Night Hawk^R
HN6200/HN6800
PowerMAX OS
Version 5.1 Release Notes

June 2001
Revision 1 October 18, 2001

0890454-5.1

READ ME BEFORE INSTALLING THIS PRODUCT

CONCURRENT COMPUTER CORPORATION
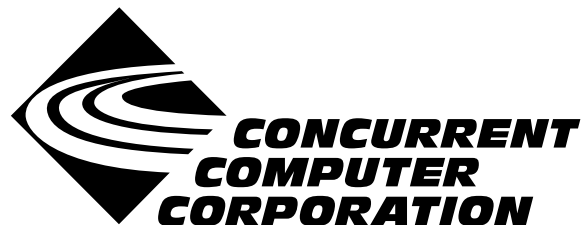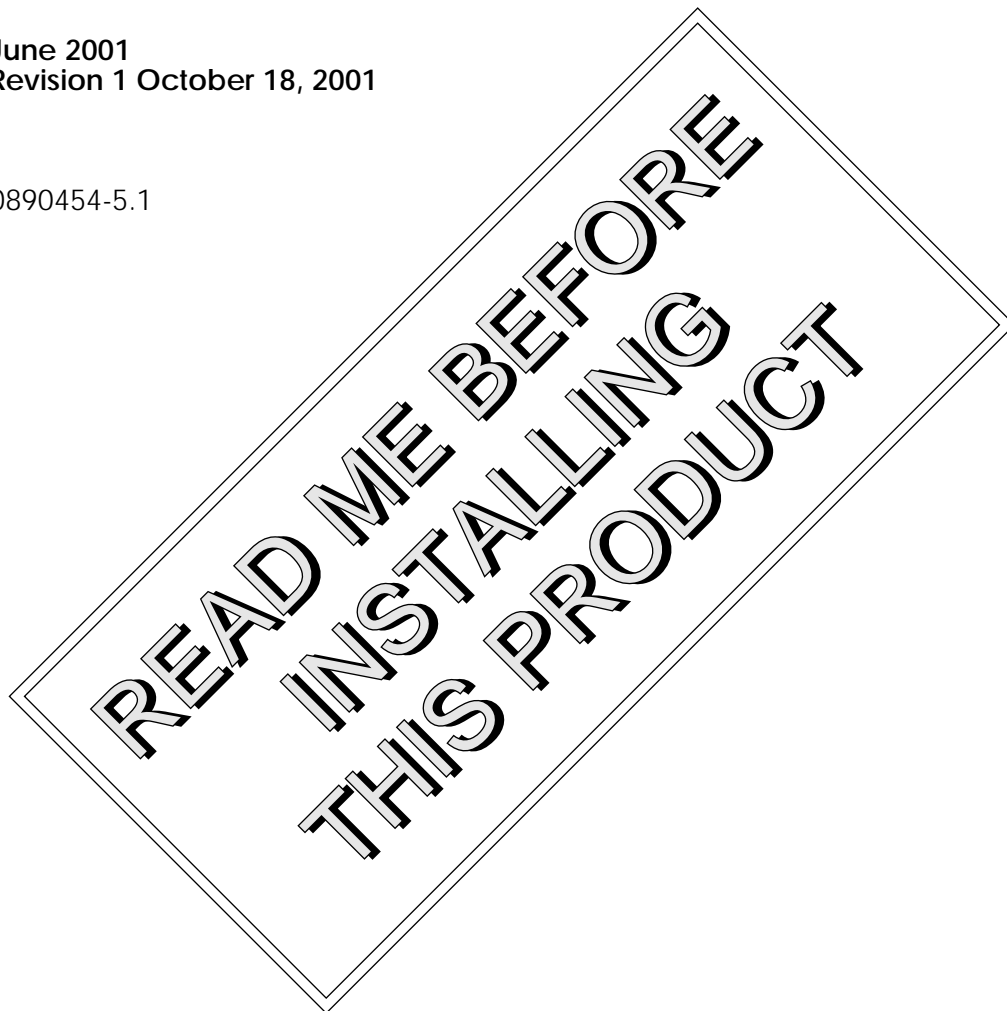
Disclaimer

The information contained in this document is subject to change without notice. Concurrent Computer Corporation has taken efforts to remove errors from this document, however, Concurrent Computer Corporation's only liability regarding errors that may still exist is to correct said errors upon their being made known to Concurrent Computer Corporation.

License

Duplication of this manual without the written consent of Concurrent Computer Corporation is prohibited. Any copy of this manual reproduced with permission must include the Concurrent Computer Corporation copyright notice.

Trademark Acknowledgments

Night Hawk is a registered trademark of Concurrent Computer Corporation.

PowerWorks, PowerMAX OS, NightStar, NightView, NightSim, NightProbe, NightTune, NightGraphics MAXAda, NightBench and ARMS are trademarks of Concurrent Computer Corporation.

Power Hawk, PowerMAXION and TurboHawk are trademarks of Concurrent Computer Corporation.

UNIX is a registered trademark and X/Motif is a trademark of The Open Group.

POSIX is a registered trademark of the Institute of Electronics and Electrical Engineers.

AltiVec and PowerStack II are trademarks of Motorola Corporation.

PowerPC is a trademark of and used under license from International Business Machine Corporation.

The X Window System is a trademark of The Open Group.

OSF/Motif is a registered trademark of the Open Software Foundation, Inc.

Synergy, VGM5 and VSS4 are trademarks of Synergy Microsystems, Inc.

SanDisk is a registered trademark of SanDisk Corporation.

Ethernet is a trademark of the Xerox Corporation.

All other trademarks are the property of the respective owners.

Contents

1.0.   Introduction

This document provides a general overview of PowerMAX OS™ Release Version 5.1(Release 5.1). Release 5.1 is designed to provide support for the following Concurrent Computer Corporation product:

- Night Hawk® HN6200/HN6800

The PowerMAX OS is based on UNIX® System V Release 4.2 MP with real-time enhancements provided by Concurrent Computer Corporation.

Release 5.1 is distributed on multiple tapes. The Base Installation tape contains standalone utilities, a bootable mini-kernel, system installation software, file system restore utility, and the base software package. The Additional Packages tape contains optional software packages that may be installed once the base package is installed.

A Service Release Tape containing updates, enhancements, patches, etc, may also be provided on an "as-released-basis" to users who are currently under a software maintenance contract.

2.0. Documentation

2.1. PowerMAX OS Software Documentation

Table 1-1 lists the PowerMAX OS documentation available from Concurrent Computer Corporation. Note that standalone release notes are available for the various platforms. The corresponding release notes will be provided with the applicable platform.

Table 1-1. PowerMAX OS Software Documentation

Manual Name	Pub. Number
PowerMAX OS Programming Guide	0890423-060
Character User Interface Programming	0890424-000
Device Driver Programming	0890425-060
STREAMS Modules and Drivers	0890426-020
User's Guide	0890428-010
System Administration Volume 1	0890429-050
System Administration Volume 2	0890430-060
Night Hawk HN6200/HN6800 PowerMAX OS Version 5.1 Release Notes	0890454-5.1
Compilation Systems Volume 1 (Tools)	0890459-050
Compilation Systems Volume 2 (Concepts)	0890460-050
PowerMAX OS Real-Time Guide	0890466-070
PowerMAX OS Documentation Overview	0890470-080
PowerMAX OS Guide to Real-Time Services	0890479-080

Copies of the Concurrent documentation can be ordered by contacting the Concurrent Software Support Center. The toll-free number for calls within the continental United States is 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822 or 1-305-931-2408.

Additionally, the manuals are available in pdf format on the Concurrent Computer Corporation web site at www.ccur.com.

Standalone product release notes are sometimes provided with software products. The release notes you receive will be at the software revision level that matches the associated software product level.

3.0. Prerequisites

3.1. Software

PowerMAX OS operating system Release 5.1 or later.

3.2. Hardware

3.2.1. CPU

HN6200/HN6800 Night Hawk (PowerPC 604^{TM} system.

3.2.2. System Console Terminal

The system console may be either:

- a video display terminal such as a Wyse 150, vt100, or comparable device connected to serial port 0 (COM1).
- a PC style keyboard/VGA/display combination, if equipped.

Supported VGA compatible devices may include the following:

- Built-in SVGA port of Power Hawk Model 640
- ACT/Technico PMC-8030 SVGA PMC card
- Matrox Millennium/Millennium II PCI cards

If equipped with a keyboard and one of these VGA compatible devices, the default console interface will be the keyboard and display. The default system console device can be modified via the console `tc` command.

3.2.3. SCSI Interface Controller

Minimum requirements are:

- VME Interface Adapter (VIA), PN 1580009. Minimum Rev is -1.
- At least one supported SCSI disk drive for system files and swap space.
- At least one supported SCSI tape device for software installation and updates.

3.2.4. Disk/Tape Drives

As stated above, each system must have a system disk and tape drive. The smallest disk drive supported for the installation disk (that contains the PowerMAX OS executable) is 2GB.

The installation disk may be on: an VIA located in slots 2 through 12; or on an ISE controller located on the processor boards; or on an HSA.

3.2.5. Ethernet Controllers

3.2.5.1. Interphase 4207 Eagle

VME-based Ethernet may be provided via the Interphase 4207 Eagle Ethernet Controller which supplies AUI connections. The minimum revision level of the ethernet controller board (PN 2010221) is Rev E.

3.2.5.2. Interphase 4221 Condor

VME-based Ethernet may be provided via the Interphase 4221 Condor Ethernet Controller which supplies AUI connections. The minimum revision level of the ethernet controller board (PN 2010316) is Rev A.

3.2.6. FDDI Controllers

3.2.6.3. Interphase Peregrine 4211

VME-based FDDI is provided via the Interphase 4211 (Peregrine 1) controller. The minimum revision level of the FDDI controller board (PN 2010225) is Rev C.

3.2.6.4. Interphase Peregrine 5211

VME-based FDDI is provided via the Interphase 5211 (Peregrine 2) controller. The minimum revision level of the FDDI controller board (PN 2010307) is Rev **-**.

3.2.7. HPS

VME-based asynchronous serial communications is provided by the High Performance Serial (HPS) interface controller. The minimum revision level of the HPS controller board (PN 2010218) is Rev D.

3.2.8. HSDE

VME-based HSD interface is provided by the High Speed Driver Enhanced (HSDE) controller. The minimum revision level of the HSDE controller board (PN 1573300) is Rev P.

3.2.9. 1553 V2

VME-based MIL-STD-1553 Version 2 (V2) interface is provided by the 1553 controller. The minimum revision level of the 1553 controller board (PN 2010209) is Rev C.

3.2.10. 1553 V5

VME-based MIL-STD-1553 Version 5 (V5) interface is provided by the 1553 controller. ABI-V5-1 is a Single Channel board, and ABI-V5-2 is a Dual Channel board.

3.2.11. DR11-W

VME-based controller with a DEC DR11-W protocol external channel interface. The minimum revision level of the DR11-W controller board (PN 2010179) is Rev **-**.

3.2.12. IEEE488

VME-based IEEE GPIB Bus Interface is provided by the IEEE488 controller. The minimum revision level of the IEEE488 controller board (PN 2010313) is Rev **-**.

3.2.13. Multiplexer VMEbus Controller (MVC)

VME-based asynchronous serial communication is provided by the Multiplexer VMEbus Controller.

3.2.14. X.25

VME-based X25 communication is provided by the VCOM-24 controller board (PN 2010305) or the VCOM-34 board (PN 2010276). The minimum revision of the VCOM-24 board is Rev A5. The minimum revision of the VCOM-34 board is Rev A3.

4.0. Available Software Packages

This section contains brief descriptions of all the available software packages. Note that your complement of available software may be less than that shown in Table 1-2 and entirely depends on the optional software packages you purchased.

The availability of a software package for **ALL** currently supported platforms is shown in Table 1-2 by a "**y**" (yes) or "**n**" (no) in the appropriate column.

Package dependencies are also specified in Table 1-2. Packages with dependencies must be installed after the packages they depend on.

When installing the optional package(s) from the system installation menu, you must also remember to select all required dependency package(s). The installation scripts will install the packages in the proper order.

Some optional packages are relocatable, that is, objects may be installed in an alternative directory other than root. When installing a relocatable package, the user will be prompted for an alternate installation path.

Note that all packages are dependent on `base`.

Table 1-2. Software Packages

Package Description	Night Hawk HN6200/ HN6800	Turbo Hawk	Power MAXION	Power Hawk Series 600	Power Hawk Series 700	Power Stack II/III	Package Name	Package Dependencies
Standard:								
Base System	y	y	y	y	y	y	base	
Advanced Commands	y	y	y	y	y	y	cmds	lp,nsu
Closed SARs	y	y	y	y	y	y	sar	
Domestic Encryption Utilities	y	y	y	y	y	y	crypt	
Fortran Libraries	y	y	y	y	y	y	hf77libs	
International Encryp-tion Utilities	y	y	y	y	y	y	crypt-int	
Kernel Debugger	y	y	y	y	y	y	kdb	
Network Support Utili-ties	y	y	y	y	y	y	nsu	
OA&M	y	y	y	y	y	y	oam	cmds
Online Manual Pages	y	y	y	y	y	y	man	
Printer Support	y	y	y	y	y	y	lp	
Program Analyzer	y	y	y	y	y	y	analyze	
Software Packaging Tools	y	y	y	y	y	y	softint	

Table 1-2. Software Packages (Cont.)

Package Description	Night Hawk HN6200/ HN6800	Turbo Hawk	Power MAXION	Power Hawk Series 600	Power Hawk Series 700	Power Stack II/III	Package Name	Package Dependencies
Standard(Cont):								
Terminfo Utilities	y	y	y	y	y	y	terminf	
Drivers:								
Condor Ethernet Driver	y	y	y	y	y	n	cnd	nsu
CD-ROM Driver	y	y	y	y	y	y	cdfs	
DR11W Driver	y	y	y	y	n	n	dr11w	
Eagle Ethernet Driver	y	y	y	y	y	n	egl	nsu
Interphase 5526 Fibre Channel Driver	n	n	n	n	n	y	fibre	
Generic CD-ROM Driver	n	n	n	y	y	n	gr	
Generic SCSI Device Driver	y	y	y	y	y	y	gs	
High Performance Serial Driver(HPS)	y	y	y	y	y	n	hps	

Table 1-2. Software Packages (Cont.)

Package Description	Night Hawk HN6200/ HN6800	Turbo Hawk	Power MAXION	Power Hawk Series 600	Power Hawk Series 700	Power Stack II/III	Package Name	Package Dependencies
Drivers (Cont):								
HSDE	y	y	y	y	y	n	hsde	
Peregrine FDDI Driver	y	y	y	y	n	n	pg	nsu
X.25 Driver	y	y	y	y	n	n	ix25	
1553 V2 ABI Driver	y	y	y	y	n	n	1553drv	
1553 V2 ABI Libraries	y	y	y	y	n	n	1533lib	1553drv
1553 V5 ABI Driver	y	y	y	y	y	n	1553v5drv	
1553 V5 ABI Libraries	y	y	y	y	y	n	1533v5lib	1553v5drv
Integral SCSI/Ethernet (ISE) - ISE SCSI Driver - ISE Ethernet Driver	y y y	n n n	n n n	n n n	n n n	n n n	ise is ie	 ise ise,nsu
NCR SCSI Driver	n	y	y	y	y	y	ncr	
DEC Ethernet Driver	n	y	y	y	y	y	dec	nsu
Symbios Fast Ethernet Driver	n	n	n	n	y	n	sym	
Internal IDE/ATA Disk Controller	n	n	n	n	n	y	ide	

Table 1-2. Software Packages (Cont.)

Package Description	Night Hawk HN6200/ HN6800	Turbo Hawk	Power MAXION	Power Hawk Series 600	Power Hawk Series 700	Power Stack II/III	Package Name	Package Dependencies
Drivers (Cont):								
Intelligent Bus Interface Module (IBIM)	y	y	y	y	n	n	ibim	
MVME300 IEEE 488	y	y	y	n	n	n	mvme300	
VIA SCSI Adapter Driver	y	y	y	y	y	n	via	
Parallel Port Driver	n	n	n	y	n	y	lpt	
Multiplexer VMEbus Controller	y	y	y	y	y	n	mvc	
National Instruments GPIB User-Level Device Driver (GPIB)	y	y	y	y	y	n	ngpib	
VMIC High Speed Data (HSD) Driver	y	y	y	y	n	n	vhsd	
TCP/IP Networking:								
Internet Utilities	y	y	y	y	y	y	inet	nsu
Commands Networking Extension	y	y	y	y	y	y	netcmds	lp,inet

Table 1-2. Software Packages (Cont.)

Package Description	Night Hawk HN6200/ HN6800	Turbo Hawk	Power MAXION	Power Hawk Series 600	Power Hawk Series 700	Power Stack II/III	Package Name	Package Dependencies
TCP/IP Networking(Cont):								
Remote Procedure Calls Utilities	y	y	y	y	y	y	rpc	inet
Network File System:								
Network File System Utilities	y	y	y	y	y	y	nfs	nsu, inet, rpc, dfs
Distributed File System Utilities	y	y	y	y	y	y	dfs	inet
Security:								
Auditing	y	y	y	y	y	y	audit	
Frequency-Based Scheduler (FBS):								
Frequency-Based Scheduler and Performance Monitor	y	y	y	y	y	y	fbs	
Frequency-Based Scheduler Manual Pages	y	y	y	y	y	y	fbsman	

Table 1-2. Software Packages (Cont.)

Package Description	Night Hawk HN6200/ HN6800	Turbo Hawk	Power MAXION	Power Hawk Series 600	Power Hawk Series 700	Power Stack II/III	Package Name	Package Dependencies
Closely-Coupled Systems and Loosely-Coupled Systems:								
Diskless	n	n	n	y	y	y	diskless	cmds, nfs, netcmds, dec, and ncr or via
Software Development:								
Concurrent Compilation System	y	y	y	y	y	y	hc	
Fortran 77 Compilation System	y	y	y	y	y	y	hf77	
C++ Compilation System	y	y	y	y	y	y	c++	
Ada Programming Support Environment (HAPSE)	y	y	y	y	y	y	ada	analyze
Ada X Interface (AXI)	y	y	y	y	y	y	axi	ada, x11
Ada Runtime Shared Libraries	y	y	y	y	y	y	ada_rts	
MAXAda Compilation System	y	y	y	y	y	y	MAXAda	analyze
AXI for MAXAda	y	y	y	y	y	y	MAXaxi	MAXAda, x11

Table 1-2. Software Packages (Cont.)

Package Description	Night Hawk HN6200/ HN6800	Turbo Hawk	Power MAXION	Power Hawk Series 600	Power Hawk Series 700	Power Stack II/III	Package Name	Package Dependencies
Software Development(Cont):								
Cross-Development Environment	y	y	y	y	y	y	xdev43nh	
							xdev43moto	
							xdev51nh	
							xdev51moto	
							xdev51synergy	
Window System:								
X Window System with OSF/Motif:	y	y	y	y	y	y	x11ipc	
							x11	x11ipc
							x11progs	x11ipc, x11
							x11dev	x11ipc, x11, x11progs
							x11pman	
							x11dman	
Metro-X X11R6 Server	n	n	y	y	y	y	metroess	
X11R6 Fonts	n	n	y	y	y	y	xfonts	

Table 1-2. Software Packages (Cont.)

Package Description	Night Hawk HN6200/ HN6800	Turbo Hawk	Power MAXION	Power Hawk Series 600	Power Hawk Series 700	Power Stack II/III	Package Name	Package Dependencies
Window System(Cont):								
Metro Link OpenGL	y	y	y	y	y	y	metrogl	
NightStar Tools:								
NightBench	y	y	y	y	y	y	nbench	x11
NightProbe	y	y	y	y	y	y	nprobe	x11
NightSim	y	y	y	y	y	y	nsim	x11
NightTrace	y	y	y	y	y	y	ntrace	x11
NightTune	y	y	y	y	y	y	ntune	x11
NightView	y	y	y	y	y	y	NightView	x11
NightView Debugger Support	y	y	y	y	y	y	Nviewp	
License Manager:								
Élan License Manager5	y	y	y	y	y	y	elan5lm	
Élan License Manager4	y	y	y	y	y	y	elan4lm	elan5lm

Table 1-2. Software Packages (Cont.)

Package Description	Night Hawk HN6200/ HN6800	Turbo Hawk	Power MAXION	Power Hawk Series 600	Power Hawk Series 700	Power Stack II/III	Package Name	Package Dependencies
Miscellaneous:								
Distributed XFS File System	y	y	y	y	y	y	xfsd	inet, rpc
Virtual Partition	y	y	y	y	y	y	vp	
VERITAS Volume Manager	y	y	y	n	n	n	vxvm	nsu

4.1. Package Descriptions

The following pages contain a brief description of **ALL** of the packages available on PowerMAX OS. Refer to Table 1-2 to determine if any given package listed is applicable to your particular system.

4.1.1. Description of Standard Packages

Base System (`base`)

> The Base System package provides the `base` set of commands and system utilities.

Printer Support (`lp`)

> Although some printing capabilities are provided in the `base` package, more advanced printing capabilities, and a wider range of printers, are included in the Printer Support package.

Network Support Utilities (`nsu`)

> The Network Support Utilities package provides the basis on which networking capabilities are built.

Terminfo Utilities (`terminf`)

> The Terminfo Utilities Package provides support for a wide variety of terminals beyond those provided in the `base` package.

Advanced Commands (`cmds`)

> The Advanced Commands package provides the remaining user and administrative commands.

Program Analyzer (`analyze`)

> This package provides utilities for performance analysis and post-linker optimization.

hf77libs (`hf77libs`)

> This package provides runtime libraries for the Fortran 77 Compilation System.

OA&M (`oam`)

> The Operations Administration and Maintenance package provides a character-based, menu-oriented interface to a wide variety of advanced, server-oriented administrative tasks.

Software Packaging Tools (`softint`)

> This package provides tools to support the development process and includes a variety of archive libraries as well as tools to create and modify packages.

Kernel Debugger (`kdb`)

> The Kernel Debugger package provides a tool to assist in the porting and debugging of kernel modules and drivers by allowing the developer to examine and control a running kernel.

Domestic Encryption Utilities (`crypt`)

> The Domestic Encryption Utilities package supports the encryption of files and other data. This package is for distribution in the United States.

International Encryption Utilities (`crypt-int`)

> Same as above but for international distribution.

Online Manual Pages (`man`)

> System manual pages (man pages) provided in an on-line format for viewing using the `man` command.

Closed SARs (`sar`)

> Software Action Reports (SARs) closed in this release can be found in file **/usr/src/PRODUCTS/SARS.CLOSED**.

4.1.2.   Description of Drivers Packages

Condor Ethernet Driver (`cnd`)

> This package supports the Condor Ethernet 4211 VME board. Up to 6 Condor Ethernet boards are supported.

CD-ROM File System (`cdfs`)

> The CD-ROM File System package provides read-only access to file systems on SCSI CD-ROM devices. ISO-9660 and High Sierra formats are supported.

Eagle Ethernet Driver (`egl`)

> This package supports the Interphase Ethernet 4207 Eagle VME board. Up to 6 Eagle Ethernet boards are supported.

Fibre Channel Driver (`fibre`)

> This driver package supports the Interphase 5526 Fibre Channel Driver in an arbitrated loop configuration. This package also provides for automatic configuration of the controller during system initialization (e.g., no **Sadapters(4)** file changes required). Up to 126 nodes, where a node may be a Fibre channel disk or a Fibre Channel adapter card attached to a computer host, may be attached to the arbitrated loop. This package also contains the **fibconfig(1M)** utility, which may be used to provide additional Fibre channel configuration control and status information.

Parallel Port Driver (`lpt`)

> This driver package supports the parallel port on the Power Hawk Series 600 systems. Only one parallel port is supported per system. This driver supports local printing to a directly connected printer.

Peregrine FDDI Driver (`pg`)

> This package supports the Interphase FDDI 4211 and 5211 Peregrine VME boards. Up to 3 Peregrine VME boards are supported.

Generic CD-ROM Device Driver (`gr`)

> The `gr` driver is a generic CD-ROM device driver. This driver supports the SCSI CD ROM device connected to the internal NCR/Symbios controller.

Generic SCSI Device Driver (`gs`)

> The `gs` driver package supports the following SCSI devices: SCSI-based CPUs, scanners, printers, media changers, and communication devices. The `gs` driver communicates with the HSA and VIA controllers/drivers using the drivers passthru options to send SCSI commands to the appropriate SCSI devices.

High Performance Serial Driver (`hps`)

> This package supports the High Performance Serial adapter, a VME board providing 16 asynchronous serial ports running up to 38400 baud, and 1 optional Centronics parallel printer port. Up to eight HPS adapters are supported.

VIA SCSI Adapter Driver (`via`)

> This driver package supports the following:

>> a. SCSI Adapter Interface (hsa), an HVME board providing mass storage capability to the system. Up to 7 Concurrent specified SCSI disks or tapes may be connected to a single HSA board. HSA boards are available only on HN6200, HN6800 and TurboHawk systems.

>> b. VME Interface Adapter for SCSI (via), a VME board providing mass storage capability to the system. Up to 30 (if using 16-bit wide SCSI) Concurrent specified SCSI disks or tapes may be connected to a single VIA board. The VIA board supports up to two optional daughter cards. Each daughter card can be one of the following:

>>> 1. SCSI-2 card that supports fast and wide single ended SCSI-2 transfers.

>>> 2. SCSI-2 differential card that supports fast and wide transfers.

> Refer to online manual page **dlvia(8)** for information on how to download VIA-board firmware.

Intelligent Bus Interface Module (`ibim`)

> This driver package supports the IBIM module. The IBIM module is a data acquisition module that acts as a host platform for a variety of analog and digital interface modules. The IBIM can support up to four daughter boards.

NCR/Symbios SCSI Driver (`ncr`)

> This driver package supports the internal SCSI controller chip and the controller chips on separate PMC or PCI cards. This package also provides for automatic configuration of the controller during system initialization (for example, no **Sadapters(4)** file changes required). Up to seven disks and/or tape drives may be connected to this SCSI bus.

Internal IDE/ATA Driver (`ide`)

> This driver package supports up to four ATA-2 or ATA-3 disk drives attached to the internal IDE/ATA disk controller on PowerStack II systems only. This package also provides for automatic configuration of the controller during system initialization (e.g., no **Sadapters(4)** file changes required). (The ability to format an IDE drive is not available. IDE drives are formatted at the factory and cannot be reformatted in the field. In addition, there is no verify function.)

DEC 21x4x Ethernet Driver (`dec`)

> This driver package supports the Ethernet controller chip on the TurboHawk, PowerMAXION, Power Hawk Series 600, PowerStack II and Motorola MCP750 processor cards along with DEC 21x4x controller chips on separate PMC or PCI

cards. Supported chips include the 21040, 21140A, 21142 and 21143. This package also provides for automatic configuration of the controller during system initialization (e.g., no **Sadapters(4)** file changes required).

100baseT (except on the 21040), 10baseT, BNC and AUI connections are supported. The Power Hawk Model 620/640 processor contains a 21140A with 10baseT or 100baseT connections. The TurboHawk has an embedded DEC 21143 with 10baseT or 100baseT connections.

Symbios Fast Ethernet Driver (`sym`)

This driver package supports the Symbios 53C885 Ethernet controller chip on the Power Hawk Series 700 SBCs.

High Speed Data Enhanced Channel Driver (`hsde`)

This package supports the HSDE Channel Interface. The HSDE provides high-speed, 32-bit parallel bidirectional link for transferring control, status and data between the HN6200/HN6800/TurboHawk (H)VMEbus system and an external device using the Encore HSD Interface Model 9132 protocol.

1553 V2 ABI Driver (`1553drv`)

This package provides a user-level device driver for the Version 2 (V2) Advanced Bus Interface (ABI) MIL-STD-1553 Adapter.

1553 V2 ABI Libraries (`1553lib`)

This package provides program interfaces that can be used within an application program. This interface is the same as those provided by the manufacturer of the 1553 board, SBS Engineering, Inc.

1553 V5 ABI Driver (`1553v5drv`)

This package provides a user-level device driver for the Version 5 (V5) Advanced Bus Interface (ABI) MIL-STD-1553 Adapter board manufactured by SBS Engineering, Inc.

1553 V5 ABI Libraries (`1553v5lib`)

This package provides program interfaces which can be used within an application program to control the ABI-VI MIL-STD-1553 Adapter. This set of interfaces is an enhanced version of the set provided by the manufacturer of the board, SBS Engineering, Inc.

Motorola MVME300 IEEE488 Interface Driver (`mvme300`)

This package supports Motorola's MVME300 IEEE 488 bus interface controller. Up to 8 MVME300 controllers are supported. Available on Night Hawk systems only.

Ikon DR11W Driver (`dr11w`)

This package supports the Ikon 10089 DR11W emulator board. Up to 16 DR11W boards are supported (8 in the Primary I/O bus, 8 in the Secondary I/O bus).

Integral SCSI/Ethernet Controller (`ise`)

Provides base support for the Integral SCSI/Ethernet daughtercards (ISE) available with HN6200/6800 systems only. Package supports up to 4 ISE cards (one per processor board), and provides for automatic configuration of ISE cards during system initialization (e.g., no **Sadapters(4)** file changes required).

ISE - SCSI (`is`)

> Provides driver support for up to seven SCSI peripherals for each configured ISE daughtercard.

ISE - Ethernet (`ie`)

> Provides Ethernet/IEEE 802.3 local area network driver support for each configured ISE daughtercard.

X.25 Driver (`ix25`)

> This package supports the X.25 high speed synchronous communication module that runs X.25/LAPB/HDLC.

Multiplexer VMEbus Controller (`mvc`)

> This package supports the Multiplexer VMEbus Controller (MVC) adapter board. The MVC adapter board provides 16 asynchronous serial ports running at up to 38.4 baud, and one optional Centronic parallel printer port. Up to four MVC adapter boards are supported.

National Instruments GPIB User-Level Device Driver (`ngpib`)

> The `ngpib` user-level device driver provides programmers with a configuration utility and library functions to access and control a National Instruments General Purpose Interface Bus (GPIB) 1014D board. This interface is a dual ported, high performance interface to the IEEE-488 bus, and is capable of controlling two independent GPIB bus configurations.

vhsd Driver (`vhsd`)

> This driver provides support for the VMIC VMIVME-5620 Intelligent HSD Emulator on the Power Hawk Series 600 systems.

4.1.3. Description of TCP/IP Networking Packages

Internet Utilities (`inet`)

> The `inet` package includes the software needed to run the `TCP/IP` network and tools such as `ftp`, `telnet`, and `rcp`. Administrative software for setting up the network is also included.

Commands Networking Extension (`netcmds`)

> The Commands Networking Extension Package extends the functionality of several basic commands by supporting the means to share printers across a network, and use additional transport mechanisms for the sending and receiving of electronic mail.

Remote Procedure Calls Utilities (`rpc`)

> The Remote Procedure Calls Utilities package supports the remote execution facility.

4.1.4. Description of Network File System Packages

Network File System Utilities (`nfs`)

> The Network File System Utilities package supports the means to transparently share resources across a network with other computers running the Network File System.

Distributed File System Utilities (`dfs`)

> The `dfs` utilities package provides a simple user interface for performing networked operations such as advertising local resources and accessing remote resources.

4.1.5. Description of Security Package

Auditing (`audit`)

> The Auditing package provides auditing facilities allowing a system administrator or security auditor to record and report all security-related events that occur on the system.

4.1.6. Description of Diskless Package

Diskless (`diskless`)

> This package provides support to configure and control multiple Power Hawk single-board computers (SBC) in a common VME chassis (referred to as a closely-coupled system). Diskless SBCs in this chassis can be booted over the VMEbus by a single server SBC in the rack, where the server SBC is attached to system disk(s) that contain the system object images for all diskless SBCs in the chassis.

> A closely-coupled system may consist of either Power Hawk 620/640 SBCs, or Power Hawk 720/740 SBCs.

> On Power Hawk 720/740 closely-coupled systems, an additional PCI-to-PCI (P0) bus also connects all SBCs in the chassis, and this P0Bus provides a high-speed connection for inter-SBC communications, including a point-to-point network interface.

> On Power Hawk 620/640 closely-coupled systems, the VMEbus is used for inter-SBC communications, including the point-to-point network interface.

> This package also supports multiple SBCs in one or more remote VME chassis connected via Ethernet (referred to as a loosely-coupled system). Diskless SBCs in these remote VME chassis can boot themselves via TFTP over the Ethernet, with a single Power Hawk server SBC providing the downloadable object images required for each diskless SBC.

4.1.7. Description of FBS Packages

Frequency-Based Scheduler and Performance Monitor (`fbs`)

> This package provides kernel support for the Frequency-Based Scheduler and Performance Monitor and Real-Time Command Processor.

Frequency Based Scheduler Manual Pages (`fbsman`)

> This package provides the man pages associated with the Frequency-Based Scheduler and Performance Monitor.

4.1.8. Description of Software Development Packages

Concurrent C Compilation System (`hc`)

> This package provides the hc C compiler. This compiler offers ANSI C compliance and support for pre-ANSI C, together with other extensions.

Fortran 77 Compilation System (`hf77`)

> This package provides the hf77 Fortran compiler and runtime libraries. This compilation system offers Fortran 77 and MIL-STD 1753 compliance, together with many popular extensions and a cross-reference tool with interface checking.

C++ Compilation System (`c++`)

> The 5.1 version of c++ utilizes Edison Design Group's C++ front end and Concurrent's proprietary Common Code Generator (CCG) technology to produce highly optimized object code tailored to the Night Hawk., Power Hawk., TurboHawk., PowerMAXION, and PowerStack systems. The 5.1 release combines into a single compiler what were previously separate C and C++ compilers. This release does not require un-installing previous versions of the C and C++ compilers.

Ada Programming Support Environment (`ada`)

> This package consists of a validated Ada compiler, library, management tools, symbolic debugger, automated build utility, Ada bindings, real-time monitoring, the Ada Real-Time Multiprocessor tasking executive (`ARMS`) and runtime support libraries.

Ada to X Interface (AXI) (`axi`)

> This package consists of the Ada to X Window system interface.

Ada Runtime Shared-Libraries (`ada_rts`)

> This package consists of the compiled and linked form of HAPSE Runtime Shared-Libraries; including the basic Ada Real-Time Multiprocessor System (ARMS) standard libraries, Ada bindings, and others.

MAXAda Compilation System (`MAXAda`)

> This package consists of the MAXAda™ tool set for the development of Ada programs on Concurrent computers under the PowerMAX OS environments. MAXAda processes the Ada language as specified by the Reference Manual for the Ada Programming Language, ANSI/ISO/IEC-8652:1995.

Ada to X Interface for MAXAda (`MAXaxi`)

> This package consists of the Ada X Window system interface for MAXAda.

Cross-Development Environment (`xdev43nh, xdev43moto, xdev51nh, xdev51moto, xdev51synergy`)

> Consists of five packages that contain all the header files and libraries required for cross development. The Cross-Development Environment allows users to develop applications from any supported Concurrent real-time computer systems targeting a specific PowerMAX OS release and architecture. (New for the PowerMAX OS 5.1 Release.)
>
> Package names and short package descriptions follow:

xdev43nh	The cross development environment package for the Night Hawk Series 6000. Includes all PowerMAX OS header files and libraries for the PowerMAX OS 4.3 release for the Night Hawk Series 6000 hardware family (Night Hawk HN6200/HN6800, PowerMAXION and TurboHawk).
xdev43moto	The cross development environment package for the Power Hawk Series 600. Includes all PowerMAX OS header files and libraries for the PowerMAX OS 4.3 release for the Power Hawk Series 600 hardware family (Power Hawk 620/640 and PowerStack II/III).
xdev51nh	The cross development environment package for the Night Hawk Series 6000. Includes all PowerMAX OS header files and libraries for the PowerMAX OS 5.1 release for the Night Hawk Series 6000 hardware family (Night Hawk HN6200/HN6800, PowerMAXION and TurboHawk).
xdev51moto	The cross development environment package for the Power Hawk Series 600. Includes all PowerMAX OS header files and libraries for the PowerMAX OS 5.1 release for the Power Hawk Series 600 hardware family (Power Hawk 620/640 and PowerStack II/III).
xdev51synergy	The cross development environment package for the Power Hawk Series 700. Includes all PowerMAX OS header files and libraries for the PowerMAX OS 5.1 release for the Power Hawk Series 700 hardware family (Power Hawk 720/740).

4.1.9. Description of X Window System Packages

X Window System Version 11, Release 6 (`x11`)

> Includes OSF/Motif commands, libraries and header files. X Window System sub-package names and short descriptions follow:

x11ipc	the libraries for ICE and ktalk only
x11	all other libraries including runtime support files they reference
x11progs	X client programs xdm, mwm, application-default files, etc.

x11dev	X program development tools, header files, imake, static libs, etc.
x11pman	man pages for all the application level programs.
x11dman	man pages for the libraries and program development tools

Metro-X Enhanced X11R6 server set and OpenGL (`metroess`, `xfonts` & `metrogl`)

Jointly developed with Metro Link, Inc., these packages provide native graphics capabilities on systems equipped with appropriate display and input devices.

Specific packages include the following:

metroess   Metro Links Enhanced X11R6 Server Set includes a powerful, configurable, host based X11R6 server derived from the X11R6/XFree86 sample servers.

xfonts   Local X11 system fonts, which are required for server installations that do not use a font server.

metrogl   OpenGL version 1.2 for Metro-X. Includes OpenGL client libraries (libGL, libGLX, libGLU), header files, sample clients, and GLX server extension for rendering OpenGL commands with the Metro-X X11R6 server.

4.1.10. Description of NightStar Tools Packages

NightBench (`nbench`)

NightBench is a graphical user interface to MAXAda, a tool set for the development of Ada programs running under the OS.

NightProbe (`nprobe`)

The NightProbe Data Monitoring package provides a real-time graphical tool for monitoring, recording, and altering program data within one or more executing programs without significantly affecting the execution of those programs.

NightSim (`nsim`)

This package provides a graphical tool for building and monitoring real-time applications that require scheduled process execution patterns. NightSim provides a GUI to the Frequency-Based Scheduler (FBS) and the Performance Monitor facilities.

NightTrace (`ntrace`)

This package provides a graphical tool for analyzing dynamic behavior of multi process and/or multiprocessor applications.

NightTune (`ntune`)

This package provides an interactive, graphical tuning and monitoring environment specifically designed for Concurrent's real-time computer systems.

NightView (`NightView`)

This package provides a general-purpose, source-level debugger for C, C++, Fortran and Ada with support for multiple processes.

NightView (`Nviewp`)

This package provides a graphical, non intrusive, source-level monitoring and debugging tool specifically designed for real-time applications.

4.1.11. Description of License Manager Packages

Élan License Manager (`elan5lm`)

> This package contains the license manager daemon and reporting commands required by license-managed applications, including all NightStar tools. This version is Y2K compliant.

Élan License Manager (`elan4lm`)

> This package is an optional 'old' (version 4) license manager daemon that can run in parallel with the new (elan5lm) license manager daemon on the same server machine in order to support serving licenses to both 'old' and 'new' tools at the same time.

4.1.12. Description of Miscellaneous Packages:

Distributed XFS File Systems (`xfsd`)

> The xfsd distributed file system package provides shared access to a single file system from multiple host systems on a multi-initiator disk bus. Unlimited read access to files in the shared file system is supported. Only limited write access to files in the shared file system is supported. A network connection between systems that access a shared file system is required for communication and synchronization of file system structures.

Virtual Partition (`vp`)

> This package provides a pseudo device driver that performs disk striping (RAID level 0) or mirrored partitioning (RAID level 1).

VERITAS Volume Manager (`vxvm`)

> This package provides the system administrator with a disk management tool. The '*System Disk Configuration*' section later in this document describes how the disk must be configured when using the VERITAS Volume Manager.

5.0. System Installation

The PowerMAX OS operating system is installed as software packages using the Software Packaging Tools. Two installation modes, Custom and Semi-Automatic, are now available. Refer to *Installation Modes* on page 29 for additional details.

5.1. Large SCSI Disk Support

Changes were introduced in a previous PowerMAX OS release to correct the support for SCSI disks larger than 8 Gigabytes. These corrections changed the location of the Geometry Block on these large disks. As a result, disks larger than 8 GB formatted using the **format(1m)** or **format(8)** standalone utility prior to Release 4.2 cannot be used to install Release 5.1 without being reformatted.

If you have a disk larger than 8 GB that was formatted using the release 4.1 version of **format()**, you must do the following:

1. Backup any data needed on any of the partitions before installation.

2. During the installation, exercise the option to format the disk and rewrite the partition information (Geometry Block).

3. Finish the Release 5.1 installation.

4. Restore any data backed up in step 1 above.

5.2. System Disk Configuration

The system disk configuration, shown in Table 1-3, is suggested for installing the system using an ufs root file system type.

Table 1-3. ufs root file system

partition	file	minimum size (formatted
0	root	100 MB
1	swap	96 MB
2	usr	500 MB
3	var	400 MB
4	----	\<remainder\>
6[1]	boot	1024 KB

The system disk configuration shown in Table 1-4 is suggested for installing the system using an xfs root file system type.

Table 1-4.   xfs root file system

partition	file	minimum size (formatted
0	/stand	60 MB
1	swap	96 MB
2	usr	500 MB
3	var	400 MB
4	----	\<remainder\>
5[2]	root	60 MB
6[1]	boot	1024 KB

1. Partition is applicable to Power Hawk and PowerStack II/III boot disks only.

2. The **xfs** root must be on partition 5. (**Caution**: cannot use VxVM with **xfs** root file system.)

Partition sizes may be increased and new partitions may be added, but the above assignments of file systems to partitions and minimum partition sizes must be maintained.

During system installation, you will be given the option of running the **format(1M)** command. You should choose this option to format the system disk and to select partition sizes.

The **format(1M)** command "partition default" automatically selects the above partition sizes for a ufs root file system.

Note that the **format(1M)** "partition default" option should not be used if an xfs root file system is selected as the default partitions automatically selected are not appropriate for the **xfs** file system.

The root and usr file systems should only contain system files and are not expected to grow much after system installation. The var file system contains system crashfiles, log(s) and temporary files. It is strongly recommended that user files be restricted from these file systems.

Swap space needs should be carefully calculated. Adding too little swap space will result in unnecessary out-of-memory conditions for your applications. Adding too much swap space will result in too much of your system's memory being locked down for swap space management. The total amount of swap space should be at least 1.5 times the size of physical memory. An initial swap partition is provided on the system disk. If this partition is insufficient, it is recommended that additional swap partitions be added, preferably on other disks.

Every Gigabyte of swap space results in 4 Megabytes of physical memory being used for swap space management. This rule demonstrates that it would be impractical to create a 9 Gigabyte swap device on a 64 Megabyte system, as this would result in more than half (36 Megabytes) of physical memory being utilized by the kernel for swap space management.

Note that partition four is left unused. This partition may be:

1. Redistributed to make the other system partitions bigger.

2. Used for additional swap space.

3. Used for user files (for example a home file system).

Use the **format(1M)** "**?**" command for help with format commands. Refer to the **format(1M)** manual page at the end of these release notes. Note that non-system disks will need to be initialized once the system is re-booted for new disk. This includes running **format(1M)** to format and partition the disk and **newfs(1M)** to initialize the file systems. Additional steps include creating a mount point directory, adding the appropriate information to **/etc/vfstab** and adding new entries to the **Device Database (DDB)**. See the *System Administration Manual* for information on disk formatting and partitioning.

Note: All disks, including the system disk, that are to be used by the VERITAS Volume Manager (VxVM) package have special configuration requirements. Note however that Power Hawk and PowerStack II/III boot disks cannot be used by VxVM. These configuration requirements are listed below.

1 Partition 5 must be left unused, and configured with a size of zero. (**Caution**: cannot use VxVM with **xfs** root file system.)

2 Partition 6 must be configured with a size of 512K bytes (1024 sectors).

Partitions 5 and 6 are used exclusively by VxVM and cannot be used by users or the system for other uses. If partitions 5 and 6 are not configured as described, then the disk *cannot* be used by VxVM.

Use the **format(1M)** command to format the disks to the specifications described above if VxVM is to be used.

5.3. Installation Modes

Early in the system installation, the operator will be asked whether to perform a custom or semi-automatic system installation. A description of each mode is provided in the following paragraphs.

5.3.1. Custom Mode of Installation

In custom mode, the operator is prompted during the installation in order to specify the values for configurable items. Custom mode is recommended if the pre-determined values assigned in the semi-automatic mode are not appropriate for your site, and/or you want the option of installing only certain packages during system installation.

5.3.2. Semi-Automatic Mode of Installation

In semi-automatic mode, the operator still specifies the basic configuration of the system however, the remainder of the installation is done with a pre-determined set of responses. Note that in semi-automatic mode, **all** the additional products on the tape are automatically installed. The operator does not have the option of installing only specific packages.

Refer to Table 1-5 for a description of the configurable items and the values that they will be assigned on a semi-automatic installation.

Semi-automatic mode can be used if the configurable values are set appropriately for the given site and all packages are being installed. Otherwise, custom mode should be used.

NOTE

Refer to Table 1-2 to determine if a given package listed in Table 1-5 is available for your particular system.

Table 1-5. Assigned Installation Values, Semi-Automatic Mode

Package Name	Configurable Item	Automatic Installation Value
base	Host nodename Root password License key - number of users License key - number of processors	Obtained during initial installation "" (null password) Automatically configured. Automatically configured.
elan5lm	Install man pages? Start license manager during system boot?	yes yes
oam	sysadm password	"" (null password)
cnd	Number of cnd adapters on primary VME bus Number of cnd adapters on secondary VME bus	1 0
dr11w	Number of dr11w adapters on primary VME bus Number of dr11w adapters on secondary VME bus	1 0
egl	Number of egl adapters on primary VME bus Number of egl adapters on secondary VME bus	1 0
hps	Number of hps adapters on primary VME bus Number of hps adapters on secondary VME bus Configure real-time driver?	1 0 no
hsde	Number of hsde adapters on primary VME bus Number of hsde adapters on secondary VME bus	1 0
ibim	Number of ibim adapters on primary VME bus Number of ibim adapters on secondary VME bus	1 0
pg	Number of pg adapters on primary VME bus Number of pg adapters on secondary VME bus	1 0
ix25	Number of ix25 adapters on primary VME bus Number of ix25 adapters on secondary VME bus Install man pages?	1 0 yes
1553drv	Number of 1553V2-ABI adapters	1

Table 1-5. Assigned Installation Values, Semi-Automatic Mode (Cont.)

Package Name	Configurable Item	Automatic Installation Value
1553v5drv	Number of 1553V5-ABI adapters	1
mvc	Number of mvc adapters on primary VME bus Number of mvc adapters on secondary VME bus Configure real-time driver?	1 0 no
mvme300	Number of mvme300 adapters	1
inet	Configure TCP listener?	yes
ada	Install directory for HAPSE	standard location relative to root directory
axi	Install directory for Ada X Interface (AXI) for HAPSE	standard location relative to root directory
MAXAda	Install directory for MAXAda	standard location relative to root directory
MAXaxi	Install directory for Ada X Interface (AXI) for MAXAda	standard location relative to root directory
x11ipc x11 x11progs x11dev x11pman x11dman	Install header files and static libraries? Install man pages? Start xdm during system boot?	yes yes yes
nbench	Install directory for NightBench	standard location relative to root directory
nprobe	Install directory for NightProbe	standard location relative to root directory
nsim	Install directory for NightSim	standard location relative to root directory
ntrace	Install directory for NightTrace	standard location relative to root directory
ntune	Install directory for NightTune	standard location relative to root directory
NightView	Install directory for NightView	standard location relative to root directory
Nviewp	Install directory for Nviewp	standard location relative to root directory

5.4.   Installation Procedure

To begin system installation on a halted system, insert the Base Installation Tape and
execute the following console commands (where c = controller or slot number, and u =
device unit designation of tape drive containing the Base Installation Tape.)

```
#>p boot 0
#>fd mt(c,u,1)  (typically, u=5 for most systems with
                      one tape drive)
#>fb
```

(**Note**: Ignore "`Device Not Responding...`" message. "`OK`" message should appear.

These commands will boot **/stand/unix** from tape. The time to complete the entire
installation will vary depending on the packages selected to be installed and the type of tape
drive being used.

As the system is brought up, the initial menu will prompt you to choose either the system
software installation program or the file system restore program. At this prompt, choose
submenu 1 INSTALL. (The file system restore program is documented in Chapter 10 of the
System Administration, Volume 2 (Pubs No. 0890430).) Prompts that require user input are
preceded by **=>**. Most prompts have defaults in parentheses that may be selected by
pressing the "**Enter**" key. At any prompt, "**?**" can be entered for help or **q** to quit
installation. If the user selects to quit installation or if a fatal error occurs, installation will
be suspended by executing a subshell. Three choices are available when exiting the
subshell:

> **CONTINUE** - repeat last operation
> **RESTART** - start over from the beginning
> **ABORT**   - return to console debugger

The installation is self-guiding, but the following configuration information is required
from the user:

Installation Mode (choose custom or semi-automatic (see *Installation Modes* on page 29)

Node name

Timezone

Time/Date

Desired file system types

System disk location (slot and unit number)

Tape drive location (if more than one drive in system)

System disk configuration (see *System Disk Configuration* on page 27)

Following are applicable to custom mode installation only:

Which additional software packages should be installed (see *Available Software Packages*
on page 6)

Configuration information requested by the various packages

If appropriate, any kernel modules from the optional packages you want to deconfigure
(that is, not link with the kernel)

Note: The installation tapes are accessed at various times and must be kept in the drive during the installation until you are instructed to install a different tape.

After all the packages have been installed, you will be given an opportunity, if in the custom installation mode, to deconfigure kernel modules from optional packages. When deconfigured, those drivers will not be linked with the kernel. Note that you must consider package dependencies when deconfiguring drivers. See Table 1-2 for software package dependency relationships.

At the end of the installation procedure, a kernel for the newly installed system will be built.

5.5.   Standalone Commands

The standalone commands are shipped on the `Base` Installation Tape as diagnostic aids. These are not required for installation.

The following standalone commands are available:

> **ls(8),format(8),cat(8),fastcopy(8),dlvia(8)**

To run the standalone commands, execute the following commands from the console terminal. (Where *c* = controller or slot number, and *u* = device unit designation of tape drive containing the Base Installation Tape.)

```
#> p boot 1
#> fd mt(c,u,1)
#> fb
  Boot
  : [cmd]
```

5.6.   Installing Additional Packages

In the semi-automatic installation mode **all** software packages provided on the Additional Packages tape are automatically installed. However, in the custom installation mode you may decide to delay installation of various packages until after a basic system configuration is installed. The procedure described below will enable you to install additional packages as you choose.

To install additional packages on an installed system, use the **pkgadd(1M)** command. **pkgadd(1M)** requires that a tape device entry be added in the Device Database with the **putdev(1M)** command.

Refer to the **pkgadd(1M)** manual page at the end of these release notes for more details. In addition, see the Chapter on "Installing Add-On Software" in the *System Administration Manual Volume 1* (Pubs No. 0890429) for more information.

Refer to the **putdev(1M)** manual page at the end of these release notes for more details. In addition, see the Chapter on "The Device Database: Adding and Removing Storage Devices" in the *System Administration, Volume 2* (Pubs No. 0890430) for more information.

The following example installs the **nsu** package from a tape device named **tape1**:

```
putdev -a tape1 volume="DAT tape" cdevice=/dev/rmt/0hf \
        desc="DAT drive 1" type=ctape

pkgadd -qld tape1 nsu
```

Multiple packages may be installed at once by specifying more than one package name, as in:

```
pkgadd -qld tape1 nsu lp cmds
```

The package(s) to install may be selected from a menu of all packages available on the tape by not specifying any package name, as in:

```
pkgadd -qld tape1
```

6.0. Rebooting The System

During the installation, a new kernel is built. When the system installation completes, the system first halts and then returns to console mode.

If a kernel was successfully built during the system installation, execute the following commands from the console (refer to the HN6800 console manual, pubs # 0830045, for information on console command syntac):

```
#>  p boot 0
#>  fd dsk(c,u)
#>  fb
```

If the kernel build fails during the installation, **/stand/unix.generic** is copied to **/stand/unix**. You may use this generic kernel to boot to single-user mode, correct the problem that caused the kernel build failure, build a new kernel and reboot. Because the generic kernel was not built using your site's specific configuration, you should not come up in multi-user mode with the generic kernel. To boot the generic kernel, execute the following commands from the console:

```
#> pboot 3
00000000
#>fd dsk(0,0,0,0)
..............................
#> fb
Set Run Mode
CPU 0   CPU 1
dsk(0,0,0,0)/.
Initialize VME
dsk(0,0,0,0)/boot
PowerMAX OS Boot Loader
Boot
/stand/unix.generic
```

Remain in single-user mode by entering the root password when prompted. Then execute the following commands:

```
# fsck -y /dev/rusr            <---| for ufs /usr filesystem only
# mount /dev/usr
# /etc/conf/bin/idbuild -B
```

After you have successfully built a kernel, you can bring the system down by executing the command init 0. The system shutdown/reboot sequence takes care of moving the newly built unix to **/stand/unix.** Then follow the above procedure on how to boot a newly installed system.

For information on how to configure and build a kernel, see the Chapter on "Configuring and Building a Kernel" in the *System Administration, Volume 2* (Pubs No. 0890430).

See the following paragraph pertaining to **xfs root** file system and the use of "non-standard" kernel names in **/stand**.

7.0. General Notes

7.1. XFS

If a new kernel is installed in **/stand** with a nonstandard name then a hard link needs to be created. An explanation for this requirement is provided in the following paragraph.

The boot program boots kernels from the file system on partition 0, which it assumes to be of type **ufs**. If this file system is also the root, then by convention the kernels are held in directory /stand within the root. However, if the root is **xfs** (partition 5), the **ufs** file system containing the kernels (and standalones) is mounted on **/stand**, which in this case is a directory in the **xfs** root. Using hard links to the directory **/stand/stand** ensures that the conventional boot path can be used.

NOTE

It is important to point out to the system administrator that if kernels with names other than "**unix**" are copied to **/stand** on a system with an **xfs root**, they should be given hard links of the same name in **/stand/stand**. This will ensure that they can be booted using the conventional path **/stand/unix.xxx**. If this link is missing, they can only be booted using **/unix.xxx**.

7.2. PCI-to-PCI Bridge Configuration

The auto configuration aspects of PCI buses define which local bus addresses are propagated on each PCI bridge. The actual PCI I/O and Memory usage is dictated by the PCI cards installed and the resources requested. This function occurs automatically at system boot time.

8.0. General Information on Release 5.1

This section provides a general overview of PowerMAX OS™ Release 5.1. As stated earlier, this release is designed primarily to provide support for the newly introduced Concurrent Computer Corporation Power Hawk™ Series 700 Systems.

8.1. Architecture Unification

As the number of hardware platforms on which PowerMAX OS operates continues to grow, enhancements made in previous releases to simplify configuration and installation on selected systems continues to be updated to accommodate new systems.

Listed below are the Board Support Packages (**bspxxxx**) for the different system types.

Architecture Family	Hardware System Types	Specific Board Support Packages	General Board Support Package
Night Hawk	Night Hawk HN6200	bsp6800	**bspall**
	Night Hawk HN6800	bsp6800	
	PowerMAXION 4-Way	bsp6400	
	PowerMAXION 8-Way	bsp6408	
	TurboHawk	bsp6800t	
Power Hawk Series 600 & PowerStack	Power Hawk Model 620	bsp2600	**bspall**
	PowerStack II/III (uniprocessor)	bspmtx	
	Power Hawk Model 640 (multiprocessor)	bsp4600	
	PowerStack II/III (multiprocessor)	bspmtx	
Power Hawk Series 700	Power Hawk Series 710 (uniprocessor)	bspvgm5	**bspall**
	Power Hawk Series 720 (dual processor)	bspvgm5	
	Power Hawk Series 740 (quad processor)	bspvss4	

The system installation scripts will automatically turn on the appropriate module (**bspxxxx**) based upon the type of hardware on which the system is being installed. It is also possible to manually select one and only one **bspxxxx** module and relink a kernel which is appropriate for that particular hardware system type (within its architecture family).

While the **bspall** module will function for any of the supported system types (within its architecture family), the specific board support package module applicable to your system is preferred as it has been optimized for that system.

8.2. Enhancements in this Release

8.2.1. AltiVec Vector Execution Unit Support

The Power Hawk 700 Series utilizes Motorola's G4 processor. This chip features the AltiVec Vector Execution Unit, which allows highly parallel operations on a new set of 128-bit wide registers. Support for the AltiVec Vector Execution Unit affects several system interfaces as outlined below.

ucontext/mcontext structure

The register state of a light-weight process (LWP) is visible via the ucontext structure. The ucontext structure contains an mcontext structure, where the actual register state of an LWP is stored. The mcontext structure (and thus the ucontext structure) has been expanded in Release 5.1 to include the AltiVec register state. The vrregset structure has been added to the mcontext structure to store the AltiVec register state. To provide backward compatibility with prior releases, two versions of the ucontext/mcontext structures are supported in Release 5.1:

> Version 1 - struct ucontext_v1 is identical to the ucontext structure prior to Release 5.1.

> Version 2 - struct ucontext is the default in Release 5.1 and contains additional storage for the AltiVec register state.

The impact on system services is grouped into two groups below: those that retrieve the context of an LWP and those that modify the context of an LWP.

Retrieving context

Two versions of the getcontext(2) system call are provided in Release 5.1. If an application is compiled on a system running Release 5.1, system call getcontext(2) is redefined in the include file **/usr/include/sys/ucontext.h** to be the getcontext_v2(2) system call. The getcontext_v2(2) system call uses Version 2 of the ucontext structure. The user should not directly call the getcontext_v2(2) system service, the correct call will be mapped via the header file. Existing binaries, which were statically or dynamically linked on a pre-Release 5.1 system, will continue to use the old getcontext service. These applications will function as expected, since they would not be using AltiVec registers.

For systems that do not have an AltiVec unit present, the getcontext_v2(2) system call returns VRF_NA (not applicable) in the vrregset.flags field of the mcontext structure.

For LWPs that do not utilize the AltiVec unit, the vrregset.flags field of the mcontext structure is zeroed.

Possible values and meaning for the vrregset.flags field:

```
#define VRF_NOTUSED 0 - not in use
#define VRF_NA 1 - not applicable to this system
#define VRF_INUSE 2 - in use
```

Modifying Context

The system calls that modify the context of an LWP include:

setcontext(2)
makecontext(2)
swapcontext(2)
_lwp_create(2)
_lwp_setcontext(2)

These services now check the version field in the mcontext structure provided by the user and act accordingly.

On PowerHawk Series 700 systems, if a Version 2 mcontext structure is passed by the calling LWP, the AltiVec state will be modified as indicated.

For LWPs that have not accessed AltiVec registers, only the **VRSAVE** register is updated in the AltiVec register portion of the mcontext for the specified LWP.

For systems that do not have an AltiVec unit, the AltiVec register fields in the mcontext structure are ignored.

Determining if a AltiVec Vector Execution Unit is Present

A system service is provided for determining if the AltiVec Vector Execution Unit is present on the system where an application is executing. The syscx(2) system service now supports the **ALTIVEC_PRESENT** command for this purpose. This service returns 1 when the system is equipped with an AltiVec vector-processing unit, and returns zero if it does not.

8.2.2. RCIM Coupled FBS Enhancements

Up until now, RCIM devices configured as distributed interrupts could only be used as Closely Coupled FBS timing devices within a single Closely Coupled cluster.

RCIM Coupled FBS support in this release removes this restriction.

Any set of SBCs that are:

- connected to the same RCIM cable, and that

- can communicate with each other over a TCP/IP connection,

may now use the same distributed interrupt RCIM device as a FBS timing device.

A new pair of device registration function calls, and a **rdevfs(4)** timing device information call:

- fbs_register_rdev(3rt | 3F77rt)

- fbs_unregister_rdev(3rt | 3F77rt)

- fbsinfo_rdev(3rt | 3F77rt)

and new **rtcp(1)** device registration and information commands:

- **rd** (register Coupled FBS timing device)
- **urd** (unregister Coupled FBS timing device)
- **vr** (view **rdevfs** file configuration information)

are provided for support of these new RCIM Coupled timing devices.

8.2.3.   VME Networking Enhancements

This enhancement adds validation checks within the VME networking system. Previously, VME bus errors could cause corrupt data in received data packets, which could in turn cause system panics or system hangs. Through validation checking, these corrupt data packets can now be detected and discarded.

8.2.4.   Inter-SBC Networking Enhancements

On Power Hawk Series 600 closely-coupled systems, additional validation checks were added to the inter-SBC VME networking system. Previously, VME bus errors could cause corrupt data in received data packets, which could in turn cause system panics or system hangs. Through validation checking, these corrupt data packets can now be detected and discarded.

These validation checks are also present in the Power Hawk Series 700 closely-coupled inter-SBC P0Bus networking system implementation. These same enhancements will help to prevent any system panics or hangs that might have otherwise resulted from P0bus bus errors that might occur during network traffic.

8.2.5.   Patch 4.3 Enhancements

The following enhancements made to PowerMAX OS Release 4.3 as a patch are available in Release 5.1.

- Significant enhancements have been made to increase the performance and reliability of the **dec**, **cnd** and **pg** networking drivers. The default settings for these devices' flow-control queue hiwater levels have been significantly increased to allow a higher number of outstanding packets to be queued to both the transmit and receive device queues. The capability to change the default setting of these device queue hiwater level in their associated **space.c** files was also added.

- The flow-control queue hiwater level in the stream head for networking devices when opened for DLPI mode was changed so that it is set to the same size as the underlying network device's queue   hiwater level.

- The limit for connectionless UDP socket transfer requests has been increased from 16 KB to 64 KB. However, the amount of user data that can now be transferred is slightly less than 64 KB because 28 bytes must be reserved for the UDP header. Therefore, the maximum user data transfer with connectionless UDP is now 65507 bytes.

- A new UDP tunable (**udp_sth_hiwat**) was added that defines the hiwater limit for the read queue's stream head when using UDP. This tunable can be modified using the **ndd(1M)** command. The

minimum value that this tunable can be set to is 64 KB, the default is 2 MB, and the maximum is 8 MB.

8.2.6. Closely-Coupled Systems

Previous PowerMAX OS releases have supported closely-coupled systems (CCS) on Power Hawk Series 620/640 systems. This PowerMAX OS release adds CCS support for Power Hawk Series 720/740 systems.

The configuration requirements for Series 700 CCS systems are:

- Only Series 720 and 740 SBCs are supported; the Series 710 SBCs are not currently supported in CCS configurations.

- All SBCs must be installed in the same VME chassis.

- All SBCs in the VME chassis must also be connected to a single P0Bus with a single P0Bus overlay board, or to one of two P0Bus overlays, where the two overlays are connected together with one Backplane P0 (BPP0) bridge board.

- One SCSI disk drive must be connected to the server SBC for storage of PowerMAX OS software.

- One SCSI CD-ROM driver must be connected to the server SBC for PowerMAX OS software installs and updates.

All of the CCS features that were previously available on Series 600 CCS systems are also provided on Series 700 CCS systems. However, there are some differences in the way that these CCS features are implemented on these two type of CCS platforms:

- On Series 700 CCS systems, the VMEbus is only used for the initial downloads of client SBC object images. After that point, all inter-SBC communications are handled across the P0Bus. This communication activity includes inter-SBC message passing, inter-SBC networking support, remote shared memory access and user-generated **read(2)** and **write(2)** activity. (Explicit user-generated **read(2)** and **write(2)** DMAs across the VMEbus are still available, when and if desired.)

- On Series 600 CCS systems, the VMEbus is used for client object downloads and for all inter-SBC communications.

- On Series 700 CCS systems, the Slave MMap shared memory method is the only method available for inter-SBC shared memory support. However, a size of up to one-fourth of the **VME_DRAM_WINDOW** tunable may be allocated as a Slave MMap memory area on each SBC, unlike the (8MB - 4KB) limit that is present on the Series 600 CCS systems. Furthermore, since the Slave MMap areas are mapped onto the P0Bus address space, these Slave MMap areas do not take away from the amount of available VMEbus A32 address space, as is the case on Series 600 CCS systems.

- On Series 600 CCS systems, there are also Master MMap Type 1 and Type 2 shared memory access methods in addition to the Slave MMap method.

- The diskless utilities for Series 700 CCS systems lets the system administrator select between either P0Bus networking or on-board Symbios Ethernet networking as the primary inter-SBC networking interface, thus provided additional system configuration flexibility.

- On Series 600 CCS systems, the VMEbus networking interface must be used as the primary inter-SBC networking interface.

8.2.7.   NFS Version 3

NFS Version 3 offers substantial enhancements over NFS Version 2. These enhancements include, but are not limited to, the following:

- Improved client write throughput.

- Reduced server load resulting in increased scalability and performance.

- Support for large (multi-gigabyte) files on NFS servers.

To address backward compatibility with the installed base using NFS Version 2, Concurrent has implemented NFS in this release so that both protocols are supported concurrently.

Changes for NFS Version 3 are shown below. See the table below additional information.

- The NFS3 maximum file size is 17,179,869,184GB - 1 = 18,446,744,073,709,551,615 bytes.

- The NFS2 maximum file size is 2GB - 1 = 2,147,483,647 bytes.

- The NFS3 <u>default</u> write size (wsize) and read size (rsize) = 32KB.

- The NFS2 <u>default</u> write size (wsize) and read size (rsize) = 8KB.

Client **PowerMAX OS Version** **and NFS Version**	Server **PowerMAX OS Version** **and NFS Version**	**Maximum (Default)** **Write and Read** **Sizes Supported**
Release 5.1 - NFS3	Release 5.1 - NFS3	32KB[1]
Release 4.3 & below - NFS2	Release 5.1 - NFS3	8KB[1]
Release 5.1 - NFS3	Release 4.3 & below - NFS2	8KB[1]
[1] Concurrent recommends users do not deviate from the default sizes specified for wsize and rsize.		

The maximum file size is of course what NFS itself is capable of. If the user's NFS mounted disk partition is 100MB, then the maximum file size allowed by NFS to that partition is 100MB.

8.2.8.   Cross-Development Environment Packages

New for the PowerMAX OS 5.1 release. These packages contain all the header files and libraries required for cross development. The cross development environment allows users to develop applications from any supported Concurrent real-time computer systems targeting a specific PowerMAX OS release and architecture.

For example, installing the "xdev43moto" software package on a Power Hawk 640 SBC running PowerMAX OS 5.1 would allow the users to develop an application from the 5.1 system targeting Power Hawk Series 600 systems running PowerMAX OS 4.3. Conversely, installing the "xdev51moto" software package on a Power Hawk 640 SBC running PowerMAX OS 4.3 would allow the users to develop an application from the 4.3 system targeting Power Hawk Series 600 systems running PowerMAX OS 5.1.

8.2.9. Miscellaneous Enhancements

1. Added PIO mode to the IDE disk driver. This enhancement was made to support SanDisk Flash IDE disks, which do not operate in DMA mode. (Note: IDE disk drives only supported on PowerStack II/III systems.)

2. Enhanced the **inetd(1M)** daemon so that the parameters: **TOOMANY, CNT_INTVL**, and **RETRYTIME** can be tuned instead of being hardcoded. This tunable parameter capability for the **inetd(1M)** daemon was implemented through the **/etc/default** mechanism. The three parameters and their default values were placed in the file "**/etc/default/inetd**". This file is read when **inetd(1M)** is invoked and the specified values are used. These values can be modified within a specific range. See manual page **inetd(1M)** for further details.

8.2.10. Miscellaneous Changes

8.2.10.1. Context Restore

Prior to Release 5.1, the context switch logic always zeroed the contents of the floating-point registers if those registers were not in use by the LWP that was about to resume execution. This was done for security reasons, that is, to prevent one LWP from viewing values in another LWP's registers. In Release 5.1, the floating-point and AltiVec registers are not zeroed when switching to a process that does not make use of these registers. This allows for faster context switch times. For security conscious sites, it is possible to enable the zeroing of both floating-point and AltiVec registers in this case by setting the CSWTCH_CLEAR_UNUSED tunable.

8.2.10.2. MAXUID Change

The maximum number of unique user and group ids allowed in release 5.1 has been raised from 60002 to 524288.

MAXUID (param.h) is a limit on the system primitive uid_t which is defined as a long. This limit is the maximum number of unique user ids allowable in the system. The MAXUID limit is also used as the limit for the maximum number of unique group ids allowable in the system (the primitive is gid_t, also defined as a long). There are a few #defines that are associated with these limits. These are: UID_MAX (limits.h - defined here with the same value as MAXUID because param.h is not reachable by the kernel), UID_NOBODY, UID_NOACCESS, and GID_NOBODY (param.h), which are defined in terms of MAXUID. MAXUID was originally defined as 60002, the new value is 524288.

UID_NOBODY and GID_NOBODY are calculated by MAXUID-1.
UID_MAX = 524288, UID_NOACCESS = MAXUID.

Note that **/etc/passwd**, **/etc/group**, and **/etc/shadow** on 5.1 systems are now different from those same files in releases prior to 5.1. Specifically, the entries for 'nobody' and 'noaccess' now show the userids 524287, and 524288, respectively.

8.2.10.3. Watchpoint Support For NightView

With release 5.1, the kernel was changed to support the new watchpoint feature that was added to NightView Release 5.3.

8.3.   C/C++ Version 5.3 Compatibility Issues

The C/C++ Release 5.3 primarily consolidates all outstanding bug fixes and incorporates some fixes of its own.

When using PowerMAX OS 5.1 headers, **varargs** and **stdarg** use a different format for the **va_list** structure. Because of this, the user should avoid passing **va_list** structures to routines that are built with previous versions of the PowerMAX OS headers.

Code build with PowerMAX OS 5.1 headers can handle both the old and the new format of **va_list**. There should therefore be no problem with old code, compiled with PowerMAX OS 5.0 and earlier headers, passing **va_list** to code compiled with PowerMAX OS 5.1 headers.

For example, a dynamically linked program that calls **vfprintf** and that is compiled under PowerMAX OS 4.3 will have no problem running on a PowerMAX OS 5.1 system, calling the **vfprintf** in the 5.1 version of **libc.so**. That same program compiled under PowerMAX OS 5.1 will not work if run on a PowerMAX OS 4.3 system. The **vfprintf** in the 4.3 version of **libc.so** will not know how to handle the new **va_list** format. For more detailed information on this C/C++ release, refer to the *C/C++ Version 5.3 Release Notes* (Pubs No. 0890497-5.3).

9.0. Optional Products Available with Release 5.1

The following option products are currently available with the distribution of PowerMAX OS Release 5.1.

9.1. NightSim Release 3.2.2

NightSim Release 3.2.2 contains the changes briefly described below. For more detailed information on the NightSim release, refer to the *NightSim Version 3.2.2 Release Notes* (Pubs No. 0890480-3.2.2).

Overview of NightSim 3.2.2 Changes:

- Loosely-Coupled System Support (Distributed FBS across multiple systems with RCIMs).

- Closely-Coupled System Support (support for Series 620/640/720/740 systems)

- Enhanced File Redirection capability

- Enhanced integration with Night View

9.2. C/C++ Release 5.3

The C/C++ Release 5.3 primarily consolidates all outstanding bug fixes, adds some fixes of its own, and is designed to install without the problems that users experienced when installing C/C++ Release 5.2. For more detailed information on the C/C++ release, refer to the *C/C++ Version 5.3 Release Notes* (Pubs No. 0890497-5.3).

Refer to the section *C/C++ Version 5.3 Compatibility Issues* on page 45 for notes on compatibility issues associated with C/C++ Release 5.3 and PowerMAX OS Release 5.1.

9.3. NightView Release 5.3

NightView Release 5.3 contains the enhancements and changes briefly described below. For more detailed information on the NightView release and its contents, refer to the *NightView Version 5.3 Release Notes* (Pubs No. 0890395-5.3).

Overview of NightView 5.3 Enhancements:

- Support for long long (64-bit integer) data types

- Support for address traps

- Utilizes the PowerPC hardware DABR register to detect references on modifications to user memory

- Enhanced auto file translation for remote & cross debugging

- Supports vector types on the Power Hawk 700 Series.
  With Release 5.3 you can now:

    - print an AltiVec vector variable or register with the print command

- see the AltiVec vector registers with the info registers command
  (NightView prints vectors as 4 hexadecimal words.)

- set the value of an AltiVec vector from a set of values with the vector
  set command

- assign one AltiVec vector to another AltiVec vector

Overview of NightView 5.3 Changes:

- Remote Debugging

- KoalaTalk

- Serial Interface

- Miscellaneous bug fixes

9.4. NightTrace Release 4.2

NightTrace Release 4.2 contains the enhancements and changes briefly described below.
For more detailed information on the NightTrace release, refer to the *NightTrace Version
4.2 Release Notes* (Pubs No. 0890398-4.2).

Overview of NightTrace 4.2 Enhancements:

- Support of Power Hawk Series 700 systems

- Additional Kernel Trace Points for tracking user pagefaults

- Verbose Report Prints Process Names

- Node Selection Dialog

- Save As... and Save Text As... Enhancements

Overview of NightTrace 4.2 Changes:

- Bugfixes to trace capture & analysis in buffer & file wraparound modes

- Intelligent Building of Kernel Display Pages

- More Robust Handling of ntrace -listing

- Save As... and Save Text As... Functionality

- Correct Evaluation of Built-in Functions

9.5. NightBench Release 2.2

NightBench Release 2.2 contains the changes briefly described below. For more detailed
information on the NightBench release and its contents, refer to the *NightBench Version 2.2
Release Notes* (Pubs No. 0890514-2.2).

Overview of NightBench 2.2 Changes:

- C/C++ Support

- Run/Debug Support

- Preferences Dialog

- Link Options / Link Method Functionality

- New Environment Save/Load Features

- Improved Introduce Source Files Dialog

- New Build Transcript Format

- Reloadable Build Settings

- New Transcript Search and Save Features in NightBench Builder

- Notification Enhancements

- Improvements in NightBench Tool Communications

- Improved Handling of Special Characters

- More Robust Handling of Create All Functionality

9.6.   NightProbe Release 2.3

NightProbe Release 2.3 contains the changes briefly described below. For more detailed information on the NightProbe release and its contents, refer to the *NightProbe Version 2.3 Release Notes* (Pubs No. 0890465-2.3).

Overview of NightProbe 2.3 Changes:

- NightProbe 2.3 was released primarily to use the Y2K compliant Elan License Manager.

- This release also contains bug fixes which allow sampling of large amounts of data (SAR HM12529), and fixes for aborts when the spreadsheet window is scrolled (SARs HM12024 and HM12383).

9.7.   NightTune Release 1.2

NightTune Release 1.2 contains the changes briefly described below. For more detailed information on the NightTune release and its contents, refer to the *NightTune Version 1.2 Release Notes* (Pubs No. 0890515-1.2).

Overview of NightTune 1.2 Changes:

- NightTune 1.2 was released primarily to use the Y2K compliant Elan License Manager version.

9.8.   MAXAda Release 3.3.1

MAXAda Release 3.3.1 contains the enhancements briefly described below. For more detailed information on the MAXAda 3.3.1 release and its contents, refer to the *MAXAda Version 3.3.1 Release Notes* (Pubs No. 0890516-3.3.1).

Overview of MAXAda 3.3.1 Enhancements:

- Support of Power Hawk Series 700 systems (710, 720, and 740)

- Support for partial parallel compilation

- Enhancements to limited NFS Support

- Ability to save/reload all environment settings

- Addition of **a.tag** cross reference utility

9.9. X11 Window System Release 6.4.2

X Window System Release 6.4.2 contains the changes briefly described below. For more detailed information on the X Window System 6.4.2 release and its contents, refer to the *X Window System Version 6.4.2 Release Notes* (Pubs No. 0890298-6.4.2).

Overview of X Window System 6.4.2 Changes:

This release is primarily a patch to the Version 6.4.1 release. The following features and changes are contained in this release:

- HyperHelp

- KoalaTalk

- Bugfixes to the /usr/bin/X11/bitmap program

9.10. Elan License Manager Release 5.0.2

Elan License Manager Release 5.0.2 contains the changes briefly described below. For more detailed information on the Elan License Manager 5.0.2 release and its contents, refer to the *Elan License Manager Version 5.0.2 Release Notes* (Pubs No. 0891055-5.0.2).

Overview of Elan License Manager 5.0.2 Changes:

License Acquisition Delay - In previous releases, the Elan License Manager could take up to 15 seconds to grant a license for a particular feature when the count of licenses in use for that feature reached the maximum allowed.

NOTE

In such circumstances, each MAXAda compilation could take, at a minimum, 15 seconds to complete.

A workaround in the past was to set the POWERWORKS_ELMHOST environment variable:

POWERWORKS_ELMHOST=@host

where host was the name of the system where the Elan License Manager was running. This workaround is no longer needed with Elan License Manager 5.0.2.

10.0.   Compatibility Issues Between Systems

PowerMAX OS for the Power Hawk Series 600/700 and PowerStack II systems, in general, provides application source and binary compatibility with PowerMAX OS on the Night Hawk HN6200/HN6800, TurboHawk and PowerMAXION systems. Application code written, compiled and/or linked on one of these platforms, will operate on the other with the following exceptions:

1   Applications written for the Night Hawk, TurboHawk and PowerMAXION systems that use the real-time clocks (RTCs) may require minor modifications to function properly on a Power Hawk system.

   Seven RTCs are available on the Power Hawk Model 620, four 32-bit timers with a 1 microsecond resolution (tick timers) and three 16-bit timers (Z8536 timers) with a 400 nanosecond resolution.

   Six RTCs are available on the Power Hawk Model 640 and PowerStack II, three 32-bit timers with a 1 microsecond resolution (tick timers) and three 16-bit timers (Z8536 timers) with a 400 nanosecond resolution. If an RCIM is installed, four additional RTCS are available with a one microsecond resolution. (Note - RCIM requires one PMC slot). Note that PMC slots are <u>not</u> available on the PowerStack III (MTX PCI Series (7-slot) systems.

   Three integral RTCs are available on the Power Hawk Series 700. Installing a Real-Time Clocks and Interrupts Module (RCIM) will add four additional rtcs. On the Series 700, all of the integral rtcs have the same fixed interrupt level, and each rtc has its own interrupt vector associated with it. The Series 700 rtcs are only accessible in the default mode. This mode provides different degrees of control over a real-time clock. Default mode enables a user to control the direction of the clock count, up or down, the clock count value, the resolution per clock count value (1, 10, 100, 1000 or 10,000 microseconds), and whether the clock automatically restarts counting when the clock count reached zero.

   On the Night Hawk, TurboHawk and PowerMAXION, the RTCs have device names in the form of: **/dev/rrtc/0c**[*x*] where *x* is between 0 and 4. On the Power Hawk and PowerStack II systems, the tick timers have similar names (**/dev/rrtc/0c**[*0-3*]) while the Z8536 timers use **/dev/rrtc/1c**[*0-2*]. If an RCIM is installed, the four additional RTCs would have the devices names in the form of: **/dev/rrtc/2c**[*0-3*]. (Note - RCIM requires one PMC slot.)

   When utilizing one of the tick timers in "default mode" no code changes are required when moving between systems, only a re-compilation of the application. When switching from the tick timer to the Z8536 timer, the application will need to be modified to handle the 400 nanosecond resolution of the timers.

   The support for "direct mode" is different on each system. Review the **rtc(7)** man page for a description of the "direct mode" functionality on each of the systems.

2   PowerMAX OS on the Night Hawk, TurboHawk and PowerMAXION systems provides the ability to generate an address translation to the Interval Timer hardware and read it directly from the user process. This is done using the **/dev/interval_timer** device. Such a POSIX-format timer does not exist on the Power Hawk Series600/700 and PowerStack II systems therefore, programs that depend upon this operation will not operate. There are two alternatives that provide compatible operation.

   a   Use the C library routines, such as **clock_gettime()**. This routine exists in the shared C library and generates instructions appropriate to the hardware platform. On the Night Hawk and PowerMAXION, this routine uses the

hardware Interval Timer while on the Power Hawk it uses the 64-bit Time Base Register. Note that conversion of the Time Base register to POSIX format takes notably longer than merely reading the Interval Timer register.

b   Use the Time Base Register exclusively for timing. This is a 64-bit register that increments at 1/4 the clock speed of the processor bus. For example, on a Power Hawk system that has a processor bus speed of 66.66MHz, it increments at the rate of 16.65MHz. This is a processor register on the PPC604 and is directly accessible from user code. However, conversion to seconds and nanoseconds may be time consuming depending upon the processor bus speed.

3   On Night Hawk, TurboHawk and PowerMAXION systems, PowerMAX OS provides a set of routines to read and/or write the hardware ipl register. This register determines the interrupt level that the processor is currently running at. The ability to read/write the ipl value is especially useful for user-level device drivers.

The **spl_map()** package includes routines to generate virtual mappings to the hardware ipl register, along with routines to write the register (*spl_request( )* and *spl_manage( )*) and a macro to do the same (*spl_request_macro( )*). Mappings are done by using the **/dev/spl** device node.

Power Hawk Series 600/700 and PowerStack II systems do not have a hardware ipl register. Instead a series of hardware registers implement the ipl functionality through a series of bit masks. Therefore, routines that map directly to the hardware ipl register will not operate on these systems. **/dev/spl** does not exist on Power Hawk systems.

Compatible operation between Power Hawk and the Night Hawk and PowerMAXION systems can be accomplished using the following techniques.

a   Do not use **mmap(3)** to map directly to the hardware register(s).

b   Use the *spl_map()*, *spl_unmap()* and *spl_manage()* library routines for all ipl register reads/writes. In this case, binary compatibility is possible between Night Hawk, PowerMAXION and Power Hawk platforms.

c   The *spl_xxx* routines are contained in libud. If the shared version of libc is used in the executables, then the shared version of libud must also be used.

d   If the **spl_request()** function or the **spl_request_macro()** is used, the program will need to be recompiled and relinked on the target machine, as the macro implementation will be different. On the Power Hawk it will be the same as the C library routine.

e   Do not depend upon hard-coded spl values, as the values used may vary between platforms. However, in all cases, an spl value of 0 allows all interrupts, and higher values prevent more and more interrupts. spl values are always less than 255.

4   Programs that depend upon the arrangement and size of kernel data structures may not work properly due to differing sizes of certain items. These would be programs that utilize **/dev/mem** or **/dev/kmem** to read and/or examine kernel structures. These programs should be recompiled on the target system to ensure functionality.

5   Power Hawk Model 620 and Model 710 systems are only single processor systems. Programs that require multiple processors, or make system calls that reference multiple processors, may not operate properly.

6   Edge Triggered Interrupts are available on the Night Hawk and PowerMAXION platforms but <u>not</u> on the Power Hawk Series 600/700 and PowerStack II systems. Therefore, there is <u>no</u> software support on the above noted systems for Edge Triggered Interrupts. If an RCIM is installed, four Edge Triggered Interrupts are available. These Edge Triggered Interrupts are compatible with those on the Night Hawk systems. (Note - RCIM requires one PMC slot.)

7   At the present time there is no support for NVRAM Global Environment Variable (GEV) on Night Hawk ('nh') or Power Hawk Series 700 platforms. The nvram system call always returns ENOSYS on these platforms and the **gev** set of commands are not present.

8   Applications written for the Power Hawk Series 700, which utilize the AltiVec instructions or registers, are not executable on the earlier Night Hawk and Power Hawk systems which are based on the 604e processor.

11.0. Service Releases

Service Releases containing updates, enhancements, patches, etc., are periodically released by Concurrent Computer Corporation on an "as-required-basis".

Descriptions of available updates and patches to Concurrent Computer Corporation software products can be examined and ordered on this site: **http://www.ccur.com/support/realtime/rt_si.htm**

To find out what patches are installed on your system enter the following command:

pkginfo -c patch | pg   (screen output example shown below)

```
    patch      base-001    Base System Patch 001 (4.3P1)
    patch      base-002    Base System Patch 002 (4.3P2)
    patch      base-003    Base System Patch 003 (4.3P3)
    patch      cmds-001    Advanced Commands Patch 001 (4.3P1)
    patch      cmds-002    Advanced Commands Patch 002 (4.3P2)
    patch      cnd-001     Condor Ethernet Driver Patch 001 (4.3P1)
    patch      cnd-002     Condor Ethernet Driver Patch 002 (4.3P2)
    patch      cnd-003     Condor Ethernet Driver Patch 003 (4.3P3)
    patch      dec-001     DEC Ethernet Driver Patch 001 (4.3P2)
    patch      dec-002     DEC Ethernet Driver Patch 002 (4.3P3)
    patch      egl-001     Eagle Ethernet Driver Patch 001 (4.3P2)
    patch      egl-002     Eagle Ethernet Driver Patch 002 (4.3P3)
    patch      fbs-001     Frequency Based Scheduler Patch 001 (4.3P1)
    patch      fbs-002     Frequency Based Scheduler Patch 002 (4.3P2)
    patch      hc-001      Concurrent C Compilation System
    patch      inet-001    Internet Utilities Patch 001 (4.3P1)
    patch      inet-002    Internet Utilities Patch 002 (4.3P2)
    patch      inet-003    Internet Utilities Patch 003 (4.3P3)
    patch      kdb-001     Kernel Debugger Patch 001 (4.3P1)
    patch      librt-001   Real-time Libraries Patch 001 (4.3P2)
    patch      man-001     On-line Manual Pages Patch 001 (4.3P1)
    patch      man-002     On-line Manual Pages Patch 002 (4.3P2)
```

12.0. Direct Software Support

Software support is available from a central source. If you need assistance or information about your system, please contact the Concurrent Software Support Center at our toll free number 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822 or 1-305-931-2408. The Software Support Center operates Monday through Friday from 8 a.m. to 7 p.m., Eastern Standard time.

Calling the Software Support Center gives you immediate access to a broad range of skilled personnel and guarantees you a prompt response from the person most qualified to assist you. If you have a question requiring on-site assistance or consultation, the Software Support Center staff will arrange for a field analyst to return your call and schedule a visit.

13.0. Manual Pages

Manual pages can be found online. The easiest way to access these is by typing `man` followed by a manual page name or a command name. Typing `man manual`, for example, will show online the manual page that contains a categorized listing of all the current manual names, publication numbers and latest revision numbers.

On-line manual pages also exist for most software and hardware manuals in this document. These manual pages provide a description of each manual and also list all related publications, where applicable. To access, type `apropos` *pubs number* that will display the manual page name. Typing `man` *manual page name* will then show on the screen the manual page you are interested in.

Printed copies of the **`format(1M)`**, **`pkgadd(1M)`** and **`putdev(1M)`** man pages that may need to be referenced during installation are provided at the end of these release notes.

(This page left intentionally blank)

format(1M) manpage

NAME

format– format program for Generic Disk driver disks

SYNOPSIS

/sbin/format [*-f* | *-l* | *-v*] *disk*

DESCRIPTION

format is a program which is used to format, verify, manage flawmaps, or define partition sizes for disk drives on HSA, or other SCSI disk controller that employs the Generic Disk driver (**gd**).

Geometry Blocks

format maintains (in memory) two copies of the disk's geometry block. The "disk geometry block" is an exact copy of the geometry block as it exists on the disk. The "current geometry block" is a copy of the geometry block which may be modified with the **"partition"** command and then written to disk. Whenever the "disk geometry block" is read from the disk, the "current geometry block" is initialized to be the same as the "disk geometry block;" if there is no geometry block on the disk, the "current geometry block" partitions are initialized to default values. **format** has several commands which manipulate the "disk geometry block" and/or the "current geometry block." For example, the **"status"** command displays the partition sizes of the "disk geometry block;" the **"partition"** command displays the partition sizes of the "current geometry block."

RUNNING FORMAT

format may be run either interactively (if no option is specified) or non-interactively (if an option is specified). Only a single disk may be specified for each invocation of the format program. Multiple disks may be formatted simultaneously by running multiple copies of **format**.

"format <disk>" starts the format program in an interactive mode. When **format** starts, it opens the raw device for slice 7 of the disk specified. The disk may either be specified as a disk "unit" number or as a full path to a raw device node for slice 7. **format** must have read/write access to the device.

In interactive mode, **format** accepts and executes commands until the program is terminated. A list of commands may be obtained by typing **"?"**. Commands may be abbreviated; only enough characters are required to uniquely identify the command.

"format <option> <disk>" starts the format program in a non-interactive mode. **format** executes the single command which is specified by the option. **format** options are:

-f format the disk (the **"format"** command)
-v verify the disk (the **"verify"** command)

After running **format,** if the disk was formatted or the partition sizes were modified, it is necessary to re-build the file systems on the disk.

COMMANDS

"? [commandname]"

If the optional parameter is not specified, this command prints a summary of all the commands. If the optional **[commandname]** parameter is specified, this command prints information about that specific command—command syntax, what the command does, and warnings associated with the command.

"format"

This is a single atomic operation, which may take 10 minutes or more on large disks. When formatting has successfully completed, the "current geometry block" and the flawmap are written to the drive description tracks.

"partition [partition#] [size]"

If no parameters are specified, this command prints the partition sizes and partition starting locations of the "current geometry block". If the optional parameters are present, this command sets the size of **[partition#]** to be **[size]** units. Typing **"partition default"** is a special form of the command, which sets all partitions sizes to default values. The partition start and size are modified only in the "current geometry block"; the "disk geometry block" is not changed.

Partition sizes are always given in the current unit, which may be modified with the **"unit"** command. The size actually reserved for a partition is always rounded up to a cylinder boundary. Each time the size of a partition changes, the starting locations of all subsequent partitions (with larger partition numbers) is also changed. This is because partition 1 always follows partition 0, etc.

"quit"

This command terminates the program. If the "current geometry block" differs from the "disk geometry block", **format** will not terminate. This means that the **"partition"** command was used to modify the geometry block, but the "current geometry block" was never written to disk. At this point it is necessary to use either the **"format"** or **"write"** command to update the geometry block on the disk, or to force a program exit by typing **"q!"**.

"read"

This command reads the geometry block from the disk. The "disk geometry block" and "current geometry block" are initialized. If there is no geometry block on the disk, the "current geometry block" is initialized with default partition sizes.

"sectorsize [size]"

This command is used to specify an alternate physical sector size to be used when formatting the disk. The size specified is in bytes. The physical sector size on the disk is not changed until the **"format"** command is issued. The default sector size is 512 bytes, unless the disk has already been formatted to a different value

"status"

This command prints detailed information about the disk, including the partition sizes and a description of how the disk is organized. Partition sizes are always given in the current unit, which may be modified with the **"unit"** command.

"unit [unit]"

This command prints/selects the unit for partition sizes. If no parameters are present, then the current unit is printed. If the optional **[unit]** parameter is present, then the current unit becomes **[unit]**. **[unit]** may be any of: **"sectors"**, **"cylinders"**, **"kbytes"**, **"megabytes"**, or **"percent"** (percent of disk). Only the first letter of **[unit]** is required.

"verify"

This command verifies the disk by reading every sector. A **"."** is printed as each cylinder is completed.

"write"

This command writes the "current geometry block" to the disk. The "disk geometry block" is set equal to the "current geometry block".

SEE ALSO

gd(7), **hsa**(7), **ise**(7), **format**(8), **errpt**(1M)

NOTES

The ability to format an IDE drive is not available. IDE drives are formatted at the factory and this cannot be redone in the field. Similarly there is no 'verify' function.

pkgadd(1M) manpage
NAME

>   **pkgadd** – transfer software package or set to the system

SYNOPSIS

>   **pkgadd [-d** *device***] [-r** *response***] [-n] [-q] [-l] [-a** *admin***]**
>   **[-p] [***pkginst1* **[***pkginst2***[...]]]**
>
>   **pkgadd -s** *spool* **[-d** *device***] [-q] [-l] [-p] [***pkginst1* **[***pkginst2***[...]]]**

DESCRIPTION

>   **pkgadd** transfers the contents of a software package or set from the distribution medium or directory to
>   install it onto the system. A package is a collection of related files and executables that can be independently
>   installed. A set is made up of a special-purpose package, referred to as a Set Installation Package (SIP), and
>   a collection of one or more packages that are members of the set. The SIP controls the installation of the set.
>
>   **pkgadd** checks that all packages listed on the command line are on the installation medium. If any of the
>   packages listed does not exist, no changes are made to the system, that is, none of the listed packages are
>   installed.
>
>   Used without the **-d** option, **pkgadd** looks in the default spool directory for the package (**/var/spool/**
>   **pkg**). Used with the **-s** option, it writes the package to a spool directory instead of in-stalling it.
>
>   Error messages are always logged (see **-l**, below). In addition, when **pkgadd** terminates, it will send mail
>   (by default, to **root**) with all the error messages and a summary of which packages installed completely,
>   partially, or not at all.
>
>   For all appropriate files, the Mandatory Access Control (MAC) and privilege information provided in the
>   **pkgmap** file will be set. MAC and privilege can only be set on files installed on a sfs-type file system (that
>   is, they will be ignored for non-sfs-type file systems). These entries are provided for upward compatibility
>   with the Enhanced Security Utilities. The following options are available.

>   **-d** *device*   Installs or copies a package/set from *device*. *device* can be: (a) the full pathname to a
>   directory, file, or named pipe (such as **/var/tmp**); (b) the device identifiers for tape or disk
>   devices (such as **/dev/rmt/*** or **/dev/dsk/***) [see **intro**(7)]; (c) a device alias (such
>   as **diskette1**); or (d) **"-"** which specifies packages in datastream format read from
>   standard input. The default device is the installation spool directory (**/var/spool/pkg**).
>
>   For device identifiers, the device specified (either by pathname or alias), must have an entry
>   in the device table (**/etc/device.tab**). If no entry exists in the device table, **pkgadd**
>   will abort.
>
>   A device alias is the unique name by which a device is known. (For example, the alias for a
>   cartridge tape drive might be **ctape1**.) The name must be limited in length to 64 characters
>   (**DDB_MAXALIAS**) and can contain only alphanumeric characters and/or any of the
>   following special characters: underscore (**_**), dollar sign (**$**), hyphen (**-**), and period (**.**). No
>   two devices in the database can share the same alias.

>   **-r** *response* Identifies a file or directory, *response*, which contains the answers to questions posed by a
>   "request script" during a previous **pkgask** session conducted in interactive mode [see
>   **pkgask**(1M)]. When *pkginst* is a package, *response* can be a full pathname or a directory;
>   when *pkginst* is a SIP, *response* must be a directory. For a complete description of request
>   scripts and response files, see your system administration or software packaging guides.

>   **-n**   Installation occurs in non-interactive mode. The default mode is interactive.

>   **-q**   Installation is performed in quiet mode. Only prompts requesting user input and error
>   messages are displayed on the screen.

>   **-l**   Error messages are not sent to the standard error output; they are only logged to **/var/**
>   **sadm/install/logs**/*pkginst*.**log**.

>   **-a** *admin*   Defines an installation administration file, *admin*, to be used in place of the default
>   administration file to specify whether installation checks (such as the check on the amount of
>   space, the system state, and so on) are done. [For a description of the format of an *admin* file,
>   see **admin**(4).] The token **none** overrides the use of any *admin* file, and thus forces
>   interaction with the user. Unless a full pathname is given, **pkgadd** looks in the **/var/**
>   **sadm/install/admin** directory for the file. By default, the file **default** in that
>   directory is used. **default** specifies that no checking will be done, except to see if there is

enough room to install the package and if there are dependencies on other packages. The **-a** option cannot be used if *pkginst* is a SIP.

-p   Do not give the initial prompt to the user to insert the distribution medium. All other prompts will continue normally.

pkginst   A short string used to designate a package/set. It is composed of one or two parts: pkg (an abbreviation for the package/set name) or, if more than one instance of that package exists, pkg plus *inst* (an instance identifier). (The term "package instance" is used loosely: it refers to all instances of *pkginst*, even those that do not include instance identifiers.)

The package name abbreviation (pkg) is the mandatory part of *pkginst*. [See **pkginfo**(1), **pkginfo**(4).]

If *pkginst* is a SIP, the SIP controls installation of the set by using request scripts and pre-install scripts. The SIP request script, not the package installation tools, is responsible for prompting the user for responses and taking the appropriate actions. If the request script fails, only the SIP will be processed. For a complete description of request and pre-install scripts, see your system administration and/or software packaging guides.

The second part (*inst*), which is required only if you have more than one instance of the package in question, is a suffix that identifies the instance. This suffix is either a number (preceded by a period) or any short mnemonic string you choose. If you don't assign your own instance identifier when one is required, the system assigns a numeric one by default. For example, if you have three instances of the Advanced Commands package and you don't create your own mnemonic identifiers (such as **old** and **beta**), the system adds the suffixes **.2** and **.3** to the second and third packages, automatically.

To indicate all instances of a package, specify **'***pkginst***.*'**, enclosing the command line in single quotes, as shown, to prevent the shell from interpreting the * character. Use the token **all** to refer to all packages available on the source medium.

-s *spool*   Reads the package into the directory *spool* instead of installing it.

USAGE

The **-r** option can be used to indicate a directory name as well as a filename. The directory can contain numerous *response* files, each sharing the name of the package with which it should be associated. This would be used, for example, when adding multiple interactive packages with one invocation of **pkgadd**. Each package that had a request script would need a *response* file. If you create response files with the same name as the package (for example, *package1* and *package2*) then, after the **-r** option, name the directory in which these files reside.

The **-n** option will cause the installation to halt if any interaction is needed to complete it.

When invoked with no *pkginst* specified on the command line, *pkgadd* only displays the names of sets if at least one SIP exists on the media. Because of this, you shouldn't include packages on the same media if some are members of sets and some are not. If you do, the packages which are not members of sets can be installed only if their *pkginst* names are provided on the command line.

The **pkgadd** command checks to see if any of the files in *pkginst* are already installed on the system and, if any are, saves this fact before continuing with installation. Later, **pkgadd** won't reinstall these files on the system. If one of the package's installation scripts removes such a file, the result will be that the file will no longer be on the system when package installation completes.

The **pkgadd** command does not encompass any files that were already compressed (that is, only those in "**.Z**" form) before being processed by **pkgmk**.

Files

/etc/device.tab	device table
/var/sadm/install/admin/default	default package administration file
/var/sadm/install/logs/*pkginst***.log**	error message log
/var/spool/pkg	default spool directory
/usr/lib/locale/*locale***/LC_MESSAGES/uxpkg**	language-specific message file [See **LANG** on **environ**(5).]

REFERENCES

admin(4), **compver**(4), **copyright**(4), **depend**(4), **installf**(1M), **intro**(7), **pkgask**(1M),
pkgchk(1M), **pkginfo**(1), **pkginfo**(4), **pkgmap**(4), **pkgparam**(1), **pkgrm**(1M), **putdev**(1M),
removef(1M), **setinfo**(4), **space**(4)

putdev(1M) manpage

NAME

   putdev – create and update the device database

SYNOPSIS

   putdev -a *alias* [*secdev=value*] [*attribute=value* [. . .]]

   putdev -m *device attribute=value* [*attribute=value* [. . .]]

   putdev -d *device* [*attribute* [. . .]]

   putdev -p *device attribute=value*[,*value . . .*]

   putdev -r *device attribute=value*[,*value . . .*]

DESCRIPTION

   The **putdev** command is used to add a new device to the Device Database (DDB), modify an existing
   device's attributes, or remove a device entry from the DDB. It also allows appending new values to attributes
   that take value-lists (separated by commas), and removal of specific values from value-lists.

   Options

   putdev takes the following options:

   -a *alias* Add a *device* to the DDB using the specified *attributes*. The *device* must be referenced by its *alias*.
   The *secdev* attribute is defined only if the Enhanced Security Utilities are installed.

   -m *device*
   Modify a *device* entry in the DDB, using the specified attribute values. If a specified attribute does
   not exist in the device entry, putdev adds the specified *attribute* to the entry. It also modifies
   attributes that already have a value with the *value* specified.

   -d *device*
   Remove a *device* entry from the DDB, when executed without the *attributes* argument. If the
   attribute argument is specified, the *attribute* and its value are deleted from the device entry.

   -p *device*
   Append the list of values to the *attribute* value-list of the *device*. If the *value* item is multiply defined
   in the input value-list or already defined in the DDB, putdev fails and prints an error message.

   -r *device*
   Remove the list of values from the *attribute* value-list, of the *device*. The command succeeds, even
   if the *value* has been removed or is not defined for the *attribute* in the DDB.

   alias   Define the alias name of the device, a value which must be unique throughout the DDB. *alias* is
   limited to 64 characters (**DDB_MAXALIAS**) and should contain only alphanumeric characters and
   any of the following special characters: **.** (period), **_** (underscore), **$** (dollar sign), and − (hyphen).

   secdev   Designate the alias of the secure device that defines all the security attributes, and is used only if the
   Enhanced Security Utilities are installed. If *secdev* is not specified during creation (**-a** option) or
   is deleted (**-d** option), the current *alias* is used as the default value of *secdev*. The validation rules
   for *secdev* are the same as those for *alias*.

   device   Designate the absolute pathname or alias name of the device whose attribute is to be added,
   modified, or removed. If *device* is a pathname, then the attributes of the alias to which it maps are
   updated.

   attribute
   Designate a device attribute to be added, modified, or deleted. This prevents an accidental
   modification or deletion of a device's alias from the DDB.

   value   Designate the value to be assigned to a device's attribute. If any of the values are invalid, then
   putdev fails and prints an error message.

   Whenever the attributes in a Device Database file are updated, the old version of the file is saved to a file with
   the capital letter "O" prefixed to the file name. If there are errors in the modification of device attributes, you
   can recover the old versions of the Device Database files.

   Attributes

   Following are all of the attributes which can be defined for a device:

   alias The unique name by which a device is known. No two devices in the database may share the same
   alias name. The name is limited in length to 64 characters (**DDB_MAXALIAS**) and should contain

only alphanumeric characters and any of the special characters: underscore (**_**), dollar sign (**$**), hyphen (**-**), and period (**.**).

bdevice
The absolute pathname to the block special device node associated with the device, if any, with maximum length of **PATH_MAX**. This attribute is optional.

bdevlist
A list of additional pathnames of block device special files which map to the same logical or secure device. Each item in the list is separated by a comma, and each must be an absolute pathname of the device special file, with a maximum length of **PATH_MAX**. Since, this attribute takes a list of values, **putdev -p | -r** can be used for this attribute. This attribute is optional.

capacity
The capacity of the device or of the typical volume, if removable.

cdevice
The absolute pathname to the character special device node associated with the device, if any, with maximum length of **PATH_MAX**. This attribute is optional.

cdevlist
It contains a list of additional pathnames of character device special files mapping to the same logical or secure device. Each item in the list is separated by a comma, and each must be an absolute pathname of the device special file, with a maximum length of **PATH_MAX**. Since, this attribute takes a list of values, **putdev -p | -r** can be used for this attribute. This attribute is optional.

cyl   Used by the command specified in the **mkfscmd** attribute.

desc   A description of any instance of a volume associated with this device (such as floppy diskette).

dpartlist
The list of disk partitions associated with this device. Used only if **type=disk**. The list should contain device aliases, each of which must have **type=dpart**.

dparttype
The type of disk partition represented by this device. Used only if **type=dpart**. It should be either **fs** (for file system) or **dp** (for data partition).

erasecmd
The command string that, when executed, erases the device.

fmtcmd
The command string that, when executed, formats the device.

fsname
The file system name on the file system administered on this partition, as supplied to the **/usr/ sbin/labelit** command. This attribute is specified only if **type=dpart** and **dparttype=fs**.

gap   Used by the command specified in the **mkfscmd** attribute.

mkfscmd
The command string that, when executed, places a file system on a previously formatted device.

mountpt
The default mount point to use for the device. Used only if the device is mountable. For disk partitions where **type=dpart** and **dparttype=fs**, this attribute should specify the location where the partition is normally mounted.

nblocks
The number of blocks in the file system administered on this partition. Used only if **type=dpart** and **dparttype=fs**.

ninodes
The number of inodes in the file system administered on this partition. Used only if t**ype=dpart** and **dparttype=fs**.

norewind
The name of the character special device node that allows access to the serial device without rewinding when the device is closed.

pathname
Defines the pathname to an i-node describing the device (used for non-block or character device pathnames, such as directories).

type   A token that represents inherent qualities of the device. Standard types include: 9-track, ctape, disk, directory, diskette, dpart, and qtape.

volname
          The volume name on the file system administered on this partition, as supplied to the **/usr/sbin/labelit** command. Used only if **type=dpart** and **dparttype=fs**.

volume
          A text string used to describe any instance of a volume associated with this device. This attribute should not be defined for devices which are not removable.

Enhanced Security Attributes

The following security attributes can be defined for a device alias if the Enhanced Security Utilities are installed:

secdev
          The alias name of the physical device or secure device, and is unique throughout the Device Database(DDB). This alias name is limited to 64 characters (**DDB_MAXALIAS**), and should contain only alphanumeric characters and the following special characters: "**_**", "**$**", "**-**" or "**.**". For a secure device alias this attribute's value is the same as the device's alias. For a logical device alias, this attribute's value is different from the device alias. By default, secdev is defined to be equal to the device's alias.

range The sensitivity Mandatory Access Control (MAC) level range of the device. It should by a *hilevel-lolevel* pair, where *hilevel* and *lolevel* are both MAC level names or fully qualified levels. The "-" character is the delimiter between *hilevel* and *lolevel*. These levels are stored in the DDB as LIDs, converted to ASCII characters. The LIDs are validated against the Label Translation Database and it is ensured that *hilevel* dominates *lolevel*, before they are saved in the DDB. This attribute must be defined.

state Determines whether the device is to be used as a private or public device. It can take any one of the values: *private*, *public*, or *pub_priv*. If it is set as *pub_priv*, then the device can either be used as private or public device. If the *startup* attribute is enabled, then the device is allocated as *private*, if the state was set to either *private* or *pub_priv*. This attribute must be defined.

mode   Determines the mode of the device. This attribute can take one of the values: *static* or *dynamic*. This attribute must be defined.

startup
          is a flag (*y[es]/n[o]*) that indicates whether the device is allocated during startup or not. This attribute is optional, and startup default value is no.

startup_level
          Defines the MAC level at which the device should be set at startup. This can be specified as a level name or fully qualified level. However, the value is saved in the DDB as an ASCII LID value. This attribute is optional.

startup_owner
          Defines the owner of the device. The value of startup_owner can be specified as the **uid** or user name followed by the access permissions. The value must be specified in the format: *uid>rwx*. If any of the read, write, or execute access is denied, that field must contain a **-** (hyphen). The > character serves as delimiter between the **uid** or user name and the access permissions. The **uid** or user name must be defined on the system (in **/etc/passwd)**, at the time this attribute is defined. This attribute is optional but must be defined if attribute **startup** is set to yes.

startup_group
          Defines the group to which the device belongs. The value of **startup_group** can be specified as the **gid** or group name followed by the access permissions. The value must be specified in the format: *gid>rwx*. If any of the read, write or execute access is denied, that field must contain a **-** (hyphen). The > character serves as delimiter between the **gid** or group name and the access permissions. The **gid** or group name must be defined on the system (in **/etc/group**), at the time this attribute is defined. This attribute is optional but must be defined if attribute **startup** is set to **yes**.

startup_other
          Defines the access permissions for *other*. The value of **startup_other** must be specified in the format: *>rwx*. If any of the read, write or execute access is denied, then that field must contain a **-** (hyphen). This attribute is optional but must be defined if attribute **startup** is set to **yes**.

ual_enable
A flag that enables or disables depending on its value the user authorization list defined in the *users* and *other* attributes. This attribute can take one of the values: **y**[es] or **n**[o]. If "**y**", then the user authorization list is checked when authorizing an user to use this device. If "**n**", then no users are authorized to use this device. This attribute is optional, and value assumed as **no** if **ual_enable** is not defined.

users The user authorization list that defines the allocation permissions for *users*. Each item is a **uid**-authorization or username-authorization pair separated by a > character. The items in the list are separated by commas. The attribute's value must be specified in the format: *uid1>n,uid2>n,uid3>y*. Each **uid** or username must be unique in a device entry, and all **uids** or usernames must be defined in **/etc/passwd**, when this attribute is defined. Since this attribute takes a list of values, **putdev -p | -r** can be used. This attribute is optional.

other Defines the authorization permissions for *other*. This attribute contains only one item and it can take one of two values: **y**[es], or **n**[o]. This attribute is optional, and its value is assumed as **no**, if *other* is not defined.

Enhanced Security Usage

The following rules and guidelines should be followed when using the putdev command in enhanced security installations.

The **alias** names of devices must be valid [see description under NOTICES] and unique throughout the DDB; otherwise, **putdev** fails.

The pathnames to device special files in attributes *cdevice*, *bdevice*, *cdevlist*, and *bdevlist* must be absolute pathnames. They cannot be repeated within an entry or occur in multiple entries. The **putdev** command checks for uniqueness of pathnames; otherwise, **putdev** fails.

Security attributes can be defined for *device*, or *alias* only if the system is configured for multilevel security; otherwise, **putdev** fails.

The MAC level values for the security level range (*hilevel-lolevel*) must be valid security level aliases or fully qualified level names defined in the Level Translation Database (LTDB); otherwise, **putdev** fails. If *hilevel* does not dominate *lolevel*, then **putdev** fails.

The *secdev* attribute requires the following special handling:

The *secdev* attribute is used to define the essential security attributes of a device, that is, those attributes required when the Enhanced Security Utilities are installed. *secdev* must be valid (see description under Security Attributes) and unique throughout the DDB; otherwise **putdev** fails.

By default, when adding a new device alias into the Device Database, if the *secdev* attribute is not defined at the command line, the new device entry is assigned a *secdev* equal to its *alias*.

The alias that defines security attributes of a device is called a secure device alias. You can define other non-security attributes for this alias, if needed. For all secure devices, *secdev* must have same value as *alias*; if *secdev* is different from *alias*, the alias is a logical alias.

The security attributes *range*, *state*, and *mode*, must be defined for every secure device alias.

When adding (using **-a**) or modifying (using **-m**) a device entry and specifying the *secdev* attribute not equal to the *alias* being added or modified, **putdev** performs the following checks in the order specified:

1. If the essential security attributes are being defined for *alias*, the command fails and displays an error message. An entry defining the essential security attributes must have the *secdev* attribute be equal to its *alias*.

2. If the essential security attributes are not being defined for *alias*, and if the specified *secdev* does not exist in the Device Database, a warning message is displayed.

3. If the essential security attributes are not being defined for *alias*, and the specified *secdev* exists in the Device Database but does not define the essential security attributes, **putdev** fails and displays an error message.

4. If the essential security attributes are not being defined for alias, and the specified *secdev* exists in the Device Database and defines the essential security attributes, then the command is successful.

You should create the secure alias before creating any logical aliases that map to the same secure alias. Similarly, you should not remove a secure device alias if any logical alias are currently mapped to that secure alias.

Additional aliases that share the security attributes defined for a secure device can be created by specifying their *secdev* to have the same value as the *alias* of the secure device. If *secdev* is not specified, and the essential security attributes are also not specified, then a logical device entry is created that does not have security attributes.

Special handling of the essential security attributes:

The essential security attributes, *mode*, *state*, and *range* must be created (using **-a** or **-m**) and deleted (using **-d**) together. Otherwise, **putdev** fails and issues an error message.

The essential security attributes of a secure alias can be modified (**-m**) separately after they are defined.

If the essential security attributes are being deleted from a device entry whose *alias* is a *secdev* attribute for at least another entry in the Device Database, then putdev fails and displays an error message.

Enhanced Security Example

The following example shows you how to create one secure device (**tapedrive1**) and two device aliases (**slowtape**, **fasttape**) that map to the secure device. (In the following example, the input is split onto three lines; you should enter the commands as one line.)

```
putdev -a tapedrive1 range="SYS_PRIVATE-SYS_PUBLIC" \
       state="public" mode="static" startup="n" ual_enable="y" \
       users="100>n,101>n" other=">y"
```

```
putdev -a slowtape secdev="tapedrive1" cdevice="/dev/tape800"
```

```
putdev -a fasttape secdev="tapedrive1" cdevice="/dev/tape1600"
```

The preceding command sequence creates one secure device alias (**tapedrive1**) with the specified security attributes for the tape drive, and two logical device aliases (**slowtape** and **fasttape**) with the specified non-security attributes in the DDB.

However, one could create one entry per device with all security attributes specified on the command line:

```
putdev -a tape1 range="hilevel-lolevel" state="public" mode="static" \
       startup="n" ual_enable="y" users="100>n,101>n" other=">y" \
       cdevlist="/dev/tape800,/dev/tape1600" desc="tape device"
```

Another example for adding a disk (say disk number 4) to the Device Database on a machine running with Enhanced Security is:

```
putdev -a disk range="SYS_RANGE_MAX-SYS_RANGE_MIN" state="public" \
       mode="static" cdevice="/dev/rdsk/4s0" bdevice="/dev/dsk/4s0" \
       mountpt="/home" desc="Disk containing the /home"
```

The DDB can be queried for any alias, or attribute value using the devattr and getdev commands.

Files

```
/etc/device.tab
/etc/security/ddb/ddb_dsfmap installed by the Enhanced Security Utilities
/etc/security/ddb/ddb_sec installed by the Enhanced Security Utilities
```

Exit Codes

If **putdev** is successful, it returns an exit code of 0. Otherwise, it returns one of the following exit codes and prints the corresponding error message:

1   incorrect usage
       USAGE: putdev -a alias [attribute=value]...

2   Device Database in inconsistent state - notify administrator

2   Device Database could not be accessed or created

3   *alias* already exists in Device Database

3   *device* does not exist in Device Database

?   "*dsf*" already exists in Device Database

6   invalid alias or invalid pathname "*device*"

4   hilevel does not dominate lolevel in attribute range

4   invalid value for attribute "*attr*"
          level= "*level*" not defined in LTDB

4   essential security attributes (range,state,mode) must be specified
          together for "alias"

6   invalid value for attribute "*attr*"
          user/uid= "*uid*" not defined in system

6   invalid value for attribute "*attr*"
          group/gid= "*group*" not defined in system

6   invalid value for attribute "*attr*"
          invalid permissions specified "*perm*"

6   invalid value for attribute "*attr*"
          invalid delimiter specified in "*value*"

6   "*value*" multiply defined for attribute "*attr*"

6   "*alias*" does not define essential security
          attributes(range,state,mode)

6   "*alias*" must be defined with essential security
          attributes(range,state,mode).

6   cannot specify security attrs for alias and map to another secdev "*alias*"

6   Device Database in use. Try again later.

3   "*alias*" not defined in Device Database

5   system service not installed
          The Enhanced Security Utilities are not installed.

NOTES

Before modification, **putdev** will back up the most recent copies of the device database files (see **Files** above), and place them (with a **.old** extension) in the same path as the original. In the event that the device database becomes corrupted, you will need to copy all of the **.old** files back.

REFERENCES

devattr(1M), **getdev**(1M)