuck.*
 - 2. Schalten Sie das System vor dem Zugriff auf den Systemkartenbehälter aus. Warten Sie zwei Minuten, bis sich die Komponenten abgekühlt haben.*
 - 3. Tragen Sie bei der Berührung von internen Komponenten ein Antistatikarmband.*
-

A.2 Remove Power from the System

Be sure to remove the AC power cord from either the wall outlet or the system cabinet. Power remains active in the RMC PIC processor when the system is powered off using only the control panel power switch. Also the CPU and memory modules have parts that operate at high temperatures. Wait 2 minutes after power is removed before touching any module.

Figure A-1 Removing Power from the System

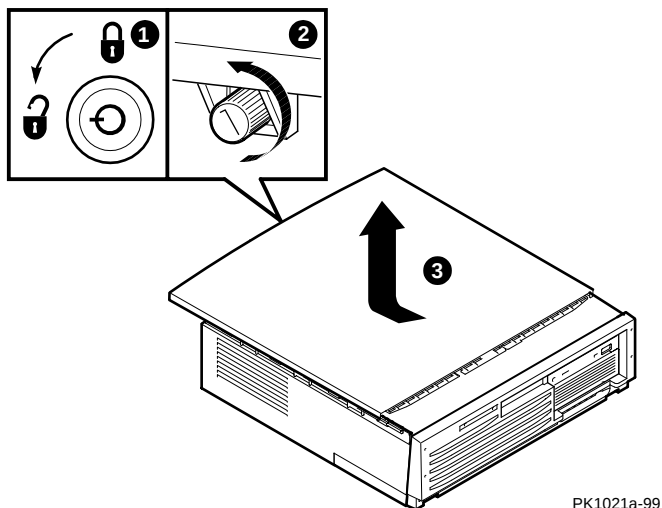


1. Turn off the system and all external peripheral devices.
2. Unplug the system from the wall outlet.
3. Wait at least 15 seconds, to allow time for the power supply capacitors to fully discharge.
4. Wait 2 minutes after power is removed before touching any module.

A.3 Open the System

Remove the cover to gain access to the system. See Figure A-2.

Figure A-2 Opening the System Cabinet



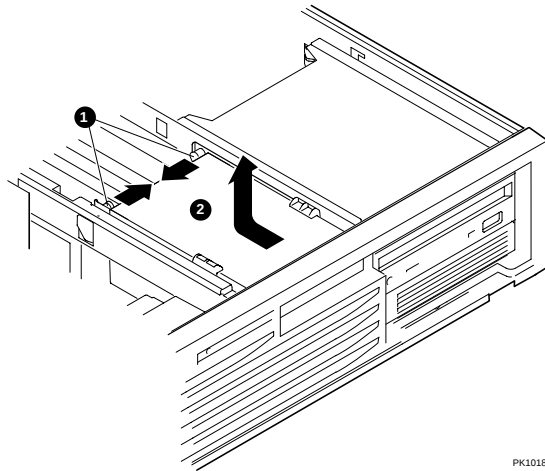
PK1021a-99

1. Unlock the top cover ❶.
2. Loosen the captive screw at the top of the bulkhead ❷.
3. Push down on the lever to the right of the screw and pull it out, perpendicular to the system.
4. Slide the cover toward the back of the system ❸. Lift it off.

A.4 Remove the Floppy Disk Enclosure

To access the jumpers, remove the floppy disk drive enclosure.

Figure A-3 Removing the Floppy Disk Enclosure

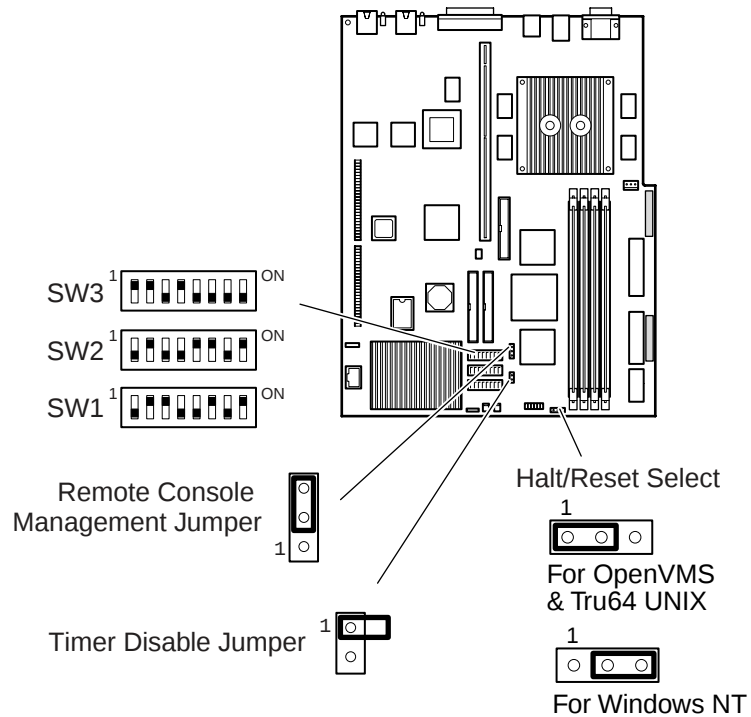


1. Hold on to the cylinder pins, which secure the floppy disk enclosure. Pull the pins toward the center (see Figure A-3 ❶).
2. Pull the floppy drive unit back and up ❷.
3. Place the unit on top of the CD-ROM drive. You do not need to remove the cables. Cable lengths have been designed to move the unit out of the way without disconnecting.

A.5 Set Jumpers

There are three jumpers on the motherboard, which you can now access. Do not change the Timer Disable jumper or any of the switches. Correct positions are given in Figure A-4.

Figure A-4 Setting Jumpers on the Motherboard



PK0246e

- The Halt/Reset Select jumper affects the operation of the control panel Halt button. You may be changing this jumper if you are switching to or from the Windows NT operating system.
- The RMC jumper enables or disables the remote management console.
- Do not change the setting of the Timer jumper or any of the switches.

Table A-1 Jumpers and Factory Default Positions

	Default position	Jumper
Halt/Reset Select	OpenVMS, Tru64 UNIX: Set for Halt	Pins 1-2
	Windows NT: Set for Reset.	Pins 2-3
RMC	Set for Enable RMC	Pins 2-3
Timer	Set for Enable Timer	Hang jumper loose from pin 1; do not jumper to pin 2.

To change the jumper on the Halt/Reset Select or RMC jumpers, wearing an anti-static strap, move the pin to the position described in Table A-1.

A.6 Restore Power

Close up the system and restore power.

1. Replace the floppy drive enclosure (see Section A.4). Check that the cables are clear and not pinched.
2. Replace the system cover, sliding it forward (see Section A.3). Secure with the handle and lock on the bulkhead.
3. Plug the AC power cord back into the system or wall outlet, whichever you disconnected (see Section A-2).
4. Push the On button on the control panel and power up.

Index

A

- Advanced CMOS setup screen, AlphaBIOS, 3-30
- Alphabios command, SRM, 2-3, 2-38
- AlphaBIOS console, 1-5
 - boot screen, 1-20, 3-2
 - checking PCI parity, 3-31
 - displaying system configuration, 3-6
 - enabling and disabling settings for memory test, 3-31
 - finding help, 3-4
 - first-level help screen, 3-4
 - key conventions, 3-4
 - running in serial mode, 3-46
 - second-level help screen, 3-5
 - setting date and time, 3-29
 - setting keyboard language, 3-29
 - setting password, 3-31
 - setup screen, 3-3
 - starting, 3-2
 - switching from SRM, 2-37
 - switching to SRM, 3-31, 3-42
- auto_action environment variable, SRM, 1-5, 2-38, 2-40
- Auxiliary power supply, RMC, 4-1

B

- Baud rate, setting for COM* ports, 2-43
- boot command, SRM, 1-6, 1-7, 1-9, 1-11, 1-12, 1-13, 1-17, 1-19, 2-3, 2-17
 - passing information to command, 2-41
 - using in secure mode, 2-25
- Boot screen, AlphaBIOS, 1-20, 3-2

- boot_file environment variable, SRM, 1-7, 1-9, 1-13, 1-15, 1-17, 2-25
- boot_flags environment variable, SRM, 2-25
- boot_osflags environment variable, SRM, 1-7, 1-9, 1-13, 1-15, 1-17, 2-38, 2-41
 - OpenVMS, 2-41, 2-42
 - Tru64 UNIX, 2-41
- boot_reset environment variable, SRM, 1-13
- bootdef_dev environment variable, SRM, 1-7, 1-9, 1-13, 1-15, 1-17, 2-25, 2-38
- Booting
 - OpenVMS, 1-12
 - from cluster disk, 1-14
 - from local disk, 1-12
 - from remote disk, 1-16
 - passing information to command, 2-41
 - Tru64 UNIX, 1-6
 - from local disk, 1-6
 - from remote disk, 1-8
- Windows NT, 1-20

C

- clear alert command (RMC), 4-23
- clear envar command, SRM, 2-3
- clear *password* command, SRM, 2-3, 2-26
- clear port command (RMC), 4-23
- CMOS setup screen, AlphaBIOS, 3-28, 3-30
- COM* ports, setting baud rate, 2-43
- com*_baud environment variable, SRM, 2-38, 2-43
- COM1 environment variables, 4-7

- Command conventions, RMC, 4-22
- Command syntax, SRM console, 2-5
- Configuring parameters, AlphaBIOS console, 3-28
- Console
 - AlphaBIOS, 1-5
 - specifying, 3-31
 - SRM, 1-5
 - switching from AlphaBIOS to SRM, 3-42
 - switching from SRM to AlphaBIOS, 2-37
- Console commands, SRM, 2-3
- Console environment variable, SRM, 2-31, 2-32, 2-38, 2-44
- Console terminal
 - specifying type, 2-44
- continue command, SRM, 2-3
 - using in secure mode, 2-25
- cpu_enabled environment variable, SRM, 2-38
- crash command, SRM, 2-3, 2-30
- Crash dump, 2-30

D

- Date, setting in AlphaBIOS console, 3-29
- deposit command, SRM, 2-3, 2-33
- Depositing data, 2-33
- Device naming conventions, SRM, 2-11
- Dial-in configuration, 4-12
- Dialing in, 4-17
- Dial-out alerts, 4-14
- disable alert command (RMC), 4-23
- disable remote command (RMC), 4-24
- display command, LFU, 1-41
- Display system configuration screen, AlphaBIOS console, 3-6
- Displaying system configuration, 2-8
 - AlphaBIOS console, 3-6
- Displaying the value or setting of an environment variable, 2-32

E

- edit command, SRM, 2-3, 2-33
- enable alert command (RMC), 4-24
- enable remote command (RMC), 4-25
- Environment variables, SRM, 2-31
 - auto_action, 1-5
 - boot_file, 1-7, 1-9, 1-13, 1-15, 1-17, 2-25
 - boot_flags, 2-25
 - boot_osflags, 1-7, 1-9, 1-13, 1-15, 1-17
 - boot_reset, 1-13
 - bootdef_dev, 1-7, 1-9, 1-13, 1-15, 1-17, 2-25
 - displaying the value or setting, 2-32
 - modifying, 2-31
 - os_type, 1-5
 - password, 2-26
 - setting, 2-31
 - summary, 2-38
- Escape sequence (RMC), 4-6
- Ethernet, specifying controller type, 2-45
- ew*0_mode environment variable, SRM, 2-38, 2-45
- ew*0_protocols environment variable, SRM, 2-38, 2-45
- examine command, SRM, 2-3, 2-33
- Examining data, 2-33
- exit command, LFU, 1-27, 1-31, 1-37, 1-40, 1-41

F

- Finding help
 - AlphaBIOS console, 3-4
 - SRM console, 2-36
- Firmware
 - updating, 1-27
 - updating from AlphaBIOS, 3-19
 - updating from CD-ROM, 1-28
 - updating from floppy disk, 1-32, 1-34
 - updating from network device, 1-38

- updating from SRM, 2-28
- updating, AlphaBIOS selection, 1-26
- updating, SRM command, 1-26

Formatting hard disk, AlphaBIOS, 3-21, 3-24, 3-26

G

Graphics monitor, specifying as console terminal, SRM, 2-44

H

halt command, SRM, 2-3, 2-28

halt in command (RMC), 4-26

Halt, remote, 4-11

hangup command (RMC), 4-13, 4-17, 4-27

Hard disk configuration screen, AlphaBIOS, 3-10

Hard disk setup screen, AlphaBIOS, 3-21

Hard disk, AlphaBIOS

- creating and deleting partitions, 3-24
- error conditions, 1-44
- formatting a FAT partition, 3-26
- setting up, 3-21

Help

- AlphaBIOS console, 3-4
- AlphaBIOS first-level help screen, 3-4
- AlphaBIOS second-level help screen, 3-5

help command, LFU, 1-41, 1-42

help command, SRM, 2-3, 2-36

help or ? commands (RMC), 4-27

I

initialize command, SRM, 2-3, 2-15, 2-35

Initializing the system, 2-35

Installing

OpenVMS, 1-18

Tru64 UNIX, 1-10

Windows NT, 1-22

Installing Windows NT, 3-32

Integrated peripherals screen, AlphaBIOS, 3-18

IOD, 1-5

K

kbd_hardware_type environment variable, SRM, 2-31, 2-32, 2-38, 2-46

Key conventions

- AlphaBIOS console, 3-4

Key mapping, AlphaBIOS in serial mode, 3-46

Keyboard

- setting language, AlphaBIOS, 3-29
- specifying hardware type, SRM, 2-46

specifying layout, SRM, 2-46

L

language environment variable, SRM, 2-31, 2-32, 2-38, 2-46

LFU

- starting, 1-26, 1-27
- starting the utility, 1-26
- typical update procedure, 1-27
- updating firmware from CD-ROM, 1-28
- updating firmware from floppy disk, 1-32, 1-34
- updating firmware from network device, 1-38

lfu command, LFU, 1-35, 1-37, 1-41, 1-42

lfu command, SRM, 1-27, 2-3, 2-28

LFU commands

- display, 1-41
- exit, 1-27, 1-31, 1-37, 1-40, 1-41
- help, 1-41, 1-42
- lfu, 1-35, 1-37, 1-41, 1-42

- list, 1-27, 1-31, 1-35, 1-37, 1-39, 1-40, 1-41, 1-42
- readme, 1-41, 1-42
- summary, 1-41
- update, 1-27, 1-31, 1-41, 1-43
- verify, 1-41, 1-43
- list command, LFU, 1-27, 1-31, 1-35, 1-39, 1-41, 1-42
- Loadable Firmware Update utility. *See* LFU
- login command, SRM, 2-3, 2-26
 - using in secure mode, 2-25

M

- Maintenance programs, AlphaBIOS, 3-44
- Memory configuration screen, AlphaBIOS, 3-17
- Memory test
 - enabling and disabling settings in AlphaBIOS, 3-31
- Modifying an environment variable, SRM, 2-31
- more command, SRM, 2-4, 2-34

N

- Naming conventions, SRM, 2-11
- Network protocols, enabling for booting and other functions, SRM, 2-45

O

- ocp_text environment variable, SRM, 2-31, 2-32, 2-39
- OpenVMS
 - booting, 1-12
 - from cluster disk, 1-14
 - from local disk, 1-12
 - from remote disk, 1-16
 - console for, 2-1
 - installing, 1-18

- specifying as default operating system, 2-47
- Operating system
 - booting OpenVMS, 1-12
 - from cluster disk, 1-14
 - from local disk, 1-12
 - from remote disk, 1-16
 - booting Tru64 UNIX, 1-6
 - from local disk, 1-6
 - from remote disk, 1-8
 - booting Windows NT, 1-20
 - installing Tru64 UNIX, 1-10
 - installing OpenVMS, 1-18
 - installing Windows NT, 1-22, 3-32
 - specifying default, 2-47
- Operating system selection setup screen, AlphaBIOS, 3-38
- Operating system selection, defined, 3-35
- Operating system selections screen, AlphaBIOS, 3-34
- os_type environment variable, SRM, 1-5, 1-21, 2-32, 2-47, 3-31

P

- Pagers, 4-15
- Parameters, configuring in AlphaBIOS console, 3-28
- password environment variable, SRM, 2-26
- Password, setting in AlphaBIOS console, 3-31
- PCI
 - enabling and disabling parity checking on bus, SRM, 2-47, 2-48
 - finding device information, AlphaBIOS, 3-14
 - parity checking in AlphaBIOS, 3-31
- PCI card cage
 - slot numbers, 2-12
- PCI configuration screen, AlphaBIOS, 3-12

- PCI options
 - slot numbers, 2-12
- pci_parity environment variable, SRM, 2-39, 2-47, 2-48
- pk*0_fast environment variable, SRM, 2-39, 2-48
- pk*0_host_id environment variable, SRM, 2-39, 2-49
- pk*0_soft_term environment variable, SRM, 2-39, 2-49
- power off command (RMC), 4-27
- power on command (RMC), 4-28
- Power-on/off, from RMC, 4-9
- Power-up, specifying console action, SRM, 2-40
- prcache command, SRM, 2-4, 2-18
- Primary operating system, AlphaBIOS, 3-36

Q

- quit command (RMC), 4-6, 4-28

R

- Reading a file, 2-34
- readme command, LFU, 1-41, 1-42
- Remote console monitor. *See* RMC
- Remote dial-in, 4-17
- Remote power-on/off, 4-9
- reset command (RMC), 4-16, 4-29
- Reset, from RMC, 4-11
- Reset, specifying console action, SRM, 2-40
- RMC
 - accessing, 4-5
 - auxiliary power supply, 4-1
 - clear alert command, 4-23
 - clear port command, 4-23
 - command conventions, 4-22
 - commands, 4-21
 - configuring call-out, 4-12
 - dial-out alerts, 4-14
 - disable alert command, 4-23
 - disable remote command, 4-24

- enable alert command, 4-24
- enable remote command, 4-25
- entering, 4-6
- escape sequence, 4-6
- exiting, 4-6
- halt in command, 4-26
- hangup command, 4-13, 4-17, 4-27
- help or ? commands, 4-27
- power off command, 4-27
- power on command, 4-28
- quit command, 4-6, 4-28
- remote halt, 4-11
- remote power-on/off, 4-11
- remote reset, 4-11
- reset command, 4-29
- resetting to factory defaults, 4-18
- send alert command, 4-29
- set alert command, 4-30
- set dial command, 4-32
- set escape command, 4-33
- set init command, 4-34
- set logout command, 4-35
- set password command, 4-35
- set user command, 4-36
- status command, 4-7, 4-36
- terminal setup, 4-3
- troubleshooting, 4-19
- RMC mode, 4-4
- Running maintenance programs, AlphaBIOS, 3-44
- Running utility programs, 3-44

S

- SCSI
 - enabling and disabling terminators, SRM, 2-46
 - enabling Fast SCSI, SRM, 2-45
 - setting controller host bus node ID, SRM, 2-46
- SCSI configuration screen, AlphaBIOS, 3-15
- send alert command (RMC), 4-32

- Serial terminal, specifying as console terminal, SRM, 2-40
- set alert command (RMC), 4-15, 4-18, 4-33
- set dial command (RMC), 4-18, 4-34
- Set dial command (RMC), 4-15
- set *envvar* command, SRM, 2-4, 2-28, 2-29
- set escape command (RMC), 4-35
- set host command, SRM, 2-4
- set init command (RMC), 4-36
- Set init command (RMC), 4-15
- set logout command (RMC), 4-37
- set password command (RMC), 4-15, 4-37
- set password command, SRM, 2-4, 2-21
- set secure command, SRM, 2-4, 2-22
- set user command (RMC), 4-38
- Setting an environment variable, 2-28
- Setting up the hard disk, AlphaBIOS, 3-21
- Setup screen, AlphaBIOS, 3-3
- show config command, SRM, 2-4, 2-8
- show device command, SRM, 1-6, 1-7, 1-9, 1-12, 1-13, 1-15, 1-17, 2-4, 2-10
- show *envvar* command, SRM, 1-12, 2-4, 2-28, 2-29
- show memory command, SRM, 2-4, 2-12
- show pal command, SRM, 2-4, 2-12
- show power command, SRM, 2-4, 2-13
- show version command, SRM, 2-4, 2-14
- Special characters, SRM console, 2-6
- SRM console, 1-5, 2-1–2-47
 - command syntax, 2-5
 - device naming conventions, 2-11
 - invoking, 2-2
 - special characters, 2-6
 - summary of commands, 2-3
 - summary of environment variables, 2-35
 - switching from AlphaBIOS, 3-31, 3-42
 - switching to AlphaBIOS, 2-34
- SRM console commands
 - alphabios, 2-3, 2-35
 - boot, 1-7, 1-9, 1-11, 1-13, 1-17, 1-19, 2-3, 2-18
 - using in secure mode, 2-22
 - clear *envvar*, 2-3
 - clear *password*, 2-3, 2-24
 - continue, 2-3
 - using in secure mode, 2-22
 - crash, 2-3, 2-27
 - deposit, 2-3, 2-30
 - edit, 2-3, 2-30
 - examine, 2-3, 2-30
 - halt, 2-3, 2-25
 - help, 2-3, 2-33
 - initialize, 2-3, 2-15, 2-32
 - lfu, 1-27, 2-3, 2-25
 - login, 2-3, 2-23
 - using in secure mode, 2-22
 - more, 2-4, 2-31
 - prcache, 2-4, 2-19
 - set *envvar*, 2-4, 2-28, 2-29
 - set host, 2-4
 - set password, 2-4, 2-21
 - set secure, 2-4, 2-22
 - show config, 2-4, 2-8
 - show device, 1-7, 1-9, 1-13, 1-15, 1-17, 2-4, 2-10
 - show *envvar*, 2-4, 2-28, 2-29
 - show memory, 2-4, 2-12
 - show pal, 2-4, 2-12
 - show power, 2-4, 2-13
 - show version, 2-4, 2-14
 - start
 - using in secure mode, 2-22
 - stop, 2-4, 2-25
 - summary of commands, 2-3
 - test, 2-4, 2-20
- SRM console environment variables

- auto_action, 2-35, 2-37
- boot_osflags, 2-35, 2-38, 2-39
- boot_osflags, DIGITAL UNIX, 2-38
- boot_osflags, OpenVMS, 2-38
- bootdef_dev, 2-35
- com*_baud, 2-35, 2-40
- commands for displaying the value or setting, 2-28
- commands for modifying, 2-28
- commands for setting, 2-28
- commands for using, 2-28
- console, 2-35, 2-40
- cpu_enabled, 2-35, 2-41
- ew*0_mode, 2-35, 2-41
- ew*0_protocols, 2-35, 2-41
- kbd_hardware_type, 2-35, 2-42
- language, 2-35, 2-43
- ocp_text, 2-36
- os_type, 1-21, 2-44, 3-31
- pci_parity, 2-36, 2-44, 2-45
- pk*0_fast, 2-36, 2-45
- pk*0_host_id, 2-36, 2-46
- pk*0_soft_term, 2-36, 2-46
- tt_allow_login, 2-36, 2-47
- SRM firmware
 - enabling and disabling login to, 2-47
- Standard CMOS setup screen,
 - AlphaBIOS, 3-28
- start command, SRM
 - using in secure mode, 2-22
- Starting AlphaBIOS, 3-2
- status command (RMC), 4-8, 4-38
- stop command, SRM, 2-4, 2-25
- Syntax
 - SRM console commands, 2-5
- System board configuration screen,
 - AlphaBIOS, 3-8
- System configuration
 - displaying in AlphaBIOS console, 3-6
 - displaying in SRM console, 2-8
- System crash dump, 2-27

- System failure, specifying console
 - action, SRM, 2-37

T

- Terminal setup (RMC), 4-3
- test command, SRM, 2-4, 2-19
- Testing the system (all operating systems), 2-19
- Time, setting in AlphaBIOS console, 3-29
- Troubleshooting
 - RMC, 4-19
- Tru64 UNIX
 - booting, 1-6
 - from local disk, 1-6
 - from remote disk, 1-8
 - console for, 2-1
 - installing, 1-10
 - specifying as default operating system, 2-47
- tt_allow_login environment variable, SRM, 2-39, 2-50

U

- update command, LFU, 1-27, 1-31, 1-37, 1-40, 1-41, 1-43
- Updating firmware
 - AlphaBIOS console, 3-19
 - from AlphaBIOS console, 1-26
 - from SRM console, 1-26
 - SRM console, 2-28
- Utility programs, 3-44
 - running from serial terminal, 3-46

V

- verify command, LFU, 1-41, 1-43
- Version selection, Windows NT, 3-34

W

- Windows NT
 - auto start, 3-38
 - booting, 1-20

designating primary operating
system, 3-36
installing, 1-22, 3-32
selecting version, 3-34

specifying as default operating
system, 2-47
starting automatically, 3-29, 3-38