

Power Hawk™ Series 900 PowerMAX OS™

Version 6.3 Release Notes

January 2006

0891089-6.3



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, VYFD, VGM5, VSS4, VAFQ, ASTRix, Raptor DX, Manta QX3, and STAR 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	1
2.0. Power Hawk Series 900 System Overview	2
2.1. Functional Overview	2
2.2. Hardware Description	3
2.3. Power Hawk Series 900 CPU Board Descriptions	4
2.3.1. Model 910 Single CPU Board	4
2.3.2. Model 920 Dual CPU Board	4
2.3.3. Model 940 Quad CPU Board	4
2.4. Power Hawk Series 900 Specifications	4
3.0. Documentation	6
3.1. PowerMAX OS Software Documentation	6
3.2. Vendor Documentation	7
4.0. Prerequisites	8
4.1. Software	8
4.2. Hardware	8
4.2.1. CPU	8
4.2.2. System Console Terminal	8
4.2.3. SCSI Interface Controller	8
4.2.4. Disk Drive	8
4.2.5. CD-ROM Drive	8
4.2.6. Ethernet Controllers	8
4.2.7. High Speed Driver Enhanced (HSDE) Controller	9
4.2.8. MIL-STD-1553 Version 5 (V5) Interface	9
4.2.9. IEEE488 GPIB Bus Interface Controller	9
4.2.10. Asynchronous Communications Multiplexer (MVC)	9
4.2.11. High-Performance Asynchronous Communications Multiplexer (MVCS)	9
4.2.12. Real-Time Clock and Interrupt Module (RCIM)	9
4.2.13. Single-Head PMC Graphics Card	10
4.2.14. Four-Head PMC Graphics Card	10
4.2.15. VMIVME-2510B 64-bit TTL I/O Board	10
4.2.16. VMIVME-3122 High-Performance 16-bit A/D Converter Board	10
4.2.17. VMIVME-4140 Intelligent 32/16-Channel Analog Output Board	10
4.2.18. PBX3 3-slot PMC Expansion Board	10
5.0. Available Software Packages	12
5.1. Package Descriptions	17
5.1.1. Description of Standard Packages	17
5.1.2. Description of Drivers Packages	18
5.1.3. Description of TCP/IP Networking Packages	21
5.1.4. Description of Network File System Packages	21
5.1.5. Description of Security Package	21
5.1.6. Description of FBS Packages	21
5.1.7. Description of Diskless Package	22
5.1.8. Description of Software Development Packages	22
5.1.9. Description of X Window System Packages	23
5.1.10. Description of NightStar Tools Packages	23
5.1.11. Description of License Manager Packages	24
5.1.12. Description of Miscellaneous Packages	24
5.1.13. Power Hawk Series 700/900 Diagnostics	25
6.0. System Installation	26

6.1.	Large SCSI Disk Support	26
6.2.	System Disk Configuration	26
6.3.	Installation Modes	28
6.3.1.	Custom Mode of Installation	28
6.3.2.	Semi-Automatic Mode of Installation	28
6.4.	Installation Procedure	30
6.5.	Installing Additional Packages	33
6.6.	Installing Service Releases	33
7.0.	Rebooting The System	35
8.0.	General Notes	37
8.1.	XFS	37
8.2.	PCI-to-PCI Bridge Configuration	37
9.0.	General Information on Release 6.3	38
9.1.	Enhancements in this Release	38
9.1.1.	Loosely-Coupled Systems	38
9.1.2.	Firmware File Repository	39
9.2.	Known Problems	39
10.0.	Compatibility Issues Between Systems	41
11.0.	Service Releases	44
12.0.	Direct Software Support	45
13.0.	Manual Pages	46

1.0. Introduction

This document provides a general overview of PowerMAX OS™ Release Version 6.3 (Release 6.3). Release 6.3 is designed to provide support for the newly introduced Concurrent Computer Corporation Power Hawk™ Series 900 systems.

There are three models in the Series 900 line. The Model 910 uses a Raptor DX/VYFD VMEbus single board computer (SBC) manufactured by Synergy Microsystems, Inc. The Model 920 is the same as the Model 910 except that it has two processors. The Model 940 uses a Manta QX3/VAFQ VMEbus single board computer (SBC) manufactured by Synergy Microsystems, Inc. (note: These release notes will refer to the Synergy Raptor DX/VYFD SBC as the "VYFD SBC" and the Synergy Manta QX3/VAFQ SBC as the "VAFQ SBC").

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

Release 6.3 is distributed on two CD-ROM discs:

- ◆ PowerMAX OS Boot CD
- ◆ PowerWorks™ Standard Products CD

The PowerMAX OS Boot CD contains a bootable mini-kernel, install and restore software. The PowerWorks Standard Products CD contains the standard software packages that may be installed once the software on the PowerMAX OS Boot CD has been installed.

An optional products disc (see below) may also be provided if optional products have been previously ordered by the user.

- ◆ PowerWorks™ Optional Products CD

A service release disc (see below) containing updates, enhancements, patches, etc, may also be provided on an "as-released-basis" to users who are currently under a software maintenance contract.

- ◆ PowerWorks™ Service Release CD

2.0. Power Hawk Series 900 System Overview

2.1. Functional Overview

The Power Hawk Series 900 is Concurrent's high-performance computer system for real-time data acquisition, simulation, and industrial systems application. The Model 910 contains a single PowerPC™ MPC7455/7457 G4 CPU while the Model 920 contains two CPUs. The Model 910 and the Model 920 are contained on a VMEbus VYFD SBC with up to 1 GB of memory on a single VME card. The Model 940 contains four PowerPC MPC7457 G4 CPUs and is contained on a VMEbus VAFQ SBC with up to 2 GB of memory on a single VME card.

The MPC7455/7457 on the Model 910/920 is a fourth generation G4 PowerPC processor which consists of a processor core, 32 KB separate L1 instruction and data caches, a 256 KB L2 cache, and an internal L3 controller with tags that support a 2 MB backside L3 cache through a dedicated high-bandwidth interface. The core is a high-performance superscaler design supporting multiple execution units, including four independent units that execute AltiVec instructions.

The MPC7457 on the Model 940 is the fourth implementation of the fourth generation G4 microprocessors. The MPC7457 is similar to the MPC7455. The following are the major differences in comparison to the 7455.

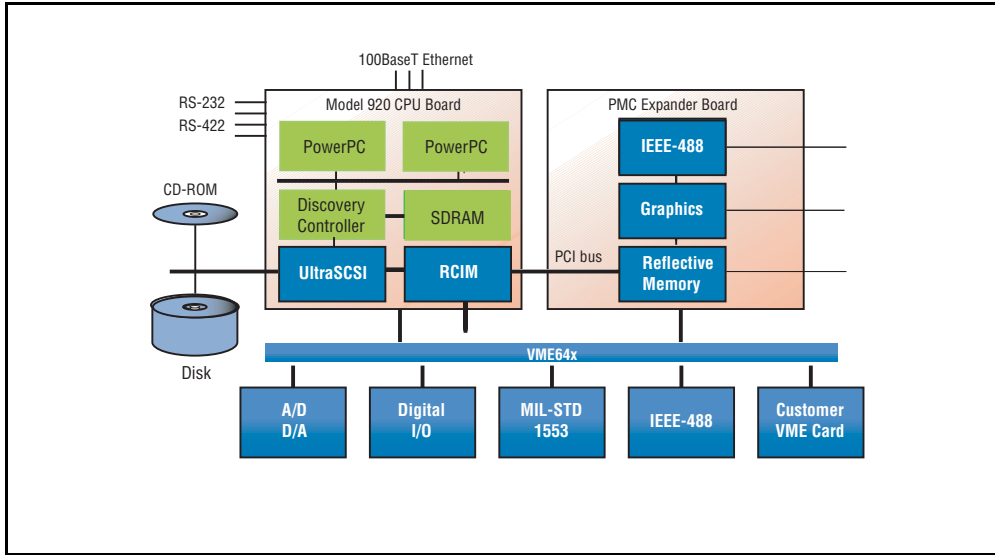
- Larger L2 cache (512 KB)
- L3 controller has increased DDR SRAM support up to 4 MB (max. of 2 MB for cache memory with the remaining 2 MB configured either as unused or as private memory).

The L3 cache is known as backside cache, because it is directly attached to the CPU via a private bus. The pre-G4 PPC processor Power Hawk systems had to negotiate for the system bus when accessing cache. This caused cache accesses to compete with memory accesses from I/O devices. This new cache scheme provides for better determinism in execution and since the cache bus is private, its speed can be increased as processor clock speeds increase.

AltiVec computational instructions are executed on four independent, pipelined AltiVec execution units. A maximum of two AltiVec instructions can be issued in order to any combination of AltiVec execution units per clock cycle, with as many as 10 AltiVec instructions executing concurrently. Depending on data size, AltiVec vectors are 4, 8 or 16 elements long. There are 32 AltiVec registers each 128-bits wide. AltiVec technology can be most accurately thought of as a set of registers and execution units added to the PowerPC architecture in an analogous manner to the addition of floating point units.

The block diagram below is of a Model 920 containing two CPUs. Model 910 is identical except it has a single CPU. Model 940 is identical but contains four CPUs.

Figure 2-1. Model 920 Block Diagram



2.2. Hardware Description

The Power Hawk 900 Series systems are available in rack mount or table-top options. The Series 900 chassis is available in 12-slot or 21-slot 6U VME configurations. The chassis contains integral peripheral bays and auto ranging power supplies. A typical Series 900 chassis with SBCs also shown, is illustrated below.

Figure 2-2. Series 900 Chassis



2.3. Power Hawk Series 900 CPU Board Descriptions

2.3.1. Model 910 Single CPU Board

The Power Hawk Model 910 is a 6U VME CPU module that contains one MPC7455/7457 PowerPC CPU with 128 MB to 1GB of main memory, 2MB of L3 cache and up to 128MB of flash memory. Also includes up to three 10/100BaseT Ethernet ports, two RS232 serial ports, nine real-time clocks, two PMC slots, and a PSCx Ultra 160/320 SCSI Interface PMC card.

2.3.2. Model 920 Dual CPU Board

The Power Hawk Model 920 is a 6U VME CPU module that contains two MPC7455/7457 PowerPC CPUs with 128 MB to 1GB of main memory, 2MB of L3 cache and up to 128MB of flash memory. Also includes up to three 10/100BaseT Ethernet ports, two RS232 serial ports, nine real-time clocks, two PMC slots, and a PSCx Ultra 160/320 SCSI Interface PMC card.

2.3.3. Model 940 Quad CPU Board

The Power Hawk Model 940 is a 6U VME CPU module that contains four MPC7457 PowerPC CPUs with 128MB to 2GB of main memory and 2 or 4MB of L3 cache. Also includes up to two 10/100/1000BaseT Ethernet Ports, four RS232 serial ports, five real-time clocks, one PMC slot, and a PSCx Ultra 160/320 SCSI Interface PMC Card.

2.4. Power Hawk Series 900 Specifications

Processor Board

- Model 910 - Single G4 PowerPC MPC7455/7457 G4
- Model 920 - Dual G4 PowerPC MPC7455/7457 G4
- Model 940 - Quad G4 PowerPC MPC7457 G4

Memory

- 128, 256, 512 MB, 1 GB, or 2GB
- 256 KB L2 and 2 MB or 4MB L3 Cache per CPU
- 128 MB User Flash
- 128 KB NVRAM and clock/calendar

I/O Interfaces

- VME64x (5-row), A32, D64
 - PCI bus to VMEbus bridge
- 2 PMC slots, 64-bit 33/66 MHz
- PSCx Ultra 160/320 SCSI Interface PMC card

Processor On-Board I/O

- Model 910/920 - up to three 10/100BaseT Ethernet ports
- Model 940 - two 10/100/1000BaseT Ethernet Ports.
- Model 910/920 - 2 RS232 serial ports
- Model 940 - 4 RS-132 serial ports.

Optional I/O Controllers

- 16-line asynchronous serial mux with RS-232, RS-422 and RS-485
- Single and quad-port 10/100BaseT Ethernet
- IEEE-488 GPIB
- MIL-STD-1553B with BC, RT and BM functions
- Graphics PMC cards
- A/D, D/A and digital I/O
- Real-Time Clock & Interrupt Module (RCIM)
- PMC Expander Modules

Peripherals

- 18, 36, 73 and 146 GB low profile 3.5" fixed and removable disk drives
- CD-ROM drive
- 4mm DAT cartridge tape drive
- ASCII terminal consoles

Chassis

- 12-slot and 21-slot 6U VME
- Integral peripheral bays
- Autoranging power supplies
- Rack mount and table-top options
- Optional ruggedized and ATR packaging

Environmental

- Operating temperature: 10° to 50° C
- Storage temperature: -40 ° to 85° C
- Relative humidity: 0% to 85%, non-condensing
- Altitude: 10,000 feet

Regulatory

- Safety: UL and CSA
- EMI/RFI: FCC Class A

3.0. Documentation

3.1. PowerMAX OS Software Documentation

Table 1-1 below lists the PowerMAX OS documentation available from Concurrent. 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	0890423-080
Character User Interface Programming	0890424-000
Device Driver Programming	0890425-070
STREAMS Modules and Drivers	0890426-020
User's Guide	0890428-010
PowerMAX OS System Administration Volume 1	0890429-080
PowerMAX OS System Administration Volume 2	0890430-090
PowerMAX OS Network Administration	0890432-050
Compilation Systems Volume 1 (Tools)	0890459-050
Compilation Systems Volume 2 (Concepts)	0890460-050
PowerMAX OS Real-Time Guide	0890466-100
PowerMAX OS Guide to Real-Time Services	0890479-110
Power Hawk 900 Series Console Reference Manual	0830060-120
Power Hawk 900 Series PowerMAX OS Version 6.3 Release Notes	0891089-6.3
Power Hawk 900 Series Diskless Systems Administrator's Manual	0891090-030

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.

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.

3.2. Vendor Documentation

Synergy commercial off-the-shelf (COTS) documentation applicable to the various Synergy Single Board Computers (SBC) are listed below. You may contact your local Synergy sales office to purchase Synergy manuals not provided with the system or visit the Synergy web site at <http://www.cwembedded.com/> for more information. See the table below for a list of Synergy manual names and applicable document numbers.

Manual Name	Synergy Document Number
Raptor DX VMEbus Dual G4, Dual PMC & StarFabric PowerPC Single Board Computer User Guide	02-0426/UG-VYFD-<REV>
MantaQX3/VAFQ VMEbus Quad G4, Single PMC & StarFabric PowerPC Single Board Computer User Manual.	815138 - Version <VER>
Synergy Microsystem STAR User Guide	03-0072/UG-STAR-<REV>
Synergy Microsystem ASTRix User Guide	03-0076/UG-ASTX-<REV>

4.0. Prerequisites

4.1. Software

Release 6.3 or later.

4.2. Hardware

4.2.1. CPU

Model 910 - single CPU SBC with minimum of 128 MB of memory.
or
Model 920 - dual CPU SBC with minimum of 128 MB of memory
or
Model 940 - quad CPU SBC with minimum of 128 MB of memory

4.2.2. System Console Terminal

Video display terminal such as a Wyse 150, vt100, or comparable device connected to Serial Port A.

4.2.3. SCSI Interface Controller

Minimum requirements are:

- At least one Synergy PSCx Ultra 160/320 SCSI Interface card.
- At least one supported SCSI disk drive for system files and swap space.
- At least SCSI CD-ROM device for software installation.

4.2.4. Disk Drive

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

The installation disk is supported on the PSCx SCSI/PMC card.

4.2.5. CD-ROM Drive

A supported SCSI CD-ROM device connected to the PSCx SCSI/PMC card controller is required for installation of system software.

4.2.6. Ethernet Controllers

4.2.6.1. Galileo GT64260 Ethernet Controller (Model 910/920)

Ethernet may be provided with 1 to 3 on-board GT64260 Ethernet controller ports.

4.2.6.2. Marvell MV64460 Ethernet Controller (Model 940)

Ethernet is provided by two on-board 10/100/100BaseT MV64460 Fast Ethernet ports.

4.2.6.3. PCI/PMC DEC Ethernet Controller

Additional networking connections are possible with selected PMC-based DEC 21143 controller chips. Contact your Concurrent representative to determine if a particular card is supported.

4.2.6.4. RAMiX 665/675 FX PMC Ethernet Controller

FX Ethernet networking connections are possible with the RAMiX 665/675 dual port 100BaseFX PMC Ethernet card.

4.2.7. High Speed Driver Enhanced (HSDE) Controller

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.

4.2.8. MIL-STD-1553 Version 5 (V5) Interface

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.

4.2.9. IEEE488 GPIB Bus Interface Controller

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 -.

4.2.10. Asynchronous Communications Multiplexer (MVC)

The 16-Line Asynchronous Communications Multiplexer with industry standard parallel port provides 16-ports of high-performance asynchronous serial communications. Supports 4-port RS-232D and RS-422 line control modules. Ports can run at speeds up through 38,400 baud.

4.2.11. High-Performance Asynchronous Communications Multiplexer (MVCS)

Supports individual line rates up to 1.5 Mbps, aggregate throughput of 48 Mbps, and independent baud rates for each port. Can be configured with up to four of any mix of four-port RS-232, RS-422 and RS-485 serial line modules. An optional 16-port I/O transition panel with RJ-45 connectors and internal cable is available.

4.2.12. Real-Time Clock and Interrupt Module (RCIM)

The RCIM is a multi-function PCI mezzanine card (PMC) that provides functions ideally suited for time-critical applications. Up to sixteen CPU boards, in the same or different chassis, can be linked via the RCIM's synchronization cable. The RCIM can be installed on any system with an available PMC slot. Refer to the *Real-Time Clock & Interrupt Module User's Guide* for more information.

4.2.13. Single-Head PMC Graphics Card

High-performance, single-head PMC graphics card. Includes a Cirrus Logic CL-GD7555 64-bit graphics controller, 4 MB display memory, two PS/2 keyboard/mouse ports, VGA BIOS, and 3-foot adapter cable. Supports true 24-bit color at resolutions up to 1280 x 1024 using Xlib.

4.2.14. Four-Head PMC Graphics Card

High-performance, four-head PMC graphics card. Includes four Cirrus Logic CL-GD7555 64-bit graphics controllers, 4 MB display memory per head, eight PS/2 keyboard/mouse ports, VGA BIOS, and 3-foot adapter cables. Supports true 24-bit color at resolutions up to 1280 x 1024 using Xlib.

4.2.15. VMIVME-2510B 64-bit TTL I/O Board

The direction of each 8-bit port is individually programmable. Separate board address decoding for control and data registers. Built-in-Test logic for fault detection and isolation. Compatible with VMIC's family of Intelligent I/O Controllers (VMIVME-90xx series).

4.2.16. VMIVME-3122 High-Performance 16-bit A/D Converter Board

Supports 64 channels of differential or single-ended wide range analog inputs.

4.2.17. VMIVME-4140 Intelligent 32/16-Channel Analog Output Board

Provides 32 high quality analog output channels with 12-bit resolution. Each output has a dedicated D/A Converter (DAC) assigned to it.

4.2.18. PBX3 3-slot PMC Expansion Board

The PBX3 Expansion board provides 3 single-width PMC slots at 66 MHz, and connects to the SBC through a stackable PMC card or adapter. The stacking design provides a PMC expansion solution without giving up the use of an on-board PMC slot.

5.0. 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	Power Hawk Series 700	Power Hawk Series 900	Package Name	Package Dependencies
Standard:				
Base System	y	y	base	
Advanced Commands	y	y	cmds	lp,nsu
Open Source	y	y	open-source	
Domestic Encryption Utilities	y	y	crypt	
International Encryption Utilities	y	y	crypt-int	
Kernel Debugger	y	y	kdb	
Network Support Utilities	y	y	nsu	
OA&M	y	y	oam	cmds
Online Manual Pages	y	y	man	
Printer Support	y	y	lp	
Software Packaging Tools	y	y	softint	
Terminfo Utilities	y	y	terminf	
Drivers:				
CD-ROM Driver	y	y	cdfs	
Generic Floppy Driver	y	y	gf	
Generic SCSI Device Driver	y	y	gs	
HSDE	y	y	hsde	
IEEE-488 Bus (GPIB) Kernel Level Driver	y	y	gpib	
1553 V5 ABI Driver	y	y	1553v5drv	
1553 V5 ABI Libraries	y	y	1533v5lib	1553v5drv
NCR SCSI Driver	y	y	ncr	

Table 1-2. Software Packages (Cont.)

Package Description	Power Hawk Series 700	Power Hawk Series 900	Package Name	Package Dependencies
Drivers (Continued):				
DEC Ethernet Driver	y	y	dec	nsu
GT64260 Ethernet Driver	y	y	gte	nsu
MV64460 Ethernet Driver	y	y	mve	nsu
Symbios Fast Ethernet Driver	y	y	sym	nsu
RAMiX PMC FOB Ethernet Driver	y	y	rmxf	nsu
Asynchronous Communications Multiplexer	y	y	mvc	
High-Performance Asynchronous Communications Multiplexer	y	y	mvcs	
National Instruments GPIB User-Level Device Driver (GPIB)	y	y	ngpib	
VMIC High Speed Data (HSD) Driver	y	y	vhsd	
V2510 User Level Driver	y	y	v2510-620	
V3122 User Level Driver	y	y	v3122-620	
V4140 User Level Driver	y	y	v4140-620	
TCP/IP Networking:				
Internet Utilities	y	y	inet	nsu
Commands Networking Extension	y	y	netcmds	lp,inet
Remote Procedure Calls Utilities	y	y	rpc	inet

Table 1-2. Software Packages (Cont.)

Package Description	Power Hawk Series 700	Power Hawk Series 900	Package Name	Package Dependencies
Network File System:				
Network File System Utilities	y	y	nfs	nsu, inet, rpc, dfs
Distributed File System Utilities	y	y	dfs	inet
Security:				
Auditing	y	y	audit	
Frequency-Based Scheduler (FBS):				
Frequency-Based Scheduler and Performance Monitor	y	y	fbs	
Frequency-Based Scheduler Manual Pages	y	y	fbsman	
Closely-Coupled Systems: (Series 700 only) Loosely-Coupled Systems: (Series 700/900)				
Diskless	y	y	diskless	nfs, netcmds, sym, gte and mve
Software Development:				
Fortran 77 Compilation System	y	y	hf77	
C++ Compilation System	y	y	c++	
MAXAda Compilation System	y	y	MAXAda	analyze
AXI for MAXAda	y	y	MAXaxi	MAXAda, x11
Cross-Development Package Set	y	y	See CrossDev package	See CrossDev package

Table 1-2. Software Packages (Cont.)

Package Description	Power Hawk Series 700	Power Hawk Series 900	Package Name	Package Dependencies
Window System:				
X Window System with OSF/Motif:	y	y	x11ipc	
			x11	x11ipc
			x11progs	x11ipc, x11
			x11dev	x11ipc, x11, x11progs
			x11pman	
			x11dman	
Metro-X X11R6 Server	y	y	metroess	
X11R6 Fonts	y	y	xfonts	
Metro Link OpenGL	y	y	metrogl	
NightStar Tools:				
NightStar	y	y	nstar	
NightBench	y	y	nbench	x11
NightProbe	y	y	nprobe, nprobeserv	x11, nstar
NightSim	y	y	nsim, nsimserver	x11, nstar
NightTrace	y	y	ntrace, ntracelog	x11, nstar
NightTune	y	y	ntune	x11
NightView	y	y	NightView	x11
NightView Debugger Support	y	y	Nviewp	

Table 1-2. Software Packages (Cont.)

Package Description	Power Hawk Series 700	Power Hawk Series 900	Package Name	Package Dependencies
License Manager:				
Élan License Manager5	y	y	elan5lm	
Élan License Manager4	y	y	elan4lm	elan5lm
Miscellaneous:				
Distributed XFS File System	y	y	xfsd	inet, rpc
Virtual Partition	y	y	vp	
Real-Time Libraries	y	y	librt	
Power Hawk Diagnostic:				
Release 9.8	y	y	n/a	

5.1. Package Descriptions

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

5.1.1. Description of Standard Packages

Base System (`base`)

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

Advanced Commands (`cmds`)

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

Network Time Protocol (`open-source`)

`xntpd` is a complete implementation of the Network Time Protocol (NTP) version 2 standard as defined by RFC 1119, and also retains compatibility with version 1 servers as defined by RFC 1059. `xntpd` does all computations in fixed point arithmetic and is entirely free of floating point code. The computations done in the protocol and clock adjustment code are carried out with high precision and with attention to the details which might introduce systematic bias into the integrations, to try to maintain an accuracy suitable for synchronizing with even the most precise external time source. `Xntp` is part of the open-source package.

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`)

The International Encryption Utilities package supports the encryption of files and other data. This package is for international distribution.

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.

Network Support Utilities (`nsu`)

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

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.

Online Manual Pages (`man`)

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

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.

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.

Terminfo Utilities (`terminf`)

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

5.1.2. Description of Drivers Packages

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.

Generic Floppy Device Driver (`gf`)

The `gf` driver is a generic CD-ROM device driver. This driver supports the SCSI Floppy device connected to the internal NCR/Symbios ControllerNCR.

Generic SCSI Device Driver (`gs`)

The `gs` driver package supports the following SCSI devices: SCSI-based CPUs, scanners, printers, media changers, and communication devices.

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 Power Hawk 700/900 System and an external device using the Encore HSD Interface Model 9132 protocol.

IEEE-488 Bus (GPIB) Kernel Level Driver (`gpib`)

The `gpib` provides a high performance Plug and Play IEEE 488 interface for the PCI Bus. The `gpib` device driver will multiplex up to four independently configured IEEE 488 buses and conforms to the IEEE STD-488.1 specification. Talker, Listener, and Controller interface functions are supported.

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.

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.

DEC 21x4x Ethernet Driver (`dec`)

This driver package supports the DEC 21143 controller chip on separate PMC cards. This package also provides for automatic configuration of the controller during system initialization (e.g., no **Sadapters (4)** file changes required).

10/100BaseT, BNC and AUI connections are supported.

Galileo/Marvell Discovery Ethernet Ports (`gte`)

Model 910/920. This driver package supports the Galileo/Marvell GT64260 Discovery system controller chip in Power Hawk 910/920 systems. The GT64260 supports up to three independent 10/100BaseT Fast Ethernet ports. Each port is fully compliant with the IEEE 802.3 and 802.3u standards and integrates the MAC function and a dual speed RMII interface.

Marvell Discovery III Ethernet Ports (`mve`)

Model 940. This driver package supports the Marvell MV64460 Discovery III System Controller chip in Power Hawk Model 940 system. The MV64460 supports up to two independent 10/100/1000BaseT Fast Ethernet ports. Each port is fully compliant with the IEEE 802.3 and 802.3u standards and integrates the MAC function with RMI (10/100BaseT) and RGMII (1000BaseT) speed interfaces.

Symbios Fast Ethernet Driver (`sym`)

This driver package supports the Symbios 53C885 Ethernet controller chip on the Power Hawk Series 700 SBCs. One 10/100BaseT Fast Ethernet port is supported on the Power Hawk 700 SBCs.

Although the Symbios Ethernet hardware is not available on Power Hawk Series 900, this package must be installed on Power Hawk Series 900 SBCs before the Diskless package may be successfully installed on Series 900 SBCs.

RAMiX 665/675 Failover Board Ethernet Adapter (`rmxf`)

Ethernet Adapter provides Ethernet driver support for RAMiX 665 and 675 boards running the RAMiX Failover Board firmware.

Asynchronous Communications Multiplexer (`mvc`)

This package supports the Asynchronous Communications Multiplexer (MVC) adapter board. The 16-Line Asynchronous Communications Multiplexer with industry standard parallel port provides 16-ports of high-performance asynchronous serial communications. Supports 4-port RS-232D and RS-422 line control modules. Ports can run at speeds up through 38,400 baud.

High-Performance Asynchronous Communications Multiplexer (`mvcs`)

This package supports the High-Performance Asynchronous Communications Multiplexer (MVCS). The MVCS adapter board supports individual line rates up to 1.5 Mbps, aggregate throughput of 48 Mbps, and independent baud rates for each port. Can be configured with up to four of any mix of four-port RS-232, RS-422 and RS-485 serial line modules.

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

The `ngpiib` 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 700/900 systems.

V2510 Driver (`v2510`)

This driver provides support for the CE2510 64-bit TTL I/O board.

V3122 Driver (`v3122`)

This driver provides support for the CE3122 64/32/16-Channel, 16-Bit High Performance Analog to Digital converter board.

V4140 Driver (v4140)

This driver provides support for the CE4140 intelligent 32/16-channel, 12-bit analog output board with D/A per channel and auto-calibration

5.1.3. 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.

5.1.4. 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.

5.1.5. 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.

5.1.6. 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.

5.1.7. 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 SBC's 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 SBC's in the chassis.

A closely-coupled system consists of multiple Power Hawk 700 Series SBC's residing in the same chassis.

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

This package also supports multiple SBC's in one or more remote VME chassis connected via Ethernet (referred to as a loosely-coupled system). Diskless SBC's 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.

Power Hawk Series 900 systems are supported in loosely-coupled configurations, but they are NOT supported in closely-coupled configurations. The Synergy-based loosely-coupled configurations may contain a mix of Series 700 and 900 systems.

5.1.8. Description of Software Development Packages

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 6.1 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 architecture of the target system. The 6.1 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.

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

Consists of 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.

5.1.9. 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.

5.1.10. Description of NightStar Tools Packages

NightStar (nstar)

This package provides a NightStar daemon that initiates individual NightStar tool servers and daemons on behalf of a (possibly remote) NightStar client tool.

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, nprobeserv)

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, nsimserver)

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, ntracelog)

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.

5.1.11. Description of License Manager Packages

Élan License Manager (elan51m)

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 (elan41m)

This package is an optional 'old' (version 4) license manager daemon that can run in parallel with the new (elan51m) license manager daemon on the same server machine in order to support serving licenses to both 'old' and 'new' tools at the same time.

5.1.12. 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).

Real-Time Libraries (librt)

The librt package provides the Real-Time libraries.

5.1.13. Power Hawk Series 700/900 Diagnostics

Software diagnostics on CD-ROM for a single Power Hawk Series 700/900 system. Includes PowerIO and PowerAT diagnostics. The Diagnostic CD is purchased separately from the OS.

6.0. 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 28 for additional details.

NOTE

A system may be booted using an installation CD-ROM disk. However, only tape is supported as the backup (restore) media. This means you can boot from a CD-ROM disc and once booted from disc, you can restore from a backup tape. However, just before the **base** package is installed, some tools used for the RESTORE function are removed to free up memory. If you need to use the RESTORE function after executing through many levels of the INSTALL menus you must reboot first from disc.

6.1. 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 Release 6.3 without being re-formatted.

If you have a disk larger than 8 GB that was formatted using the Release 4.1 or earlier 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 6.3 installation.
4. Restore any data backed up in step 1 above.

6.2. System Disk Configuration

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

Table 1-3. ufs root file system

partition	file	Disk < 9GB Default Partition size	Disk > 9GB Default Partition size
0	root	200 MB	500 MB
1	swap	500 MB	2000 MB
2	usr	500 MB	2000 MB

Table 1-3. ufs root file system (Cont.)

partition	file	Disk < 9GB Default Partition size	Disk > 9GB Default Partition size
3	var	500 MB	2000 MB
4	----	<remainder>	<remainder>
6 ²	boot	1024 KB	1024 KB

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

Table 1-4. xfs root file system

partition	file	Disk < 9 GB Default Partition Size	Disk > 9GB Default Partition size
0	/stand	100 MB	100 MB
1	swap	500 MB	2000 MB
2	usr	500 MB	2000 MB
3	var	500 MB	2000 MB
4	----	<remainder>	<remainder>
5 ¹	root	200 MB	500 MB
6 ²	boot	1024 KB	1024 KB

1. The `xfs` root must be on partition 5.
2. Partition is applicable to Power Hawk Series 700/900 boot disks only.

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.

The total amount of swap space for MAXAda development systems should be at least 3 times the size of physical memory. This larger swap space compared to runtime systems is required because of the virtual memory expansion of the compiler.

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.

6.3. 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.

6.3.1. 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 predetermined 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.

6.3.2. 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 predetermined set of responses. Note that in semi-automatic mode, **all** the additional products on the release distribution media 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 supported 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) no. no.
elan5lm	Install man pages? Start license manager during system boot?	yes yes
oam	sysadm password	"" (null password)
hsde	Number of hsde adapters on VME bus	1
1553v5drv	Number of 1553V5-ABI adapters	1
mvc	Number of mvc adapters on VME bus Configure real-time driver?	1 no
inet	Configure TCP listener?	yes
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
nstar	Install directory for NightStar	standard location relative to root directory
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

6.4. Installation Procedure

It is recommended that the user read the *Power Hawk Series 900 Console Reference Manual* (Pubs No. 0830060) beforehand in order to be familiar with the Series 900 console commands used throughout these release notes.

The Power Hawk Model 920/940 platform uses two monitor ROM programs called **STAR** (**S**elf **T**est **A**nd **R**un) and **ASTRix** (**A**dvanced **S**elf **T**est **R**eboot **I**nitialize **eX**ecute) that provide basic board initialization, configuration and diagnostic capability. The STAR utility will execute whenever the system is powered up or is reset. STAR and ASTRix are used to configure the basic system and to load the initial boot image from distribution media. The individual commands used in STAR and ASTRix are documented in the Synergy Microsystem STAR User Guide (03-0072/UG-STAR), and the Synergy Microsystem ASTRix User Guide (03-0076/UG-ASTX) respectively.

NOTE

The PCI Memory Space layout, which is setup with the STAR config command, MUST be set to the proper values before a PowerMAX OS kernel can be successfully booted. The STAR config command should be used to verify that the PCI Memory parameters are set to the following default values:

```
PCI 0 Start Addr (ex. 0x80000000): 0x80000000 =
PCI 0 Size (ex. 0x40000000 (1GB)): 0x30000000 =
PCI 1 Start Addr (ex. 0xB0000000): 0xC0000000 =
PCI 1 Size (ex. 0x10000000 (256MB)): 0x40000000 =
```

Please consult the Power Hawk Series 900 Console Reference Manual (Pubs No. 0830060) for more information on this subject.

STAR/ASTRix commands are used to boot the installation image from the distribution media. The time to complete the entire installation will vary depending on the packages selected to be installed and the type of drive being used.

It may be necessary to use operator intervention if the PowerMAX OS Console Software has been previously loaded into flash memory. If the "STAR0>" prompt does not appear on the console terminal at power up, it will be necessary to power cycle or reset the board and use <CTRL>c to obtain a STAR prompt, and then enter the following commands to load the PowerMAX OS installation image.

```
STAR0>config
...
Change the
Enable "post" script:           Y =
to N
STAR0> astrix
Y: CPU 1 standing by.
```

```

Synergy Microsystems ASTRix Rev: 1.01.50 Jan 10 2006 08:58:19

*> scsi-drivers

Using scsi/scsi_mod.o
Using scsi/sd_mod.o
Using cdrom/cdrom.o
Using scsi/sr_mod.o
Using scsi/sym53c8xx.o
Loaded SCSI modules.
Now mount your drive by typing
(for example) mount /dev/sda1 /mnt .

*> boot /dev/cdrom1
Using /lib/modules/2.4.19-Asmp/misc/reboot.o

Synergy Microsystems ASTRix Rev: 1.01.50 Jan 10 2006 08:58:19

PowerMAX OS Kernel Decompressor
...
(OS is now booting)

```

General Notes

The **cdfs** (CD-ROM File System) package (including the generic CD-ROM driver (**gr**)) is always automatically installed following the **base** package installation.

A system may be booted using an installation CD-ROM disc. However, only DAT tape is supported as the backup (restore) media. This means a user can boot from a CD-ROM disc and once booted from the CD-ROM disc, the user is then able to restore from a backup tape.

As a side note, just before the **base** package is installed, some tools used for the **RESTORE** function are removed to free up memory. If you the user wants to use the **RESTORE** function after executing many levels of the **INSTALL** menus you must first reboot from CD-ROM disc or DAT tape, whichever is applicable.

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 28)

Node name

Timezone

Time/Date

Desired file system types

System disk location (slot and unit number)

Installation media device location

System disk configuration (see *System Disk Configuration* on page 26)

NOTE

At this point in the installation sequence, the system disk is initialized and you will be prompted on-screen to insert the PowerWorks Standard Products media.

License key for number of users

License key for number of processors

Following are applicable to custom mode installation only:

Which additional software packages should be installed (see *Installing Additional Packages* on page 33)

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 media in use is accessed at various times and must be kept in the drive during the installation until you are instructed to install different media.

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.

NOTE

When installation is completed the STAR/ASTRix Automatic initialization procedure contained in the PowerMAX OS Console Reference Manual (0830060) should be completed. This will place the Console Software in flash memory to improve boot time of the system.

6.5. Installing Additional Packages

In the semi-automatic installation mode **all** software packages provided on the PowerWorks Standard Products CD 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.

To install additional packages on an installed system, use the **pkgadd (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.

The following procedure will enable you to install additional packages as you choose.

NOTE

The system must be configured with the **gr** (generic CD-ROM device driver) **gr (7)** and **cdfs** (CD-ROM filesystem) modules and, in addition, an entry must exist for the CD-ROM device nodes in **/etc/conf/node.d/gr**.

In the example shown below, the CD-ROM device is **/dev/cd/0** and the CD-ROM disc is mounted under **/mnt/cdrom**.

1. Insert the PowerWorks Optional Products disc into the CD-ROM drive and mount the file system as shown below.

```
mkdir /mnt/cdrom
```

```
mount -F cdfs -r /dev/cd/0 /mnt/cdrom
```

2. Get a list of the packages on the CD-ROM disc and verify that the package you wish to install is on the CD-ROM disc.

```
pkginfo -d /mnt/cdrom/pkgs.dstream
```

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

```
pkgadd -d /mnt/cdrom/pkgs.dstream nsu lp cmds
```

4. The package(s) to be installed may be selected from a menu of all the packages available on the CD-ROM disc by not specifying any package name, as in:

```
pkgadd -d /mnt/cdrom/pkgs.dstream
```

6.6. Installing Service Releases

To install packages on the PowerWorks Service Release CD use **pkgadd (1M)**. 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.

The following procedure(s) enable you to install the packages on the PowerWorks Service Release CD as you choose.

NOTE

The system must be configured with the **gr** (generic CD-ROM device driver) **gr (7)** and **cdfs** (CD-ROM filesystem) modules and, in addition, an entry must exist for the CD-ROM device nodes in **/etc/conf/node.d/gr**.

In the example shown below, the CD-ROM device is **/dev/cd/0** and the CD-ROM disc is mounted under **/mnt/cdrom**.

1. Insert the PowerWorks Service Release disc into the CD-ROM drive and mount the file system as shown below.

```
mkdir /mnt/cdrom  
mount -F cdfs -r /dev/cd/0 /mnt/cdrom
```

2. Get a list of the packages on the CD-ROM disc and verify that the package you wish to install is on the CD-ROM disc.

```
pkginfo -d /mnt/cdrom/pkgs.dstream
```

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

```
pkgadd -d /mnt/cdrom/pkgs.dstream nsu base-001 \  
base-002 base-003
```

4. The package(s) to be installed may be selected from a menu of all the packages available on the CD-ROM disc by not specifying any package name, as in:

```
pkgadd -d /mnt/cdrom/pkgs.dstream
```

5. View the service release README File to get information about the contents of the service release.

```
pkginfo -Rd /mnt/cdrom/pkgs.dstream
```

7.0. Rebooting The System

During the installation, a new kernel is built. When the system installation completes, the system first halts and then returns to the STAR0> prompt. Enter the following commands to then load and execute the console processor:

```
STAR0>config
...
...
Change the
Enable "post" script:           Y =
to Y

STAR0> astrix           (loads ASTRix)
* scsi-drivers         (loads the SCSI device support)
* boot /dev/sda       (loads console processor into memory,
                        resets the board, and then starts its
                        execution)
```

In the above example, the 'a' in `/dev/sda` assumes that the system disk is disk 'a' in the ASTRix environment.

If a kernel was successfully built during the system installation, execute the following commands from the console (refer to the *Power Hawk Series 900 Console Reference Manual*, Pubs No. 0830060, for information on console command syntax):

```
#0>y.
#0>pboot 0
#0>fd dsk(c,u,p,b,l)
#0>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:

```
#0>pboot 3
00000000
#>fd dsk(c,u,p,b,l)
.....

#0>fb
Set Run Mode
CPU 0 CPU 1
dsk(0,0,0,0,0)/.
Initialize VME
dsk(0,0,0,0,0)/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
```

NOTE

If the system appears to be hung during the reboot to the new kernel, power cycle the machine and repeat the steps listed above.

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 section pertaining to `xfst` file system and the use of “non-standard” kernel names in `/stand`.

8.0. General Notes

8.1. 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`.

8.2. 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.

9.0. General Information on Release 6.3

This section provides a general overview of PowerMAX OS™ Release 6.3.

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 continue 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
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	
Power Hawk Series 900	Power Hawk Model 910 (single processor)	bspvyfd	bspall
	Power Hawk Model 920 (dual processor)	bspvyfd	
	Power Hawk Model 940 (quad processor)	bspvafq	

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.

9.1. Enhancements in this Release

9.1.1. Loosely-Coupled Systems

Previous PowerMAX OS releases have supported closely-coupled (CCS) and loosely-coupled systems (LCS) on Power Hawk Series 600/700 systems.

This 6.3 release provides loosely-coupled (LCS) support for Power Hawk Model 900 Series systems.

Power Hawk 900 Series may be used as the server or client(s) in a LCS configuration that contains any mix of Power Hawk Series 700 and Series 900 systems.

The configuration requirements for a Power Hawk Series 900 LCS configuration are:

- All Series 700 SBCs should have a Symbios (*sym*) Ethernet interface connection to the server SBC, and all Series 900 SBCs should have a GT64260 (*gte*) or MV64460 (*mve*) Ethernet interface connection to the server SBC.
- The server SBC may be a Series 700 system or a Series 900 system with ethernet access to all of the configured client Series 700/900 SBCs.
- 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.

NOTE

Release 6.3 does NOT provide any support for closely-coupled (CCS) configurations for Power Hawk Series 900 systems.

9.1.2. Firmware File Repository

In this release of PowerMAX OS various STAR firmware files have been gathered together into a central location. These firmware files can be used to modify the STAR firmware on an Series 900 system. The files will be distributed with the OS and stored in `/var/sadm/STAR/<filename>`. Along with the firmware files is a `Series900_FirmwareUpdate.rdme` file which documents how to modify the firmware on a Series 900 system.

Although these files will now exist locally on the user's end system, it is still advisable to contact Concurrent Customer Support prior to changing any firmware file.

It is intended that (going forward) this repository of firmware files will be maintained with up-to-date versions.

9.2. Known Problems

The following is a list of known problems which Concurrent is working to resolve. If a work around is available, then it is listed in the *Known Workaround* column below. These problems are deemed to have low impact upon the use of the operating system.

Table 1-6. PowerMAX OS 6.3 Known problems

Bugzilla Number	Summary	Known Workaround
597	Intermittent errors when booting from CD.	The problem is currently under investigation. Continued attempts succeed.
600	Package install errors received when attempting base installation.	The problem is under investigation. If the SCSI is in PMC Slot 1 and something (anything) else is in PMC Slot 2, then the problem does not occur.
3205	MVE Ethernet port Gigabit performance is too slow.	Power Hawk 940 Systems still have inferior performance numbers on the 1000BaseT (Gigabit). The performance numbers for the 10BaseT and 100BaseT are excellent.
4831	ftw() errors during boot if the swap space size is larger than 4096 MB.	There are data type limitations which must be corrected to allow for swap space sizes greater than 4096 MB.

10.0. Compatibility Issues Between Systems

PowerMAX OS for the Power Hawk Series 600/700/900 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 not available on the PowerStack III (MTX PCI Series (7-slot) systems).

Three integral RTCs are available on the Power Hawk Series 700, nine integral RTCs are available on the Power Hawk Series 910/920, and 5 integral RTCs are available on the Power Hawk Series 940. Installing a Real-Time Clocks and Interrupts Module (RCIM) will add four additional RTCs. On the Series 700/900, all of the integral RTCs have the same fixed interrupt level, and each RTC has its own interrupt vector associated with it. The Series 700/900 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 Series 600 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]`. On the Power Hawk Series 700 systems, the tick timers have device names in the form of `/dev/rrtc/0c[0-3]`, the Power Hawk Series 910/920 systems have tick timers device names in the form of `/dev/rrtc/0c[0-8]` and Power Hawk Model 940 has device names in the form of `/dev/rrtc/0c[0-4]`. If an RCIM is installed, the four additional RTCs would have the device 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 `rrtc(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 Series 600/700/900 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 600/700/900 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, platform-specific spl values, as these spl values may vary between platforms. Note that the `spl_manage()` calls make use of generic `PL_xx` spl values that are compatible across all PowerMAX platforms. Regardless of which type of spl values you use, an spl value of 0 allows all interrupts to occur, and higher spl values prevent increasingly more interrupts from occurring. The 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, Model 710, and Model 910 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 not on the Power Hawk Series 600/700/900 and PowerStack II systems. Therefore, there is no 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') platforms. The `nvr` 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 700/900, 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.
9. The physical CPU addresses that are used to access the physical VME bus address spaces (A32, A24, A16) differ between Power Hawk Series 700 and 900 systems. Therefore, applications or user-level device drivers that access VME bus address spaces directly from user-space should use the `vme_address(3C)` library routine to obtain the correct physical CPU addresses to use for creating virtual address translations to VME bus addresses.
10. On other Power Hawk platforms, kernel tunables define the address range for VME A32 address space. However, on Power Hawk Series 900 platforms the address range for VME A32 address space is directly determined from the PCI Memory Space parameters in the STAR config command. Please consult the *Power Hawk Series 900 Console Reference Manual* (Pubs No. 0830060) section "PCI Memory Space Configuration" for more information on this subject.

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/isd_support.asp

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. The Software Support Center operates Monday through Friday from 8 a.m. to 5 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.

You may also submit a request for assistance at any time by using the Concurrent Computer Corporation web site at [http:// www.ccur.com/isd_support_contact.asp](http://www.ccur.com/isd_support_contact.asp) or by sending an email to support@ccur.com.

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.

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 rebuild 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, i.e., 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.

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 installing 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. Device table entry must have volume attribute set initially via **putdev** command, otherwise, **pkgadd** will abort. Typically usage:

```
# putdev -a ctape1 cdevice=/dev/rmt/0m volume="volume=""
```

(See **putdev** for additional information.)

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. 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 instantiations 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 preinstall 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 preinstall 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 uncompress any files that were already compressed (that is, only those in “.Z” form) before being processed by **pkgmk**.

The **pkgadd** command, when invoked with the **-d device** option to add packages from the *device* to the system, will copy files from the *device* to the default temporary directory: **/var/tmp**. If the **/var** file system does not contain enough free space to complete this command successfully, then the command user may set and export the environment variable, **TMPDIR**, to some temporary directory on a file system that does contain enough free space and re-execute the command. Once the command completes successfully, the temporarily extracted files are removed from the system.

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) .]
/var/tmp	default location of TMPDIR

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)**

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).

erasescmd

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 **type=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).

removable

This should be set to true to identify a device as a removable one (such as a tape device).

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 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 be 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>rxw*. 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>rxw*. 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: *>rxw*. 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 **ua1_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 *devlist* must 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 or 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/rdisk/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_dsfnmap 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)

This page intentionally left blank

