



PowerHub 7000/8000 Software Reference Manual

MANU0167-02 - Rev. A - July 27, 1998

Software Version PH_FT 5.0.x

FORE Systems, Inc.

1000 FORE Drive
Warrendale, PA 15086-7502
Phone: 724-742-4444
FAX: 724-742-7742

<http://www.fore.com>

Legal Notices

Copyright © 1995-1998 FORE Systems, Inc. All rights reserved. FORE Systems is a registered trademark, and *ForeRunner*, *ForeView*, *ForeThought*, *ForeRunnerLE*, *PowerHub*, and *CellPath* are trademarks of FORE Systems, Inc. All other brands or product names are trademarks of their respective holders.

U.S. Government Restricted Rights. If you are licensing the Software on behalf of the U.S. Government (“Government”), the following provisions apply to you. If the Software is supplied to the Department of Defense (“DoD”), it is classified as “Commercial Computer Software” under paragraph 252.227-7014 of the DoD Supplement to the Federal Acquisition Regulations (“DFARS”) (or any successor regulations) and the Government is acquiring only the license rights granted herein (the license rights customarily provided to non-Government users). If the Software is supplied to any unit or agency of the Government other than DoD, it is classified as “Restricted Computer Software” and the Government’s rights in the Software are defined in paragraph 52.227-19 of the Federal Acquisition Regulations (“FAR”) (or any successor regulations) or, in the cases of NASA, in paragraph 18.52.227-86 of the NASA Supplement to the FAR (or any successor regulations).

Printed in the USA.

No part of this work covered by copyright may be reproduced in any form. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

This publication is provided by FORE Systems, Inc. “as-is” without warranty of any kind, either express or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose. FORE Systems, Inc. shall not be liable for any errors or omissions which may occur in this publication, nor for incidental or consequential damages of any kind resulting from the furnishing, performance, or use of this publication.

Information published here is current or planned as of the date of publication of this document. Because we are improving and adding features to our products continuously, the information in this document is subject to change without notice.

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (October 1988) and FAR 52.227-19 (June 1987).

The VxWorks software used in the Mini Loader is licensed from Wind River Systems, Inc., Copyright ©1984-1996.

FCC CLASS A NOTICE

WARNING: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void this user’s authority to operate this equipment.

NOTE: The PowerHub 7000/8000 has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15, FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

DOC CLASS A NOTICE

This digital apparatus does not exceed Class A limits for radio noise emission for a digital device as set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le present appareil numerique n’emet pas de bruits radioelectriques dépassant les limites applicables aux appareils numeriques de la class A prescrites dans le reglement sur le brouillage radioelectrique edicte par le ministere des Communications du Canada.

VCCI CLASS 1 NOTICE

この装置は、第一種情報処理装置（商工業地域において使用されるべき情報処理装置）で商工業地域での電波障害防止を目的とした情報処理装置等電波障害自主規制協議会(VCCI)基準に適合しております。

従って、住宅地域またはその隣接した地域で使用すると、ラジオ、テレビジョン受信機等に受信障害を与えることがあります。

取扱説明書に従って正しい取り扱いをして下さい。

This equipment is in the Class 1 category (Information Technology Equipment to be used in commercial and/or industrial areas) and conforms to the standards set by the Voluntary Control Council For Interference by Information Technology Equipment aimed at preventing radio interference in commercial and/or industrial areas. Consequently, when used in a residential area or in an adjacent area thereto, radio interference may be caused to radios and TV receivers, etc. Read the instructions for correct handling.

CE NOTICE

Marking by the symbol **CE** indicates compliance of this system to the EMC (Electromagnetic Compatibility) directive of the European Community and compliance to the Low Voltage (Safety) Directive. Such marking is indicative that this system meets or exceeds the following technical standards:

- EN 55022 - "Limits and Methods of Measurement of Radio Interference Characteristics of Information Technology Equipment."
- EN 50082-1 - "Electromagnetic compatibility - Generic immunity standard Part 1: Residential, commercial, and light industry."

SAFETY CERTIFICATIONS

ETL certified to meet Information Technology Equipment safety standards UL 1950 3rd Edition, CSA22.2, No. 950-95, EN 60950 (1992) and IEC 950, 2nd Edition.

CANADIAN IC CS-03 COMPLIANCE STATEMENT

NOTICE: The Industry Canada label identifies certified equipment. This certification means that the equipment meets certain telecommunications network protective, operational and safety requirements. The Industry Canada label does not guarantee the equipment will operate to the user's satisfaction.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. In some cases, the company's inside wiring associated with a single line individual service may be extended by means of a certified connector assembly (telephone extension cord). The customer should be aware that compliance with the above conditions may not prevent degradation of service in some situations.

Repairs to certified equipment should be made by an authorized Canadian maintenance facility designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

Caution: Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

TRADEMARKS

FORE Systems is a registered trademark, and *ForeView* and *PowerHub* are trademarks of FORE Systems, Inc. All other brands or product names are trademarks or registered trademarks of their respective holders.

Table of Contents

List of Figures

List of Tables

Preface

Chapter Summaries.....	i
Related Publications.....	iii
Technical Support.....	iv
Typographical Styles.....	v
Important Information Indicators.....	vi
Laser Notice.....	vii
Safety Precautions.....	viii
Modifications to Equipment.....	viii
Placement of a FORE Systems Product.....	viii
Power Cord Connection.....	viii
Command Syntax.....	x

CHAPTER 1 Features Overview

1.1	Intelligent Packet Switching.....	1 - 1
1.1.1	Packet Engine.....	1 - 1
1.1.1.1	Packet Engine 1 (PE1).....	1 - 2
1.1.1.2	Packet Engine 2 (PE2).....	1 - 2
1.1.2	Network Interface Modules (NIMs).....	1 - 2
1.1.2.1	Intelligent Modules.....	1 - 2
1.1.2.2	ATM Modules.....	1 - 3
1.1.2.3	FDDI Modules.....	1 - 3
1.1.2.4	Ethernet Modules.....	1 - 3
1.1.2.5	Fast Ethernet (FE) Modules.....	1 - 4
1.2	Software Features.....	1 - 5
1.2.1	Multiprocessor Optimization.....	1 - 5
1.2.2	Boot Sources.....	1 - 6
1.2.3	Command-Line Interface.....	1 - 6
1.2.4	File-Management System.....	1 - 6
1.2.5	Concurrent Command Line Sessions.....	1 - 6
1.2.6	Configuration Files.....	1 - 6
1.2.7	Parameter Files.....	1 - 7
1.2.8	Automatic Segment-State Detection.....	1 - 7

Table of Contents

1.2.9	Segment Statistics	1 - 8
1.2.10	Traffic Monitoring	1 - 8
1.2.11	Virtual Local Area Networks (VLANs)	1 - 8
1.2.12	Bridging and Routing	1 - 8
1.2.12.1	Bridge Table and Cache	1 - 8
1.2.12.2	802.1d	1 - 9
1.2.12.3	Spanning-Tree	1 - 9
1.2.12.4	IPX Translation Bridging	1 - 9
1.2.12.5	IP Routing	1 - 9
1.2.12.5.1	Routing Information Protocol (RIP)	1 - 10
1.2.12.5.2	Open Shortest Path First (OSPF)	1 - 10
1.2.12.6	AppleTalk Routing	1 - 10
1.2.12.7	IPX Routing	1 - 10
1.2.12.8	DECnet Routing	1 - 10
1.2.13	Route Protocol Statistics	1 - 11
1.2.14	Security Filters	1 - 11
1.3	Network Management Features	1 - 12
1.3.1	Network Management System (NMS)	1 - 12
1.3.2	Management Information Base (MIB) Agents	1 - 12
1.3.3	ForeView	1 - 12
 CHAPTER 2 Software Subsystems		
2.1	Firmware	2 - 1
2.2	Runtime Software	2 - 4
 CHAPTER 3 PowerHub Files		
3.1	File Types	3 - 1
3.2	System Software	3 - 3
3.2.1	PowerHub 7000	3 - 3
3.2.2	PowerHub 8000	3 - 3
3.2.3	Other Files	3 - 4
3.2.4	Created Files	3 - 4
 CHAPTER 4 Command-Line Interface		
4.1	Using the User Interface (UI)	4 - 1
4.1.1	Runtime Prompt	4 - 1
4.1.2	Entering and Editing Command Lines	4 - 3
4.2	Command Syntax	4 - 4
4.2.1	Verb Objects	4 - 4
4.2.1.1	set and unset	4 - 4
4.2.1.2	define and undefine	4 - 4
4.2.1.3	attach and detach	4 - 4

4.2.1.4	add and delete.	4 - 5
4.2.1.5	enable and disable	4 - 5
4.2.1.6	show and clear	4 - 5
4.2.2	Noun Objects	4 - 5
4.2.2.1	config.	4 - 5
4.2.2.2	status.	4 - 5
4.2.2.3	stats.	4 - 6
4.2.2.4	interface.	4 - 6
4.2.2.5	route	4 - 6
4.2.2.6	cache.	4 - 6
4.2.3	Parameters	4 - 6
4.2.3.1	Keyword Parameters	4 - 6
4.2.3.2	Positional Parameters	4 - 6
4.3	On-line Help	4 - 7
4.3.1	Syntax Help	4 - 8
4.3.2	Help Set	4 - 9
4.3.3	Help Show	4 - 10

CHAPTER 5 Global Commands

5.1	Accessing Global Commands	5 - 1
5.1.1	Alias	5 - 1
5.1.2	Checksum.	5 - 2
5.1.3	Copy	5 - 3
5.1.4	Default-Device	5 - 4
5.1.5	Directory	5 - 4
5.1.6	Format	5 - 5
5.1.7	Help	5 - 6
5.1.8	History	5 - 7
5.1.9	History Characters	5 - 7
5.1.10	Logout	5 - 8
5.1.11	Port Number Mode	5 - 9
5.1.12	Return Code Prompt	5 - 9
5.1.13	Read Environment	5 - 10
5.1.14	Rename	5 - 11
5.1.15	Remove	5 - 11
5.1.16	Save Environment	5 - 12
5.1.17	Show Configuration Example	5 - 12
5.1.18	Set TTY	5 - 13
5.1.19	Set User	5 - 14
5.1.20	Subsystems	5 - 14
5.1.21	Timed Command	5 - 15
5.1.22	Type	5 - 16
5.1.23	Unalias	5 - 16

CHAPTER 6 System Commands

6.1	Accessing the System Subsystem	6 - 2
6.1.1	Baud	6 - 2
6.1.2	Bootinfo	6 - 3
6.1.3	Card Swap	6 - 3
6.1.4	Config	6 - 5
6.1.5	Convert Config.	6 - 6
6.1.6	Date	6 - 6
6.1.7	Data Carrier Detect	6 - 7
6.1.8	Ethernet Address.	6 - 8
6.1.9	ID Prom	6 - 8
6.1.10	Memory	6 - 9
6.1.11	Password	6 - 10
6.1.12	Read Configuration	6 - 11
6.1.13	Reboot	6 - 11
6.1.14	Save Configuration	6 - 12
6.1.15	System Location	6 - 12
6.1.16	System Name	6 - 13
6.1.17	Temperature	6 - 13
6.1.18	TTY2	6 - 14
6.1.19	Uptime	6 - 15
6.1.20	Version	6 - 15

CHAPTER 7 Media Commands

7.1	Displaying Bridge-Related Configuration	7 - 2
7.2	Inter-Segment Statistics	7 - 4
7.3	Ethernet LED Modes	7 - 5
7.4	Port Monitoring	7 - 7
7.4.1	How Port Monitoring Works	7 - 7
7.4.2	Performance Considerations and Operation Notes	7 - 8
7.4.3	Packet Modifications	7 - 9
7.5	Monitoring a Segment	7 - 14
7.6	Operating-Mode	7 - 16
7.6.1	Full-Duplex and Half-Duplex Modes	7 - 16
7.6.2	Auto-negotiation	7 - 17
7.6.3	10/100 FEMA Values	7 - 17
7.6.4	Setting the Operating Mode for Ethernet and 10/100 FEMA	7 - 18
7.6.5	Displaying the Operating Mode Configuration	7 - 19
7.6.5.1	Troubleshooting	7 - 19
7.7	UTP Port Receiver Status	7 - 20
7.8	Displaying Port-Level Statistics	7 - 21

7.9	Configuring Packet Forwarding on Segments	7 - 22
7.10	Segment Names	7 - 23
7.11	Segment-State Detection	7 - 24
7.11.1	Automatic Segment-State Detection	7 - 24
7.11.1.1	Software Behavior When Disabled	7 - 25
7.11.1.2	Default Setting	7 - 25
7.11.1.3	Disabled on AUI	7 - 26
7.11.1.4	Segment-State Detection on 10Base-T	7 - 26
7.11.1.5	Explicitly Disabling Unused Segments	7 - 27
7.12	Segment-State Detection Threshold	7 - 28
7.13	Status	7 - 30
7.14	Statistics	7 - 31

CHAPTER 8 NVRAM Commands

8.1	NVRAM Configuration Commands	8 - 1
8.1.1	Boot Order	8 - 1
8.1.2	My Internet Protocol Address	8 - 2
8.1.3	My Subnet Mask	8 - 3
8.1.4	File Server IP Address	8 - 3
8.1.5	Gateway IP Address	8 - 4
8.1.6	Crash Reboot	8 - 5
8.1.7	Slot Segments	8 - 5
8.2	RIPv2 Authentication	8 - 8

CHAPTER 9 Host Commands

9.1	Accessing the Host Subsystem	9 - 2
9.2	Displaying the Configuration	9 - 3
9.3	Keep Alive Delay	9 - 5
9.4	Keep Alive Interval	9 - 6
9.5	Ending (Killing) a TCP Connection	9 - 7
9.6	Statistics	9 - 8
9.7	Status	9 - 10

CHAPTER 10 Bridge Commands

10.1	Accessing the Bridge Subsystem	10 - 2
10.2	Aging	10 - 3
10.3	Bridging	10 - 4
10.4	Bridge Table	10 - 6
10.5	Cache	10 - 9
10.6	Configuration	10 - 10

Table of Contents

10.7	Allocate Memory	10 - 12
10.8	Bridge Groups	10 - 13
10.9	IPX Bridge Translation	10 - 15
10.9.1	Encapsulation Types	10 - 16
10.9.2	Configuration Requirements	10 - 17
10.10	Learning	10 - 18
10.11	Relearn Log	10 - 19
10.12	Spanning Tree	10 - 20
10.12.1	Fast-Hello Time	10 - 22
10.12.2	High- and Low-Utilization Percentage	10 - 22
10.13	Statistics	10 - 23
10.14	Status	10 - 24
CHAPTER 11 Fiber Distributed Data Interface (FDDI)		
11.1	Accessing the FDDI subsystem	11 - 1
11.2	Concentrator	11 - 2
11.3	DAC	11 - 3
11.4	NVRAM	11 - 3
11.5	Target Token Rotation Time (TREQ)	11 - 5
11.6	Time Transmission Variable (TVX)	11 - 6
11.7	Reset Count	11 - 7
11.8	FDDI MIB Variables	11 - 8
11.9	Statistics	11 - 10
CHAPTER 12 SNMP Commands		
12.1	Accessing the SNMP Subsystem	12 - 2
12.2	SNMP Community	12 - 3
12.3	Standard Traps	12 - 4
12.4	Enterprise-Specific Traps	12 - 6
12.4.1	SNMP Configuration	12 - 7
12.5	Displaying Statistics	12 - 9
12.6	Adding an SNMP Manager	12 - 11
12.7	Using SunNet Manager	12 - 13
CHAPTER 13 TFTP Commands		
13.1	Accessing the TFTP Subsystem	13 - 2
13.2	Considerations	13 - 3
13.2.1	TFTP Commands and UNIX Read/Write Permissions	13 - 3
13.2.2	Path Names	13 - 4

13.2.3	File-Naming Conventions	13 - 4
13.2.4	Remote File Names	13 - 5
13.3	TFTP Commands	13 - 6
13.3.1	Setting the Default Server	13 - 6
13.4	Downloading a File	13 - 7
13.5	Uploading a File	13 - 9
13.6	Read Configuration	13 - 12
13.7	Save Configuration	13 - 13

CHAPTER 14 Telnet Commands

14.1	Accessing the Telnet Subsystem	14 - 1
14.2	Opening a Telnet Session	14 - 2
14.3	Closing a Telnet Session	14 - 3
14.4	Viewing Telnet Status	14 - 4

APPENDIX A Configuration Defaults

APPENDIX B Netboot Options

B.1	Choosing a Netbooting Method	B - 1
B.2	The Boot Process	B - 2
B.2.1	Point-to-Point	B - 2
B.2.2	Cross Gateway--Boot Helper Service Used	B - 4
B.2.3	Cross-Gateway--No Boot Helper Service Used	B - 5
B.3	Configuration Options	B - 6
B.3.1	TFTP Server	B - 6
B.3.2	BOOTP Server	B - 6
B.3.3	Intervening Gateway	B - 7
B.3.4	Client PowerHub	B - 8
B.3.5	Using the Same Boot Definition File with Multiple Switches	B - 9
B.3.6	Sharing Methods	B - 10
B.3.6.1	MAC-Address Method	B - 11

Index

List of Figures

Figure 2.1 PowerHub 7000 Boot Screen Display..... 2 - 1
Figure 2.2 PowerHub 8000 Boot Screen Display 2 - 2
Figure 4.1 Command Line 4 - 1

List of Tables

Table 2.1	Subsystems	2 - 4
Table 7.1	Packet Modifications On Monitoring Segment	7 - 10
Table 7.2	Segment-State Detection Methods.	7 - 25
Table 7.3	Automatic Segment-State Detection Default Settings	7 - 25
Table 7.4	Segment Level Statistic Parameters.	7 - 31
Table 10.1	Configuration Arguments	10 - 10
Table 10.2	IPX Translation Bridging Encapsulations	10 - 16
Table 12.1	Standard Traps	12 - 4
Table 12.2	Enterprise-Specific Traps	12 - 6
Table 12.3	SunNet Manager Utilities	12 - 13
Table A.1	Boot PROM Commands	A - 1
Table A.2	Global Subsystem Commands	A - 1
Table A.3	ATALK Subsystem Commands	A - 2
Table A.4	Bridge Subsystem Commands	A - 2
Table A.5	DECnet Subsystem Commands	A - 3
Table A.6	Host Subsystem Commands	A - 4
Table A.7	IP Subsystem Commands	A - 5
Table A.8	IP Multicast Subsystem Commands	A - 6
Table A.9	IP/OSPF Subsystem Commands	A - 7
Table A.10	IP/RIP Subsystem Commands	A - 7
Table A.11	IPX Subsystem Commands	A - 8
Table A.12	TFTP Subsystem Commands	A - 8
Table B.1	Point-to-Point Netbooting	B - 3
Table B.2	Helper-Assisted Netbooting	B - 4
Table B.3	Cross-Gateway Netbooting — No Boot Helper Service	B - 5
Table B.4	Boot Definition Macro Commands	B - 9

List of Tables

Preface

This manual describes the *PowerHub 7000/8000* user interface and commands used to configure and manage the *PowerHub 7000/8000*. Refer to the *PowerHub 7000/8000 Filters Reference Manual* for details on creating and applying filters to control traffic received and transmitted by the PowerHub. Refer to the *PowerHub 7000/8000 Protocols Reference Manual* for details on configuring the communications protocols supported by the PowerHub.

Chapter Summaries

Chapter 1 - Features Overview - Describes the software features of the *PowerHub 7000* and *PowerHub 8000*.

Chapter 2 - Software Subsystems - Describes the Packet Engine boot PROM commands and the software subsystems available in the *PowerHub 7000/8000*.

Chapter 3 - PowerHub Files - Provides information on the files that are shipped installed on the *PowerHub 7000/8000* and the files created within the *PowerHub 7000/8000*.

Chapter 4 - Command-Line Interface - Describes how to interpret user interface screens. The command syntax is discussed with details on the more common noun/verb command combinations. A discussion is provided on the various ways to obtain on-line help while in a user session.

Chapter 5 - Global Commands - Describes the commands that are available in the global command subsystem of the *PowerHub 7000* and *PowerHub 8000*. Global commands are commands that are available throughout the *PowerHub 7000/8000*.

Chapter 6 - System Commands - Describes the commands available from the system subsystem. The system subsystem commands are commands that are used to control overall system parameters and environment.

Chapter 7 - Media Commands - Describes commands available in the media subsystem. Media subsystem commands are used to control physical media and bridging configuration.

Chapter 8 - NVRAM Commands - Describes the commands available in the nvram subsystem. The nvram subsystem commands can be used to make changes to the order in which the system boots as well as to configure the segments that a module slot can support.

Preface

Chapter 9 - Host Commands - Describes the commands available in the host subsystem. The host subsystem commands are used to set or display various TCP, TELNET and UDP information.

Chapter 10 - Bridge Commands - Describes the commands available in the bridge subsystem. The bridge subsystem commands are used to display and control bridging parameters in the *PowerHub 7000/8000*.

Chapter 11 - Fiber Distributed Data Interface (FDDI) - Describes the commands available in the fddi subsystem. The fddi subsystem describes the command necessary to configure the *PowerHub 7000/8000* to support various fddi modules and interfaces. Refer to the *PowerHub 7000/8000 Installation and Maintenance Manual* for a discussion on the physical makeup of fddi subsystems.

Chapter 12 - SNMP Commands - Describes the commands available in the snmp subsystem. The snmp commands are used to configure the *PowerHub 7000/8000* to respond to commands from a system-level management system, i.e., *ForeView* or HP *OpenView*.

Chapter 13 - TFTP Commands - Describes the commands available in the tftp subsystem. The tftp subsystem commands provide an way to transfer files to/from the *PowerHub 7000/8000*. Additionally, tftp commands can be used to configure the system to be booted remotely from a tftp boot server.

Chapter 14 - Telnet Commands - Describes the command available in the telnet subsystem. The telnet subsystem provides commands to initiate an outbound telnet session from the *PowerHub 7000/8000*.

Appendix A - Configuration Defaults - Provides tables outlining the configuration defaults applied to commands in the *PowerHub 7000/8000*.

Appendix B - Netboot Options - Provides instruction on configuring netboot options for the *PowerHub 7000/8000*.

Related Publications

The following publications are referred to throughout this manual and comprise the PowerHubASN-9000 Reference manual set.

- *PowerHub 7000/8000 Release Notes*, MANU0254-05, June 1, 1998.
- *PowerHub 7000/8000 Installation and Maintenance Manual*, MANU0166-02, June 1, 1998.
- *PowerHub 7000/8000 Filters Reference Manual*, MANU0168-02, June 1, 1998.
- *PowerHub 7000/8000 Protocols Reference Manual*, MANU0271-02, June 1, 1998.

Technical Support

In the U.S.A., customers can reach FORE Systems' Technical Assistance Center (TAC) using any one of the following methods:

1. Select the "Support" link from FORE's World Wide Web page:

<http://www.fore.com/>

2. Send questions, via e-mail, to:

support@fore.com

3. Telephone questions to "support" at:

800-671-FORE (3673) or 724-742-6999

4. FAX questions to "support" at:

724-742-7900

Technical support for customers outside the United States should be handled through the local distributor or via telephone at the following number:

+1 724-742-6999

No matter which method is used to reach FORE Support, customers should be ready to provide the following:

- A support contract ID number
- The serial number of each product in question

All relevant information describing the problem or question

Typographical Styles

Throughout this manual, all specific commands meant to be entered by the user appear on a separate line in bold typeface. In addition, use of the Enter or Return key is represented as <ENTER>. The following example demonstrates this convention:

```
cd /usr <ENTER>
```

File names that appear within the text of this manual are represented in the following style: "...the `fore_install` program installs this distribution."

Command names that appear within the text of this manual are represented in the following style: "...using the `flush-cache` command clears the bridge cache."

Subsystem names that appear within the text of this manual are represented in the following style: "...to access the `bridge` subsystem..."

Parameter names that appear within the text of this manual are represented in the following style: "...using `<seg-list>` allows the segments to be specified for which to display the specified bridge statistics."

Any messages that appear on the screen during software installation and network interface administration are shown in `Courier` font to distinguish them from the rest of the text as follows:

```
.... Are all four conditions true?
```

Important Information Indicators

To call attention to safety and otherwise important information that must be reviewed to ensure correct and complete installation, as well as to avoid damage to the FORE Systems product or to the system, FORE Systems utilizes the following *WARNING/CAUTION/NOTE* indicators.

WARNING statements contain information that is critical to the safety of the operator and/or the system. Do not proceed beyond a **WARNING** statement until the indicated conditions are fully understood or met. This information could prevent serious injury to the operator, damage to the FORE Systems product, the system, or currently loaded software, and is indicated as follows:

WARNING!



Hazardous voltages are present. To reduce the risk of electrical shock and danger to personal health, follow the instructions carefully.

CAUTION statements contain information that is important for proper installation/operation. Compliance with **CAUTION** statements can prevent possible equipment damage and/or loss of data and are indicated as follows:

CAUTION



Damaging to the equipment and/or software is possible if these instructions are not followed.

NOTE statements contain information that has been found important enough to be called to the special attention of the operator and is set off from the text as follows:



To change the value of the LECS control parameters while the LECS process is running, the new values do not take effect until the LECS process is stopped, and then restarted.

Laser Notice

Class 1 Laser Product:
This product conforms to
applicable requirements of
21 CFR 1040 at the date of
manufacture.

Class 1 lasers are defined as products which do not permit human access to laser radiation in excess of the accessible limits of Class 1 for applicable wavelengths and durations. These lasers are safe under reasonably foreseeable conditions of operation.



The Laser Notice section applies only to products or components containing Class 1 lasers.

Safety Precautions

For personnel protection, observe the following safety precautions when setting up equipment:

- Follow all warnings and instructions marked on the equipment.
- Ensure that the voltage and frequency of the power source matches the voltage and frequency inscribed on the equipment's electrical rating label.
- Never push objects of any kind through openings in the equipment. Dangerous voltages may be present. Conductive foreign objects could produce a short circuit that could cause fire, electric shock, or damage to the equipment.

Modifications to Equipment

Do not make mechanical or electrical modifications to the equipment. FORE Systems, Inc., is not responsible for regulatory compliance of a modified FORE product.

Placement of a FORE Systems Product

CAUTION



To ensure reliable operation of the FORE Systems product and to protect it from overheating, openings in the equipment must not be blocked or covered. A FORE Systems product should never be placed near a radiator or heat register.

Power Cord Connection

WARNING!



FORE Systems products are designed to work with single-phase power systems having a grounded neutral conductor. To reduce the risk of electrical shock, do not plug FORE Systems products into any other type of power system. Contact the facilities manager or a qualified electrician if unsure of what type of power is supplied to the building.

WARNING!



FORE Systems products are shipped with a grounding type (3-wire) power cord. To reduce the risk of electric shock, always plug the cord into a grounded power outlet.

Command Syntax

The following expressions are used in this manual when describing command syntax:

AaBbCcDd A term that is being defined. Example:
IP Helper is an enhancement to the **ip** subsystem that allows a system to be boot from a server separated from the boot client by a gateway.

AaBbCcDd A command name. Commands are case-sensitive; they should always be issued in lowercase. Example:

dir

| 1) Separates the full and terse forms of a command or argument:

- The full form is shown on the left of the |.
- The terse form is shown on the right of the |.

Example:

dir | ls

When the command or argument is entered, either the full form or terse form may be used. In this example, either **dir** or **ls** can be used.

2) Separates mutually exclusive command arguments. Example:

active-ama|aa cset p[rimary]|b[ackup] <slot>|all

In this example, the command **active-ama|aa** can accept either **active-ama** or **aa**, but not both.

[] Enclose optional command arguments or options. Example:

active-ama|aa [show] [linemode|lm] <slot>|all

In this example, the [] enclose an optional argument. The command can be issued without the argument(s) shown in []. However, if specified, the argument must be one of the two options listed between the [].

<AaBbCcDd> Indicates a parameter for which a value is supplied by the operator. When used in command syntax, *<italics>* indicates the value to be supplied. Example:

savecfg *<filename>*

In this example, *<filename>* is a parameter for which a value must be supplied when the command is issued.

AaBbCcDd Indicates a field name or a file name.

An example of a field name is when booting the software, the `login:` prompt is displayed.

A filename example is when booting the software, the system looks for a file name `cfg`.

Indicates text (commands) displayed by the software or typed at the command prompt. To distinguish output generated from a command, the typed input is shown in bold typeface. Example:

```
16:PowerHub:system# bootinfo
Tue Jan 20 15:46:25 1998 start
Tue Jan 20 15:46:34 1998 nvram boot order: fm
boot device: m
17:PowerHub:system#
```

In this example, the user enters **bootinfo** and the software responds with:

```
Tue Jan 20 15:46:25 1998 start
Tue Jan 20 15:46:34 1998 nvram boot order: fm
boot device: m
17:PowerHub:system#
```

Preface

CHAPTER 1

Features Overview

This chapter provides an overview of the major features of the *PowerHub* 7000/8000. The features discussed include:

- Intelligent Packet Switching
- Software
- Network Management

1.1 Intelligent Packet Switching

Much of the packet switching in the PowerHub is performed by the Packet Engine (PE). The PE is the centralized packet processing and forwarding engine of the PowerHub. When a packet is received on a segment, the packet is forwarded to the PE and placed in Shared Memory where it is examined and either dropped or forwarded, as applicable. The PowerHub 7000 utilizes the first generation Packet Engine, Packet Engine 1 (PE1), while the PowerHub 8000 utilizes the second generation Packet Engine, Packet Engine 2 (PE2).

1.1.1 Packet Engine

The processors on board the PE contain the bridging and routing engines that intelligently examine packet headers for bridging and routing, and modifying them as required for routing. When a non-intelligent Network Interface Module (NIM) receives a packet from one of its ports, it places the packet on the Packet Channel and transfers it directly to the shared packet-buffer memory on the PE.

The Main CPU (MCP) in the PE examines the source and destination addresses in the packet to determine the segments to which the packet needs to be forwarded and the modifications, if any, to be made to the packet. After the necessary modifications are performed, the Input Output/Processor (IOP) queues the packet for transmission on the appropriate destination port(s).

The PE is also responsible for maintaining complete routing and bridging tables. Caches of route and bridge tables are distributed to intelligent NIMs, which make forwarding decisions locally and use the IOPs to queue the packets to the appropriate NIM.

1.1.1.1 Packet Engine 1 (PE1)

PE1 contains the following major features:

- Supports all currently supported NIMs.
- Contains three 40MHz RISC (64bit internal-32bit external) processors, each with specialized functions: one MCPU and two IOPs. Installing a packet Accelerator adds another MCPU, increasing the number of processors to four, similar to the PE2 (refer to Section 1.2.1).
- Supports the two 800Mbps packet channels of the Packet-Channel Backplane found in the PowerHub 7000 for a peak bandwidth of 1.6 Gbps. These high-speed channels are implemented and controlled through the incorporation of ten proprietary ASIC devices.

1.1.1.2 Packet Engine 2 (PE2)

PE2 contains the following major features:

- PE2 is backwards compatible with PE1.
- Supports all currently supported NIMs.
- Contains four 100/150MHz RISC (64bit internal-32bit external) processors, each with specialized functions: two MCPUs and two IOPs.
- Supports the four 800Mbps packet channels of the Packet-Channel Backplane found in the PowerHub 8000 for a peak bandwidth of 3.2 Gbps. These high-speed channels are implemented and controlled through the incorporation of ten proprietary ASIC devices.

1.1.2 Network Interface Modules (NIMs)

The PowerHub supports various interfaces through the use of Network Interface Modules (NIMs). Some NIMs are termed as Intelligent NIMs (INIMs). The following paragraphs provide an overview of the supported NIMs, grouped by interface type. For detailed descriptions of supported NIMs, refer to the *PowerHub 7000/8000 Hardware Reference Manual*.

1.1.2.1 Intelligent Modules

The following NIMs are termed Intelligent NIMs (INIMs). INIMs have the ability to make packet handling and forwarding decisions. These INIMs contain processor and intelligence (firmware) that can relieve the respective PE some of the workload of handling packets. These INIMs forward packets directly to ports that are physically located on the same INIM.

- PowerCell 700 ATM module
- Single, Dual, Universal Single and Universal Dual FDDI modules
- 6x1 Fast Ethernet (6x1FE) module

- 2x8 Fast Ethernet (2x8FE) module

1.1.2.2 ATM Modules

The PowerCell 700 ATM INIM supports up to two ATM Media Adapters (AMAs). These AMAs can support various physical (PHY) ATM interfaces. The interfaces available include OC-3 Single-Mode Fiber (SMF), OC-3 Multimode Fiber (MMF) and OC-3 Unshielded Twisted-Pair (UTP). If two AMAs are installed, one can be configured as a primary port while the other can be configured as a backup port.

1.1.2.3 FDDI Modules

The FDDI modules are available in both Single and Dual configurations. Each configuration is available with multi-mode MIC, single-mode ST and UTP connectors. Universal Single and Dual modules with the same adapters types are also available. Additionally, there are 1x6 and 1x16 FDDI Concentrator modules. These are available with multi-mode mini MICs or UTP connectors.

1.1.2.4 Ethernet Modules

Ethernet modules are available in the following varieties:

6x1 Universal Ethernet Module (UEM)	Provides six slots for installation of Ethernet Media Adapters (EMAs). Any combination of the following EMA types can be installed on the UEM: AUI (10Base-5), 10Base-FL (FOIRL-compatible), 1-Base-FB, BNC (10Base-2), MAU (Media Access Unit), 10Base-T (UTP).
10x1 10Base-FL	Provides 10 independent 10Base-FL segments; connection for each segment is provided by multimode ST connectors.
13x1	Provides twelve 10Base-T connectors and one slot for installation of a Fast Ethernet Media Adapter (FEMA); the FEMA types are the same as those for the 6x1FE module.
16x1	Provides 16 independent 10Base-T segments. Connections for each segment is provided by an RJ-45 connector.
4x4 Microsegment Repeater	Provides four independent 10Base-T segments. Each segment is further divided into four ports and connection to each port is provided by an RJ-45 connector.

- | | |
|----------------------------------|--|
| 4x6 Microsegment Repeater | Provides four independent 10Base-T segments. Each segment is further divided into six ports and connection to each port is provided in a 50-pin Champ-style connector. |
|----------------------------------|--|

1.1.2.5 Fast Ethernet (FE) Modules

Fast Ethernet (FE) modules are available in the following varieties:

- | | |
|--------------|---|
| 6x1FE | Provides six FE interfaces in the form of individually installed Fast Ethernet Media Adapters (FEMAs). The FEMAs are available with 100Base-TX, 100Base-FX, or 100Base-T4 connectors. |
| 13x1 | Provides twelve 10Base-T connectors and one slot for installation of a FEMA. The FEMA types are the same as those for the 6x1FE. |
| 4x8 | Provides |

1.2 Software Features

The following software features are supported in the *PowerHub 7000* and *8000*. This section describes the features that can be found in the PowerHub software. The focus of this section is on system management, rather than configuration and management of network interfaces. The following subjects are discussed:

- Multiprocessor Optimization
- Boot Sources
- Command-Line Interface
- File Management System
- Concurrent Command-Line Sessions
- Configuration Files
- Parameter Files
- Automatic Segment-State Detection
- Segment Statistics
- Traffic Monitoring
- Virtual Local Area networks (VLANs)
- Bridging and Routing
- Route Protocol Statistics
- Security Filters

1.2.1 Multiprocessor Optimization

Multiprocessor optimization minimizes the latency caused in the normal packet-forwarding functions due to the processing of management events. By moving these processing-intensive functions to a separate MCPU, the latency of packets in the fast path can be kept to a minimum.

This feature is dependent on having a PE1 with a Packet Accelerator installed. With the accelerator installed, there are four CPUs available. Without the multiprocessor optimization feature, only three CPUs are used. This feature makes use of the fourth CPU by splitting the functions of the single MCPU.

Multiprocessor Optimization moves all of the fast-path packet processing to one MCPU and retains the slow path and management functions on the other MCPU. Multiprocessor optimization automatically detects the presence of an Accelerator Card at boot time and operates in the appropriate mode. Without the Accelerator Card, the system uses only one MCPU for all functions.

1.2.2 Boot Sources

The PowerHub can be configured to boot from one or a combination of up to three sources: floppy diskette (fd) (PowerHub 7000), Flash Memory Module (fm) (PowerHub 7000)/Compact Flash Card (fc) (PowerHub 8000), or a TFTP/BOOTP file server. The PowerHub 8000 has only two boot sources: Compact Flash Card or a TFTP/BOOTP file server. Failure of the primary boot source can be prevented by configuring a boot order in Non-Volatile Random Access Memory (NVRAM).

1.2.3 Command-Line Interface

The PowerHub is managed through a DOS/UNIX-like command-line user interface. Commands can be issued from a management terminal attached to directly through a TTY connection on the PE or indirectly through an in-band TELNET connection. Refer to *Chapter 2, Software Subsystems* for a discussion of the software subsystems. Refer to the appropriate section of this manual for discussions of the commands available in each subsystem. Refer to the *PowerHub 7000/8000 Protocols Software Reference Manual* for discussion of the protocol-related subsystems commands.

1.2.4 File-Management System

The PowerHub contains global commands to display, copy, rename, and remove files stored on a floppy diskette, in the Flash Memory Module of the PowerHub 7000, or in the Compact Flash Card of the PowerHub 8000. The file management global commands provide the ability to calculate checksum values of files (`checksum`) and display directory and volume information (`dir | ls`). Text files on a PowerHub can also be displayed to the operator console using the `type | cat` command. Additionally, the Flash Memory Module in a PowerHub 7000 can be reformatted if necessary.

1.2.5 Concurrent Command Line Sessions

Up to four management sessions can be open at the same time. The primary session is always the session on TTY1, a second TTY session can be opened on TTY2. In addition, up to two TELNET sessions can be open simultaneously.

1.2.6 Configuration Files

Configuration changes effected through software commands can be preserved by saving the changes in a configuration file. Changes saved to the file name `cfg` are automatically applied and, following a software reboot, provided the `cfg` file is present on the boot source applied to the new session.

1.2.7 Parameter Files

Commands can be issued to modify parameters that control user sessions. These parameters include scroll control, TELNET control characters, command aliases, and timed commands. If session parameters are not saved in environment files, these parameters will be lost when the session is closed.

Environment files can be saved so that the same conditions can be made available in another user session. The environment file can then be read (loaded), reinstating the session parameter changes that were stored in the environment file.

If an environment file is saved under the name `root.env`, it is automatically loaded whenever the system is logged into under `root` status. Likewise, environment files saved under the name `monitor.env` are automatically loaded when logging on with `monitor` status or if the user level is changed from `root` to `monitor` during a session.

1.2.8 Automatic Segment-State Detection

When enabled, Automatic Segment-State Detection senses when a link (or something configured on the link) is “bad” or “down.” When a “bad” or “down” link is detected on a particular port, the state of the segment is reflected in the software’s interface tables. *ForeView* Network Management software allows link types to be enabled or disabled on a particular port. Through *ForeView* the state of the following link types can be learned:

- AUI
- MAU RPTR
- MAU
- BNC
- BNCT
- 10Base-T
- Fiber
- Unknown



To disable automatic segment state detection on a UTP port, rename the configuration file to something other than `cfg` and then reboot the system.

1.2.9 Segment Statistics

Access method and protocol statistics related to segment and packet activity can be displayed. For example, state-change statistics for individual segments can be displayed to show how many times a particular segment has gone up or down since the software was last booted. Statistics related to protocols are briefly described in Section 1.2.13.

1.2.10 Traffic Monitoring

Port activity can be monitored at regular intervals. For example, statistics of packet activity or packet errors and collisions on a particular port can be monitored and graphed.

1.2.11 Virtual Local Area Networks (VLANs)

A Virtual Local Area Network (VLAN) is a collection of segments that share the same group name or interface address. Layer-2 VLANs are created by creating a bridge group. The software comes with a default bridge group called `default` that contains all installed PowerHub segments.

Layer-3 VLANs can be created by assigning the same IP, IPX, or AppleTalk interface address to multiple segments. When the software determines a packet is to be sent to a Layer-3 VLAN assigned to multiple segments, the software forwards a copy of the packet on each segment. From a physical perspective, when this happens, a separate packet is sent to each physical interface. From a logical standpoint, however, the forwarded packet has been forwarded onto its single destination network or subnet, irrespective of how many physical interfaces that network or subnet is configured on.

1.2.12 Bridging and Routing

The `bridge` subsystem contains commands for configuring and managing the PowerHub as an IEEE 802.1d bridge. Up to 32 network (bridge) groups can be defined, each containing any subset of PowerHubsegments.

1.2.12.1 Bridge Table and Cache

The software maintains a bridge table containing the MAC-layer hardware addresses of devices to which the PowerHub is able to bridge packets. The software maintains this table by automatically adding new entries and deleting unused entries. In addition, individual entries can be added or removed, including entries that support multi-homed hosts.

Following is an example of a bridge table. Although only a handful of bridge entries are shown in this example, the bridge table usually contains many entries.

```
98:PowerHub:bridge# bt
```

```

Bridging table (aging time = 60 minutes)
Ethernet-address  Seg  Rule  Flags
00-60-08-b0-97-04  2.1  none
00-00-ef-03-9a-b0  --   none  system permanent
08-00-20-7d-e1-7d  2.1  none
.
.
.
00-a0-24-17-3d-9a  2.1  none
00-a0-98-00-09-d3  2.1  none
00-a0-d1-01-ed-7f  2.1  none
ff-ff-ff-ff-ff-ff  --   none  permanent bmcast

Total entries: 97, Learned entries: 95, Permanent Entries: 2
99:PowerHub:bridge#

```

In addition to the bridge table, the software maintains a bridge cache of the most recently used source-destination pairs. A source-destination pair contains a packet's source and destination MAC-addresses. The bridge cache provides a fast path for the bridging software and gives an at-a-glance view of current bridging activity. The bridge cache can be displayed to see the source-destination pairs that are frequently used.

1.2.12.2 802.1d

The PowerHub can be used “right out of the box” as an 802.1d Bridge. The designation 802.1d refers to the IEEE specification for this type of bridge. For more information regarding 802.1d bridging, refer to Request for Comments (RFCs) 1493 and 1525.

1.2.12.3 Spanning-Tree

The bridge software includes implementation of the 802.1d Spanning-Tree (ST) algorithm. When enabled, the software identifies and “breaks” loops in the network without requiring configuration changes. Commands in the `bridge` subsystem allow fine-tuning of the ST parameters to fit network needs.

1.2.12.4 IPX Translation Bridging

IPX translation bridging allows one or more IPX networks that span FDDI and Ethernet segments using different packet encapsulations to be configured. This type of bridging is different from 802.1d bridging, which bridges packets based on the MAC-layer hardware address of the devices in the network.

1.2.12.5 IP Routing

Commands in the `ip` subsystem allow segments to be configured for IP routing. Using `ip` commands, IP interfaces can be assigned to individual segments. The IP routing software also supports IP VLANs, enabling a single IP subnet that spans multiple segments to be defined.

The following subsections describe major features of the `ip` subsystem. Refer to the *PowerHub 7000/8000 Protocols Reference Manual* for more information about these features and the `ip` commands.

1.2.12.5.1 Routing Information Protocol (RIP)

The `ip/rip` subsystem commands enable the PowerHub to perform IP routing. Using commands in this subsystem, RIP parameters such as `talk` and `listen` can be configured on a segment-by-segment basis. Statistics for RIP packets can also be displayed.

1.2.12.5.2 Open Shortest Path First (OSPF)

The `ip/ospf` subsystem contains commands that can be used to configure the PowerHub as an Open Shortest Path First (OSPF) router. OSPF is a routing protocol that enables each participating router to use a topological map of the network to route packets. OSPF routers exchange route information using link state advertisements (LSAs). An LSA is a packet that reports the link state (up or down) of a router's interfaces that are attached to devices in the OSPF network.

1.2.12.6 AppleTalk Routing

The `atalk` subsystem contains commands that can be used to configure PowerHub segments for AppleTalk Phase-2 routing. AppleTalk zones and interfaces can be defined as well as commands to ping AppleTalk nodes.

1.2.12.7 IPX Routing

The PowerHub can be configured and managed as an IPX router. In addition, the software provides management information on IPX routers and servers through implementation of IPX Routing Information Protocol (RIP) and Service Advertisement Protocol (SAP). RIP or SAP `talk` and `listen` parameters can be enabled selectively on a per-segment basis to control the flow of RIP and SAP updates.

1.2.12.8 DECnet Routing

The `dec` subsystem contains commands for configuring the PowerHub to perform DECnet Phase IV routing. Depending on the configuration of the network, the system can be configured to function as a Level-1 or Level-2 router. DECnet statistics for the system (in its capacity as a DECnet node) and for the individual segments configured as DECnet interfaces can also be displayed.

1.2.13 Route Protocol Statistics

The PowerHub can gather statistics for the following Internet routing protocols:

- AppleTalk
- Bridge
- DECnet
- IP
- IPM
- IPX
- OSPFv2
- RIP
- SNMP
- TCP
- TCP/IP

1.2.14 Security Filters

Filters can be defined and applied to segments or protocol interfaces to control the traffic sent and received on the segments or interfaces. The following types of filters can be defined:

- Bridge filters
- Host (TCP) filters
- IP filters
- IP route filters (RIP and OSPF)
- AppleTalk filters
- IPX RIP and SAP filters

1.3 Network Management Features

The PowerHub has a rich management environment providing comprehensive support for Simple Network Management Protocol (SNMP) as well as local RS-232 and Telnet console support. *ForeView* graphical network management software provides true point-and-click device configuration and runs on a variety of popular management stations.

1.3.1 Network Management System (NMS)

The Network Management System (NMS) manages the PowerHub by sending requests to a software module, or agent, to change the value of one or more variables on the device. For example, an agent reports data such as the number of packets sent, received or dropped on that device. Then, the managed device and the NMS use SNMP as the common protocol language to exchange the information requested by the NMS.

1.3.2 Management Information Base (MIB) Agents

Management Information Base (MIB) agents contain definitions of all resources (represented by managed objects) within the MIB that are managed by a network management system (NMS). The managed object has properties that hold values such as routing table information and error counters.

1.3.3 ForeView

ForeView is a graphical-based management application tool for managing the PowerHub. With a point-and-click interface, *ForeView* provides access to PowerHub functions at both the system and segment level. *ForeView* can control the PowerHub, monitor errors, control bridge and routing configuration parameters, and display, print, and save statistics.

ForeView integrates the PowerHub system, bridge, and router features into a single application with access and control of all information from one location. It also contains fault management features to troubleshoot, analyze, and monitor multiple Ethernet or FDDI segments using a single network analyzer.

Statistics are shown in graphical formats, and the physical attributes, such as model and segment type are displayed on the front panel of a graphical representation of the PowerHub. The graphical representation is displayed when *ForeView* is started. For more information about the *ForeView* Network Management application, refer to the *ForeView Network Management User's Manual*.

CHAPTER 2

Software Subsystems

This chapter describes the PowerHub platform software, which controls the operation of the PowerHub. This software is comprised of firmware, located in the Boot PROM on the Packet Engine, and runtime software, which is loaded into the Flash Memory Module of the *PowerHub 7000* or the Compact Flash Card of the PowerHub 8000.

2.1 Firmware

Firmware commands are available to the user if the normal boot process is aborted at the 5-second delay prompt. A typical boot screen display is shown in Figure 2.1. Figure 2.2 shows a typical PowerHub 8000 boot screen display.

```
Starting Packet Engine ...
Prom version: pelp-3.0.0 (7887) 1998.05.06 13:01
I-cache 16KB OK
Entering cached code
I-cache execution OK
D-cache 4KB OK
SRAM 128KB OK
DRAM 24MB OK
Shared Memory 4MB OK
Entering Monitor

FORE Systems PowerHub 7000
FlashInit: found 4MB Flash Memory Module
Board Type: 7PE , CpuType: MCPUCPU, Instance: 1
Ethernet address: 00-00-ef-03-9a-b0

(normal start)

Hit any key now to abort boot [4]:

<PROM-7PE>
```

Figure 2.1 - PowerHub 7000 Boot Screen Display

```
FORE Systems PowerHub 8000 Packet Engine
Prom version: pe2p-2.0.0 (7846) 1998.05.06 13:02
MCPU 1
MCPU local RAM tests:  basic, byte, burst, address, data
MCPU shared RAM tests: basic, byte, address, data
PACKET DESC RAM tests: basic, byte, address, data
PACKET DATA RAM tests: basic, byte, address, data
MCPU local RAM - 4MB
MCPU shared RAM - 32MB
PACKET DESC RAM - 512KB
PACKET DATA RAM - 8MB
Entering Monitor
LOCK switch: UNLOCKED
ACTIVE/STANDBY switch: ACTIVE
Chassis: 10 slots
W Bus:
Slots that are equipped:
Slots that are equipped and latched:
Z Bus:
Slots that are equipped:
Slots that are equipped and latched:
X Bus:
Slots that are equipped:
Slots that are equipped and latched:
Y Bus:
Slots that are equipped: 2 1
Slots that are equipped and latched: 2 1
Board Type: PE2, CpuType: MCPU, Instance: 1
Breaks enabled
Ethernet address: 00-00-ef-06-7d-c0

(normal start)

Hit any key now to abort boot [1]:

<PROM-8PE>
```

Figure 2.2 - PowerHub 8000 Boot Screen Display

The available Boot PROM commands, that the user would normally access, include:

```
<PROM-7PE> ?
COMMANDS:
boot:          boot|b [-n] [fd|net|fm]
copy file:    copy|cp <src-file> <dest-file>
              -or-
              copy|cp <src-file> [<src-file>...] <device>
ethaddr:     ethaddr|ea
help:        help|? [COMMAND]
expert help:  ??
ls:          ls|dir
more:        more [-[<rows>]] f1 [f2...[fn]]
nvram:       nvram [set|unset|show <variable> [<value>]]
```

```

("nvram set bo" sets disk/net boot order)
remove file:    rm|del [-f] f1 [f2...[fn]]
rename file:    rename|ren <oldfilename> <newfilename>
zmodem receive: zreceive|zr|rz [--27abcehtw] [<filename>]
zmodem send:    zsend|zs|sz [--27abehkLlNnoptwXYy]
<PROM-7PE>

```

Additional commands can be found under expert help, which is accessed by entering two question marks (??). Most of the commands located under expert help should not be used unless directed by FORE Systems TAC. Some of the commands can be used to upgrade the Packet Engine firmware. Any commands pertaining to normal operation of the PowerHub are discussed in detail, in the *PowerHub 7000/8000 Hardware Reference Manual*. Commands used by FORE Systems TAC are not discussed.

2.2 Runtime Software

The commands used to configure or exercise PowerHub features are grouped into subsystems. Each subsystem contains commands pertaining to a particular aspect of PowerHub configuration or management. Issuing the `subsystems|ss` command, displays a list of all available subsystems. Issuing `help` or `global help` from a system prompt, displays a list of commands that can be executed at the current prompt. The following subsystems are supported in the PowerHubUI.

Table 2.1 - Subsystems

Subsystem	Description	Refer To
global	System-wide commands	Chapter 5, "Global Commands"
system	Display and manage hardware configuration items, manage file, save, and load configuration files. Default subsystem when powering on the system.	Chapter 6, "System Commands"
media	Define information about physical links.	Chapter 7, "Media Commands"
nvramp	Non-Volatile Random Access Memory (NVRAM)	Chapter 8, "NVRAM Commands"
host	Define and display TELNET control characters, display active TCP connections and UDP agents	Chapter 9, "Host Commands"
bridge	Bridging, Spanning-Tree, and IPX translation bridging	Chapter 10, "Bridge Commands"
fddi	Fiber Distributed Data Interface (FDDI)	Chapter 11, "Fiber Distributed Data Interface"
snmp	Simple Network Management Protocol (SNMP)	Chapter 12, "SNMP Commands"
tftp	Trivial File Transfer Protocol (TFTP)	Chapter 13, "TFTP Commands"
telnet	Outbound Telnet	Chapter 14, "Telnet Commands"
atalk	AppleTalk	PowerHub 7000/8000 Protocols Reference Manual

Table 2.1 - Subsystems

Subsystem	Description	Refer To
atm	Asynchronous Transfer Mode (atm)	PowerHub 7000/8000 Protocols Reference Manual
dec	DECnet	PowerHub 7000/8000 Protocols Reference Manual
ip	Internet Protocol (IP), IP/RIP, IP/OSPF	PowerHub 7000/8000 Protocols Reference Manual
ipx	IPX, IPX/RIP and IPX/SAP	PowerHub 7000/8000 Protocols Reference Manual

As noted in the 'Refer To' column, the subsystems that deal with setting up interface protocols are explained in the *PowerHub 7000/8000 Protocols Reference Manual*. Additionally, those commands dealing with setting up filters are explained in the *PowerHub 7000/8000 Filters Reference Manual*.

Software Subsystems

CHAPTER 3

PowerHub Files

This chapter describes the software used by the PowerHub and the files that are shipped with the PowerHub. The user is advised to contact FORE Systems TAC if it is necessary to upgrade any of the system software or firmware.

3.1 File Types

The following types of software are utilized:

Packet Engine Boot PROM Contains firmware used by the Packet Engine when it is booted. From this PROM, configuration values, including the boot source, can be changed and stored in NVRAM. Refer to the *PowerHub 7000/8000 Hardware Reference Manual* for details on the Boot PROM commands. The boot PROM prompt is displayed as:

```
<PROM-PE1>, for the PowerHub 7000  
or <PROM-PE2>, for the PowerHub 8000
```

System software Sometimes called “runtime software.” The runtime software is accessed from the runtime command prompt. Refer to the appropriate chapter of this manual for detailed information on the commands available in the subsystems of the system software. Refer to the *PowerHub 7000/8000 Protocols Reference Manual* for details on configuring protocols. Refer to the *PowerHub 7000/8000 Filters Reference Manual* for details on configuring filters. The default runtime prompt is displayed as:

```
1:PowerHub:
```

INIM PROM Intelligent Network Interface Modules (NIMs) (such as FDDI, 6x1FE, and PowerCell ATM modules) contain a PROM whose firmware is used by the module when it is booted. The NIM PROM cannot be interacted with directly.

PowerHub Files

Runtime PROM Contains runtime features used by intelligent NIMs.
The runtime PROM firmware is stored on the NIMs.

3.2 System Software

The current version of PowerHub firmware and software is shipped already installed. All required software and firmware is installed on the Packet Engine and all installed NIMs. The following sections describe the software and firmware that may be installed on the PowerHub. Specific software/firmware actually installed depends on the PowerHub model and installed INIMs.

3.2.1 PowerHub 7000

- PE1 System software image file: this file resides on the boot source and gets loaded when the system is loaded.
- ATM-PE1 Runtime PROM image for PowerCell 700: An instance of the appropriate file resides in a PROM on the intelligent module itself. The software is automatically booted.
- FDDI-PE1 Runtime PROM image for FDDI modules: note that FDDI Concentrator modules do not have runtime PROMs. An instance of the appropriate file resides in a PROM on the intelligent module itself. The software is automatically booted.
- FETH-PE1 Runtime PROM image for 6x1FE module: An instance of the appropriate file resides in a PROM on the intelligent module itself. The software is automatically booted.

3.2.2 PowerHub 8000

- PE2 System software image file: this file resides on the boot source and is loaded when the system is booted.
- ATM-PE2 Runtime PROM image for PowerCell 700: An instance of the appropriate file resides in a PROM on the intelligent module itself. The software is automatically booted.

- FDDI-PE2 Runtime PROM image for FDDI modules: note that FDDI Concentrator modules do not have runtime PROMs. An instance of the appropriate file resides in a PROM on the intelligent module itself. The software is automatically booted.
- FETH-PE2 Runtime PROM image for 6x1FE module: An instance of the appropriate file resides in a PROM on the intelligent module itself. The software is automatically booted.

3.2.3 Other Files

Additional files that may be installed on the PowerHub are files that are used for testing the system.

- bootdef Used by the system when the software is booted to identify the name of the system software image, configuration file, and/or boot source.
- dispcfg Configuration file that runs a series of commands that display system configuration information and statistics. This file is useful to assist FORE Systems TAC in diagnosing configuration problems.

3.2.4 Created Files

In addition to the files that are shipped installed on the system, the following files can be created and saved during a session:

- cfg Configuration file: created when issuing the **system savecfg cfg** or **tftp savecfg cfg** command. The configuration file can be saved under any DOS-compatible filename, but the configuration must be manually loaded unless the user also edits the bootdef file to contain the configuration file name.
- root.env Environment file for root sessions: created when issuing the **saveenv root.env** command. Environment files can be saved under any DOS-compatible filename but must be manually loaded.

<code>monitor.env</code>	Environment file for monitor sessions: created when issuing the <code>saveenv monitor.env</code> command. Environment files can be saved under any other DOS-compatible filename but must be manually loaded.
<code>powerhub.dmp</code>	Dump file: created if a system crash is experienced.
<code>iop1.dmp</code>	Another type of dump file the software can produce when a crash is experienced.
<code>iop2.dmp</code>	Another type of dump file the software can produce when a crash is experienced.



The dump (.dmp) files should be supplied to FORE Systems TAC when reporting system crashes. The contents of these dump files assist TAC determine the cause of the crash.

PowerHub Files

CHAPTER 4

Command-Line Interface

This chapter describes the User Interface (UI) command-line, command syntax, various ways to display on-line help and levels of on-line help that are available.

4.1 Using the User Interface (UI)

The user interface (UI) comes up by default in the **system** subsystem when initially loaded. The following sections describe the subsystem command line prompt and how to issue commands.

4.1.1 Runtime Prompt

Regardless of whether the system is being accessed through one of the TTY (RS-232) ports or through an active TELNET session, the command prompt is displayed as shown in Figure 4.1:

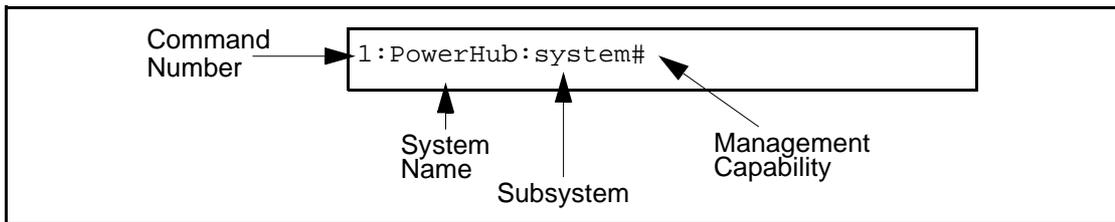


Figure 4.1 - Command Line

As shown in Figure 4.1, the command prompt contains four components:

- Command Number** Sequential line number of commands executed during the active session (similar to a command number in the UNIX C-shell). In this example, the command number is 1.

When a carriage return (Enter key) is issued, the PowerHub attempts to execute the command entered at the command prompt. A message, or data (if requested), is displayed, then a new command prompt is displayed. The number in the command prompt increments by one from the previous command prompt.

System Name Name assigned to this PowerHub. The default name can be changed by using the `system sysname` command (refer to Chapter 6).

Subsystem Name of the current subsystem. Commands issued at the command prompt must either be global commands or commands available within the current subsystem. In this example, the subsystem is `system`, the initial subsystem.

Access Level Indicates the level of access granted for this session.

>Indicates monitor level capability. *Monitor* capability is a display-only capability. Statistics and configuration information can be displayed. Commands that could change the configuration, clear statistics, or modify internal tables are not allowed.

#Indicates root level capability. *Root* capability allows commands to be issued that can change the configuration and clear statistics.

In this example, the current session is at the root access level.



If a session is started and the `login:` prompt is displayed, root or monitor must be entered, followed by a password before being allowed to proceed. Refer to Chapter 6 for information on changing or assigning passwords.

4.1.2 Entering and Editing Command Lines

All commands are entered at the command prompt using a workstation, terminal, or PC as a management station. The workstation, terminal, or PC must be attached to one of the TTY ports or connected through an active TELNET session.

Commands and arguments are case-sensitive and should be entered only as shown in the manual or on-line help. Commands must not exceed 128 characters in length. The keys used to edit and issue commands are the standard keys used on most UNIX workstations:

- To issue a command, enter the command and any options or arguments (if needed or required) at the command prompt, then press the `<Enter>` key.
- To erase individual characters in a command, use the `<Backspace>` or `<Delete>` key, or the `EraseChar` character assigned in the TELNET session (`<Ctrl+H>`).
- To cancel an entire line of input, use the reassign character (`<Ctrl+U>`).
- To control the scrolling of output on the terminal, use `<Ctrl+S>` to stop the flow and `<Ctrl+Q>` to resume the flow.

The global command `stty` provides options and arguments to change the key sequences used during TELNET sessions. The key sequences for the current session or the default key sequences used for all sessions can also be displayed or changed. Refer to Chapter 5, for information on the `stty` command.

4.2 Command Syntax

The command syntax is comprised of verbs, nouns and parameters. Understanding what each syntax component can do is important in understanding how to issue commands.

4.2.1 Verb Objects

The command verbs described in the following paragraphs consist of:

- `set/unset`
- `define/undefine`
- `attach/detach`
- `enable/disable`
- `show/clear`

4.2.1.1 `set` and `unset`

`set` and `unset` apply or remove settings. Some examples include:

- Boot order (the device from which the system attempts to load the runtime software).
- Scroll control (`stty`) parameters
- Timed commands
- Routing protocols
- Specific bridging and routing protocol features

When the `set` or `unset` verb is prepended by `c`, `p`, `n` or `s`, it applies only to specific segments (`c,p`) or networks (`n,s`).

4.2.1.2 `define` and `undefine`

`define` and `undefine` are primarily used when creating or deleting filter templates, rules and/or filters themselves. Refer to the PowerHub 7000/8000 *Filters Reference Manual* for detailed information on the use of `define` and `undefine`.

4.2.1.3 `attach` and `detach`

`attach` and `detach` are primarily used to apply or remove templates, rules and/or filters to segments or interfaces. Refer to the PowerHub 7000/8000 *Filters Reference Manual* for detailed information on the use of `define` and `undefine`.

4.2.1.4 add and delete

add and **delete** are used to add or delete objects to or from tables or to add or delete interfaces. Examples include bridge-table entries, protocol interface-table entries (IP, AppleTalk, IPX, and DECnet), and route-table entries.

4.2.1.5 enable and disable

enable and **disable** turn on or off specific features. Examples include bridging and protocol routing, specific IP routes and IP Helpers. When the **enable** or **disable** verb is prepended by **p**, **n** or **s**, the verb applies only to specific segments (**p,s**) or networks (**n,s**).

4.2.1.6 show and clear

show and **clear** are used to display or clear configuration information, tables, caches, and statistics. With these verbs, configuration information or statistics can be displayed, cleared, or reset to zero (0).

4.2.2 Noun Objects

The command structure contains many nouns. Some of the more commonly used nouns are described in the following paragraphs. This is a non-inclusive list of available nouns. Entering **help|?** at a subsystem prompt displays a complete list of nouns available in the respective subsystem.

4.2.2.1 config

The **config** command displays configuration information relative to the current subsystem on hardware and interfaces. In general, the **config** (**config** [**show**]) command displays parameters that have been configured through the software. Notice the [**show**] verb is enclosed in brackets. Whenever a noun, verb or part of a noun or verb is displayed in brackets, the portion in brackets is not required for the command to be executed. This portion of the command is assumed if omitted from the command line. Entering **config** or **config show** produces the same end result.

4.2.2.2 status

The **status** command displays the current status of the hardware such as segment up/down status, port status, the current bridge status of segments (bridging enabled or disabled, Spanning Tree enabled or disabled). In general, configuration parameters are displayed when the **status** (**status** [**show**]) command is issued.

4.2.2.3 stats

The **stats** command displays the statistics related to the current, or specified, subsystem. When **stats**, (**stats** [**show**]) is issued, statistics related to the area in the current subsystem (or specified subsystem if different from the current) are displayed. For example, the **stats** command issued from within the **ip** subsystem displays the current IP, ARP, and ICMP packet statistics.

4.2.2.4 interface

The **interface** | **it** command configures an interface in the current protocol. Typically there are **add**, **delete**, and **show** verbs accompanying this noun.

4.2.2.5 route

The **route** | **rt** command is used to manually add a route to an IP, IP Multicast, AppleTalk, IPX, or DECnet protocol session. Typically there are **add**, **delete**, **show**, **enable**, and **disable** verbs accompanying this noun.

4.2.2.6 cache

The **cache** command displays or clears cache entries in the current subsystem. The bridge subsystem and all routing protocol subsystems contain a cache command. The cache provides a “fast-path” entry, which is used as a shortcut to bridge or route packets. When a bridge table or route table is in the fast path, the PowerHub does not need to perform all the bridging or routing processing that it normally performs in order to bridge or route a packet. Each cache is maintained by placing in it the most recently used source/destination MAC-address pairs (for bridging) or protocol interface addresses (for routing).

4.2.3 Parameters

Most command nouns and verbs can be modified with parameters to further define what the command is to accomplish. Where the parameter falls in relation to the noun and/or verb depends on the type of parameter. The PowerHub supports keyword and positional parameters.

4.2.3.1 Keyword Parameters

Keyword parameters can be entered at any point following the verb.

4.2.3.2 Positional Parameters

Positional parameters must be entered in a specific position following the use of the verb. The need for positional parameters in the UI is infrequent because the software uses keywords to determine the function being performed. When the need for a positional parameter arises, the software provides a response (**usage:**) statement, showing the proper syntax.

4.3 On-line Help

Entering **help|?** at the command prompt, displays the commands that can be executed at that level. The example below shows a sample display produced by this command:

```
2:PowerHub:system# help
```

```
system subsystem:
```

```

    baud                      readcfg|rdcfg
    bootinfo|bi              reboot
    card-swap|cs            savecfg|svcfg
    config                   syslocn
    convert-config|ccfg     sysname
    date                     temperature|temp
    dcd-detection|dcd       tty2
    ethaddr|ea              uptime
    idprom|idp              version|ver
    passwd
```

```
type 'global help' for global commands
```

```
type 'shex' to show an example of configuration
```

```
3:PowerHub:system#
```

As shown in the display below, entering **global help** displays the global commands, and entering **shex** provides examples on configuring the PowerHub. This information is displayed whenever **help|?** is entered from any command prompt. Entering **global help** and **shex** results in the following displays:

```
236:PowerHub:system# global help
```

```
global subsystem:
```

```

    alias                    rcprompt
    checksum                 readenv|rdenv
    copy|cp                  rename|mv
    default-device|dd       rm
    dir                      saveenv|svenv
    format|fmt              show-config-example|shex
    help|?                  stty
    history|hi              su
    histchars               subsystems|ss
    logout|bye              timedcmd|tc
    ls                      type|cat
    pnm                     unalias
```

Command-Line Interface

```
237:PowerHub:system# shex
```

The following shows a short example to configure ip interface

```
ip vlan add 200.200.200.200 2.1 (add a vlan on segment 2.1)
ip it add 200.200.200.200 200.200.200.200 (add an ip interface)
ip enable (enable ip forwarding)
```

The following shows some commands in subsystem "bridge"

```
bridge br penable 2.1 ("port" enable bridging on seg. 2.1)
bridge br pdisable 2.2 ("port" disable bridging on seg. 2.2)
bridge st enable (enable spanning tree)
bridge st disable (enable spanning tree)
```

In summary, there may be "enable/disable" and their derives such as "penable/pdisable", "senable/sdisable", and etc. to set a particular feature on and off. Use "help [cnps]enable" and "help [cnps]disable" in each subsystem to see what can be set on/off.

```
238:PowerHub:system#
```

4.3.1 Syntax Help

Syntax help is provided when an incomplete or incorrect command is entered. Entering a command that requires additional options or arguments without providing these options or arguments, or entering them incorrectly, will prompt a usage statement indicating the correct parameter syntax. For example, entering `idprom|idp` at the `system` command prompt displays the following usage statement:

```
3:PowerHub:system# idp
usage:
    idprom|idp [show] <slot number>|all
4:PowerHub:system#
```

Detailed help on a command can be provided by entering `help|?` and the specific command. The example below shows a sample display produced by this command:

```
240:PowerHub:system# ? readcfg

readcfg|rdcfg [-v] <file or device name>

    -v : verbose: print each command as it is executed

Reads the configuration from the specified file or device
For auto-configuration on boot up, use the file name 'cfg'
NOTE: The last line in any configuration file must be the string
'endcfg' or 'ecfg'
```

```
241:PowerHub:system#
```

The display produced by this help command shows the syntax required to successfully execute the command. The display also provides a brief description of the options/arguments used with the command and explains what the command is intended to perform. Depending on the command, additional information may also be provided, as shown in the `readcfg` command above.

With some commands, there are numerous command verbs available. Such commands as `interface` may contain the `add`, `delete`, and `show` verbs. Entering `help|? interface` results in a display similar to the following:

```
242:PowerHub:ip# ? interface
```

```
Help available for:
```

```
it|interface add <vlanid> <ipaddr>[/<prefixlen>|<mask>]
      [ ift[type] b[c] | n[bma] | [p[top] <nbr_addr>] ]
it|interface del[ete] [-p] <vlanid> <ipaddr>|all
it|interface [show] [<disprestrictors>]
```

You may obtain more detailed help by giving additional parameters

```
243:PowerHub:ip#
```

The previous display shows all of the syntax available for use with the `ip interface` command. Additional help on each of the various options can be obtained by entering `help|? interface` and the verb as shown in the following example.

```
243:PowerHub:ip# ? interface add
```

```
it|interface add <vlanid> <ipaddr>[/<prefixlen>|<mask>]
      [ ift[type] b[c] | n[bma] | [p[top] <nbr_addr>] ]
```

```
Add an IP interface to the given vlan. If <mask> is not specified
then "natural" subnet mask (class A, B, or C address mask) for the
IP address is used. Interface type can be one of broadcast, nbma
and ptop. Neighbor address must be specified only for ptop type.
If interface type is not specified, broadcast is assumed by default.
```

```
244:PowerHub:ip#
```

4.3.2 Help Set

Entering `help|? set` at any subsystem prompt displays those commands along with global commands available in the present subsystem that use the command verb `set` as part of the command syntax. Use of this help option can save the user time in searching for the particular `set` command to perform a particular function.

Command-Line Interface

```
245:PowerHub:system# ? set
```

```
Help available for:
```

```
pnm set multi|old
pnm [show]
default-device|dd set <device>
baud set tty1|tty2 1200|2300|4800|9600|19200
baud [show]
date set [YYMMDD]hhmm[.ss]
date [show]
syslocn set <location>
syslocn [show] <location>
sysname set <location>
sysname [show] <location>
```

You may obtain more detailed help by giving additional parameters

```
246:PowerHub:system#
```

4.3.3 Help Show

Like the **help|? set** option discussed in the previous paragraph, entering **help|? show** at any subsystem prompt displays those commands, and the global commands, available in the present subsystem that use the command verb **show** as part of the command syntax. Use of this help option can save the user time in searching for the particular **show** command to perform a particular function.

```
246:PowerHub:system# ? show
```

```
Help available for:
```

```
default-device|dd [show]
baud set tty1|tty2 1200|2300|4800|9600|19200
baud [show]
bootinfo|bi [show]
date set [YYMMDD]hhmm[.ss]
date [show]
dcd-detection|dcd enable|disable
dcd-detection|dcd [show]
idprom|idp [show] <slot number>|all
promver|pv [show]
syslocn set <location>
syslocn [show] <location>
sysname set <location>
sysname [show] <location>
temperature|temp [show] <slot number>|all
uptime [show]
version|ver [show] [<slot-number>|all]
config [show]
```

You may obtain more detailed help by giving additional parameters

```
247:PowerHub:system#
```


CHAPTER 5

Global Commands

This chapter discusses the use of global commands that are available within the PowerHub run-time software from any system prompt. Global commands are those commands which can be executed from any subsystem in the PowerHub.

5.1 Accessing Global Commands

Entering `global help` at any system prompt displays the global commands. The following display shows the global commands:

```
4:PowerHub:system# global help
```

```
global subsystem:
```

```
alias
checksum
copy|cp
default-device|dd
dir
format|fmt
help|?
history|hi
histchars
logout|bye
ls
pnm
rcprompt
readenv|rdenv
rename|mv
rm
saveenv|senv
show-config-example|shex
stty
su
subsystems|ss
timedcmd|tc
type|cat
unalias
```

```
5:PowerHub:system#
```

The following paragraphs describe the function performed by each global command and the syntax required.

5.1.1 Alias

The `alias` command is used to create a shortened version of a command. The syntax for this command is as follows:

```
alias [<name> [<command>]]
```

where

[<name> Specifies a name for the specified command. If no command is specified, the alias command displays the command assigned to that alias. If no name is specified, all defined aliases are displayed.

[<command>]] Specifies a command to be executed whenever the specified alias is entered at a command prompt.



Aliases can be removed by using the **unalias** command (Section 5.1.23) when logging out of the current session (**logout|bye**) (Section 5.1.10) or rebooting the system. Aliases can not be saved with the **system savecfg|svcfg** command (refer to Chapter 6) but can be saved using the global **saveenv|svenv** command (see Section 5.1.13).

The following example defines the alias **ia**, which replaces the **interface add** command, whenever entered. The alias shown is used to add an ip interface on vlan test using ip address 144.132.55.65.

```
43:PowerHub:system# alias ia interface add
Added ia: interface add
44:PowerHubASN-9000:system# ip ia test 144.132.55.65
Vlan test, Addr 144.132.55.65, Subnet mask 255.255.0.0, type bcast Added
45:PowerHub:system#
```

5.1.2 Checksum

The **checksum|sum** command is used to calculate the checksum of files on the default or specified device. Calculating the checksum of all files when they are initially loaded and comparing this information, should problems start to occur, could point to possible corruption of the source file(s). Replacing the file(s) with clean copies may correct the problem. For each new code release, FORE Systems TACTics Online distributes checksum information against which you will be able to verify the validity of the code by using the **checksum** command. Each patch release has a corresponding patch release note which identifies each fix delta incorporated as well as the checksum values for the operational code modules (eg. 7pe, 7atm, 7feth, 7fdd).

The syntax for the **checksum|sum** command is:

```
checksum|sum [<device>] <filename>
```

where

- [<device>]** The default-device is used unless a device is specified. Specify **fd:** for the floppy diskette or **fm:** for the Flash Memory Module in the PowerHub 7000. The only device on a PowerHub 8000 is the Compact Flash Card (**fc:**).
- <filename>** Specify the name of a file on the device to calculate the checksum of the file.

The following example displays the checksum value of the Packet Engine runtime software image (**pe1**) located on the default-device, i.e. the Flash Memory Module (**fm:**).

```
24:PowerHub:system# checksum pe1
0x425367b2 FM:-PE1
25:PowerHub:system#
```

5.1.3 Copy

The **copy** | **cp** command is used to copy files from one device to another or to make additional copies of files on the default device or specified device. The syntax of this command is as follows:

```
copy|cp [default-device|<device>]<file1> <file2>
```

where

- [default-device|<device>]** Specifies the source device: **fd**, **fm**, or for the PowerHub 8000, **fc** only. If no device is specified the default-device is assumed.
- <file1>** Specifies the source filename.
- <file2>** Specifies the destination filename.



There is only one device available on the PowerHub 8000, the Compact Flash Card (**fc:**). Use of this command is, therefore, limited to copying files within the Compact Flash Card.

The following example copies the default configuration file (**cfg**) from the Flash Memory Module (default-device) to the floppy disk (**fd:**) renaming the file to **cfg1**.

Global Commands

```
39:PowerHub:system# copy cfg fd:cfg1
copy: copying 'cfg' to 'fd:cfg1'
40:PowerHub:system#
```

5.1.4 Default-Device

The `default-device|dd` command is used to set the default device for file operations within the current session. Subsequent file names that do not include a device name are automatically referred to this device. The syntax for this command is as follows:

```
default-device|dd set <device>
```

where

<device> The specified device can be **fd** for the floppy diskette or **fm** for the Flash Memory Module. Since the PowerHub 8000 has only one device, this command is not applicable.

The following example sets the default-device to the floppy diskette (**fd**) (PowerHub 7000 only).

```
43:PowerHub:system# dd set fd
default device set to FD:
44:PowerHub:system#
```

5.1.5 Directory

The `dir|ls` commands are used to display a listing of the contents of the files on the default or a specified device. These commands present a DOS or UNIX like listing of the contents of the files. The syntax for this command is as follows:

```
ls|dir [default-device|<device>] [<filespec>]
```

where

[default-device|<device>] Specifies a device if the default-device is not the device being queried. The device can be **fd**, **fm**, or for the PowerHub 8000, **fc** only.

<filespec> Specifies the files to display. The asterisk (*) wild card can be used to display all files of a particular type.

The following example displays the directory contents of the default-device, Flash Memory Module, on a PowerHub 7000.

```
1:PowerHub:system# dir
```

```

Volume in device is 4MB FLASH
ATM-PE1 598747 5-08-1998 9:36a
FDDI-PE1149841 5-08-1998 9:37a
FETH-PE177784 5-08-1998 9:37a
PPU-PE1 58365 5-13-1998 1:05p
PE1P PRM 588918 5-13-1998 1:05p
EXPERT 0 5-13-1998 1:09p
BOOTDEF PE1 28 5-08-1998 9:33a
PE1 1446688 5-08-1998 9:35a
BOOTDEF PPU 36 5-13-1998 1:04p
GFC 36779 5-15-1998 11:06a
CFG 34950 5-19-1998 12:57p
MREBOOT LOG 0 5-21-1998 1:31p
12 File(s) 1069568 bytes free

```

```
2:PowerHub:system# ls
```

```

598747 5-08-1998 9:36a FM:ATM-PE1
28 5-08-1998 9:33a FM:BOOTDEF.PE1
36 5-13-1998 1:04p FM:BOOTDEF.PPU
34950 5-19-1998 12:57p FM:CFG
0 5-13-1998 1:09p FM:EXPERT
149841 5-08-1998 9:37a FM:FDDI-PE1
77784 5-08-1998 9:37a FM:FETH-PE1
6779 5-15-1998 11:06a FM:GFC
0 5-21-1998 1:31p FM:MREBOOT.LOG
1446688 5-08-1998 9:35a FM:PE1
588918 5-13-1998 1:05p FM:PE1P.PRM
58365 5-13-1998 1:05p FM:PPU-PE1

```

5.1.6 Format

CAUTION



Use extreme care when executing this command. This command removes **all** files from the specified device. Ensure that any configuration (**cfg**) files have been saved to floppy disk (**copy cfg fd:cfg**) or transferred to a configured tftp boot server and that access to the run-time software is available either from floppy diskette (PowerHub 7000) or a tftp host.

This command is not applicable to the PowerHub 8000.

Global Commands

The **format|fmt** command can be used to format the Flash Memory Module (**fm:**) of the PowerHub 7000 or the Compact Flash Card (**fc:**) of the PowerHub 8000. The syntax for this command is as follows:

```
format|fmt <device>
```

where

<device> Specify **fm:** (Flash Memory Module).

The following example formats the Flash Memory Module.

```
63:PowerHub:system# format fm:
WARNING: Formatting will erase all data from the device.
Are you sure you want to continue? y
Erasing .....
Format complete
Restarting FlashFileSystem
Restarting FlashFileSystem complete
64:PowerHub:system#
```

5.1.7 Help

The **help|?** command is used to display the commands in the current, or specified, subsystem as well as syntax help for a specified command. Refer to Chapter 4 for detailed information on On-Line Help. The syntax for this command is as follows:

```
help|? <word> [<word> ... ]
```

where

<word> Specifies a command for which to obtain help. If no **<word>** is specified, a listing of the available commands is displayed.

[<word> ...] Specifies additional words that are normally used in conjunction with the first word specified, e.g., **interface add**.

The following example displays the help information for the **ip interface add** command. Notice that this command was executed from within the **system** subsystem and that the help command (?) was entered after specifying the **ip** subsystem.

```
17:PowerHub:system# ip ? interface add

it|interface add <vlanid> <ipaddr>[/<prefixlen>|<mask>]
    [ if[type] b[c] | n[bma] | p[top] <nbr_addr> ]

Add an IP interface to the given vlan. If <mask> is not specified
```

then "natural" subnet mask (class A, B, or C address mask) for the IP address is used. Interface type can be one of broadcast, nbma and ptop. Neighbor address must be specified only for ptop type. If interface type is not specified, broadcast is assumed by default.

```
18:PowerHub:system#
```

5.1.8 History

The **history|hi** command is used to display a history of the last 21 commands executed during the current session. The syntax for this command is as follows:

```
history|hi
```

The following example displays the last 6 commands executed. The line numbers on the left indicate the commands in the order they were entered. Using this display, it is possible to reenter a command by entering an exclamation mark followed by the command line number. In this example, entering (!3) will re-execute the **stty** command. The last command can also be re-entered with a double exclamation mark (!!).

```
6:PowerHub:system# hi
1 ver
2 idp
3 stty
4 ?
5 baud
6 hi
7:PowerHub:system#
```

5.1.9 History Characters

The **histchars** command can be used to display the current history characters or set different history characters. The syntax for this command is as follows:

```
histchars [<ch1>[<ch2>]]
```

The following example displays the current (default) history characters.

```
9:PowerHub:system# histchars
history sub: !          quick sub: ^
10:PowerHub:system#
```

For each session, a history of the 32 most recently issued commands is maintained. The history commands can be used to display, to reissue, or edit and reissue commands. To reissue or edit commands listed in the command history, use the history control characters. The default history control characters are:

- ! History-prefix character.
- ^ Quick-substitution character.

The history control characters can be used to form commands to be reissued (or modified and reissued) from the command history. The history commands used to edit and reissue commands listed in the command history are discussed below. The syntax is shown using the default history characters.

- !! Repeats the previous command.
- !**<n>** Repeats a command listed in the command history, where **<n>** indicates the number of the command as listed in the history display.
- !**<i>** Issues a previously issued command, where **<i>** is the offset back from the current command. For example, the command `!-1` gives the same results as `!!`, which reissues the previous command.
- !**<substring>** Repeats a previous command that begins with the string identified by **<substring>**.
- ^<old>^<new>** Modifies then reissues the previous command. **<old>** indicates the string to be replaced with **<new>**.

5.1.10 Logout

The `bye|logout` command is used to log out of the current session. If the lock switch, located on the front panel of the Packet Engine is set to Lock (refer to the *PowerHub 7000/8000 Hardware Reference Manual* for detailed information on setting/changing the Lock Switch and/or Lock Switch Jumpers), the interface displays the `login:` prompt and the next session requires the user to login at either the root or monitor level and to enter the appropriate password to gain access. Additionally, the command line counter is reset to 1. The syntax for this command is:

`bye|logout`

The following example shows the use of the `logout` command.

```
18:PowerHub:system# logout
```

5.1.11 Port Number Mode

The **pnm** command is used to change the way port numbers are entered and displayed. The syntax for this command is as follows:

```
pnm set multi|old
pnm [show]
```

where

multi|old Specifies to set either the multi-part (<slot>.<seg>) or the old-style (*vport*) port numbering scheme.

The following examples show the results of using the **pnm** command. First the current state is displayed, followed by a display of a configured vlan. The the port numbering mode is set to the old style and the same vlan is displayed. Notice that the Segment List field in the first vlan display shows the segment as <slot>.<port> and in the second vlan display the Segment List displays the virtual port. The multi-part numbering uses the physical segments on the installed cards, counting from the card in slot 1 up to the last segment on the last card installed in the system. The virtual port (<*vport*>) numbering system consecutively numbers each segment available in the system. Using the multi-mode scheme, the vlan shown is on the first segment of the card in slot 2.

```
12:PowerHub:ip# pnm
pnm:      multi
13:PowerHub:ip# vlan
VLAN      State Segment List
-----
techpubs  up      2.1

VLAN Count: 1
14:PowerHub:ip# pnm set old
15:PowerHub:ip# vlan
VLAN      State Segment List
-----
techpubs  up      33

VLAN Count: 1
16:PowerHub:ip#
```

5.1.12 Return Code Prompt

The **rcprompt** command is used to enable or disable printing of command return codes for commands executed automatically from a script. This feature is intended primarily for automated interactions with the command-line interface. The syntax for this command is as follows:

```
rcprompt enable|disable
```

where

enable|disable Enables or disables printing of command-return codes in the next UI prompt. Return codes are displayed with **0** for successfully executed commands and with **F** for unsuccessful commands.

The following example enables the return code prompt and displays the results.

```
33:PowerHub:system# rcprompt enable
00000000:34:PowerHub:system# hi
35 ip
36 it
37 vlan
36 pnm set multi
37 hi
00000000:38:PowerHub:system#
```

5.1.13 Read Environment

The **readenv|rdenv** command is used to execute the file environment *<file>* in the context of the current UI session. The syntax for this command is:

```
readenv|rdenv [default-device|<device>]<file>
```

where

[default-device|<device>] Specifies the device to read the environment file from: **fd**, **fm**, or for the PowerHub 8000, **fc** only. If no device is specified, the environment file is read from the default device.

<file> Specifies an environment file stored on the specified device. The default is no device specified.

The following example reads environment file *myenv* from the default device.

```
57:PowerHub:system# rdenv myenv
nui

#
# stty
#
stty rows 24
stty -more

#
# aliases
```

```
#
#
# timed commands
#
58:PowerHub:system#
```

5.1.14 Rename

The **mv** | **rename** command is used to rename files located on either a specified device or the default-device. The syntax for this command is as follows:

```
rename | mv [default-device | <device>] <file1> <file2>
```

where

- [default-device|<device>]** Specifies the device the source or destination file is located on: **fd**, **fm**, or for the PowerHub 8000, **fc** only. If no device is specified, the source file is assumed to be on the default-device.
- <file1>** Specifies the filename of the file to be renamed.
- <file2>** Specifies the new filename to be applied to the file.

5.1.15 Remove

The **rm** command deletes files from the default-device or a specified device. The syntax for this command is as follows:

```
rm [-i] [-f] [default-device | <device>] <filespec>
```

where

- [-i]** Overrides the **-f** flag, presenting a prompt before removing each file. The prompt provides an opportunity to cancel the request to remove the file. If **-f** or **-i** is not specified, **-i** is the default.
- [-f]** Specifies forced deletion. Delete without confirmation.
- [default-device|<device>]** Specifies the device the source file is located on: **fd**, **fm**, or for the PowerHub 8000, **fc** only. If no device is specified, the source file is assumed to be on the default-device.

<filespec> Specifies the file, or files, to be deleted from the default-device or specified device. Use of the asterisk (*) is allowed.

5.1.16 Save Environment

The **saveenv** | **svenv** command saves the current system environment to the default-device or specified device. The syntax for this command is as follows:

```
saveenv | svenv [default-device | <device>] <file>
```

where

[default-device|<device>] Specifies the device the environment file is being saved to: **fd**, **fm**, or for the PowerHub 8000, **fc** only. If no device is specified, the file is saved on the default-device.

<file> Specifies the name of the environment file to be saved.

5.1.17 Show Configuration Example

The **show-config-example** | **shex** command displays to the user console examples of typical configurations. There are no arguments or options for this command. The syntax for this command is as follows:

```
show-config-example | shex
```

```
169:PowerHub:ip/rip# shex
```

The following shows a short example to configure ip interface

```
ip vlan add 200.200.200.200 2.1 (add a vlan on segment 2.1)
ip it add 200.200.200.200 200.200.200.200 (add an ip interface)
ip enable (enable ip forwarding)
```

The following shows some commands in subsystem "bridge"

```
bridge br penable 2.1 ("port" enable bridging on seg. 2.1)
bridge br pdisable 2.2 ("port" disable bridging on seg. 2.2)
bridge st enable (enable spanning tree)
bridge st disable (enable spanning tree)
```

In summary, there may be "enable/disable" and their derives such as "penable/pdisable", "senable/sdisable", and etc. to set a particular feature on and off

Use "help [cnps]enable" and "help [cnps]disable" in each subsystem to see what can be set on/off.

```
170:PowerHub:ip/rip#
```

5.1.18 Set TTY

The **stty** command is used to set or display tty parameters. The syntax for this command is:

```
stty [-d[efault]] [-t <tty>] [rows <#>] [-|[+]more]
[-|[+]dcd] [<speed>] [erase <c>] [kill <c>] [werase <c>]
[intr <c>] [rprnt <c>] [stop <c>] [start <c>]
```

where

-default	Sets <speed> in NVRAM for the tty port.
-t <tty>	Specifies the tty port to apply the baud rate change. Used with the -default option.
rows <number>	Specifies the number of rows to display on the terminal.
[+]more	Enable paging of long displays.
-more	Disable paging of long displays.
[+]dcd	Enable dcd-detection.
-dcd	Disable dcd-detection.
tabs	Output tabs unchanged.
-tabs	Expand tabs to spaces on output.
<speed>	Specifies the baud rate to be used for tty. Used with default and -t options.

NOTE: Specifying -default also causes the speed change to occur in NVRAM. If -default is not specified, then the speed change affects the tty.

erase <c>	Sets erase character for telnet sessions.
kill <c>	Sets line erase for telnet sessions.
werase <c>	Sets word erase for telnet sessions.
intr <c>	Sets interrupt character for telnet sessions.
rprnt <c>	Sets reprint line for telnet sessions.
stop <c>	Sets xoff flow control for telnet sessions.
start <c>	Sets xon flow control for telnet sessions.

Global Commands

The following example sets the baud rate of `tty1` to the default of 9600 baud and the baud rate of `tty2` to 4800 baud. This is followed by a display of the current tty settings.

```
7:PowerHub:system# stty -default -t tty2 4800
8:PowerHub:system# stty
TTY          Current Baud Rate          NVRAM Baud Rate
1             9600                               9600
2             -----                               4800

rows: 24
more: disabled

dcd-detection is currently disabled.
9:PowerHub:system#
```

5.1.19 Set User

The `su` command can be used to nest access levels within the current session from `root` to `monitor`. The syntax for this command is as follows:

```
su [root|monitor]
```

where

[root|monitor] Changes the access level of the current session to either a `root` or `monitor` session. If the current session is `root`, the only available option is `monitor`, and vice versa. Entering `logout` returns the session to the `root` access level. When nested in a `monitor` session, the user has no `root` privileges. The default is `root`.

The following example switches the current user session from `root` to `monitor`. An attempt is then made to change the session back to `root`. An error message is presented. Then the `monitor` session is exited by entering `logout`.

```
1:PowerHub:system# su monitor
Ok
2:PowerHub:system> su root
Cannot nest 'su' commands: use 'logout' to end sub-session.
3:PowerHub:system> logout
4:PowerHub:system#
```

5.1.20 Subsystems

The `subsystems|ss` command displays a list of available subsystems to the console. The syntax for this command is:

subsystems | ss

The following example displays the results of entering the **ss** command.

```
4:PowerHub:system# ss
atalk atm atm/1483bridged atm/1483routed atm/clip atm/clippvc atm/foreip atm/lan
e atm/mps atm/nhs atm/mpc bridge dec fddi host ip ip/rip ip/ospf ip/mcast ipx ip
x/rip ipx/sap media nvramp snmp system tftp telnet
5:PowerHub:system#
```

5.1.21 Timed Command

The **timedcmd|tc** command is used to define a timed command. Timed commands can be defined to automatically issue any command string at regular intervals. A timed command is similar to an alias (see Section 5.1.1 for more information on defining an alias), except that it is automatically executed at a specific interval. Commands and aliases can be defined as timed commands. Each user session can contain up to eight timed commands. As with aliases, timed commands are local to the current session. Timed commands can be saved in environment files (see Section 5.1.16 for more information on saving environment files). If no argument is specified, a display of all defined timed commands is presented. The syntax for this command is as follows:

```
timedcmd|tc add <id> <time> <cmd>
timedcmd|tc del[ete] <id>
timedcmd|tc enl[enable] <id>
timedcmd|tc dis[able] <id>
```

where

- add** Associates <cmd> with <id> and interval <time>. If <id> specifies a running timed command, it remains running, but if <id> specifies an idle timed command, it remains idle.
- del[ete]** Delete timed command <id>.
- enl[enable]** Enable timed command <id>, that is, start it running.
- dis[able]** Disable timed command <id>, that is, stop it from running.

The following example creates a timed command to display the configured elans every 10 seconds. The timed command is then enabled and, following the display, deleted.

```
38:PowerHub:system# timedcmd add elansh 10 atm/lane elan all
Added elansh: 10 secs, atm/lane elan all (timer not running)
39:PowerHub:system# timedcmd enl elansh
elansh: started at 10 seconds interval
```

Global Commands

```
40:PowerHub:system#
Segment  Elan Name                Sel Byte      Mode      State
-----
1.1      -auto                          0x00         Auto      Down
40:PowerHub:system# timedcmd del elansh
elansh: stopped and deleted
41:PowerHub:system#
```

5.1.22 Type

The **type|cat** command can be used to display a file located on the default-device or a specified device to the console. This command works similarly to the UNIX cat or DOS type commands. The syntax for this command is as follows:

```
cat|type [default-device|<device>]<filename>
```

where

[default-device|<device>] Specifies the device on which the file to be displayed is located on: **fd**, **fm**, or for the PowerHub 8000, **fc** only. If no device is specified, the file is assumed to be on the default-device.

<filename> Specifies the name of the file to be displayed.

5.1.23 Unalias

The **unalias** command removes a previously defined alias definition (see Section 5.1.1). The syntax for this command is as follows:

```
unalias <name>
```

where

<name> Specifies the alias definition to be removed.

CHAPTER 6

System Commands

This chapter describes the commands in the `system` subsystem. These commands are used to display or control various system-level settings or conditions. The commands available within the `system` subsystem are:

```
51:PowerHub:system# ?
```

```
system subsystem:
```

```
baud                                passwd
bootinfo|bi                        readcfg|rdfcg
card-swap|cs                       reboot
config                             savecfg|svcfg
convert-config|ccfg               syslocln
date                               sysname
dcd-detection|dcd                 temperature|temp
ethaddr|ea                        tty2
idprom|idp                        uptime
mem                                version|ver
```

```
type 'global help' for global commands
```

```
type 'shex' to show an example of configuration
```

```
2:PowerHub:system#
```

6.1 Accessing the System Subsystem

The `system` subsystem is the default subsystem entered when the PowerHub completes the boot process. To access the `system` subsystem from any other subsystem, enter `system` from the current runtime prompt.

6.1.1 Baud

The `baud` command is used to set or display the baud rate on either the TTY1 or TTY2 RS-232 port. However, before setting the baud rate associated with TTY2, the port must be enabled using the `tty2` command (see Section 6.1.18). The syntax for this command is as follows:

```
baud set tty1|tty2 1200|2300|4800|9600|19200
baud [show]
```

where

<code>set</code>	Specifies that the baud rate is to be set. Sets the specified baud rate for the specified port.
<code>tty1 tty2</code>	Specifies the port to be set.
<code>1200 2300 4800 9600 19200</code>	Specify the desired baud rate to be applied to the specified port.

The newly specified rate is stored in non-volatile random access memory (NVRAM) and takes effect immediately. It is retained across logins and power cycles. The following examples display the current baud rate selections, then TTY2 is enabled and the baud rate is set to 9600.

```
43:PowerHub:system# baud
TTY      Baud Rate
1        9600
2        19200
44:PowerHub:system# tty2 enable
45:PowerHub:system# baud set tty2 9600
Changed tty2 baud rate to 9600; written to nvram
46:PowerHub:system#
```



If the Lock Switch is unlocked when booting the system, the TTY ports use the default baud rates (9600 for TTY1 and 1200 for TTY2), regardless of the baud rates stored in NVRAM.

6.1.2 Bootinfo

The `bootinfo|bi` command is used to display the contents of the boot log. The syntax for this command is as follows:

```
bootinfo|bi [show]
```

After the system run-time software is loaded, the following information is logged in memory as the boot log.:

- The date and time the system was started.
- The date, time and nvram bootorder (see Chapter 8 on setting the nvram boot order).
- The boot device used to boot. The value can be f (floppy diskette or c Compact Flash Card), m (Flash Memory Module) or n (network). This value shows the boot source actually used, which may differ from the boot order specified in NVRAM.

The following example displays the boot log.

```
7:PowerHub:system# bootinfo
Thu Feb 26 15:18:08 1998 start
Thu Feb 26 15:18:17 1998 nvram boot order: fm
boot device: m
8:PowerHub:system#
```

6.1.3 Card Swap

The `card-swap|cs` command is used whenever it is necessary to remove and re-install a NIM so that the configuration manager can deactivate traffic to ports/segments on that NIM. These operations can be accomplished while the system is operating if the module being installed matches exactly the module that was removed. Refer to the PowerHub 7000/8000 *Hardware Reference Manual* for detailed procedures on removing and replacing NIMs.



NIMs can only be swapped when the chassis contains at least one redundant power module. Refer to the PowerHub 7000/8000 *Hardware Reference Manual* for information about power redundancy. The `card-swap` command is only to be used with Network Interface Modules (NIMs).



The NIM being installed must be of the same type as the one removed. If the replacement NIM is of a different type than the one that was removed, it will be necessary to power down the system, remove the card, insert the new card and then power on the system. This sequence loads the ID PROM information of the cards currently installed into the configuration manager.

The syntax for this command is as follows:

```
card-swap|cs enable|disable <slot>
card-swap|cs [show]
```

where

enable|disable Specifies whether to enable, i.e. insert, or disable, i.e. remove, a module. Enable states that the card is restored to the system.

<slot> Specifies which slot is being enabled/disabled.

The following example shows the result of executing the `card-swap|cs` command with no options.

```
4:PowerHub:system# cs
Slot          Status
1:           Actively in service.
2:           Actively in service.
3:           Not present during boot!
4:           Actively in service.
5:           <<< Packet Engine CPU >>>
6:           Not present during boot!
7:           Not present during boot!
8:           Not present during boot!
9:           Not present during boot!
10:          Not present during boot!
5:PowerHub:system#
```

The following example removes the ATM PowerCell module in slot 1. This is followed by installing the ATM PowerCell module back to slot 1. Notice that following the execution of the `enable` command, the appropriate runtime module is reloaded to the new module.

```

84:PowerHub:system# cs disable 1
Card 1 removed.
85:PowerHub:system# cs enable 1

GINIM BOOTCard 1 inserted.
: slot 1, image "FM:atm pel"
86:PowerHub:system#

```

6.1.4 Config

The **config** command is used to display the current system configuration. This command displays information pertaining to the physical configuration of the system. There are no options or arguments to this command. The following example displays the system information for a 5-slot PowerHub 7000.

```

3:PowerHub:system# config
Accelerator board is present. Accelerator IOP is being used.
Installed DRAM Size: 24 MB
tty1: 9600 baud
tty2: 4800 baud
PE: slot 5
PM1: present and good
PM2: not present
PM3: not present
PM4: not present
 04/51 MM/MM
 02/33 UTP UTP UTP UTP UTP UTP
      UTP UTP UTP UTP UTP UTP
      UTP UTP UTP UTP
 01/01 OC3-MF OC3-MF OC3-MF OC3-MF OC3-MF OC3-MF
      OC3-MF OC3-MF
4:PowerHub:system#

```

The following information is displayed:

- Whether or not a Packet Accelerator is present on the Packet Engine and if the accelerator input/output processor (IOP) is in use or not.
- The amount of dynamic random access memory (DRAM) installed on the Packet Engine.
- The current baud rates assigned to the TTY1/TTY2 ports.
- The slot occupied by the Packet Engine, indicated by PE. In this example, the Packet Engine is in slot 5, the top slot in a 5-slot chassis.
- The presence and status of power modules, indicated by PM1, PM2, PM3, and PM4.

- The slot number and starting segment number of the modules in each slot, and the media type in use in each segment position. The row beginning 01/01 displays the configuration of the module in slot 1, beginning with segment 1. Empty NIM slots are not displayed.

6.1.5 Convert Config

The `convert-config|ccfg` command reads a configuration file from pre-*Forethought* PowerHub software and saves it to a *ForeThought* 5.1.0 compatible format. The syntax for this command is as follows:

```
convert-config|ccfg [default-device|<device>]<oldui filename>
                    [default-device|<device>]<newui filename>
```

where

<code>[default-device <device>]<oldui filename></code>	Specifies the device, or the default-device if no device is specified, and the filename of the pre- <i>ForeThought</i> software configuration file.
<code>[default-device <device>]<newui filename></code>	Specifies the device, or the default-device if no device is specified, and the filename of the <i>ForeThought</i> software configuration file.

The following example converts a configuration file (`cfg`), which is maintained on a floppy diskette, to `cfg2`, which is also stored on a floppy diskette.

```
92:PowerHub:system# ccfg fd:cfg fd:cfg2
##### Switching to OLDUI #####
##### Reading old config file: fd:cfg #####

##### Switching to ZUI #####
##### Saving config file: fd:cfg2 #####
93:PowerHub:system#
```

6.1.6 Date

The `date` command is used to display or set the system date and time. The syntax for this command is as follows:

```
date set [YYMMDD]hhmm[.ss]
        date [show]
```

where

`set` Sets the specified date and/or time.

[YYMMDD]hhmm>[.ss] Specifies the year (*YY*), month (*MM*), day (*DD*), hour (*hh*), minute (*mm*), and, optionally, the seconds [*.ss*]. To set the time, but not the date, specify *<hhmm>[.<ss>]*. If the seconds argument is used, make sure to use the period (.) in front of the seconds. If the number of seconds is not specified, the value is set to 00. (The software reads this argument from right to left, so any additional arguments can be specified with *<hhmm>*. For example, specifying *<DDhhmm>* also specifies the day. Note that the arguments must be specified in the order shown. For example, *<YYhhmm>* or *<DDMMYYhhmm>* cannot be entered.) If no arguments are specified, the current date and time is displayed.

The following examples show the command to display the current system date and time, the command to set a new system date and time, and the command to re-display the time set for verification.

```
117:PowerHub:system# date
Thu Feb 26 13:35:08 1998
118:PowerHub:system# date set 9802261436.30
date set to: Thu Feb 26 14:36:30 1998
119:PowerHub:system# date
Thu Feb 26 14:36:33 1998
120:PowerHub:system#
```

6.1.7 Data Carrier Detect

The **dcd-detection|dcd** command is used to enable or disable data-carrier detection. If entered with no arguments, the current state of data-carrier detection is displayed. The syntax for this command is as follows:

```
dcd-detection|dcd enable|disable
dcd-detection|dcd [show]
```

where

enable|disable Specifies whether to enable or disable data-carrier detection. If no argument is entered, the current state of data-carrier detection is displayed.

The following examples show a display of the current state of data-carrier detection, the disabling of data-carrier detection, and the re-enabling of data-carrier detection:

System Commands

```
135:PowerHub:system# dcd
dcd-detection is currently disabled.
136:PowerHub:system# dcd disable
dcd-detection disabled
137:PowerHub:system# dcd enable
dcd-detection enabled
138:PowerHub:system#
```

6.1.8 Ethernet Address

The `ethaddr|ea` command is used to display the Ethernet Mac-layer address of the PowerHub. The syntax for this command is as follows:

```
ethaddr|ea [show]
```

The following example displays the Ethernet MAC-layer address.

```
141:PowerHub:system# ethaddr
Ethernet address: 00-00-ef-03-9a-b0
142:PowerHub:system#
```

6.1.9 ID Prom

The `idprom|idp` command is used to display the ID PROM information recorded during the last power on cycle. The Packet Engine and NIMs contain a special PROM called the ID PROM. The ID PROM contains identification information and power requirements for the respective module. The syntax for this command is:

```
idprom|idp [show] <slot number>|all
```

where

`<slot number>|all` Specifies the slot containing the module for which the ID PROM information is requested or all, displaying the ID PROM information for all installed modules.

The following example displays the results produced by this command. In this example, information is displayed for the module in slot 1, a PowerCell 700 module.

```
77:PowerHub:system# idp 1

Card Type: PowerCell 700 (ATM)
Serial #: 633020637
Model: 7401-00
Revision: K
Issue: 3
Deviation: <not set>
```

```
Power Requirements:
  8000 mA at 5V
  10 mA at 12V
```

```
78:PowerHub:system#
```

The ID PROM display shows the following information:

Card Type:	The module currently installed in the specified slot.
Serial #:	The serial number of the module.
Model:	The model number of the module.
Revision:	The revision level of the module.
Issue:	The module issue number.
Deviation:	If applicable, the factory-assigned deviation number. Only some modules have deviation numbers.
Power Requirements:	The maximum amperage (milliamps) required by the module at +12-volts, +5-volts, or +3.3-volts, as applicable.

Some older revisions of the Packet Engine and NIMs do not contain ID PROMs. If the `idprom` command is issued against such a module, or an empty card slot, the following message is displayed:

```
78:PowerHub:system# idp 3
unable to read IDPROM information from slot 3
79:PowerHub:system#
```

6.1.10 Memory

The `mem` command is used to display the amount of memory that is installed on an Intelligent Network Interface Module (INIM) or the Packet Engine. The Packet Engine memory can also be noted when booting the system. The syntax for this command is:

```
mem [show] <slot-number>
```

where

```
<slot-number> The slot being queried.
```

The following example displays the amount of memory installed on the INIM located in slot 1. The INIM in slot one is an ATM PowerCell module.

```
3:PowerHub:system# mem 1
memory is 32MB
4:PowerHub:system#
```

6.1.11 Password

The `passwd` command is used to change the system password associated with “root” or “monitor” logins. The syntax for this command is:

```
passwd [root|monitor]
```

where

root|monitor Indicates the management capability for which the password is being changed.

The steps to change a password are:

1. Issue the `passwd` command, specifying the appropriate management level (root or monitor) capability. A prompt is displayed to enter the new password.
2. Enter the new password to be assigned to this management level. If no password is to be set, press Enter.
3. A prompt is then presented to re-enter the password (Re-enter new password:) previously entered.
4. Re-enter the password that was entered at the `New password:` prompt, press Enter.



This prompt is not displayed if the Lock Switch is in the unlocked position (U) or the Lock Switch jumper is set to Unlock. Instead, the `New password:` prompt is displayed.

5. The message “Password changed” is displayed to confirm that the password was changed.

The following example changes the password for the monitor access level.

```
11:PowerHub:system# passwd monitor
New password:*****
Re-enter new password:*****
Password changed
12:PowerHub:system#
```

For security reasons, the input shown above with asterisks does not appear when entered in response to the prompts. Passwords are limited to 13-characters in length. Remember that passwords are not required if the Lock Switch is in the unlocked (U) position. If the password is forgotten, turn the Lock Switch off, log in and enter a new password, then turn the Lock Switch on again.

6.1.12 Read Configuration

The `readcfg|rdcfg` command is used to load a configuration file. Even if the system finds and loads a configuration file when the software is booted, additional configuration files can be loaded during a session using the `readcfg` command. The syntax for this command is as follows:

```
readcfg|rdcfg [-v] [default-device|<device>]<file>
```

where

- `[-v]` Optional argument that displays each command in the configuration file to the user console as it is read.
- `[default-device|<device>]` Specifies the device where the configuration file is stored: `fd`, `fm`, or in PowerHub 8000, `fc` only. If no device is specified, the `default-device` is assumed.
- `<file>` Specifies the name of the configuration file to be read.



The new configuration information does not undo the configuration information contained in the default `cfg` file. Instead, the new configuration is added to the current configuration until the system is powered down or rebooted. The additional configuration information can be saved with the current configuration information by issuing the `savecfg` command (see Section 6.1.14).

6.1.13 Reboot

The `reboot` command is used to manually reboot the PowerHub. The `reboot` command performs a cold restart of the PowerHub. During a cold restart, the Packet Engine conducts a power-on self-test to check its various hardware components. Following successful completion of the power-on self-tests the system software is loaded. The syntax for this command is as follows:

```
reboot
```

6.1.14 Save Configuration

The `savecfg|svcfg` command is used to save the current configuration to a file on a specified device. The default filename for this file is `cfg`, but any filename can be used. If configuration changes are saved to a file other than `cfg`, the file must be loaded after the software is loaded using the `readcfg|rdcfg` command (see Section 6.1.12). The syntax for this command is:

```
savecfg|svcfg <file or device name>
```

where

<file or device name> Specifies the name under which the configuration file is to be saved.

The following example saves the current configuration changes to the default file `cfg` on the default-device.

```
17:PowerHub:system# savecfg cfg
overwrite cfg? y
18:PowerHub:system#
```

6.1.15 System Location

The `syslocn` command is used to optionally identify a particular PowerHub. The system location can be used by a Simple Network Management Protocol (SNMP) management station to identify this particular system. The syntax for this command is:

```
syslocn set <location>
syslocn [show] <location>
```

where

set Sets the specified location.

<location> Specifies the location of the PowerHub. Any alphanumeric string up to 24 characters in length can be specified. The location name cannot contain spaces. If a location is not specified, the location name of the current system issuing the command is displayed.

The following example shows the current system location, “Undefined,” and changes the location variable to “TechPubs.”

```

19:PowerHub:system# syslocn
Current system location is: Undefined

20:PowerHub:system# syslocn set TechPubs
System location set to:
TechPubs
21:PowerHub:system#

```

6.1.16 System Name

The **sysname** command can be used to change the displayed PowerHubsystem command prompt. The default system name is PowerHub. The syntax for this command is:

```

sysname set <sysname>
sysname [show] <sysname>

```

where

set	Sets the name specified.
<sysname>	Specifies a name to be assigned to this PowerHub. Any alphanumeric string up to 24 characters in length can be specified. The name cannot contain spaces. If a name is not specified, the current system name is displayed.

The following example shows how to display the current system name and to change the name variable. The new system name is defined as “PHswitch.”

```

22:PowerHub:system# sysname
Current system name is: PowerHub
23:PowerHub:system# sysname set PHswitch
System name set to 'PHswitch'.
24:PHswitch:system#

```

6.1.17 Temperature

The **temperature|temp** command is used to read the temperature sensor on board each module. Each module type contains an on board temperature sensor that reads the temperature of the module with an accuracy of plus or minus 0.5° C. The syntax of this command is:

```

temperature|temp [show] <slot number>|all

```

where

<slot number>	Specifies the slot for which to display the temperature.
----------------------------	--

all Displays the temperature for all installed modules.

In the following examples, the temperature of all installed modules is displayed followed by the temperature of the module currently installed in slot 1.

```
4:PowerHub:system# temperature all
slot 5, temp 44 degrees C
slot 4, temp 39 degrees C
slot 2, temp 34.5 degrees C
slot 1, temp 34.5 degrees C
5:PowerHub:system# temp 1
slot 1, temp 34.5 degrees C
6:PowerHub:system#
```

Note that the PowerHub is designed to operate over a range of external ambient temperatures. An additional temperature rise inside the chassis is taken into account in the design of the product.

Some older revisions of the Packet Engine and NIMs do not contain an ID PROM. If the **temperature** command is issued against a module that does not contain an ID PROM or against a slot that does not contain a NIM, the system displays the following message:

```
6:PowerHub:system# temp 3
slot 3, temp not available
7:PowerHub:system#
```

6.1.18 TTY2

The **tty2** command is used to enable or disable the TTY2 port, located on the Packet Engine. The TTY2 port must be enabled before setting or changing the associated baud rate with the **baud** command (see Section 6.1.1). The syntax for this command is as follows:

```
tty2 enable|disable
```

where

enable|disable Enables or disables the tty2 port.

In the following examples, an attempt is made to change the baud rate of the TTY2 port to 9600 baud. An error is displayed. The TTY2 port is then enabled and another attempt is made to change the baud setting. This results in a message displaying that the baud rate was changed and written to nvram. The TTY2 port is then disabled. A message is displayed indicating that the TTY2 port is now closed.

```
13:PowerHub:system# baud set tty2 9600
Failed to change tty2 baud rate to 9600
14:PowerHub:system# tty2 enable
15:PowerHub:system# baud set tty2 9600
```

```

Changed tty2 baud rate to 9600; written to nvram
16:PowerHub:system# tty2 disable
tty2 is now closed
17:PowerHub:system#

```



If the Lock Switch is unlocked when booting, the TTY ports use the default baud rates (9600 for TTY1 and 1200 for TTY2), regardless of the baud rates stored in NVRAM.

6.1.19 Uptime

The **uptime** command is used to display how much time has elapsed since the last time the software was loaded. There are no parameters for the **uptime** command. The syntax for this command is as follows:

```
uptime [show]
```

The following example shows that the system has been up for 15 hours, 8 minutes and 52 seconds.

```

19:PowerHub:system# uptime
Elapsed time since last reboot: 15 hours, 8 minutes, 52 seconds
20:PowerHub:system#

```

6.1.20 Version

The **version|ver** command displays the version level of software currently running on the installed modules. The syntax of this command is as follows:

```
version|ver [show] [<slot-number>|all]
```

where

- <slot-number>** Lists the version of software on the card in the slot specified.
- all** Lists the version of software on the Packet Engine and all installed Intelligent NIMs. If no parameter is specified, version information for the Packet Engine firmware is displayed.

The following examples show typical displays of the use of this command.

System Commands

21:PowerHub:system# **ver**

Card Type: Packet Engine - 40MHz
Serial #: 633020265
Model: 7101-01
Revision: C
Issue: 2
Deviation: <not set>

PowerHub Version: PH ple FT_5.0.0 @6933 1998.02.18 09:20

PROM Version: pelp-3.0.0 (7887) 1998.05.06 13:01

00000000:12:PowerHub:system# ver 1

Card Type: PowerCell 700 (ATM)
Serial #: 633020637
Model: 7401-00
Revision: K
Issue: 3
Deviation: <not set>

Runtime Version: PH7000-FT_5.0.0 atm-pel (7973) 1998.05.07 11:26

Prom Version: PH7000-7atmp-1.3 (s1.8) 1998.03.10 11:39

00000000:13:PowerHub:system# ver all

COPYRIGHT NOTICE #####
Copyright © 1994-1998 FORE Systems, Inc., as an unpublished work. This notice does not imply unrestricted or public access to these materials which are a trade secret of FORE Systems, Inc. or its subsidiaries or affiliates (together referred to as "FORE"), and which may not be reproduced, used, sold or transferred to any third party without FORE's prior written consent.

All rights reserved.

Slot 5

Card Type: Packet Engine - 40MHz
Serial #: 633020265
Model: 7101-01
Revision: C
Issue: 2
Deviation: <not set>

PowerHub Version: PH7000_FT_5.0.0 pel (7973) 1998.05.07 11:19

```

PROM Version: PH7000-pe1p-3.0.0 (7887) 1998.05.06 13:01
##### Slot 4 #####

Card Type: UTP 13x1 Interface Module
Serial #: 98125515
Model: 7350-01
Revision: K
Issue: <not set>
Deviation: <not set>

##### Slot 3 #####

Card Type: 6x1 FE
Serial #: 97516928
Model: 7360-00
Revision: P
Issue: <not set>
Deviation: <not set>

Runtime Version: PH7000-FT_5.1.0 feth-pe1 (7973) 1998.05.07 11:30
Prom Version: PH7000-7fep-1.2 (s1.7) 1997.03.03 15:33

##### Slot 2 #####

Card Type: UTP 16x1 Interface Module
Serial #: 632027371
Model: 7202-00
Revision: G
Issue: 1
Deviation: <not set>

##### Slot 1 #####

Card Type: PowerCell 700 (ATM)
Serial #: 633020637
Model: 7401-00
Revision: K
Issue: 3
Deviation: <not set>

Runtime Version: PH7000-FT_5.0.0 atm-pe1 (7973) 1998.05.07 11:26
Prom Version: PH7000-7atmp-1.3 (s1.8) 1998.03.10 11:39

```

The information provided in the **ver** command contains:

System Commands

Card Type:	The module currently installed in the slot.
Serial #:	The serial number of the module.
Model:	The model number of the module.
Revision:	The revision level of the module.
Issue:	The module issue number.
Deviation:	If applicable, displays the factory-assigned deviation number. Only some modules have deviation numbers.
Runtime Version:	Displays the installed software version, with the software build number, date and time.
PROM Version:	Displays the installed PROM version information, which includes the version number with build date and time.

CHAPTER 7

Media Commands

This chapter describes the `media` subsystem commands. The `media` subsystem commands relate to the physical media and bridging configuration information. The commands within the `media` subsystem are:

```
24:PowerHub:media# help
```

```
media subsystem:
```

```
config                segment
isstats               segmentname | segname | name
ledmode | lm          ssd
monitor              ssdthreshold | ssdt
operating-mode | om  status
portreceive | pr     stats
portstats
```

```
25:PowerHub:media#
```

7.1 Displaying Bridge-Related Configuration

The `config` command is used to display the current port and segment configuration and displays bridge-related information. The syntax of this command is as follows:

```
config [show] [<params>] [<disp-restrictors>]
```

where

<params> Specifies a comma separated list of parameters, where;

monitor Displays whether port monitoring is enabled or disabled on the specified segment or segment list.

segment Displays whether forwarding is enabled or disabled on the specified segment or segment list.

[port]receive Displays whether the UTP port receivers are enabled or disabled. All available ports are displayed whether a segment is specified or not.

ssd Displays the status of automatic segment state detection on the specified segment, or command-separated list of segments.

[segment]names Displays the segment names for the specified segment or command-separated list of segments. If no segment is specified, the segment names for all available segments are displayed.

portstats Displays whether port statistics are enabled or disabled.

isstats Displays whether inter-segment statistics collection is enabled or disabled.

<disp-restrictors> The display restrictors is limited to a segment or a comma separated list of segments.

The following example displays the bridge-related configuration information for port 2.1.

```
48:PowerHub:media# config 2.1

Port Monitoring
-----
Packets...
not being monitored on segment 2.1
```

Forwarding status of segments

2.1 :enabled

UTP port receiver enable/disable status

Slot 4: .

Slot 2:

Slot 1:

.

Automatic segment state detection

Segment 2.1 : enabled (currently good)

Segment names

2.1 : Port_33

Port level statistics collection: currently disabled.

Inter-Segment Statistics collection is disabled

49:PowerHub:media#

7.2 Inter-Segment Statistics

If statistic collection is enabled, the `isstats show` command can be used to display, clear, enable, or disable the statistics collection for packets between segments on installed PowerHub NIMs. The syntax of this command is as follows:

```
isstats [show] [<params>] [<disp-restrictors>]
isstats clear|enable|disable
```

where

<params> Specifies a comma-separated list of packets (**p**) or octets (**o**).

<disp-restrictors> Specifies an optional list of segments from which to collect statistics. Specify a `fr[om]=<seglist>` and `to=<seglist>` list of segments.

The following example displays the inter-segment statistics for packets sent from segment 1.1 to segment 2.1.

```
52:PowerHub:media# isstats p fr=1.1 to=2.1
Segment to segment statistics collection is disabled
FROM          TO>          2.1
1.1 : pkts          0

53:PowerHub:media#
```

7.3 Ethernet LED Modes

The `ledmode` command is used to configure or display the operating mode of the traffic LEDs (C/X and A/R) on all types of Ethernet modules, except the Universal Ethernet Module (UEM). LEDs are set as a group for the entire module, not on a segment-by-segment basis. The operating mode information displayed by the LEDs on installed Ethernet modules can be set to reflect either transmission collision and activity or transmit and receive activity.

Set the LEDs to C and A (transmission collision and activity) or to X and R (packet transmit/receive). The LNK LED cannot be configured and always shows link status information for the corresponding segment. (For a description of the information indicated by each setting, refer to the *PowerHub 7000/8000 Hardware Reference Manual*.) The default `xr` configuration should be used except in networks that experience a large number of collisions. Note that collisions do not occur on segments that are configured for full-duplex operation (see Section 7.6 for details on the `operating-mode | om` command).



The LEDs on the UEM always indicate receive, transmit, and collision separately.

The syntax for this command is:

```
ledmode | lm [show] [<slot>]
ledmode | lm set <slot> ca | xr
```

where

- set** Set the LEDs on the specified slot to either `c/a` or `x/r`. The slot specified must be a slot containing an Ethernet module.
- [<slot>]** Specify which slot to set the specified configuration or which slot to display the current configuration. If no slot is specified, all slots are displayed

If a NIM slot is specified, but not a LED setting, the current setting for the specified module is shown. If a NIM slot is not specified, the current LED settings for all Ethernet modules (except UEMs) in the chassis are shown.

Media Commands

ca|xr Configures the LEDs to reflect either transmit collisions and activity (transmit and receive) or packet transmit and receive activity. The default is **xr**.

The following example displays the current Ethernet module LED settings. Notice that only the module in slot 2 is an Ethernet module.

```
55:PowerHub:media# lm  
Slot 01: LED not configurable on this module type.  
Slot 02: xr (leds reflect transmit and receive activity)  
Slot 04: LED not configurable on this module type.  
56:PowerHub:media#
```

The following example sets the LED mode settings to **ca** on the installed Ethernet module.

```
59:PowerHub:media# lm set 2 ca  
Slot 02: ca (leds reflect collision and activity)  
60:PowerHub:media# )
```

7.4 Port Monitoring

Port monitoring allows the use of a protocol analyzer (such as a Sniffer, LANalyzer, or Network Pharaoh) connected to a PowerHub segment to monitor the traffic on any other segment or set of segments. Rather than separately attaching the analyzer to each segment to be monitored, the analyzer can be attached to one segment, then Port Monitoring commands can be used to select the segments to monitor.

The segments being monitored can be changed without moving the analyzer to another segment. In addition, traffic on more than one segment can be monitored simultaneously, without the need to use multiple analyzers on multiple segments. To enable port monitoring, perform the following steps:

1. Log on at the `root` access level.
2. Disable forwarding on the segment to which the analyzer is attached, issuing the `segment penable|pdisable <segment-list>` command, where `<segment-list>` specifies the segment, or segments, to which the analyzer is to be attached (refer to Section 7.9 for more information on the `segment` command).



If the Port-Monitoring feature is being used frequently, configure one segment for Port Monitoring to prevent having to disable forwarding each time this feature is used.

3. Enable port monitoring with the `monitor set` command (refer to Section 7.5 for descriptions of the options and parameters available for use with the `monitor set` command).

7.4.1 How Port Monitoring Works

Conceptually, port monitoring “copies” packets from the monitored segments to the monitoring segments. Actually, packets are not really copied because this would dramatically reduce performance. Instead, a pointer to the packet buffer containing a monitored packet is placed on the transmit queue for the monitoring segment(s), and the packet buffer is freed up only after it has been transmitted both to its normal destination and to the monitoring segment(s). For monitoring purposes, packets are classified into three types:

Incoming A packet that is received on the monitored segment. An incoming packet might or might not be forwarded, according to the usual bridging and routing rules.

Forwarded	A packet received on one segment, then transmitted on the monitored segment.
Generated	A packet transmitted on the monitored segment as required by the internal protocol stacks. This includes outgoing TCP packets in TELNET sessions, UDP packets for RIP updates and SNMP replies, ARP requests and replies, ICMP packets for various IP routing errors, Spanning-Tree hello and topology-change packets, and various packets generated by the IPX, AppleTalk, and DECnet protocol stacks.

Port monitoring monitors packets regardless of any filters defined on the monitoring segment. This includes any filters that normally block traffic from the monitored segment to the monitoring segment. Filters defined on the monitored segment remain in effect. In addition, incoming packets are monitored regardless of a segment's Spanning-Tree state (blocked or forwarding) or the enabled state (enabled or disabled) of the monitored segment.

7.4.2 Performance Considerations and Operation Notes

In general, port monitoring does not adversely affect performance on the monitored segments or other segments. However, if the monitored traffic load is greater than the capacity of the monitoring segment, then not all monitored packets are successfully queued. Packets not queued onto the monitoring segment for this reason are still delivered to their normal destinations.

When multiple segments are monitored, packets from all segments are queued onto the monitoring segment in the approximate order in which they were received, forwarded, or generated. Note that if a packet is "incoming" on one monitored segment and "forwarded" on another monitored segment, only one copy of the packet is queued onto the monitoring segment.

When outgoing (forwarded or generated) and incoming packets are monitored on a segment, they might not appear on the monitoring segment in the same order in which they appear on the monitored segment. This can happen because an outgoing packet is queued for transmission on the monitoring segment at the same time that it is queued for transmission on the monitored segment, not when it is actually transmitted. Therefore, it is possible for one or more packets to be received on the monitored segment and queued after the outgoing packet on the monitoring segment, even though they appear on the monitored segment before the outgoing packet is actually transmitted. However, the order of packets within either the incoming stream or the outgoing stream on the monitored segment is preserved on the monitoring segment.



Incoming runt packets, giant packets, and packets with FCS or frame-alignment errors are not monitored. Long (larger than 1518 bytes) packets on FDDI segments are fragmented and the fragments appear on the protocol analyzer. This applies even if the monitoring segment and monitored segment are both FDDI segments.



Do not use the monitoring segment for routing or any other purpose except monitoring. The monitoring segment should not have any devices connected to it other than a protocol analyzer. Other types of connected devices (workstations, servers, and so on) can get very confused by packets from monitored segments.

7.4.3 Packet Modifications

During normal bridging and routing, certain packets are modified before being forwarded. For example, both the MAC-layer and network-layer (routing) headers in routed packets are modified. Moreover, when packets are forwarded from FDDI to Ethernet, or vice versa, it modifies the packets accordingly.

With port monitoring, the modified packet, not the original packet, is transmitted to the monitoring segment. As a result, the packet displayed by the protocol analyzer is the modified packet. The way the packet is modified depends upon the segment type (Ethernet or FDDI) and the forwarding algorithm used, as summarized in Table 7.1.

Table 7.1 - Packet Modifications On Monitoring Segment

Traffic Type	Monitored Segment Type	Monitoring Segment Type	Packet Is...
Bridged Description: Forwarded, or incoming but not forwarded.	Ethernet	Ethernet	U
	Ethernet	FDDI	T
	FDDI	Ethernet	T
	FDDI	FDDI	TT/U
Routed Description: Forwarded.	Ethernet	Ethernet	M, I, R
	Ethernet	FDDI	M, I, R, T
	FDDI	Ethernet	M, I, R, T
	FDDI	FDDI	M, I, R, TT/U
Routed Description: Incoming but not forwarded.	Ethernet	Ethernet	I
	Ethernet	FDDI	I, T
	FDDI	Ethernet	I, T
	FDDI	FDDI	I, TT/U
Generated Description: Generated.	Ethernet	Ethernet	U
	Ethernet	FDDI	T
	FDDI	Ethernet	U
	FDDI	FDDI	U
KEYI=IP TTL and checksum changed M=MAC address changed R=Routing header changed U=Unmodified T=Translated TT/U=Double Translated but Unchanged.			

The modifications made to packets appearing on the monitoring segment are further explained by the following key:

- U** The packet is not changed in any way. If the packet also undergoes a double-translation (denoted in Table 7.1 by TT), this means the packet is double-translated, but the resulting packet is identical to the packet before double translation.

- M** The destination MAC address is changed to the address of the next hop. The source MAC address is changed to the address of the PowerHub.
- I** If the packet is an IP packet, certain fields in the IP header are changed. Specifically, the TTL field is decremented and the IP-header checksum is incremented. The IP header and payload are otherwise unmodified.
- R** Certain fields in the network header (for example, the IP header) might be changed, depending upon the routing protocol:

AppleTalkThe hop count is increased by one. Also, if the packet contains a checksum, the checksum is changed appropriately.

IP If the packet has an options field specifying source routing or route tracing, the appropriate modifications are made. If IP security options (RFC 1108) are used, then option fields may be added to or removed from the header. In rare cases, adding option fields causes the packet to exceed 1518 bytes, and consequently the packet becomes fragmented.

IPXThe only IPX field that is changed in the header is the “Transport Control” field. This field is incremented by 1 for each router that the packet passes through. (This field is similar to the TTL field in the IP header). Note, however, that the MAC header can change in many different ways.

In the simplest case, where there is no header translation, the MAC header is changed as follows:

src-mac-addr Changed to the address of the PowerHub.

dst-mac-addr Changed to the address of either the destination node or the next hop gateway.

When header translation is involved, in addition to the two fields above, the header type changes from the configured type for the receiving network to the configured type for the destination/next-hop network. The four different encapsulation types used for IPX are IEEE 802.3 (Raw), IEEE 802.2 (LLC), IEEE 802.2 (SNAP) and Ethernet-II.

DECnetThe following fields in the MAC header are changed:

*src-mac-addr*Changed to the address of this router.

*dst-mac-addr*Changed to the address of either the destination node or the next hop gateway.

In addition, a change is made to the DECnet long data packet headers (these are normal data packets). The long data header contains a “flags” field which is modified as follows:

If the source and destination nodes of the packet are both on the same segment, the “INTRA ETHERNET PKT” bit is set.

If the source and destination nodes are on different segments, the “INTRA ETHERNET PKT” bit is cleared.

If the destination node is not reachable and the sender has set the “RETURN TO SENDER REQ” bit, the PowerHub clears this bit and sets the “RETURNING TO SENDER” bit. In this case, the MAC header is changed as follows:

*src-mac-addr*Changed to the PowerHub MAC-layer hardware address.

*dst-mac-addr*Changed to the MAC-layer hardware address of the sender.

- T The packet undergoes translation between Ethernet and FDDI formats. In the case of long FDDI IP packets (larger than 1518 bytes), the packet also undergoes IP fragmentation. (Long non-IP packets are not monitored.)

TT/U The packet undergoes a double translation, from FDDI to Ethernet and back. The end result normally appears unchanged, except for fragmentation in the case of long IP packets. (Long non-IP packets are not monitored.)

7.5 Monitoring a Segment

The `monitor` command is used to monitor one or more segments on the PowerHub. Before monitoring, disable forwarding on the segment to which the monitored traffic is being sent. Use the `segment` command (see Section 7.9) to disable forwarding on a segment. After disabling the segment, issue the `monitor` command to begin monitoring. If multiple `monitor` commands are issued, their effect is cumulative. That is, the PowerHub monitors **all** of the traffic specified by **all** of the commands. The syntax for the `monitor` command is as follows:

```
monitor set [from <monitor-spec>] [to <monitor-spec>] on <seglist>
           monitor [show] [<seglist>]
           monitor clear
```

where

- | | |
|------------------------------------|---|
| set | Sets monitoring on the specified segment or segments. |
| [from <monitor-spec>] | <p>Specifies which segments are to be monitored. Packets entering through segments identified by this variable are copied to the monitoring segment. The <i><monitor-spec></i> can be one of the following options:</p> <ul style="list-style-type: none"> internalpacket from internal protocol stacks. <i><seglist></i>packets from the specified segments. *packets from all segments anypacket from internal protocol stacks. This is the default option and the same as the “internal” option. |
| [to <monitor-spec>] | <p>Specifies which segments are to be monitored. Packets leaving through segments identified by this variable are copied to the monitoring segment. The <i><monitor-spec></i> can be one of the following options:</p> <ul style="list-style-type: none"> internalpacket to internal protocol stacks. <i><seglist></i>packets to the specified segments. *packets to all segments anypacket from internal protocol stacks. This is the default option and the same as the “internal” option. |

- <seglst>** Indicates the segments to which the monitored traffic is to be sent. For most applications of Port Monitoring, this segment list contains just one segment.
- clear** Clears all port monitoring parameters previously set.



Packets destined for internal protocol stacks cannot be differentiated from other incoming packets. To exclude packets destined for internal protocol stacks, use the “*” option in the “[to <monitor-spec>]” field and filter by MAC address with external monitoring equipment.

7.6 Operating-Mode

The `operating-mode|om` command is used to set the operating mode of installed Ethernet or Fast Ethernet modules or media adapters. For general information about Ethernet and Fast Ethernet Modules and adapters, see chapter 8 in the *PowerHub 7000/8000 Installation and Maintenance Manual*. The syntax for the `operating-mode|om` command is described below:

```
operating-mode|om [show] [<seglist>|all]
operating-mode|om set <seglist>|all fdx|lbk|flbk|declbk|normal|hdx
    For 10/100 Fast Ethernet ports:
operating-mode|om set <seglist>|all <om-config-mode>
```

where

[<seglist> all]	Specifies a segment, segment list, or all segments to set the operating mode. The specified segments can be a range of segments separated by a hyphen or a comma-separated list of segments.
set	Sets the operating mode specified on the specified segments.
fdx lbk flbk declbk normal hdx	Sets the specified segment(s) to the desired operating mode where: fdx Full duplex lbk Local loopback to the on board chip. flbk Full duplex loopback to the on board chip. declbk Loopback to the on board DEC chip. normal Half-duplex without loopback. hdx Half-duplex

7.6.1 Full-Duplex and Half-Duplex Modes

The default operating mode for the 100Base-TX, the 100Base-FX, and 10/100 FEMA is full-duplex mode. In full-duplex mode, these FEMA types are capable of transmitting and receiving simultaneously. Moreover, segments being used in full-duplex mode do not experience collisions. The exception to the full-duplex mode default is when the 10/100 is connecting to a legacy device that does not support auto-negotiation or to a device whose auto-negotiating function has been turned off. In this case, the 10/100 defaults to half-duplex. To over-ride this default, you must turn off the auto-negotiation and manually set the speed and mode to coincide with that of the connecting device.

The alternative to full-duplex mode is half-duplex mode. In half-duplex mode, the FEMA can transmit and receive, but not simultaneously. At any given moment, the FEMA is either transmitting or receiving (or is inactive).

The maximum bandwidth available on a Fast Ethernet segment operating in half-duplex is 100 Mb/s. Thus a maximum of 100Mb/s can be used to either send or receive.

The operating mode for one or more FEMA's can be changed by using the `media operating-mode` command. Refer to the *PowerHub 7000/8000 Software Reference Manual* for more information on setting the operating mode.

CAUTION



Note that if you are connecting to a device that either does not support auto-negotiation or whose auto-negotiating function has been turned off, you must configure mode and speed manually. Inconsistent mode settings on connecting devices may result in data corruption.

7.6.2 Auto-negotiation

Auto-negotiation enables the 10/100 FEMA to operate at either 10Mb/s or 100Mb/s by detecting the operating speed of the device at the other end of the connection. With auto-negotiation it is not necessary to manually configure the operating speed or mode. Auto-negotiation configures the FEMA automatically to operate at the same speed and mode as the device it is connecting to. The exception to this auto-negotiation feature is the case noted in the caution above when a 10/100 FEMA connects to a legacy device without auto-negotiation or to a device whose auto-negotiating function has been disabled. In this case, the auto-negotiation on the 10/100 FEMA must be turned off to prevent the 10/100 from automatically defaulting to half-duplex. After turning off auto-negotiation, set the speed and mode manually to match those of the connecting devices.

7.6.3 10/100 FEMA Values

For 10/100 Fast Ethernet Media Adaptor (FEMA) ports, the following values can be entered with the operation mode configuration command values.

`<om-config-mode>` Specifies the operating mode for the configured 10/100 FEMA ports where one of the following values must be entered:

```
fdx100_onSets full duplex, 100Mbps, auto-  
negotiation enabled
```

```
fdx10_onSets full duplex, 10Mbps, auto-negotiation  
enabled
```

```
hdx100_onSets half duplex, 100Mbps, auto-  
negotiation enabled
```

```
hdx10_onSets half duplex, 10Mbps, auto-negotiation  
enabled
```

```
fdx100_offSets full duplex, 100Mbps, auto-  
negotiation disabled
```

```
fdx10_offSets full duplex, 10Mbps, auto-negotiation  
disabled
```

```
hdx100_offSets half duplex, 100Mbps, auto-  
negotiation disabled
```

```
hdx10_offSets half duplex, 10Mbps, auto-negotiation  
disabled
```

```
lbcLoopback on (enabled)
```

```
nolbcLoopback off (disabled)
```



The various loopback options should only be used when attempting to isolate problems on the Ethernet/Fast Ethernet modules supported. These should also only be used in conjunction with assistance from FORE Systems TAC.

7.6.4 Setting the Operating Mode for Ethernet and 10/100 FEMA

Attempting to set a port to an operating mode that is not supported by that particular segment results in an error message as shown in the first example below. The next two examples set the operating mode on Ethernet segments. Note that all **operating mode** commands are entered from the media subsystem.

```
38:PowerHub:media# om set 1.1 hdx  
Segment 1.1 : May set operating mode only on AUI, 10T, 10FL, 100TX, 10/100 and  
100FX ports.  
39:PowerHub:media# om set 2.2 fdx
```

```
Segment 2.2 : fdx
```

```
40:PowerHub:media# om set 2.3 normal
Segment 2.3 : normal
```

The following examples set the operating mode on 10/100 FEMA segments. The first example sets the 10/100 FEMA port to full duplex, 100Mbps, with autonegotiation enabled:

```
41:PowerHub::media# om set 2.4 fdx100_on
```

In the second example, the 10/100 FEMA is connecting to a legacy device sending packets at 10Mb/s in full-duplex mode. The legacy device does not support autonegotiation. The settings on the port will need to be as follows: full-duplex mode, 10 MB, and auto-negotiation off. The command below shows this configuration on port 2.5.

```
42:PowerHub: media# om set 2.5 hdx10_off
```

7.6.5 Displaying the Operating Mode Configuration

The `om [show][<seglist>]|all` command shows current operating mode configurations on the segments specified.

```
2:PowerHub:media# om 2.1-2.15
```

Segment	2.1 : hdx	100MB	Autoneg ON	10/100	status:	Undetermined
Segment	2.2 : fdx	100MB	Autoneg ON	10/100	status:	Fdx, 100Mbps
Segment	2.3 : hdx	10MB	Autoneg ON	10/100	status:	Hdx, 10Mbps
Segment	2.4 : hdx	10MB	Autoneg ON	10/100	status:	Fdx, 10Mbps
Segment	2.5 : fdx	10MB	Autoneg OFF	10/100	status	Fdx, 10Mbps

7.6.5.1 Troubleshooting

If you are experiencing delays or corruption of data, first verify that the configuration is set for optimal performance. One of the ways to detect the source of data transfer problems is to view the data transfer statistics. To view the statistics display, enter the `stats` command from the media subsystem. Any errors reported in this display may indicate that the settings on ports receiving and transmitting data are not set to accommodate the settings on devices they are connecting to.

7.7 UTP Port Receiver Status

The `portreceive|pr` command enables or disables the receivers on specified Unshielded Twisted Pair (UTP) ports and can control each port independently. Enabling or disabling the UTP port receivers has no effect on the port transmitters.



This command is only valid if a 4x4 or a 4x6 module is installed.

The syntax for this command is as follows:

```
portreceive|pr penable|pdisable <port-list>
```

where

- penable|pdisable** Enables or disables the port receiver on the specified port or port list.
- <port-list>** Specifies the port, port list, or range of ports to be enabled or disabled. This variable can be a port name, a comma separated list of ports, or a dash-separated range of ports.

The following example disables the port receivers on segments 4.1 through 4.4. This is followed by a configuration display of the port receiver status, indicating that all port receivers are disabled. Then the port receivers on segment 4.3 are enabled and the resultant port receiver configuration is again displayed, indicating that the receivers on segment 4.3 are enabled.

```
7:PowerHub:media# pr pdisable 4.1-4.4
Okay
8:PowerHub:media# config receive
UTP port receiver enable/disable status
Slot 4: ---- ---- ---- ----
Slot 2: . . . . .
Slot 1: . . . . .
9:PowerHub:media# pr penable 4.3
Okay
10:
UTP port receiver enable/disable status
Slot 4: ---- ---- YYYY ----
Slot 2: . . . . .
```

7.8 Displaying Port-Level Statistics

The `portstats` command is used to display statistics on 4x4 or 4x6 Ethernet Intelligent Network Interface Modules (INIMs). Before displaying port-level statistics, the `portstats enable` command must be issued. The syntax of the `portstats` command is as follows:

```
portstats [show] [<display-restrictor>]
portstats enable|disable
```

where

- [<display-restrictor>]** Optional display restrictors which can be used to specify the segment, list of segments, or range of segments for which statistics are to be displayed.
- enable|disable** Specifies whether to enable or disable statistics collection. Disabling statistics resets the counters to zero.



Enabling the collection of port-by-port statistics places an extra load on the PowerHub processors. It is recommended that port-by-port statistics be left disabled, except when this function is necessary.

7.9 Configuring Packet Forwarding on Segments

The **segment** command is used to enable or disable forwarding of packets on segments. Under certain circumstances it is desirable to disable the transmission and reception of packets on a specific segment. For example, when using the port monitoring feature (see Section 7.4) it is not desirable for the segment receiving the monitored packet to receive or transmit any other traffic. The syntax for this command is as follows:

```
segment penable/pdisable <segment-list>
```

where

<segment-list> Specifies a segment, dash-separated range of segments, or a comma-separated list of segments.

The following example disables segment forwarding on segment 1.16.

```
55:PowerHub:media# segment pdisable 1.16  
Segment 1.16: disabled  
56:PowerHub:media#
```

7.10 Segment Names

The `segmentname | name` command can be used to change the default port names of ports in the PowerHub. When the system first boots and assigns segments, segment numbers are assigned from bottom to top and the segments are named starting with “Port_1.” The syntax for this command is as follows:

```
segmentname | name sset <name> <seglist>
segmentname | name [show] [<seglist>]
```

where

- sset** Sets the segment name for the specified segment or segments.
- <name>** Specifies the name to use as a replacement for the default “Port_x” name. This variable is not required when using the **show** argument. The assigned name cannot exceed 23 characters or contain any spaces.
- <seglist>** Specifies the segment number of the segment to be renamed. This variable must be a single segment number when renaming a segment. This variable may be a dash-separated range or a comma-separated list when used with the **show** argument. When the **show** argument is used and the **<seglist>** variable is not used, all segments are displayed.

The following example renames segment 2.1 from Port_33 to Marketing, indicating that this segment is connected to the Marketing department.

```
49:PowerHub:media# segmentname sset Marketing 2.1
Segment 2.1 named: Marketing
51:PowerHub:media# segmentname show 2.1
Segment names:
2.1 : Marketing
52:PowerHub:media#
```

7.11 Segment-State Detection

The `ssd` command is used to set Segment State Detection on specified segments. The syntax for this command is as follows:

```
ssd [show] [<seglist>]
ssd penable|pdisable [<seglist>]
```

where

<seglist> Specifies the segment, or segments, to display the current segment state detection state. If no segments are specified, all segments are displayed.

penable|pdisable Specifies to either enable or disable segment state detection on the specified segment, or segments.

The following example displays the current state of segment 1.16 and then disables segment state detection on segment 1.16:

```
78:PowerHub:media# ssd 1.16
Automatic detection of ports state:
Segment 1.16: enabled (currently bad)
79:PowerHub:media# ssd pdisable 1.16
Segment 1.16 : disabled
80:PowerHub:media#
```

7.11.1 Automatic Segment-State Detection

Automatic segment-state detection recognizes if a segment is down and automatically disables bridging and routing on that segment. When it has been detected that the state of a segment has changed, the segment is disabled (taken out of service) and the software is marked to denote the change. The updated segment state is displayed when the `ssd` command is issued.



If automatic segment-state detection is disabled on a segment, the segment's state is always reported as "good" and interface states are always reported as "up." For information about the state of a segment or interface, enable automatic segment-state detection on that segment.

The method used to determine whether a segment is down differs depending upon the type of segment. Table 7.2 lists the methods used to determine the state of each type of segment.

Table 7.2 - Segment-State Detection Methods

Segment Type	Segment is Determined To Be Down If
10Base-FB	No link-test pulses are present on this segment.
10Base-FL	No link-test pulses are present on this segment.
100Base-FX	No data or idle symbols are being received on this segment.
100Base-TX	No data or idle symbols are being received on this segment.
ATM	The ELAN goes down or the physical link to the AMA goes down.
AUI	No packets are received and a “loss of carrier” is detected T times over a 1-second period, where T is specified by the <i><threshold></i> argument on the <code>ssdthreshold</code> command. Note that the PowerHub does not send test packets, but relies on client and network management traffic to detect carrier loss. The state is changed to “up” if at least one packet is received. When automatic segment-state detection is first enabled (for example, when the PowerHub is booted), each AUI segment begins in the “down” state but is changed to the “up” state as soon as it receives packets.
FDDI	The attachment configuration of the segment is “isolated.”
MAU	No AUI cable carrying +12-volt current (standard for AUI) is connected to the MAU.
UTP	No link-test pulses are present on this segment.

7.11.1.1 Software Behavior When Disabled

When a segment is disabled, no packets are bridged or routed on that segment. Bridging and routing do not occur regardless of whether the segment is disabled by automatic segment-state detection or by issuing the `segment disable` command (see Section 7.9).

7.11.1.2 Default Setting

The default setting for the automatic segment-state detection differs depending upon the segment type. Table 7.3 lists the default setting for each segment type.

Table 7.3 - Automatic Segment-State Detection Default Settings

Segment Type	Default
10Base-FB	Enabled

Table 7.3 - Automatic Segment-State Detection Default Settings

Segment Type	Default
10Base-FL	Enabled
10Base-T (UTP)	Enabled
100Base-FX	Enabled
100Base-TX	Enabled
ATM	Enabled
AUI	Disabled
FDDI	Enabled
MAU	Enabled

As shown in Table 7.3, all segment types except AUI and BNC/BNCT have automatic segment-state detection enabled by default. In general, automatic segment-state detection should be left at the factory default settings.

7.11.1.3 Disabled on AUI

When automatic segment-state detection is enabled, the AUI segments are not enabled until the segments receive traffic. In most configurations, the AUI segments are connected to devices that are prepared to generate traffic. However, connecting AUI segments to AUI segments on another PowerHub (if the other PowerHub has automatic segment-state detection enabled on these segments) results in no traffic exchanged by the segments. Each end of the segment waits to receive traffic before becoming enabled. As a result, neither end of the segment becomes enabled and no traffic is exchanged.

If the device at the other end of the AUI segment is prepared to generate traffic, enable automatic segment-state detection on the segment. When the segment receives traffic from the other device, the segment is enabled.

7.11.1.4 Segment-State Detection on 10Base-T

Automatic segment-state detection should be enabled for all 10Base-T (UTP) segments even if they are not going to be used. If automatic segment-state detection is disabled, the Ethernet controllers on the corresponding 10Base-T segments do not stop using the forwarding buffer for those segments. Instead, they fill their transmit buffers even though no traffic needs to be forwarded. Full buffers can negatively affect packet throughput.

7.11.1.5 Explicitly Disabling Unused Segments

For AUI, BNC, and BNCT segments, heuristics are used to determine the segment state. Occasionally, electronic noise can make an AUI, BNC, or BNCT segment appear active when it is not. When this occurs, automatic segment-state detection believes the segment is active and does not disable it. Accordingly, it is recommended that segments be explicitly disabled when removing them from service. See Section 7.9 for information about the `segment` command.

7.12 Segment-State Detection Threshold

The `ssdthreshold|ssdt` command sets segment-state thresholds for AUI and BNC segments. The syntax for this command is as follows:

```
ssdthreshold/ssdt sset <value> <seglist>
```

where

<value> For AUI segments, specifies the “loss-of-carrier threshold”; that is, the number of times a loss of carrier must be detected in a one-second period for the segment to be considered down and, therefore, to be disabled.

For BNC segments, specifies the “idle period threshold”; that is, the number of seconds during which the segment must remain idle to be considered down and therefore to be disabled by software.

<seg-list> Specifies the segment(s) for which automatic segment-state detection is to be enabled or disabled. If all is specified, the detection state is changed for all segments in the chassis.



Automatic segment-state detection must be enabled on all 10Base-T segments. The Ethernet controllers refuse to transmit packets on any segment that does not have a “good” link status. As a result, buffers can become “stuck” on the output queue of 10Base-T segments that do not have a “good” link status. This can adversely affect performance of the rest of the system. Enabling automatic segment-state detection, buffers can prevent buffers from being enqueued on the segments and allow any enqueued buffers to be recovered if the segments go down.

An example of setting the segment-state detection threshold for port 2.1 is shown below:

```
82:PowerHub:media# ssdt sset 10 2.1
Segment 2.1 : enabled (currently bad)
83:PowerHub:media#
```

This display shows information appropriate to each segment type. Because BNC segments are determined to be down if they are idle for the period specified by *<threshold>* (in this case, 5 seconds), their “idle period threshold” is shown. The “loss of carrier” threshold 10 seconds (in this case) is listed because AUI segments are determined to be down when a “loss of carrier” is detected the number of times specified by *<threshold>* in a one-second period.

The other types of Ethernet segments are determined to be down in the absence of regular link-test pulses, or data or idle symbols. FDDI segments are down if the attached configuration of the segment is “isolated.” ATM segments are down if the ELAN on the segment goes down or the physical link to the PHY goes down. In these cases, no threshold is shown.

7.14 Statistics

The **stats** command is used to display media-level statistics. If port statistics are being collected (see Section) they are displayed in place of segment-wide statistics unless the **-s** flag is specified. Segment-level statistics can be cleared but not disabled. When segment-level statistics are cleared, they are reset to zero and immediately begin to increment as packets are received and sent. The syntax of the **stats** command is as follows:

```
stats [show] [-p|-s] [<params>] [<display-restrictor>]
stats clear
```

where

[-p|-s] Optionally display either port or segment level statistics, if available. If one of these options are not specified, both port and segment statistics, if available, are displayed.

[<params>] Specifies a comma-separated list of the following statistics or “all” if none are specified:

pi, po, oi, oo, bpi, bpo, pu, rbe, xbe, fcs, fa, c, rc, tc, q, gp, cu, lc, er, tm

Table 7.4 defines the above parameters.

Table 7.4 - Segment Level Statistic Parameters

Parameter	Description
pi	packets in
po	packets out
oi	octets in
oo	octets out
bpi	broadcast packets in
bpo	broadcast packets out
pu	peak utilization
rbe	receive buffer errors
xbe	transmit buffer errors
fcs	frame check sequence errors
fa	frame alignment errors
c	segment collisions

Table 7.4 - Segment Level Statistic Parameters

Parameter	Description
rc	port collisions
tc	transmit collisions
q	output queue length
gp	giant packets
cu	current utilization
lc	local carrier
er	excessive retries
tm	table miss

[<display-restrictor>] Optionally specify a specific segment, range of segments, or a comma-separated list of segments.

clear If specified, clears all port and segment-level statistics.

The following examples show the display received when requesting the peak utilization of statistics:

```
30:PowerHub:media# stats -s pu
```

```
Peak util
```

```
4.1 :          0%

3.1 :          43%          0%          18%          43%          0%          18%

2.1 :          20%          6%          6%          6%          6%          6%
      6%          6%          6%          6%          7%          6%
      6%          6%          6%          6%

1.1 :          0%          0%          0%          0%          0%          0%
      0%          0%          0%          0%          0%          0%
      0%          0%          0%          0%          0%          0%
      0%          0%          0%          0%          0%          0%
      0%          0%
```

CHAPTER 8

NVRAM Commands

This chapter explains the commands within the NVRAM subsystem which are used to make changes in the NVRAM subsystem and booting parameters. The following commands are available in the NVRAM subsystem:

nvr_{am} subsystem:

```
bo                               gwip
myip                             crashreboot
mysm                             slotsegs[n]
fsip                             md5key[keyid]
```

8.1 NVRAM Configuration Commands

The NVRAM commands are used to display, set, or clear parameters that affect the PowerHub boot order, IP addresses of the system, file server and gateway, the system behavior in the event of a system crash, the number of segments each installed module is to support, and a RIPv2 key (password).

8.1.1 Boot Order

The **bo** command is used to display, set or clear the default booting order. The syntax of this command is as follows:

```
bo set <value>
bo [show]
bo unset
```

where

- | | |
|----------------------|--|
| set | Sets the specified boot order. The boot order designates the order of sources from which the system attempts to boot. |
| <value> | Specifies the actual boot order. Values can be f (Floppy Diskette in the PowerHub 7000), or c (Compact Flash Card in the PowerHub 8000), m (Flash Memory Module) or n (network via a tftpboot server). The boot order can be set to any order. |

unset Unsets the specified boot order. Causes the boot order to default to the Floppy Diskette Compact Flash Card.

If more than one boot source is specified, the system attempts to boot in the order specified. For example: if **fm** is entered, the system attempts to boot from the Floppy Diskette (Compact Flash Card in the PowerHub 8000) and then the Flash Memory Module in the PowerHub 7000).



If more than one boot source is specified, the runtime and configuration files on each should match to prevent an erroneous configuration from being loaded.

The following examples show the results of the various boot order command options:

```
15:PowerHub:nvram# bo
bo                (not set, defaults to "f")
16:PowerHub:nvram# bo set fm
17:PowerHub:nvram# bo
bo                fm (floppy,flash-module)
18:PowerHub:nvram#
```

8.1.2 My Internet Protocol Address

The **myip** command is used to display, set or clear the IP address of the PowerHub. The syntax of this command is as follows:

```
myip [show]
myip set <ipaddr>
myip unset
```

where

- set** Sets the IP address assigned to the PowerHub.
- <ipaddress>** Specifies the IP address assigned to the PowerHub.
- unset** Unsets and clears the IP address currently assigned.

The following examples display the current assigned IP address and finds that it is unset. The IP address is then set and displayed to verify it was properly set.

```

21:PowerHub:nvram# myip
myip                (not set)
22:PowerHub:nvram# myip set 169.144.86.54
23:PowerHub:nvram# myip
myip                169.144.86.54
24:PowerHub:nvram#

```

8.1.3 My Subnet Mask

The **mysm** command is used to set the subnet mask of the system. The syntax for this command is:

```

mysm [show]
mysm set <ipaddr-mask>
mysm unset

```

where

set	Sets the IP subnet mask for the system.
<ipaddr-mask>	Specifies the IP subnet mask.
unset	Unsets and clears the IP subnet mask.

The following examples display the current assigned subnet mask and finds that it is unset. The subnet mask is then set and displayed to verify it was properly set.

```

28:PowerHub:nvram# mysm
mysm                (not set)
29:PowerHub:nvram# mysm set 255.255.255.0
30:PowerHub:nvram# mysm
mysm                255.255.255.0
31:PowerHub:nvram#

```

8.1.4 File Server IP Address

The **fsip** command is used to display, set, or clear the file server IP address associated with the system. The syntax for this command is as follows:

```

fsip [show]
fsip set <ipaddr>
fsip unset

```

where

set	Specifies whether to show, set, or clear the IP address of the file server.
------------	---

<ipaddress> A file server's IP address.

unset

The following examples display the current file server IP address and finds that it is unset. The file server IP address is set and displayed to verify it was properly set.

```
36:PowerHub:nvram# fsip
fsip          (not set)
37:PowerHub:nvram# fsip set 169.144.86.49
38:PowerHub:nvram# fsip
fsip          169.144.86.49
39:PowerHub:nvram#
```

8.1.5 Gateway IP Address

The **gwip** command is used to display, set or clear the IP address of the gateway router. The syntax for this command is as follows:

```
gwip [show]
gwip set <ipaddr>
gwip unset
```

where

set Sets the IP address of the gateway router.

unset Unsets and clears the IP address of the gateway router.

<ipaddress> Specifies the IP address of the intervening router (gateway).

The following examples display the current assigned subnet mask and finds that it is unset. The subnet mask is then set and displayed to verify it was properly set.

The following examples display the use of these commands. In the first example, values are configured into the hub's NVRAM to support "semi-prescient" netbooting.

```
45:PowerHub:nvram# gwip
gwip          (not set)
46:PowerHub:nvram# gwip set 169.144.86.1
47:PowerHub:nvram# gwip
gwip          169.144.86.1
48:PowerHub:nvram#
```

8.1.6 Crash Reboot

The **crashreboot** command is used to instruct the system to reboot automatically following a system crash. The syntax of this command is as follows:

```
crashreboot [show]
crashreboot set
crashreboot unset
```

where

- set** When set, the PowerHub automatically attempts a reboot following an unexpected system crash. The default is **set**, which causes a reboot to be attempted following a system crash. Do not change this setting unless instructed to do so by FORE Systems TAC.
- unset** Clears the crash reboot behavior. If this parameter is not set, the system pauses at the Boot PROM prompt (<prompt-7PE>) and waits for manual intervention.

```
52:PowerHub:nvram# crashreboot
crashreboot      (not set)
53:PowerHub:nvram# crashreboot set
54:PowerHub:nvram# crashreboot
crashreboot      (set)
55:PowerHub:nvram#
```

8.1.7 Slot Segments

The **slotsegs** command is used to allocate segments to specific slots. Slot segments are set by default and there is no need to manually set them. However, since the larger chassis can be populated with NIMs totalling more segments than the recognized chassis maximum, it may be necessary to manually unset segments that are not in use in order to ensure that all segments in use are allocated. The maximum allowable segment count for a 5-slot PowerHub 7000/8000 is 96 segments. The 10- or 15-slot PowerHub 7000/8000 has a maximum segment count of 128 segments. The syntax for this command is as follows:

```
slotsegs [show]
slotsegs[<n>] [show]
slotsegs[<n>] set <segment-count>
slotsegs[<n>] unset
```

where

- [<n>]** Specifies the slot number for which segments are being allocated. The brackets are required around the slot number.
- set** Sets the specified number segments for the specified slot.
- <segment-count>** Specifies the number of segments being allocated to the specified slot.
- unset** Clears the segment count for the specified slot.

In the following example, a 5-slot PowerHub 7000 has segments allocated as follows:

- Slot 1 has thirty-two segments allocated (PowerCell 700 ATM module).
- Slot 2 has sixteen segments allocated (16x1 Ethernet Module).
- Slot 3 has two segments allocated (Single FDDI module).
- Slot 4 has six segments allocated (Universal 6x1 Ethernet Module).
- Slot 5 is unset and contains the Packet Engine. The Packet Engine slot could also be set to zero (0) segments.

Following the segment allocation display, the total allocated (reserved) segment count for the chassis is displayed. Since this is a 5-slot chassis, the number of reserved segments is less than the maximum allowable segment count of 96. The segment count for the ATM PowerCell in slot 1 is then decreased to sixteen segments and the setting is then verified.

```
57:PowerHub:nvram# slotsegs
slotsegs[ 1]    32
slotsegs[ 2]    16
slotsegs[ 3]     2
slotsegs[ 4]     6
slotsegs[ 5]   (not set)
slotsegs[ 6]   (not set)
slotsegs[ 7]   (not set)
slotsegs[ 8]   (not set)
slotsegs[ 9]   (not set)
slotsegs[10]   (not set)
slotsegs[11]   (not set)
slotsegs[12]   (not set)
slotsegs[13]   (not set)
slotsegs[14]   (not set)
slotsegs[15]   (not set)
slotsegs[16]   (not set)
slotsegs[17]   (not set)
slotsegs[18]   (not set)
slotsegs[19]   (not set)
slotsegs[20]   (not set)
Total segments reserved: 56
```

```
58:PowerHub:nvram# slotsegs[1] set 16
59:PowerHub:nvram# slotsegs[1]
slotsegs[ 1]    16
Total segments reserved: 40
60:PowerHub:nvram#
```



If an INIM has zero (0) slotsegs configured, the software image for that INIM will not be loaded during the booting process and the INIM will appear to be dead or bad. Setting a segment value for the INIM using the `slotsegs[<n>] set <segment-count>` command and then rebooting loads the image.

8.2 RIPv2 Authentication

RIPv2 supports encrypted packet transmission using the MD5 algorithm to authenticate route and table updates. The MD5 algorithm allows packets to be encrypted at a source PowerHub and decoded at a destination PowerHub containing the same encryption key and key-string (password). Because the keyID is not transmitted over the network but is set at each end, it reduces the likelihood of a successful attack on the network.

MD5 authentication is only supported in RIPv2. It does not work in RIPv1. RIPv2 must be enabled and running on all interfaces that require authentication. Additionally, RIPv2 authentication is not supported on interfaces that are configured for both RIPv1 and RIPv2; interfaces must be configured for RIPv2 only.

The MD5 key must be set up on the PowerHubs at both sides of the connected interfaces in order for the authentication to take place. The keyid and the key-string must be the same on both PowerHubs. Refer to RFC-2082 for a discussion on RIPv2 authentication using the MD5 encryption algorithm. The syntax for the `md5key` command is:

```
md5key [show]
md5key[<keyid>] [show]
md5key[<keyid>] set <key-string>
md5key[<keyid>] unset
```

where

[keyid] Specifies the number, or identifier, of the MD5key. The number must be a whole number between 1 and 255. There is no space between `md5key` and the keyid when the command is entered. Brackets around the keyid are part of the command and must be included.

set Sets a specific keyid.

unset Unsets a specific keyid.

<key-string> Specifies the password to be used for encryption. The maximum password length is 16 characters.

unset

Examples of the `md5key` command are shown below:

```
65:PowerHub:nvram# md5key[1] set powerhub
66:PowerHub:nvram# md5key
md5key[ 1]      (set)
Total keys reserved: 1
```

```
67:PowerHub:nvram# md5key[1]
md5key[ 1]      (set)
Total keys reserved: 1
68:PowerHub:nvram#
```

In this example, key 1 is set with the keyID (password) of “powerhub.” This is the only time the keyID is displayed.

NVRAM Commands

CHAPTER 9

Host Commands

This chapter describes the commands in the `host` subsystem and discusses how to use these commands to perform the following tasks:

- Display the TCP configuration settings.
- Display the TCP table.
- Display TCP, TELNET, and UDP statistics.
- Clear TCP, TELNET, and UDP statistics.
- Set the connection time.
- Set the keep-alive interval.
- Kill a TCP connection.
- Display the UDP table.

The `host` subsystem includes an implementation of the Transmission Control Protocol (TCP) stack, a connection-oriented, industry-standard protocol for moving data between nodes in a network environment. In particular, TCP is used by TELNET, a program that allows workstations to communicate using either an in-band or outbound network connection. To define TCP filters, refer to the *PowerHub 7000/8000 Filters Manual*.

9.1 Accessing the Host Subsystem

To access the `host` subsystem, issue the following command at any runtime command prompt:

```
host
```

The following commands are available in the host subsystem:

```
config                kill
filter               stats
kainterval|kai       status
kadelay|kad          template
```

The filter and template commands are discussed in the *PowerHub 7000/8000 Filters Reference Manual*.

9.2 Displaying the Configuration

The `config` command is used to display configuration parameters used by the host subsystem. The syntax for this command is:

```
config [show] tcp|fi[lters]|ru[les]|tem[plates] [<disprestrictions>]
```

where

`tcp|fi[lters]|ru[les]|tem[plates]` Specifies whether to display the TCP configuration, configured host filters, rules or templates.

`<disprestrictions>` Optionally, the configuration information pertinent to a specific segment, or segment list, can be displayed. If no restrictions are supplied, the specified configuration is displayed for all segments.

The following example shows the displays associated with each of the required command line arguments.

```
104:PowerHub:host# config tcp
TCP Configuration
```

```
-----
Round Trip Algorithm:          vanj
Min Rexmit Interval:          1000 ms
Max Rexmit Interval:          64000 ms
Max Connections Allowed:      2
connection-idle-time:         20 minutes
keep-alive interval [kainterval]: 75 seconds
keep-alive delay [kadelay]:    1200 seconds
Time to disconnect on idle conn: 30 minutes 0 seconds
```

```
105:PowerHub:host# config fi
Host Filter Template Definitions
Filter    Templates
```

```
-----
Host Receive Filter attachments
Segment   Filter
```

```
107:PowerHub:host# config ru
Host Filter Template Definitions
Filter    Templates
```

```
108:PowerHub:host# config tem
```

```
T#          Source IP address/mask      Destination IP address/mask  ipproto
TCP/UDP source port      dest port      TCP conreq      Action
=====
```

Host Commands

The first example displays the following information about the current TCP configuration parameters:

- The round-trip algorithm used is the Van Jacobson algorithm.
- The minimum retransmit interval is 1,000 milliseconds.
- The maximum retransmit interval is 64,000 milliseconds.
- The maximum number of simultaneous TELNET (TCP) connections that can be supported is two.
- The connection-idle time is 20 minutes.
- The keep-alive interval is 75 seconds.
- The keep-alive delay is 1200 seconds.
- The time allowed before an idle connection is automatically disconnected is 30 minutes. This value is based on the values of the connection-idle time and the keep-alive interval.
- The other examples display any configured host filters, rules or templates. This particular system has no filters, rules, or templates configured.

9.3 Keep Alive Delay

The `kadelay|kad` command is used to specify how long a TELNET connection can remain idle before keep-alive packets are sent. The keep alive delay is specified in seconds. The syntax for this command is as follows:

```
kadelay|kad set <time>
```

where

<time> Specify in seconds the number of minutes to allow a TCP (TELNET) connection to remain idle before sending keep-alive packets. The range is 5 to 30 minutes; the default is 20 minutes.

The following example sets the keep alive delay to six minutes (360 seconds).

```
120:PowerHub:host# kad set 360  
121:PowerHub:host#
```

9.4 Keep Alive Interval

The `kainterval|kai` command is used to specify how often keep-alive packets are sent before a connection is closed. The syntax for this command is as follows:

```
kainterval|kai set <time>
```

where

<time> Specifies how often keep-alive packets are sent before a connection is closed. The range is 30 to 240 seconds; the default is 75 seconds.

The following example sets the keep alive interval to 180 seconds, or 3 minutes.

```
122:PowerHub:host# kai set 180
123:PowerHub:host#
```

9.5 Ending (Killing) a TCP Connection

The `kill` command is used to end a TCP connection other than the active session. This command must be issued from a session other than the active one. The syntax for this command is as follows:

```
kill <connection-id>
```

where

<connection-id> Specifies the ID assigned to the session when the session was established. To determine what the connection ID is, use the `status tcp` command to display the active TCP connections. The connection IDs are listed under Conn ID.

The following example checks the current TCP connections, using the `status tcp` command, and then kills the connection labeled Conn ID 16.

```
192:PowerHub:host# status tcp
```

```

Active TCP Connections
Conn Id  Rem IP Addr      Rem Port  Loc IP Addr  Loc Port  Conn. State
-----  -
16      169.144.86.49    23        169.144.86.54 1494      ESTABLISHED

```

```
193:PowerHub:host# kill 16
```

```
194:PowerHub:host#
```

9.6 Statistics

The **stats** command is used to display or clear statistics on TCP, TELNET, and UDP packets. TCP and UDP statistics are a superset of the corresponding statistics provided in the SNMP MIB. (There is no TELNET MIB.) The software maintains two types of stats for TCP, TELNET, and UDP statistics counter:

- Count since last statistics clear.
- Count since last system reset.

The syntax for this command is as follows:

```
stats clear [-i] [-t] tcp|tel[net]|udp|all
stats [show] [-i] [-t] tcp|tel[net]|udp|all
```

where

clear	Specifies to clear the specified statistics or all statistics.
-i	Valid only for TCP. Displays, or clears, TCP statistics such as connections established, dropped, closed, etc.
-t	Displays total statistic count since last system reset of the specified protocol, or all, if specified.
tcp tel[net] udp	Specifies for which type of protocol to clear the statistics. If all is specified, statistics for all protocols are displayed or cleared.

The following example displays the statistics for all protocols, TCP, Telnet and UDP.

```
144:PowerHub:host# stats
Telnet Data Statistics (count since last stats clear):
Pkts Rcvd From Net:          2308
Pkts Sent To Net:            17070
Bytes Rcvd From Net:         2933
Bytes Sent To Net:           45790
Bytes Rcvd From CLient:      43303
Bytes Sent To Client:        2556
Conn Opens Rcvd:              9
Conn Rejects Sent:           0
Conn Aborts Sent:            0
Conn Aborts Rcvd:            1

TCP Connection & Pkt statistics (count since last stats clear):
Active Opens:                 4
Passive Opens:                9
Failed Conn Attempts:         13
```

```
Resets In Estb State:          0
Current Open Conns:           4
Segments Received:            6402
Segments Sent:                 7567
Rexmitted segments:           0
Segments Rcvd With Err:       1
Resets Sent:                   37
Short Segments Rcvd:          0
```

UDP statistics: count since last stats clear

```
Datagrams received:           738
Unknown destination ports received: 11
Errors received:               0
Datagrams discarded:           0
Datagrams sent:                746
```

145:PowerHub:host#

9.7 Status

The **status** command is used to display active TCP and/or UDP connections. The syntax for this command is as follows:

```
status [show] tcp|udp
```

where

tcp|udp Specifies to display either the TCP or UDP active connections. If neither is specified, both are displayed.

Following is an example of the TCP table:

```
197:PowerHub:host# status
```

```

                Active TCP Connections
Conn Id  Rem IP Addr  Rem Port  Loc IP Addr  Loc Port  Conn. State
-----  -
17       169.144.86.49  23       169.144.86.54  1495     ESTABLISHED **
List of registered UDP clients:
161 snmp
520 rip
198:PowerHub:host#
```

For each TCP connection, the following information is displayed:

Conn ID	A unique integer that identifies the connection. This identifier can be used to terminate the connection using the kill <connection-id> command.
Rem IP Addr	The IP address of the remote device that initiated the connection.
Rem Port	A process port number for the remote device (management station). Note that the process port number is unrelated to the physical port or segment numbers. It is assigned by the remote operating system.
Loc IP Addr	The IP address of the local device. This is always the PowerHub.
Loc Port	A process port number. Unrelated to the physical port or segment numbers. It is a “well-known” port number used by the TELNET process.

Conn. State The connection state of the standard TCP state machine:

CLOSEDCLOSING
 CLOSE-WAITESTABLISHED
 FIN-WAIT-1FIN-WAIT-2
 LAST-ACKLISTEN
 SYN-RECEIVEDSYN-SENT
 TIME-WAIT

Most of these states are never displayed by the `status tcp` command because they occur for a very brief time. Connections in the CLOSED or LISTEN state are not displayed.

The current TELNET session (if connected through TELNET) is indicated by two asterisks (**) following the table entry for that session.

The status information for UDP clients includes the number of registered SNMP and RIP clients. The numbers and names are “well-known” UDP protocol port numbers and names as defined in RFC 1700.

The UDP ports listed in this display indicate that agents for processing UDP packets are sent to UDP protocol ports 161 and 520. In other words, the following types of UDP packets are supported:

- SNMP
- IP RIP

Host Commands

CHAPTER 10 Bridge Commands

The PowerHub contains implementations of IEEE 802.1d bridging and the 802.1d Spanning-Tree protocol. This chapter describes the `bridge` subsystem commands that can be used to perform the following tasks:

- Display the bridge configuration
- Display and manage the bridge table (includes changing the aging interval for dynamic (learned) entries)
- Display, add, and delete bridge groups
- Display the bridging status of a segment
- Enable, disable, and configure Spanning-Tree Protocol
- Display or clear packet, bridge, and segment statistics
- Display and clear the bridge cache

10.1 Accessing the Bridge Subsystem

To access the `bridge` subsystem, issue the following command from any command prompt:

bridge

The following commands are located in the bridge subsystem:

aging	ipx-br-translation ibt
bridging br	learning learn
bt	lrule
cache	relearn-log rl
config	spantree st
filter	stats
getmem	status
group	template

Commands related to the configuring of filters; `filter`, `lrule`, and `template` are discussed in the *PowerHub 7000/8000 Filters Reference Manual*.

10.2 Aging

The **aging** command is used to set or unset (disable) the bridge table aging time. Aging is a mechanism that periodically clears learned entries from the table. Only dynamic entries (entries learned and not configured manually) are aged by the software. Static entries (those created by the user) do not age.

At the interval specified (the aging interval), the software determines which of the learned entries in the table have not been used recently. Each learned entry that has not been used during the specified interval is marked aged. This value shows up in the Flags column of the bridge table.

If an entry marked aged is used during the next aging interval, the aged flag is removed and the entry remains in the table. However, if an entry marked aged is unused during the next interval, the entry is removed from the table. The syntax for this command is as follows:

```
aging set <time>
aging unset
```

where

<time> Specifies the aging time to clear learned entries in seconds. Aging time must be specified in integrals of 60 seconds. Default is 60 minutes (3600 seconds).

In the following examples, the aging time is displayed (60 minutes), then disabled (unset) and set to 30 minutes.

```
9:PowerHub:bridge# aging
Bridge table aging time: 60 minutes
10:PowerHub:bridge# aging unset
Aging time specified is short. Shorter Aging time
may affect Powerhub performance
Bridge Table aging turned off
11:PowerHub:bridge# aging set 1800
Bridge Table aging time set to 30 minutes
12:PowerHub:bridge#
```

10.3 Bridging

The `bridging|br` command is used to display, enable and disable bridging. If the command is entered without an argument, the current bridging status is displayed. The syntax for this command is as follows:

```
bridging|br [show]
bridging|br pen[penable] <segment-list>|all
bridging|br pdis[able] <segment-list>|all
```

where

pen[penable] Enables bridging on the specified segment, or segment list.

pdis[able] Disables bridging on the specified segment or segment list.

<segment-list>|all Specifies the segments on which to enable or disable bridging. Specify one segment, a comma-separated list of segment, and/or ranges of segments. If **all** is specified, bridging is enabled or disabled on all available segments.

The following example displays the bridging status of all segments:

```
14:PowerHub:bridge# br
Port      Bridging Status
1.1       Disabled
1.2       Disabled
1.3       Disabled
1.4       Disabled
1.5       Disabled
1.6       Disabled
1.7       Disabled
1.8       Disabled
1.9       Disabled
1.10      Enabled
1.11      Enabled
1.12      Disabled
1.13      Enabled
1.14      Enabled
1.15      Enabled
1.16      Enabled
2.1       Disabled
2.2       Disabled
2.3       Disabled
2.4       Disabled
2.5       Disabled
```

```
2.6      Disabled
2.7      Disabled
2.8      Disabled
2.9      Disabled
2.10     Enabled
2.11     Enabled
2.12     Enabled
2.13     Enabled
2.14     Enabled
2.15     Enabled
2.16     Enabled
4.1      Disabled
15:PowerHub:bridge#
```

The following example disables bridging on segments 1.10, 1.11 and 1.13 through 1.16:

```
28:PowerHub:bridge# br pdis 1.10,1.11,1.13-1.16
29:PowerHub:bridge#
```

10.4 Bridge Table

The `bt` command is used to add, delete, display or clear bridge table entries. The bridge table contains information about attached devices. Entries in the bridge table are used to bridge packets. Entries can be added to the table automatically or manually.

Each time the bridging engine receives a packet, it checks the packet's source address against the MAC addresses listed in the bridge table. If the address is not listed in the table, an entry is added to the table. The entry contains the source device's MAC address, the segment number on which the packet was received, and other information used for bridging.

Static entries are created using the `add` argument. A static entry is manually added to the bridge table, rather than learned by the bridge table. Static entries are not subject to aging and remain in the bridge table until removed. Moreover, they are saved in the configuration file when the configuration is saved (see Chapter 6). The syntax for this command is as follows:

```
bt add <ethaddr> <seglist>
    bt delete <ethaddr>
bt [show] [-h] [-m] [-t] [<disprestrict>]
    bt clear
```

where

- | | |
|-------------------------------|--|
| <ethaddr> | Specifies the MAC-layer address of the device to add or delete bridge-table entries. Specify the address as six hyphen-separated two-digit hexadecimal octets (ex: <code>08-00-20-0f-a5-ab</code>). |
| <seglist> | Specifies the segment or segments to add to the bridge table. |
| [-h] | Displays the hash displacements for the specified entries. |
| [-m] | Displays entries for multi-homed hosts. |
| [-t] | Displays the total number of entries in the table. The total is comprised of all learned and permanent (static) entries. This argument also shows how many entries remain available in the bridge pool; that is, the number of entries for which the table still has room. |
| [<disprestrict>] | Optional display restrictions of
<code>a[ddr]=<ethaddr> <ethpat></code>
<code>[[seg[ment[s]]]=<seglist></code> |


NOTE

Because the **-h** and **-m** options display specific entries in the bridge table, they cannot be used with the **-t** option, which displays total bridge entries.

The following examples clear the bridge table entries. The bridge table is then displayed in its entirety and then with the **-t** and **-h** optional arguments.

```
14:PowerHub:bridge# bt clear
Ok
15:PowerHub:bridge# bt

Bridging table (aging time = 60 minutes)
Ethernet-address  Seg  Rule  Flags
00-00-ef-03-9a-b0  --  none  system permanent
00-20-48-08-8f-85  2.1  none
00-20-48-04-ef-a7  2.1  none
ff-ff-ff-ff-ff-ff  --  none  permanent bmcast

Total entries: 4, Learned entries: 2, Permanent Entries: 2
16:PowerHub:bridge# bt -t

Total entries: 6, Learned entries: 4, Permanent Entries: 2
Total entries in free pool 8186
17:PowerHub:bridge# bt -h

Bridging table (aging time = 60 minutes)
Ethernet-address  Seg  Rule  Flags
Hash: 273, collision displacement: 0
00-a0-98-00-09-d3  2.1  none
Hash: 319, collision displacement: 0
08-00-20-1f-fa-fa  2.1  none
Hash: 9b3, collision displacement: 0
00-00-ef-03-9a-b0  --  none  system permanent
Hash: 1594, collision displacement: 0
00-00-ef-04-86-90  2.1  none
Hash: 17ad, collision displacement: 0
00-20-48-08-8f-85  2.1  none
Hash: 17cb, collision displacement: 0
00-20-48-04-ef-a7  2.1  none
Hash: 1ffd, collision displacement: 0
ff-ff-ff-ff-ff-ff  --  none  permanent bmcast

Total entries: 7, Learned entries: 5, Permanent Entries: 2
18:PowerHub:bridge#
```

The bridge table contains the following information for each entry:

Ethernet-address The MAC-layer hardware address of the device.

Seg (Segment)	The segment to which the network joining the device is attached. If the MAC-layer hardware address belongs to a multi-homed host, the segment number is shown as MH.
Rule	The number of a logical filtering rule applied to packets forwarded to or from this address. Refer to the <i>PowerHub 7000/8000 Filters Reference Manual</i> for information about defining rules.
Flags	<p>Certain flags are maintained in order to use and manage addresses in the bridge table. For example, entries such as the address of the PowerHub are marked, and entries that haven't been used recently are flagged for possible deletion (aging).</p> <p>Each entry in the bridge table can have one or more of the following flags:</p> <p>bmcastA broadcast/multicast address.</p> <p>permanentMost often, this flag indicates that the address is a static entry. Otherwise, it is a switch-defined entry.</p> <p>spanning-treeThe industry-standard (IEEE 802.1d) multicast address used by the Spanning-Tree algorithm.</p> <p>systemThe factory-configured MAC-layer hardware address of the PowerHub.</p> <p>blankIn a typical application, most entries in the bridge table have none of the preceding flags set. Such entries are learned addresses that have been seen at least once since the last time the bridge table was aged.</p>

10.5 Cache

The **cache** command is used to display or clear bridge cache entries. Each time the bridging engine bridges a packet, it creates an entry in the bridge cache containing the packet's destination and source Ethernet MAC address. The bridge cache is frequently updated with the most recently used source-destination pairs and provides a fast path for bridge traffic resulting in increased performance. The bridge cache can be used for at-a-glance information about the current bridge traffic in the network. The syntax for this command is as follows:

```
cache [show] [<disprestrict>]
      cache clear
```

where

[<disprestrict>] Specifies the segments for which to display the cache entries. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

The following example shows a brief list of cache entries.

```
17:PowerHub:bridge# cache
Bridging cache:
Port 01: Dest: 08-00-20-08-70-54, Source: 08-00-20-0f-dd-99
        Dest: 00-00-6b-82-3f-34, Source: 08-00-20-0f-6c-96
        Dest: 08-00-20-08-85-69, Source: 08-00-20-0f-dd-99
        Dest: 08-00-20-08-70-54, Source: 08-00-20-0f-6c-96
Port 02: Dest: 00-00-6b-82-3f-34, Source: 08-00-20-0e-ae-03
        Dest: 00-00-94-06-79-12, Source: 08-00-20-10-56-53
        Dest: 08-00-20-08-85-69, Source: 00-00-6b-82-3f-34
        Dest: 08-00-20-08-70-54, Source: 08-00-20-0e-ae-03
```

Listing continues

```
Port 21: empty
Port 22: empty
Port 23: empty
Port 24: empty
18:PowerHub:bridge#
```

10.6 Configuration

The `config` command is used to display the bridge configuration parameters. The syntax for this command is as follows:

```
config [show] [<params>] [<disprestrict>]
```

where

- <params>** Specifies the configuration parameters to display. Specify an argument, a comma-separated list of arguments, or `all` for all arguments. Table 10.1 lists the arguments that can be specified. The default is `all`.
- <disprestrict>** Specifies the segment or segment list to display the configuration information

Table 10.1 - Configuration Arguments

Argument	Description
<code>vars</code>	The aging time for entries in the bridge table. This also shows if learning is enabled.
<code>groups</code>	The currently defined network bridge groups.
<code>templates</code>	All defined logical filtering templates (refer to <i>PowerHub 7000/8000 Filters Reference Manual</i>).
<code>rules</code>	All defined logical filtering rules (refer to <i>PowerHub 7000/8000 Filters Reference Manual</i>).
<code>filters</code>	The packet-forwarding restrictions for all segments. This includes the source and destination logical filtering rules and whether or not learned entries are blocked (refer to <i>PowerHub 7000/8000 Filters Reference Manual</i>).
<code>st</code>	All configured Spanning-Tree Algorithm parameters.

The following example shows the type of information displayed by the `config` command when issued without arguments. Some areas have been shortened for brevity.

```
347:PowerHub:bridge# config
Spanning Tree
Status :                Disabled
System Priority :       8000
Spanning Tree Address : 01:80:c2:00:00:00
My Bridge Address :    00:00:ef:03:9a:b0
```

```

Max Age :                21
Hello Time :             4
Forward Delay :         16
Sending Fast Hellos :   Disabled
Fast Hello Params :     Hello Time: 1 sec, High Util: 70%, Low Util: 50%
    
```

Segment	Prio	Path Cost	Designated Bridge	Des Seg	Des Cost	Sta Chngs
1.1	80	-	-	-	-	-
.						
4.1	80	-	-	-	-	-

```

Bridge learning
segment 1.1: on
.
Segment 4.1: on
    
```

Bridge table aging time: 60 minutes

Bridge Groups:

Name	Segment List
default	1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.14, 1.15, 1.16, 1.17, 1.18, 1.19, 1.20, 1.21, 1.22, 1.23, 1.24, 1.25, 1.26, 1.27, 1.28, 1.29, 1.30, 1.31, 1.32,

Filter templates

Number	Offset(dec)	Mask(hex)	Comparator(hex)
099	004	00000000	00000000

Filter rules

Number	Description
163	99

Filters applied

Segment	Transmit	Receive
1.1	-	-
.		
4.1	-	-

348:PowerHub:bridge#

10.7 Allocate Memory

The `getmem` command is used to allocate memory for bridge table MIB processing. The syntax for this command is as follows:

```
getmem [br]mib
```

The following examples shows the use of this command.

```
392:PowerHub:bridge# getmem mib
Memory allocated for Bridge table MIB processing.
393:PowerHub:bridge#
```

10.8 Bridge Groups

The **group** command is used to define (set) or clear (unset) network groups. Network groups are a specific subset of network segments among which packets can be bridged, creating a Layer-2-only VLAN. A packet from one segment in the network group can be bridged only to the other segments in the network group. Up to 32 network groups can be defined. Group membership can overlap segments and each segment can belong to all, some, or none of the network groups.

As shipped from the factory, the bridging engine contains one network group known as **default**. All attached segments automatically belong to this network group. The group is added to the configuration file when the configuration is saved (see Chapter 6).



When the configuration file is saved, the default group is automatically added to the configuration file. If the configuration requires that not all segments belong to a common network group (for example, if groups were defined with restricted sets of segments), be sure to delete the default group before saving the configuration file.

The syntax for this command is as follows:

```
group pset <groupname> <seglst>
group punset <groupname>
```

where

- <groupname>** Specifies the name of the network group. Specify any alphanumeric string up to 15 characters in length.
- <seglst>** Specifies the segment(s) that belongs to the network group. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments. If **all** is specified, all segments are added to the network group.



To create a new **default** group, specify **all** or list all the segments as the **<seglst>**. If a **<seglst>** is specified instead of **all**, and the **<seglst>** does not include all the segments, a network group called **old_default** is created. This default group is stored in the configuration file when saved.

The following example creates a bridge group called **pubs** using segments 2.1 through 2.6:

```
57:PowerHub:bridge# group pset pubs 2.1-2.6
Group pubs with ports 2.1-2.6: added
58:PowerHub:bridge#
```

10.9 IPX Bridge Translation

The `ipx-br-translation|ibt` command is used to configure one or more IPX networks that span across FDDI and Ethernet segments using different packet encapsulations. Without altering the configurations of individual devices, IPX translation bridging enables Ethernet and FDDI devices with different encapsulation types to communicate with each other. This feature is especially useful if the IPX network consists largely of Ethernet devices using 802.3 encapsulation, the default encapsulation type in Novell IPX software versions 2.2 through 3.11.¹ However, if the network name is not in the IBT table, IPX translation bridging does not occur and normal bridging does. This section describes the commands that are used to display, add, delete, enable or disable IPX translation bridging. The syntax for this command is as follows:

```
ipx-br-translation|ibt [show] [<network>]|[-t]
ipx-br-translation|ibt add <network> <ethernet-encap> <fddi-encap>
ipx-br-translation|ibt delete <network>|all
ipx-br-translation|ibt enable
ipx-br-translation|ibt disable
```

where

- | | |
|-------------------------------|---|
| [<network>] [-t] | Optionally specifies the IPX network number(s) that were added for translation bridging between Ethernet and FDDI. The [-t] option displays the total number of entries in the IPX translation table. |
| <network> | Specifies which IPX network number to add to the IPX translation table for the specified encapsulation type. |
| <ethernet-encap> | Specifies the encapsulation type to be used for Ethernet packets. Packets bridged from FDDI to this network number are converted to this encapsulation. Specify one of the following:
enetEthernet Type II
802.3Raw 802.3
802.2802.3 with an LLC header
snap802.3 with LLC and SNAP headers |

¹ If the FDDI device does not support 802.3, bridging between the Ethernet devices and the FDDI device, standard IPX bridging is not allowed. IPX Translation bridging must be used.



The default Ethernet encapsulation type for use in Novell IPX, versions 2.2 through 3.11, is 802.3. The default for versions 3.12 through 4.x is 802.2.

<fdi-encap>

Specifies the encapsulation type to be used for packets translated to FDDI. Specify one of the following:

802.3Raw 802.3

802.2802.3 with an LLC header

snap802.3 with LLC and SNAP headers



The default FDDI encapsulation type for use in Novell IPX, versions 2.2 through 3.11, is 802.3. The default for versions 3.12 through 4.x is 802.2.

<network>|all

Specifies the IPX network number, or all, for deletion.



IPX translation bridging is independent of but mutually exclusive with IPX routing. It is recommended that IPX translation bridging and IPX routing not both be enabled. However, if both IPX translation bridging and routing are enabled, IPX routing takes precedence over IPX translation bridging.

10.9.1 Encapsulation Types

When IPX translation bridging is used, the Ethernet and FDDI encapsulation types to be used on each IPX network are specified. For each IPX network number, both the Ethernet and FDDI encapsulation types to be used on that network can be specified. Table 10.2 lists the combinations of encapsulation types that can be specified.

Table 10.2 - IPX Translation Bridging Encapsulations

	ENET	802.2	802.3*	SNAP
FDDI		∩	∩	∩
Ethernet	∩	∩	∩	∩

Table 10.2 - IPX Translation Bridging Encapsulations

	ENET	802.2	802.3*	SNAP
* The FDDI “raw” encapsulation is 802.3-like and is listed as “802.3” in table and command descriptions. However, this encapsulation is not identical to the 802.3 format on Ethernet since it does not include an explicit length field. Refer to the <i>PowerHub 7000/8000 Protocols Reference Manual</i> for the format of each type of encapsulation.				

For further information about IPX bridging over FDDI and packet encapsulation information, see *PowerHub 7000/8000 Protocols Reference Manual*.

10.9.2 Configuration Requirements

Although IPX translation bridging is easy to configure, the following conditions must be met:

- The servers attached to the segments in an IPX translation bridging network must be configured to have the same network number as the “IPX translation-bridging” network number. If a server’s network number cannot be changed to correspond to the IPX translation-bridging network, change the defined network number to match the server.
- Servers and clients must be configured to have the same encapsulation type as the type specified for the appropriate medium in the IPX translation-bridging network. For example, a client attached to an Ethernet segment must be configured to use the same Ethernet encapsulation type as the one defined for the corresponding IPX translation-bridging network. However, if encapsulation types on the server or client cannot be changed, the encapsulation types of the client or server can be configured on the PowerHub.

The following example displays the contents of the IPX bridging table:

```
49:PowerHub:bridge# ibt
IPX Translation Bridging: Disabled
IPX Network      Ethernet Encap      FDDI Encap
-----
          100      802.2              802.2/SNAP
```

```
Total entries: 1
50:PowerHub:bridge#
```

The following example adds IPX bridging network 200 with Ethernet Type II encapsulation and FDDI snap encapsulation:

```
53:PowerHub:bridge# ibt add 200 enet snap
IPX network 200 added to the translation table
54:PowerHub:bridge#
```

10.10 Learning

The `learning|learn` command is used to enable or disable bridge learning on specified segments. When bridge learning is enabled, MAC addresses from received packets are recorded. The learned MAC addresses are used to return packets to those destinations. By default, bridge learning is enabled when the system is loaded. The syntax for this command is as follows:

```
learning|learn pen[penable] <seglist>|all
learning|learn pdis[able] <seglist>|all
```

where

<code>pen[penable]</code>	Enables bridge learning on the specified segments.
<code>pdis[able]</code>	Disables bridge learning on the specified segments.
<code><seglist></code>	Specifies the segment(s) on which bridge learning is to be enabled or disabled. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments. Specify <code>all</code> if bridge learning is to be enabled or disabled for all segments.

The following examples disable bridge learning on a segment, bridge learning is then enabled on that segment.

```
12:PowerHub:bridge# learn pdis 2.1
Learned disabled on segment 2.1
13:PowerHub:bridge# learn penl 2.1
Learning enabled on segment 2.1
14:PowerHub:bridge#
```

10.11 Relearn Log

The `relearn-log | r1` command is used to display a log of learned MAC addresses on different segments. Also displayed are the previous segments the MAC address was learned on. The syntax for this command is as follows:

```
relearn-log | r1
```

The following example shows the relearn log is empty.

```
16:PowerHub:bridge# r1
Bridge Relearn Log is EMPTY
17:PowerHub:bridge#
```

10.12 Spanning Tree

The `spantree|st` command is used to enable, disable, and set Spanning-Tree algorithm options. The Spanning-Tree algorithm is a mechanism that logically eliminates physical loops in a bridged network. For example, if bridges are configured in such a way that broadcast/multicast packets are eventually forwarded back to the bridge that first sent them, the network contains a loop. Unless the network topology or bridges are re-configured to break this loop, or implement a mechanism to logically break the loop, broadcast/multicast packets are forwarded from bridge to bridge indefinitely, clogging the network. Whenever a segment's state is changed, either by automatic segment-state detection or by a user-interface command, the Spanning-Tree algorithm adjusts the network topology accordingly. When the Spanning-Tree algorithm is enabled, using the `spantree` command (see Section 10.12), the following Spanning-Tree parameters can be fine tuned:

- Bridge priority
- Segment priority
- Timer threshold
- Spanning-Tree path cost
- Fast hello-time thresholds (if the fast hello-time feature is enabled)

The first four parameters are always used; the last one is optional. The following sections describe how to adjust these parameters. The syntax for this command is as follows:

```

spantree|st en[enable]|dis[able]
spantree|st en[enable]|dis[able] fast-hello
spantree|st set maxage <time>
spantree|st set hello <time>
spantree|st set fwddelay <time>
spantree|st set fast-hello <time>
spantree|st set high-util <percentage>
spantree|st set low-util <percentage>
spantree|st set bridge-priority|bp <priority>
spantree|st sset seg-priority|sp <priority> <seglist>
spantree|st sset path-cost|pc <path-cost> <seglist>

```

where

`en[enable]|dis[able]` Specifies whether the Spanning-Tree algorithm or fast-hellos are to be enabled or disabled. The default is `disable`.

- <time>** Specifies the time to be set for:
maxage of the bridge-timer threshold. The range is 6-40 seconds. The default is 21 seconds.
hello time of the bridge-timer threshold. The range is 1-10 seconds. The default is 4 seconds.
fwdelay time of the bridge-timer threshold. The range is 4-30 seconds. The default is 16 seconds. The fast hello (1-10) default is 1 second.
- <percentage>** Specifies the percentage to be set for:
high-util (range is 1-100%). The default is 70%
low-util (range is 1 - 100%). The default is 50%.
- <priority>** Specifies the hexadecimal priority level assigned to:
bridge-priority (range 0 to **FFFF**). The default is 80 hex
seg-priority (range 0 to **FF**). The default is 80 hex.
- When specifying seg-priority, a separate priority must be assigned for each segment specified.
- <seglist>** Specifies the segments to which priority has been assigned. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.
- path-cost** Specifies the cost of the path. Specify a value from 1 to 65535. The default is 100 for 10Mb/s Ethernet segments and 10 for FDDI and Fast Ethernet segments. You must specify a separate path cost for each segment.

To display the current settings for these parameters, issue the following command:

```
config st
```

Following is an example of the display produced by the **config st** command, shortened for brevity:

```
62:PowerHub:bridge# config st
```

```
Spanning Tree
Status :                Disabled
System Priority :       8000
Spanning Tree Address : 01:80:c2:00:00:00
My Bridge Address :    00:00:ef:03:9a:b0
Max Age :              21
Hello Time :           4
Forward Delay :        16
```

Bridge Commands

```
Sending Fast Hellos :      Disabled
Fast Hello Params :      Hello Time: 1 sec, High Util: 70%, Low Util: 50%
```

```
Segment  Prio  Path Cost  Designated Bridge  Des Seg  Des Cost  Sta Chngs
-----  -
1.1      80    -          -                  -        -        -
.
.
4.1      80    -          -                  -        -        -
63:PowerHub:bridge#
```

10.12.1 Fast-Hello Time

Under heavy network traffic, Spanning-Tree hello packets are not transmitted at regular hello-time intervals. Such irregular time intervals can delay the transmission of hello packets. If hello packets are delayed past a certain time value, called the maximum age, the Spanning-Tree state can change. If the segment state is “blocking,” and hello packets are not received before the Max Age time value, the Spanning-Tree state changes to “listening” and then to “learning.”

This feature is by default disabled. Issue the `config st` command to display the Spanning-Tree settings, then check the value in the Sending Fast Hellos field.

10.12.2 High- and Low-Utilization Percentage

If the fast hello timer feature is enabled, when a segment’s utilization exceeds an upper-end value (<*high-util*>), the software automatically compensates for the increased traffic by using fast hello time to transmit hello packets. The fast hello time is less than the normal (configured) hello time. When all segments’ utilizations drop below a lower-end value, <*low-util*>, the hello time reverts to normal (either previously configured or system defaults).

The high and low utilization percentage values specify the upper-end value of segment utilization. If segment utilization exceeds this value and the fast hello timer feature is enabled, the software automatically compensates for the increased network traffic.

10.13 Statistics

The `stats show` command is used to display or clear bridge statistics of bridge table misses. When the `clear` argument is specified, all bridge statistics are cleared to zero (0), then statistics gathering begins. Once cleared, the statistics shown in the `stats` displays show the counts since the most recent clear, rather than since the most recent reboot. The syntax for this command is as follows:

```
stats clear
stats [show]
```

The following example shows the bridge statistics entries for bridge table misses:

```
64:PowerHub:bridge# stats
Table misses
04/49      0      0      -      -      -      -
02/33      0      0      0      0      0      0
01/01      0      0      0      0      0      0
65:PowerHub:bridge#
```

10.14 Status

The **status show** command is used to display the bridge spanning tree status for each segment. When this command is issued, the bridge status can differ depending on whether bridging or routing are enabled on particular segments. The syntax for this command is as follows:

```
status [show]
```

In the following example, a subset of segments is displayed, for brevity, showing the Spanning Tree status of those segments.

```
66:PowerHub:bridge# status
Segment   Segment Name           Spanning-tree
-----
1.1      Port_1                 forwarding
1.3      Port_3                 disabled
1.32     Port_32                disabled
2.1      Port_33                forwarding
2.2      Port_34                disabled
2.5      Port_37                forwarding
2.16     Port_48                disabled
4.1      Port_51                disabled

67:PowerHub:bridge#
```

For bridge or VLAN traffic to be forwarded on the segment, the Spanning-Tree state must be forwarding. The Spanning-Tree state does not affect routed traffic on the segment.

Note that the Spanning Tree state blocking does not indicate a problem in your network. As described in Section 10.12, the Spanning-Tree algorithm breaks loops in the bridge network by blocking certain segments. The columns in this display show the following information:

Segment The segment number listed in this column corresponds to the physical location of the segment in the PowerHub chassis. Use the **system config show** command to display information about a segment's physical location in the chassis. See your *PowerHub Installation and Maintenance Manual* for more information about this command.

If the segment number is followed by ** (two asterisks), then bridging has been disabled by the **bridging** command on that segment. Note that the bridging command does not affect routing. In this example, bridging has been disabled on segments 1.1, 1.3, 2.3, and 2.4.

Segment Name The description assigned to each segment. You can change the description using the `media sset segment name` command. See the *PowerHub Hardware Installation and Maintenance Manual* for more information about this command.

Spanning-tree The Spanning-Tree algorithm automatically causes segments to forward or block traffic based on the network topology. When the Spanning-Tree algorithm is enabled, this column shows one of four states:

listening
learning
blocking
forwarding
disabled

The `listening` and `learning` states occur when you first enable the Spanning-Tree feature or when your network topology changes. The `blocking` state indicates that packets are not being forwarded. The `forwarding` state indicates that packets can be forwarded on the segment. The `disabled` state indicates that the segment has been disabled using the `segment` command.

In this example, the Spanning-Tree feature is blocking bridge traffic on segment 1.5. The Spanning-Tree state has no effect on routing. However, this state does affect VLANs because traffic is bridged within VLANs rather than routed.

Bridge Commands

CHAPTER 11

Fiber Distributed Data Interface (FDDI)

This chapter explains the commands used to display, configure, and adjust parameters related to the Fiber Distributed Data Interface (FDDI) connections. The following functions available through the `fddi` subsystem are described:

- Attaching FDDI concentrators to a FDDI segment to enable a Dual Attached Concentrator (DAC).
- Detaching a FDDI concentrator.
- Adjusting hardware timer.
- Displaying the concentrator configuration.
- Displaying current FDDI-specific statistics.
- Displaying values of a specified group of FDDI MIB objects.



FDDI Concentrators can only be installed in the 10- and 15-slot PowerHub 7000/8000 chassis. The 5-slot chassis can accept FDDI Network Interface Modules.

11.1 Accessing the FDDI subsystem

To access the commands in the `fddi` subsystem, issue the following command from any PowerHub runtime command prompt:

```
fddi
```

The subsystem commands listed below are discussed in this chapter:

```
fddi subsystem:
```

```
concentrator | con          tvx
dac                       resetct | src
nvram                     smtmib
treq                      status
```

11.2 Concentrator

The `concentrator` command is used to attach or detach FDDI Concentrator modules to a single FDDI segment. Up to four FDDI Concentrator modules can be attached to a single FDDI segment, for a total of up to 64 FDDI Concentrator ports. The second FDDI concentrator channel can be used to attach up to four concentrators to an additional FDDI segment. Up to two DAS (Dual Attachment Stations) segments can be configured as DACs, provided they are part of the same FDDI module. Note that a single FDDI Concentrator module cannot be attached to two DAS segments (DACs). Refer to the *PowerHub 7000/8000 Installation and Maintenance Manual* for information about FDDI Concentrator Modules.



The `concentrator attach` command resets the FDDI module to attach the A and B port of the FDDI segment and the M ports of the FDDI Concentrator to the same ring. This reset causes a slight delay of approximately 10 seconds while the ring is reconfigured to add the newly attached FDDI Concentrator module.

The syntax for this command is as follows:

```
concentrator|con attach|detach [<slot-list> to <fddi-segment>]  
concentrator|con [show]
```

where

<code>attach detach</code>	Attach or detach a concentrator to the FDDI segment.
<code><slot-list> to <fddi-seg></code>	Specifies the slots that contain the Concentrator modules to attach to the specified FDDI segment. Up to three slot numbers can be specified. Separate the slot numbers with spaces.
<code><fddi-seg></code>	Specifies the FDDI segment to which the FDDI Concentrator Module is to be attached.

In the following example, FDDI Concentrator modules in NIM slots 4 and 5 are attached to FDDI segment 32. This command configures FDDI segment 32 as a DAC.

```
4:PowerHub:fddi# concentrator attach 4 5 to 32  
5:PowerHub:fddi#
```

If additional Concentrator modules are being attached to a DAC, use the **attach** option to add the module. In the following example, the Concentrator module in NIM slot 6 is attached to the DAC in segment 32, which is already managing the Concentrator module in NIM slots 4 and 5.

```
5:PowerHub:fddi# concentrator attach 6 to 32
6:PowerHub:fddi#
```

The example below shows the FDDI Concentrator configuration created in the example above.

```
6:PowerHub:fddi# concentrator
Concentrator modules are in slots:
DAC      Concentrator modules
32       4 5 6
7:PowerHub:fddi#
```

11.3 DAC

The **dac** command is used to display information about the FDDI Dual Attach Concentrators installed. The syntax for this command is as follows:

```
dac [show]
```

The following example shows the display produced by the **dac** command.

```
187:PowerHub:fddi# dac
There is no DAC configured
188:PowerHub:fddi#
```

11.4 NVRAM

The **nvr**am **show** command is used to display the FDDI Dual Attach Concentrator non-volatile random access memory (nvram). The syntax for this command is:

```
nvram show
```



Unlike other display commands, the **nvr**am command requires the **show** argument. If the **show** argument is not used, the active subsystem changes to the **nvr**am subsystem.

Fiber Distributed Data Interface (FDDI)

The following example shows the display produced by the `nvrAm show` command.

```
118:PowerHub:fddi# nvrAm show
bo                fm (floppy,flash-module)
locbdfile         bootdef
netbdfile         bootdef1
myip              169.144.86.54
mysm              255.255.255.0
fsip              169.144.86.49
gwip              169.144.86.1
crashreboot       (set)
oldui             (not set)
slotsegs[ 1]     32
slotsegs[ 2]     16
slotsegs[ 3]     (not set)
slotsegs[ 4]     2
slotsegs[ 5]     0
slotsegs[ 6]     (not set)
slotsegs[ 7]     (not set)
slotsegs[ 8]     (not set)
slotsegs[ 9]     (not set)
slotsegs[10]     (not set)
slotsegs[11]     (not set)
slotsegs[12]     (not set)
slotsegs[13]     (not set)
slotsegs[14]     (not set)
slotsegs[15]     (not set)
slotsegs[16]     (not set)
slotsegs[17]     (not set)
slotsegs[18]     (not set)
slotsegs[19]     (not set)
slotsegs[20]     (not set)
Total segments reserved: 50
md5key[ 1]       (set)
Total keys reserved: 1
```

11.5 Target Token Rotation Time (TREQ)

The **treq** command is used to set the target token rotation time (TTRT) variable (T_REQ). The TTRT specifies the amount of time each FDDI station holds on to the FDDI token. If the T_REQ timer for a segment is adjusted, the amount of time each device attached to the FDDI segment holds on to the token can be changed. The syntax for this command is as follows:

```
treq pset <time>|default <portlist>
```

where

<time>|default Specifies, in milliseconds, the new value for the hardware timer. The time must be followed by an “m” to indicate milliseconds.

Enter 4 to 167 milliseconds. An integer value must be specified; decimal numbers are truncated after the decimal. The default is 167 (milliseconds).

If **default** is specified, the timer resets to the default value.

<portlist> Specifies the FDDI segments for which to adjust a hardware timer.

Examples of the use of the **treq** command is shown below:

```
20:PowerHub:fddi# treq pset 25m 3.1
Segment 3.1 treq set to 25 millisecond
21:PowerHub:fddi# treq pset default 3.1
Segment 3.1 treq set to 167 milliseconds
22:PowerHub:fddi#
```

11.6 Time Transmission Variable (TVX)

The **tvx** command sets the valid time transmission variable (TVX). The syntax for this command is as follows:

```
tvx pset <time>|default <portlist>
```

where

<time>|default Specifies the new value for the hardware timer. The time entered must be followed by a “u” to indicate microseconds or an “m” to represent milliseconds.

If adjusting the TVX timer, specify from 2621 - 5200 microseconds (2.6 to 5.2 milliseconds). A number expressed up to three decimal places may be specified. Numbers with more than three decimal places are truncated after the third decimal. The default is 2621 (2.00 milliseconds).

<portlist> Specifies the FDDI segmen(t)s for which to adjust a hardware timer.

The examples below illustrate the use of the **tvx** command:

```
47:PowerHub:fddi# tvx pset 4700u 3.1
Segment 3.1 tvx set to 4.00 milliseconds
48:PowerHub:fddi# tvx pset 4m 3.1
Segment 3.1 tvx set to 4.00 milliseconds
49:PowerHub:fddi# tvx pset default 3.1
Segment 3.1 tvx set to 2.00 milliseconds
50:PowerHub:fddi#
```

11.7 Reset Count

The `resetct | src` command is used to display the fddi reset count for all fddi port(s). The syntax for this command is as follows:

```
[show] resetct | src
```

The following example displays the FDDI reset counter:

```
160:PowerHub:fddi# resetct
FDDI Reset Count:
    PORT 49:          0
161:PowerHub:fddi#
```

11.8 FDDI MIB Variables

The `smtmib` command is used to display the FDDI Concentrator MIB variables for specified FDDI port(s). The syntax for this command is as follows:

```
smtmib [show] [<group>] [<disprestrict>]
```

where

[<group>] Can be one of:

smt Displays the smt FDDI MIB objects.

mac Displays the MAC FDDI MIB objects.

port Displays the Port FDDI MIB objects.

priv Displays the Private FDDI MIB objects.

all Displays the FDDI MIB objects in all groups.

[<disprestrict>] Specifies the FDDI segment(s) to display the MIB objects. A single segment, a comma-separated list of segments, or a hyphen-separated list of segments can be specified.

The following example shows the display produced by the `smtmib` command with no arguments.

```

151:PowerHub:fddi# smtmib

SMT variables of port# 4.1:
-----
Station ID:                00-00-00-00-ef-03-9a-b0
Number of MACs in this station: 1
Number of Non-Master ports:  2
Number of Master Ports:     0
Paths available:            primary, secondary
Attachment configuration:   isolated

MAC variables of port# 4.1
-----
Upstream Neighbor:        00-00-f8-00-00-00
MAC Address:              00-00-ef-03-9a-b0
Downstream Neighbor:     00-00-f8-00-00-00
T-Req:                    167 milliseconds
T-Neg:                    167 milliseconds
T-Max:                    167 milliseconds
TvxValue:                 2.621 milliseconds
MAC Frame Count:         1
MAC Transmit Count:      0
MAC Copied Count:        0
MAC Error Count:         0
MAC Lost Count:          0
MAC Ring Operation Count: 1
RMT State:               ring-operational

PORT variables of port# 4.1
-----
FDDI Port 1:
Port PC Type:            A-port
Port PC Neighbor:        None
FDDI Port 2:
Port PC Type:            B-port
Port PC Neighbor:        None

Private MIB of port# 4.1
-----
Trace Signal Counter      0
My Claim Counter          1
My Beacon Counter         0
Other Beacon Counter      0
TRT Expired Counter       0
Duplicate Claim Counter    0
152:PowerHub:fddi#

```

11.9 Statistics

The **status** command is used to display statistics of FDDI modules. This command displays information that is tracked with the FDDI counters.



Statistics tracked with the **status** command are not specific to a FDDI segment but the module as a whole.

In addition to the standard Ethernet and FDDI packet statistics available, the **bridge stats** command (see Chapter 10) can be used to display statistics that apply specifically to the FDDI modules. For each FDDI module, the software maintains counters for the following FDDI packet statistics:

- Number of packets forwarded locally (from one FDDI segment to the other) on the module. This number is always 0 for the Single FDDI and the Universal Single FDDI module.
- Number of packets forwarded to the Packet Engine because the FDDI Engine could not make a forwarding decision for the packet.
- Number of packets locally filtered by the FDDI module.
- Number of fragmented packets.
- Number of un-fragmented packets.
- Number of large IP packets fragmented by the FDDI module. A large IP packet is one that exceeds the Ethernet MTU (maximum transmission unit) size.
- Number of large IP packets that needed to be fragmented but could not be fragmented and were, therefore, dropped by the FDDI module. This can occur if the packet's No Fragment bit is set, or if the switching engine temporarily runs out of resources for fragmenting packets.



Statistics for specific segments are available using the **smtmib priv** command. These statistics do not apply to the FDDI Concentrator modules.

The syntax for this command is as follows:

```
status [show] counter <slot>
```

where

<slot> Specifies the slot in which the FDDI module is installed. If the slot number is not known, use the **system config** command to display the slot locations of the FDDI modules. Alternatively, the slot number can be verified by visually checking the slot-number label located to the left of the modules in the PowerHub chassis.

The example below shows the information displayed by this command:

```
157:PowerHub:fddi# status counter 4

FDDI Counters of slot# 4:
-----
Number of Packets Forwarded to FDDI                0
Number of Packets Forwarded to Packet Engine       0
Number of Packets Filtered                          0
Number of Packets Fragmented                       0
Number of Packets Not-Fragmented                   0
Number of IP Packets Forwarded Locally              0
Number of IP Packets Dropped (BAD CHKSUM)          0
158:PowerHub:fddi#
```

Fiber Distributed Data Interface (FDDI)

CHAPTER 12 SNMP Commands

The PowerHub contains an implementation of Simple Network Management Protocol (SNMP). SNMP uses User Datagram Protocol (UDP), an industry-standard connectionless protocol used to send and receive packets between a managed PowerHub and other devices. This chapter describes the commands located in the `snmp` subsystem and shows how to perform the following tasks:

- Display the SNMP configuration.
- Add an SNMP management community.
- Add an SNMP manager.
- Delete an SNMP management community.
- Delete an SNMP manager.
- Display SNMP packet statistics.
- Clear SNMP packet statistics.

In addition, this chapter describes how to set up files for use with SunNet Manager to access the PowerHub Management Information Bases (MIBs).

Using a third-party SNMP application, the PowerHub MIB objects can be accessed for information about the PowerHub. The software contains implementation of standard MIBs and the PowerHub Proprietary MIB.

12.1 Accessing the SNMP Subsystem

To access the `snmp` subsystem, issue the following command at any runtime command prompt:

```
snmp
```

The following commands are available in the `snmp` subsystem:

```
snmp subsystem:
```

```
community|com  
config
```

```
manager|man  
stats
```

12.2 SNMP Community

The `community|com` command is used to add or delete SNMP community settings. The default configuration includes the standard default SNMP community, `public`, which has read-only access. Up to eight SNMP communities can be supported at any one time. The syntax for this command is as follows:

```
community|com add <community-name> [ro|rw]
community|com delete|del <community-name>
```

where

- add** Specifies that the named community is to be added to the configuration.
- del[ete]** Specifies that the named community is to be deleted from the configuration.
- <community-name>** Specifies the community name to be added or deleted.
- [ro|rw]** Specifies the community's access as read-only (**ro**) or read-write (**rw**). The default is read-only access.

The following example illustrates adding an `admin` community with read-write access:

```
77:PowerHub:snmp# community add admin rw
78:PowerHub:snmp#
```

The following command deletes the `admin` community:

```
79:PowerHub:snmp# community del admin
80:PowerHub:snmp#
```

12.3 Standard Traps

SNMP specifications define a series of standard traps of which the PowerHub implements those listed below in Table 12.1:

Table 12.1 - Standard Traps

Trap	Conditions	OID and MIB objects	Variables
coldStart	The device has been power-cycled.	generic 0	
linkDown	This trap is produced when a link goes down due to a secure address violation, network connection error, or an explicit management disable action. The trap frame carries the index value of the port.	generic 2	ifIndex
linkUp	This trap is generated when a port is re-enabled. The trap frame contains the index value of the affected port.	generic 3	if Index
authenticationFailure	This trap is generated when the switch receives an SNMP message that is not accompanied by a valid community string.	generic 4	
fddiRing-Wrap	FDDI ring wrap has occurred.	1.3.6.1.2.1.10.15.73.1 (fddimib-SMT1)	fddiSMTIndex
newRoot	Indicates the sending agent has become new root of Spanning Tree.	1.3.6.1.2.1.17.1 (dot1dBridge 1)	dot1dBaseBridge-Address

Table 12.1 - Standard Traps

Trap	Conditions	OID and MIB objects	Variables
topology-Change	Sent by a bridge when any of its configured ports transitions from Learning to Forwarding state, or from Forwarding to Blocking state.	1.3.6.1.2.1.17.2 (dot1dBridge 2)	dot1dBaseBridge-Address

12.4 Enterprise-Specific Traps

Table 12.2 shows the enterprise-specific traps implemented in the PowerHub and lists the conditions that cause these traps to be generated, their OIDs, MIB objects, and variables:

Table 12.2 - Enterprise-Specific Traps

Trap	Conditions	MIB object	Variables
atmLinkUp	The specified interface has just left the <i>down</i> state. Slot ID is reported by trap.	1.3.6.1.4.1.326.2.6.1.1.2.1 (alatm 1)	alAtmAMAActual Use alATMCurrentAMA Type
atmLinkDown	Indicates a link is down and reports the slot ID for the downed interface.	1.3.6.1.4.1.326.2.6.1.1.2.2 (alatm 2)	alAtmAMASlotNumber alAtm PreviousAMA alAtmPreviousAMA-Type
atmCutOver	Reports a cut over from primary to backup port or vice versa.	1.3.6.1.4.1.326.2.6.1.1.2.3 (alatm 3)	alAtmAMASlotNumber, alAtmAMAActual Use, alAtmCurrentAMA-Type, alAtmPreviousAMA, alAtmPreviousAMA-Type
atmBootUp	Indicates start up and reports slot Id.	1.3.6.1.4.1.326.2.6.1.1.2.4 (alatm 4)	alAtmAMASlotNumber, alAtmAMAActual-Use, alAtmCurrentAMA-Type

Table 12.2 - Enterprise-Specific Traps

Trap	Conditions	MIB object	Variables
atmFault	Indicates a series of five or more consecutive atmLinkdowns have occurred. Once atmFaults occur, atmLinkdown traps will not be sent.	1.3.6.1.4.1.326. 2.6.1.1.2.5 (alatm 5)	alAtmAMASlotNumber
powerFailure	Indicates power failure has occurred in the reported slot.	1.3.6.1.4.1.326. 2.6.1.1.1.1 (alchassis 1)	alPSNumber
boardFailure	Indicates an intelligent card failure in the reported slot.	1.3.6.1.4.1.326. 2.6.1.1.1.2 (alchassi 2)	alSlotNumber
alLoginFailure	The login failure trap indicates that the login failed due to some error condition during the login process.	1.3.6.1.4.1.326. 2.6.2.1.15.1 (alsystem 1)	alLastLoginFailureTimeDate; alLastLoginSourceAddress; alLastLoginFailureUserId; alLastLoginFailureReason

12.4.1 SNMP Configuration

The `config` command is used to display the current SNMP configuration of communities and managers. The syntax for this command is as follows:

```
config [show] [-l] [<community-name>]
```

where

[-l] Optionally specifies the list of managers and trap configurations.

[<community-name>] Optionally specifies the community to display.

The following example displays the default SNMP configuration. Notice that no options are specified.

SNMP Commands

```
74:PowerHub:snmp# config
```

```
Community                Access
```

```
-----
```

```
public                   ro
```

```
75:PowerHub:snmp#
```

12.5 Displaying Statistics

The **stats** command is used to display and clear statistics on SNMP packets transmitted and received. These statistics are a superset of the corresponding statistics provided in the SNMP table of MIB-II. Two copies of each SNMP statistics counter are maintained:

- Count since last clear.
- Count since last reset.

The syntax for this command is as follows:

```
stats [show] [-t]
stats clear
```

[-t] Displays statistics since the last reset.

clear Clears all SNMP statistics.

The following example shows the **stats** command used both with and without the **[-t]** argument:

```
82:PowerHub:snmp# stats
SNMP packet statistics (count since last stats clear):
Packets Rcvd:          96   Packets Sent:          93
Bad Version Rcvd:      3   Bad Comm Name Rcvd:    0
Bad Comm Uses Rcvd:    0   ASN Parse Err Rcvd:    0
Bad Type Rcvd:         0   Too Big Rcvd:         0
No Such Name Rcvd:     0   Bad Values Rcvd:      0
Read Onlys Rcvd:      0   Gen Errs Rcvd:        0
Total vars Req:       417   Total vars Set:        0
Get Req Rcvd:         0   GetNext Req Rcvd:     93
Set Req Rcvd:         0   Get Resp Rcvd:        0
Traps Rcvd:           0   Too Big Sent:         0
No Such Name Sent     0   Bad Values Sent:      0
Read Onlys Sent:      0   Gen Errs Sent:        0
Get Req Sent:         0   GetNext Req Sent:     0
Set Req Sent:         0   Get Resp Sent:       93
Traps Sent:           0
```

```
83:PowerHub:snmp# stats -t
SNMP packet statistics (Total count since last system reset):
Packets Rcvd:          96   Packets Sent:          93
Bad Version Rcvd:      3   Bad Comm Name Rcvd:    0
Bad Comm Uses Rcvd:    0   ASN Parse Err Rcvd:    0
Too Big Rcvd:         0
No Such Name Rcvd:     0   Bad Values Rcvd:      0
Read Onlys Rcvd:      0   Gen Errs Rcvd:        0
Total vars Req:       417   Total vars Set:        0
Get Req Rcvd:         0   GetNext Req Rcvd:     93
```

SNMP Commands

Set Req Rcvd:	0	Get Resp Rcvd:	0
Traps Rcvd:	0	Too Big Sent:	0
No Such Name Sent	0	Bad Values Sent:	0
Gen Errs Sent:	0		
Get Req Sent:	0	GetNext Req Sent:	0
Set Req Sent:	0	Get Resp Sent:	93
Traps Sent:	0		

84:PowerHub:snmp#

12.6 Adding an SNMP Manager

The `manager` | `man` command is used to add or delete SNMP managers. Each community can include up to 16 managers. The SNMP manager entries include an IP address. This IP address should be the SNMP management station that any configured traps are to be sent to. The syntax for this command is as follows:

```
manager|man add <community-name> <IP-addr> [trap|notrap]
manager|man delete|del <community-name> <IP-addr>|all
```

where

add	Specifies that the SNMP manager at <IP-addr> be added and associated with the specified <community-name>.
<community-name>	Specifies the community name to which a SNMP manager is to be added.
<IP-addr>	Specifies the IP address of the SNMP manager.
[trap notrap]	Optional flag, indicating whether the SNMP manager should receive traps or not. If the manager should receive traps, use (trap). If the manager should not receive traps, use (notrap). The default is notrap .
delete	Deletes the SNMP manager specified by <IP-addr> and <community-name> or all configured SNMP managers.
<IP-addr> all	Specifies the IP address of the SNMP manager to be deleted. If all is specified, all configured SNMP managers are deleted.

In the following example, an attempt is made to add a SNMP manager at IP address 169.144.86.49 to the admin SNMP community. Since the community doesn't exist, an error message is generated. The community is then created (`com add`). The attempt to add the SNMP manager is made again and is successful. The SNMP configuration is then displayed.

```
96:PowerHub:snmp# man add admin 169.144.86.49
```

```
ERROR: Community admin not found. Cannot add manager 169.144.86.49.
```

```
97:PowerHub:snmp# com add admin rw
```

```
98:PowerHub:snmp# man add admin 169.144.86.49
```

```
99:PowerHub:snmp# config
```

```
Community
```

```
Access
```

SNMP Commands

```
-----  
public          ro  
admin           rw  
100:PowerHub:snmp#
```

Additionally, SNMP managers can be deleted by deleting the community they are attached to. Do this with care, as deleting the community deletes all managers attached to that community.

12.7 Using SunNet Manager

If SunNet Manager is being used to access the MIBs, the following types of files must be prepared for each MIB:

- Schema
- Trap
- OID.

Table 12.3 lists the utilities and file names in SunNet Manager used to prepare these files.

Table 12.3 - SunNet Manager Utilities

schema	A MIB converted from ASN.1 format.	mib2schema	<MIB-name>.schema
trap	Active traps for a particular MIB.	mib2schema	<MIB-name>.trap
OID	Object Identify file. Translates the Object Identifiers used by SNMP to communicate into the identifiers that SunNet Manager understands.	mib2schema	<MIB-name>.oid

*Where <MIB-name> is the name of the MIB.

SNMP Commands

CHAPTER 13 TFTP Commands

The `tftp` subsystem contains the PowerHub implementation of TFTP (Trivial File-Transfer Protocol). Use the `tftp` subsystem commands to perform the following tasks:

- Set a default TFTP server IP address.
- Display the default TFTP server IP address.
- Unset the default TFTP server IP address.
- Download or display a file stored on a TFTP server.
- Upload a file to a TFTP server.
- Load (activate) a configuration file stored on a TFTP server.
- Save a configuration file to a TFTP server.

To make use of the commands in this subsystem, a TFTP server must be configured to support TFTP file transfers. The procedures for configuring a TFTP server depend upon the particular type of server being used. Refer to the appropriate server documentation for specific configuration information.

Also, the segment connecting to the TFTP server must have an IP interface defined on it. For information about adding an IP interface, refer to the PowerHub 7000/8000 *Protocols Manual*.



The TFTP protocol provides no authentication for any services, including downloading or changing files stored on the TFTP server. If the TFTP server is configured to allow `tftp` commands to be used, anyone with access to the server can download or change files.

13.1 Accessing the TFTP Subsystem

To access the `tftp` subsystem, issue the following command at any runtime command prompt:

```
tftp
```

13.2 Considerations

The TFTP commands work with many types of TFTP servers, including servers running UNIX, DOS, Windows NT or OS/2. The following considerations apply to TFTP servers that are running UNIX, a very common platform for TFTP. Regardless of the platform used, consult the appropriate server documentation regarding either of the following:

- File permissions (not applicable to some operating systems).
- Conventions for pathnames and file names.

If problems are experienced while uploading or downloading files between the PowerHub and TFTP server, they can often be resolved by verifying whether read and write access to the server is required and how file names need to be specified.

13.2.1 TFTP Commands and UNIX Read/Write Permissions

To use TFTP commands to upload or download files, the proper UNIX read/write permissions must be setup on the TFTP server. On most servers, permissions are controlled separately for users, groups, and “others.” The TFTP server considers the PowerHub to be among the “others.” It is recommended that an outbound Telnet session be established with the TFTP boot server. Refer to *Chapter 14* for details on opening an outbound Telnet session.

Read/write access to PowerHub files and directories can be controlled on the TFTP server by setting the read and write permissions. On most UNIX systems, permissions can be displayed using the UNIX `ls` command. Following is an example of the permissions information displayed for a file on a typical UNIX TFTP server.

```
$ ls -l
total 3
-rw-rw---- 1 mrspat      622 Jul 19 15:09 Lab1.env
-rw-rw-r-- 1 ethan        643 Jul 19 15:11 Lab2.env
-rw-rw-w-- 1 sascha        611 Jul 19 15:13 Lab3.env
-rw-rw-rw- 1 stripie       698 Jul 19 15:15 Lab4.env
-rw-rw-rw- 1 tiger        698 Jul 19 15:15 Lab5.env
```

The text shown in bold is the permission information for each file for the “others” category.

- In this example, no read or write permissions are enabled on `Lab1.env`. Consequently, this file cannot be uploaded or downloaded with this name using the TFTP commands.
- Read, but not write, permission is granted to the file `Lab2.env` for others. This file can be downloaded or displayed, but a file using this name cannot be uploaded.
- The file `Lab3.env` cannot be downloaded. However, it can be uploaded.

- Finally, `Lab4.env` and `Lab5.env` have both read and write permissions enabled. These files can be uploaded or downloaded.

The UNIX `chmod` command can be used on most UNIX systems to change read/write permissions. From an open outbound Telnet session, issue the `chmod`, or appropriate, command to change the read/write permissions after creating the zero-file length file. Refer to the UNIX shell documentation for details.

13.2.2 Path Names

Depending upon the TFTP server configuration, path names may need to be specified when using the TFTP commands.

On some servers, when the TFTP commands are used to upload or download files, the PowerHub understands file names according to where the server is accessed. Only those files located in the directory accessed by the PowerHub, or in a subdirectory of that directory, can be uploaded or downloaded if specified.

For example, suppose the TFTP server is configured to allow access to the server at a directory called TFTP.

```
TFTP
  fore
    ph
      ethan.env
      sascha.env
```

All directories below the TFTP directory are considered part of the pathname for the files stored there. Relative to the PowerHub, the pathname for the files `ethan.env` and `sascha.env` is `fore/ph`. To download `ethan.env`, the following command would be issued:

```
get -a fore/ph/ethan.env ethan.env
```

where

-a	Specifies net-ASCII mode. (Files are transferred in binary mode by default.)
fore/ph/ethan.env	Source file name with path information.
ethan.env	Destination file name. This name must be in the DOS file-naming format (filename.ext).

13.2.3 File-Naming Conventions

Local file names are optional when using the `get` command. The local file name can be omitted if the file name is eight characters or fewer in length with an extension no longer than three characters and the name does not need to be changed.

Suppose the PowerHub has access to the TFTP server at the TFTP directory, as shown in the following example:

```
TFTP
  fore
    ph
      sascha.env
      ethan.env
      lotsofdots
```

If the `get` command is issued, without specifying the local file name (`sascha.env`), an error message is displayed on the PowerHub.

To download the file, `lotsofdots`, a local file name fitting the DOS file naming conventions must be specified, as shown in the following example, where `lotsofdots` is renamed `spots`.

```
get -a fore/ph/lotsofdots spots
```

13.2.4 Remote File Names

Some TFTP servers require that the remote file name exist on the server before allowing anything to be written to that file name. If this is required, create a zero-length file on the server (Unix `touch <filename>` command), specifying the name of the remote file name that is to be used with the `put` or `savecfg` commands.

Also, on some TFTP servers, files that are overwritten on the server are not properly truncated. When overwriting an existing file on the TFTP server, if the older version of the file is longer than the new file, the older version is not truncated properly by the server. As a result, the new version of the file contains part of the older version of the file. Do one of the following to verify that the new version completely replaces the older version of a file: Remove the older version of the file, then save the new version.

- If the server requires that the file name be present on the server before copying it, create a zero-length file with the new name then save the file under the new name. After the new file is copied to the server, delete the older version of the file and rename the new file as desired.

13.3 TFTP Commands

The commands described in the following sections allow a particular TFTP server to be specified in the file operations described in this chapter. These following commands are discussed:

tftp subsystem:

```
server          readcfg|rdcfg
get             savecfg|svcfg
put
```

13.3.1 Setting the Default Server

The **server** command is used to specify the default TFTP server. The syntax for this command is as follows:

```
server [show]
server set <ipaddr>
server unset
```

where

- set** When specified, sets the TFTP server to the specified <ipaddr>. If no <ipaddr> or the verb unset is specified, the current server configuration is displayed.
- <ipaddress>** Specifies the IP address of the TFTP server to use as the default. Specify the address in dotted-decimal notation.
- unset** Deletes the current TFTP server configuration.

In the following examples, the current server address is shown as unset and then a server address is specified.

```
6:PowerHub:tftp# server
server: (not set)
7:PowerHub:tftp# server set 169.144.85.49
8:PowerHub:tftp# server
server: 169.144.85.49
9:PowerHub:tftp#
```

Only one active TFTP server can be configured at a time. Setting a new default TFTP server IP address replaces the existing TFTP server IP address.

13.4 Downloading a File

The `get` command is used to transfer a file from the configured TFTP server to a local disk file or displayed on the local terminal. The syntax for this command is as follows:

```
get [-h <host>] [-a] <remote-file> [<local-file>|tty]
```

where

[-h <host>] Specifies the IP address (in dotted-decimal notation) of the TFTP server. If this argument is not specified, the default server is used. The default server is specified using the `set server` command. (See Section 13.3.1.)

[-a] Forces the transfer to take place in net-ASCII transfer mode rather than octet mode. Octet mode transfers the file, including end-of-line characters, exactly as it is stored on the server. Net-ASCII changes the end-of-line characters to be compatible with the display or storage device that receives the file.

Use the default (octet-mode) to download software image files (ex: 7f, 7pe, 7atm, and so on). Use the net-ASCII mode to download configuration files, environment files, and other text files.

If the file is to be displayed on the management terminal (by specifying `tty` as the local file name), omit this argument. The file is automatically transferred in net-ASCII format.

<remote-file> Specifies the name of the remote file. Specify the name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called `transfer`, and this directory is specified as the TFTP home directory, do not specify `transfer` as part of the file name.

[<local-file>|tty] If this argument is not specified, the PowerHub assumes the same file name on the server. The pathname (if any) must be included with the file name.

If a *<local-file>* argument is omitted, or a local file name is specified, the file is written to a local storage device.



If a local device is not specified, the file is written to the default-device. To specify a device, preface the file name with **fm:** (Flash Memory Module; PowerHub 7000) or **fd:** (Floppy Diskette). On the PowerHub 8000, no device is required. If the system was booted over the network, the **fm:** is the default device.

If the file name on the server is an invalid pathname on the PowerHub, an error message is displayed.

If **tty** is specified, the file is not downloaded but an image of the file is displayed on the management terminal. The file can be displayed from within a TTY (RS-232) session or a TELNET session.

If a TFTP server name is not specified and no default server name has been configured, an error message is displayed. To configure a default server name, use the **server set** command. (See Section 13.3.1.)

13.5 Uploading a File

The `put` command is used to transfer a file to the configured TFTP server. The syntax for this command is as follows:

```
put [-h <host>] [-a] <localfile> [<remote-file>]
```

where

-h <host> Specifies the IP address, in dotted-decimal notation, of the TFTP server. If this argument is not specified, the default TFTP server is used. The default TFTP server is specified using the `set server` command. (See Section 13.3.1.)

-a Forces a net-ASCII transfer. If not specified, octet mode is used to transfer the file. Octet mode transfers include end-of-line characters, transferring the file exactly as it is stored on the server. Net-ASCII changes the end-of-line characters to be compatible with the display or storage device that receives the file.

Use the default (octet-mode) to download software image files (ex: 7f, 7PE, ppu. 7PE, and so on). Use the net-ASCII mode to download configuration files, environment files, and other text files.

<local-file> Specifies the local file name.



If a local device is not specified, the file is written to the default-device. To specify a device, preface the file name with **fm:** (Flash Memory Module; PowerHub 7000) or **fd:** (Floppy Diskette). On the PowerHub 8000, no device is required. If the system was booted over the network, the **fm:** is the default device.

- <remote-file>** Specifies the name of the file as it is to appear on the server. Specify the name that is meaningful to the TFTP program on the server. For example, if the name, with the path of the server, contains a subdirectory called `transfer` and this directory is specified as the TFTP home directory, do not specify `transfer` as part of the file name.

The following procedure shows the steps required to upload the default configuration file (`cfg`) to the configured TFTP server from the PowerHub.

1. Open an outbound Telnet session with the remote TFTP host.

```
29PowerHub:tftp# telnet open 169.144.86.49
Trying 169.144.86.49...
Connected to 169.144.86.49.
Escape character is '^Y'.
```

```
SunOS UNIX (fabrique)
```

```
login: username
```

```
Password:
```

```
Last login: Mon Mar  2 07:57:39 from username
```

```
SunOS Release 4.1.4 (GENERIC) #2: Fri Oct 14 11:08:06 PDT 1994
```

2. On the remote TFTP host, change to the `tftpboot` directory.

```
fabrique-username:51=> cd /
fabrique-:52=> cd tftpboot
```

3. In the `tftpboot` directory, create a zero-length file called `cfg`.

```
fabrique-tftpboot:56=> touch cfg
```

4. Escape back to the PowerHub (Ctrl+Y).

```
fabrique-tftpboot:57=> Escape to Command line mode. Type 'open' to return.
```

5. Issue the `put` command to upload the file. Notice the use of the `-a` option, since the `cfg` file is an ASCII file.

```
31PowerHub:tftp# put -a cfg
```

```
tftp: Peer generated error
```

```
protocol error: Permission denied: Access violation
```

6. Notice that an error was received. To correct the error, re-open the outbound Telnet session and display the read/write permissions for the `cfg` file created in step 3.

```
32PowerHub:tftp# telnet open
```

```
fabrique-tftpboot:57=> ls -al
total 3
```

```
drwxrwxrwx  3 root          512 Mar  2 08:12 .
drwxr-xr-x 26 root          1024 Jan 19 13:27 ..
-rw-r--r--  1 username      0 Mar  2 08:12 cfg
drwxrwxrwx  4 username      512 Mar  2 08:02 fore
```

- Note that the `cfg` file does not have write permissions. Issue the UNIX `chmod` command to change the read/write permissions to allow writing to the file and then display the files to verify the read/write permissions were changed.

```
fabrique-tftpboot:58=> chmod 777 cfg
fabrique-tftpboot:59=> ls -al
total 3
drwxrwxrwx  3 root          512 Mar  2 08:12 .
drwxr-xr-x 26 root          1024 Jan 19 13:27 ..
-rwxrwxrwx  1 username      0 Mar  2 08:12 cfg
drwxrwxrwx  4 username      512 Mar  2 08:02 fore
```

- Escape back to the PowerHub (Ctrl+Y) and re-attempt the upload.

```
fabrique-tftpboot:60=> Escape to Command line mode. Type 'open' to return.
33PowerHub:tftp# put -a cfg
169.144.86.49:cfg: 28553 bytes
```

- Notice that this time the transfer was successful. Re-open the outbound Telnet session and display the files present to verify a successful transfer

```
34PowerHub:tftp# telnet open
fabrique-tftpboot:60=> ls -al
total 31
drwxrwxrwx  3 root          512 Mar  2 08:12 .
drwxr-xr-x 26 root          1024 Jan 19 13:27 ..
-rwxrwxrwx  1 username      28553 Mar  2 08:13 cfg
drwxrwxrwx  4 username      512 Mar  2 08:02 fore
fabrique-tftpboot:61=>
```

In the following example, an environment file is uploaded to a UNIX server in a subdirectory (`fore/configs`) of the `tftpboot` directory.

```
40PowerHub:tftp# put myenv fore/configs/myenv
169.144.86.49:fore/configs/myenv: 92 bytes
41PowerHub:tftp#
```

Notice that a pathname is specified with the file name in this example. Ensure a pathname that is meaningful to the TFTP program is specified.

13.6 Read Configuration

The `readcfg|rdcfg` command is used to read (load) a PowerHub configuration file that is stored on a remote TFTP server. The syntax for this command is as follows:

```
readcfg|rdcfg [-v] [-h <host>] <remote-file>
```

where

- `-v` Displays the configuration commands as they are executed.
- `-h <host>` Specifies the IP address of the TFTP server. If not specified, the default TFTP server is used. (The default TFTP server is specified using the `server set` command (see Section 13.3.1).
- `<remote-file>` Specifies the name of the configuration file to read. Specify a name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called `configs` and this directory is specified as the TFTP home directory, do not specify `configs` as part of the file name.

As with the `get` command, if a host server name is not specified and no default server name has been configured, an error message is displayed.

The following example reads (loads) the configuration file (`cfg`) from the remote `tftpboot` server in the `fore/configs` directory. An outbound Telnet session is opened first to effect the transfer. The outbound Telnet session is not closed during this process.

```
67:PowerHub:tftp# readcfg cfg
System name set to 'PowerHub'.
System location set to:
Undefined
dcd-detection disabled
Bridge Table aging time set to 60 minutes
.
.
.
68:PowerHub:tftp#
```

13.7 Save Configuration

The `savecfg|svcfg` command is used to save the current configuration to a remote TFTP host. To save a configuration to a remote TFTP host, make sure a TFTP file already exists on the host to which the configuration can be saved. The syntax for this command is as follows:

```
savecfg|svcfg [-h <host>] <remote-file>
```

where

- `-h <host>` Specifies the IP address of the TFTP server. (The default server is specified using the `set server` command (see Section 13.3.1).
- `<remote-file>` Specifies the name of the configuration file to be saved. Specify a name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called `configs` and this directory is specified as the TFTP home directory, do not specify `configs` as part of the file name.

On UNIX-based TFTP servers, if write permission for “others” is not enabled for the configuration file name or the directory to which the file is being written, a message such as the following is displayed:

```
16:PowerHub:tftp# savecfg cfg
tftpWrite: Peer generated error
tftp: Permission denied: Access violation
17:PowerHub:tftp#
```

If this error is received, check the file and directory permissions for “others” on the TFTP server.

If the UNIX-based server requires that the file name already exist, but the file does not yet exist on the server, a message such as the following is displayed:

```
18:PowerHub:tftp# savecfg cfg
tftpWrite: Peer generated error
tftp: File not found: File not found
19:PowerHub:tftp#
```

The following example shows a successful execution of the `savecfg` command to the default remote host directory.

```
70:PowerHub:tftp# savecfg cfg
71:PowerHub:tftp#
```

TFTP Commands

14.2 Opening a Telnet Session

The **open** command is used to open an outbound telnet client session to a remote host whose IP address is specified <ipaddr>. The optional parameter, [<TCP port>], specifies a remote telnet server listening TCP port. The default value is TCP port 23. The syntax for this command is as follows:

```
open <ipaddr> [<TCP port>]
```

where

<ipaddr> Specifies the IP address of the remote device to open the Telnet session.

[<TCP port>] Optionally used to specify a TCP port if the default TCP port 23 is not used.

Entering **open <ipaddr>** from the telnet system prompt displays the following information:

```
80:PowerHub:telnet# open 169.144.86.49
Trying 169.144.86.49...
Connected to 169.144.86.49.
Escape character is '^Y'.
```

```
SunOS UNIX (fabrique)
```

```
login:
```



Only two telnet client sessions can be opened from the PowerHub at a time.

When the **open** command is executed with a valid <ipaddr>, the system attempts to connect to the requested address and the information is echoed back to the console. Additionally, a keyboard shortcut is available to return to the PowerHub system prompt. Pressing the Control key with the 'Y' (Ctrl+Y) key from the active host acts as a shell command returning the user to the PowerHub system prompt.

14.3 Closing a Telnet Session

The `close` command closes the current telnet client session. The syntax for this command is as follows:

```
close
```

To use the `open` command to exit from an active telnet client session, the `Ctrl+Y` keyboard shortcut must be used to shell back to the PowerHub system prompt. The system responds as shown below:

```
86:PowerHub:telnet# close
Telnet session Disconnected.
87:PowerHub:telnet#
```

It is also possible to exit from the active telnet client session by logging out of the connected host. This terminates the session and returns the user to the PowerHub system prompt as shown below:

```
SunOS Release 4.1.4 (GENERIC) #2: Fri Oct 14 11:08:06 PDT 1994
fabrique-dspreadb:51=> logout
88:PowerHub:telnet#
```

14.4 Viewing Telnet Status

The `status` command displays the current telnet client status information. The syntax for this command is as follows:

status

Issuing **status** from the telnet system prompt displays the following information:

```
82:PowerHub:telnet# status
Connected to 169.144.86.49
Escape character is '^Y'.
83:PowerHub:telnet#
```



To execute the status command with an open telnet client session, the user must escape to the PowerHub user interface with the `Ctrl+Y` keyboard shortcut.

This appendix lists the PowerHub configuration defaults. The purpose of this appendix is to provide information about what is already configured in the software. This information will assist in diagnosing and troubleshooting any potential problems. Configuration defaults are listed by subsystem.

Table A.1 - Boot PROM Commands

Command and Description
zreceive zr rz [-+27abcehtw] [<file-name>] t Sets the receive timeout to N/10 seconds (10 <= N <= 1000). The default is 100 ; that is, 10 seconds.
zsend zs sz [-+27abehkLlNnoptwXYy] <file-name> t Sets the receive timeout to N/10 seconds (10 <= N <= 1000). The default is 600 ; that is, 60 seconds.

Table A.2 - Global Subsystem Commands

Command and Description
su [root monitor] Used to change the userid to the root or monitor. The default is the root.
rm [-f] [-i] <filespec> [<filespec>...] Overrides the -f (Force) flag, presenting a prompt before removing each file. If -f or -i is not specified, -i is the default.

Table A.3 - ATALK Subsystem Commands

Command and Description
<p><code>enable disable atalk</code></p> <p>Specifies whether AppleTalk routing is to be enabled or disabled. The default is disable.</p>
<p><code>ping [-t <timeout>] [-size <pktsize>] <net>.<node></code></p> <p><code>[-t <timeout>]</code> Optionally specifies the number of seconds to wait to receive a reply packet from the specified node. The default is 15 seconds.</p> <p><code>[-size <pktsize>]</code> If the <code><timeout></code> argument is used, optionally specifies the size of the echo packet sent to the node. The default is 64 bytes.</p>

Table A.4 - Bridge Subsystem Commands

Command and Description
<p><code>config [show] [<argument-list> all]</code></p> <p>Specifies the configuration parameters to display. The default is all.</p>
<p><code>bt [show] [<seglist> all] [<ethaddr>] [-t] [[-h] [-m]]</code></p> <p>Specifies the segment(s) to display bridge table entries. The default is all.</p>
<p><code>set aging [<time>]off</code></p> <p>Specifies the aging time to clear learned entries in seconds, complex time (hh:mm:ss) or tiny time (microseconds or milliseconds). Default is set to 60 minutes.</p>
<p><code>set aging [<time>]off</code></p> <p>Specifies the aging time to clear learned entries in seconds, complex time (hh:mm:ss) or tiny time (microseconds or milliseconds). Default is set to 60 minutes.</p>
<p><code>enable disable spantree</code></p> <p>Specifies whether the Spanning-Tree algorithm is to be enabled or disabled. The default is disable.</p>
<p><code>spantree st set bridge-priorit bp <priority></code></p> <p>Specifies the Spanning-Tree bridge priority. The default is 8000 (hex).</p>

Table A.4 - Bridge Subsystem Commands

Command and Description
spantree st sset seg-priorit sp <priority> <seglst> Specifies the Spanning-Tree segment priority. The default is 8000 (hex).
spantree st sset path-cost pc <path-cost> <seglst> Specifies the cost of the path. The default is 100 for 10Mb/s Ethernet segments, and 10 for FDDI and Fast Ethernet segments.
spantree st set maxage <time> Specifies the maximum age, in seconds. The default is 21 seconds
spantree st set hello <time> Specifies the hello time, in seconds. The default is 4 seconds.
spantree st set fwddelay <time> Specifies the forward delay, in seconds. The default is 16 seconds.
spantree st set high-util <percentage> Specifies the upper-end value of segment utilization. This value is a percentage in the range of 1 to 100. The default is 70%.
spantree st set low-util <percentage> Specifies the upper-end value of segment utilization. This value is a percentage in the range of 1 to 100. The default is 50%.

Table A.5 - DECnet Subsystem Commands

Command and Description
set max-node-num mn <value> This determines the number of nodes. The default is 255.

Table A.6 - Host Subsystem Commands

Command and Description
set kadelay kad <minutes> Specifies how many minutes a TCP (TELNET) connection remains idle before sending keep-alive packets. The default is 20 minutes.
set kainterval kai <seconds> Specifies how often keep-alive packets are sent before ending a connection. The default is 75 seconds.

Table A.7 - IP Subsystem Commands

Command and Description
<p>interface add <vlanid> <ipaddr>[/<prefixlen> <mask>][ift[ype] b[c] n[bma] p[top] <nbr_addr>]</p> <p>Allows a standard IP subnet mask to be used. If a particular network uses IP subnet addressing, then the subnet mask should be specified here using dotted-decimal notation. Otherwise, the system uses a default subnet mask equal to the “natural” subnet mask for the particular class of address.</p> <p>[br[oadcast] 0 1 Specifies the style of broadcast address on a segment-by-segment basis. The default is br1.</p> <p>[met[ric] <metric> Specifies an additional cost of using the subnet interface. The default is one.</p>
<p>route enable disable <destination> <gw-ipaddr> <metric> <segment></p> <p>Specifies whether to enable or disable IP routing. The default is disable.</p>
<p>arp set show unset age <time></p> <p><time> Specifies (in minutes) a new aging interval or turns aging off. The default is 5 minutes.</p>
<p>ping pi [-t <timeout>] [-size <size>] <ipaddr></p> <p>[-t <timeout>] Specifies how many seconds to wait for a response from the specified device. The default is 5 seconds.</p> <p>[-size <size>] Specifies the packet length. Specify any length from 64 through 1472 bytes. The default is 64 bytes.</p>
<p>ipdefaultttl ittl set <value></p> <p>Specifies the new TTL time in hops. The default is 16 hops.</p>
<p>enable disable send-icmp-redirect sir</p> <p>Specifies whether you are enabling or disabling ICMP redirect messages. The default is enable.</p>

Table A.8 - IP Multicast Subsystem Commands

Command and Description
<p>it interface add <ipaddr> [met[ric]<metric>] [thresh[old]<thresh>]</p> <p>[met[ric]<metric></p> <p>Specifies an additional cost (measured in hops to the destination) of using the interface. The default is 1.</p> <p>[thresh[old]<thresh></p> <p>Specifies the minimum time-to-live (TTL) value that an IP Multicast packet must have before it is forwarded over this interface. The default is 1.</p>
<p>tunnel add [-s] loc[al]<local-addr> rem[ote]<remote-addr> [met[ric]<mv>] [thresh[old]<tv>]</p> <p>[met[ric]<mv></p> <p>Specifies an additional cost (extra hops to the destination) of using the virtual interface with which this tunnel is associated. The default is 1.</p> <p>[thresh[old]<tv>]</p> <p>Specifies the minimum time-to-live (TTL) value that an IP Multicast packet must have before it can be forwarded through the tunnel. The default is 1.</p>
<p>enable disable ipm</p> <p>Specifies whether to enable or disable IP Multicast forwarding. The default is disable.</p>
<p>penable pdisable transmit <segment-list></p> <p>Specifies whether to enable or disable IP Multicast forwarding. The default is penable.</p>
<p>enable disable multicast-aware-bridging</p> <p>Specifies whether to enable or disable multicast-aware-bridging. The default is disabled.</p>
<p>enable disable fwd-pkts-with-srcrt-option fps</p> <p>Specifies whether enabling or disabling source-route filtering. The default is enable.</p>

Table A.9 - IP/OSPF Subsystem Commands

Command and Description
<p>asbd enable disable</p> <p>Specifies whether to enable or disable Autonomous System Border router. The default is disable.</p>
<p>auto-vlink enable disable</p> <p>Specifies whether to enable or disable the automatic virtual-link feature. The default is enable.</p>
<p>area add <area-id> [<auth-type>] [stub-area-cost sac <cost>]</p> <p><auth-type> md5 m</p> <p>Specifies that MD5 authentication is required for OSPF packets sent within this area. The default is none (no authentication).</p> <p>stub-area-cost sac <cost></p> <p>The OSPF software configures the default route automatically.</p>

Table A.10 - IP/RIP Subsystem Commands

Command and Description
<p>rip-bridging rb [enable disable]</p> <p>Enables or disables the RIP bridging feature. The default is disable.</p>

Table A.11 - IPX Subsystem Commands

Command and Description
<p>interface it add <segmentlist> <network> [mtu <mtu>] [met[ric] <metric>] [encap enet 802.3 802.2 snap]</p> <p>[mtu <mtu>] Specifies the maximum transmission unit (number of octets) for packets forwarded on this segment.</p>
<p>enable disable [ipx] Specifies whether enabling or disabling IPX forwarding. The default is disable.</p>
<p>set ripsap-ctrl rsct [normal n vlan v]</p> <p>normal n Specifies that RIP and SAP updates are generated on a per-segment basis. This is the default.</p>
<p>penable pdisable type20-port-forwarding tpfw <seglist> Specifies whether to enable or disable type-20 packet forwarding. The default is penable (enabled).</p>

Table A.12 - TFTP Subsystem Commands

Command and Description
<p>get -a fore/ph/ethan.env ethan.env Specifies net-ASCII mode. Files are transferred in binary mode by default.</p>
<p>get [-h <host>] [-a] <remote-file> [<local-file> tty] Specifies the IP address of the TFTP server in dotted-decimal notation. If no TFTP server is specified, the default server is used. The default server is specified using the set server command.</p>
<p>put [-h <host>] [-a] <localfile> [<remote-file>] Specifies the IP address of the TFTP server in dotted-decimal notation. If no TFTP server is specified, the default server is used. The default server is specified using the set server command.</p>

APPENDIX B

Netboot Options

This appendix describes the netbooting process in detail and describes how a common boot definition file can be shared among multiple PowerHubs. For additional netbooting options information, see the *PowerHub 7000/8000 Installation and Maintenance Manual*.

The PowerHub implementation of netbooting uses the Boot Protocol (BOOTP) and Trivial File Transfer Protocols (TFTP). PowerHub netbooting is designed to be fully compliant with RFCs 951, 1048 and 1350. The PowerHub can netboot over any type of Ethernet segment on a 13x1, 16x1, 4x4, 4x6 or Ethernet AMA on a Universal Ethernet Module.

After configuring the PowerHub for netbooting, the netboot process can begin by booting (or rebooting) the system, using any of the following methods:

- Press the reset switch (RST), located on the front of the Packet Engine.
- Issue the `reboot` command.
- Issue the `boot (b)` command at the `<PROM-7pe>` prompt.
- Turn the power supply off, then back on.

B.1 Choosing a Netbooting Method

The boot process differs depending on whether the client PowerHub and server are on the same subnet or different subnets. Accordingly, netbooting process depends on the network configuration.

Point-to-point netbooting can be used if the client PowerHub and the BOOTP/TFTP server are on the same subnet. The subnet can be a single segment or multiple segments connected by bridges. Point-to-point netbooting is the simplest to implement. Point-to-point netbooting is recommend when the client PowerHub and the BOOTP/TFTP server are all on the same subnet. If the client PowerHub and server are on different subnets, do one of the following:

- Implement a boot helper service on the client PowerHub subnet. The PowerHub provides a service called IP Helper which can forward UDP packets (including BOOTP packets) between a netboot client and a remote server.
- Manually configure information (such as the client and server IP addresses and the IP address of the gateway) in NVRAM. NVRAM contains a battery backup and retains its data across power cycles. The contents of NVRAM are not lost, even if the system is powered down.

The boot parameters configured in NVRAM override the corresponding parameters returned by the BOOTP server. If the PowerHub is to bypass the BOOTP process, configure all applicable boot values in NVRAM.

B.2 The Boot Process

The netbooting process takes place in the following phases:

BOOTP	BOOTP packets are exchanged. (The BOOTP phase is bypassed if the applicable boot parameters are configured in NVRAM.)
BOOTDEF	Boot definition file is received via TFTP from server and parsed. The boot definition file specifies the configuration file and system software to be used.
IMAGE	Image files (system software) are received via TFTP from server and executed.
CONFIG	Configuration file is received via TFTP from server and executed.
RUN-TIME	Normal run-time operation begins.

The last four phases are identical for each netbooting implementation. However, the first phase (BOOTP) differs according to mode of implementation. For reference, the tables in the following sections summarize the netbooting process for each method of netbooting. It is not necessary to know the netbooting phases in detail to implement netbooting, but these tables can help troubleshoot problems in netbooting implementation.

B.2.1 Point-to-Point

The following table summarizes the netbooting process used when the PowerHub and BOOTP server are on the same subnet.

Table B.1 - Point-to-Point Netbooting

Phase	Process
BOOTP	<p>BOOTP broadcast packet sent out each Ethernet segment. The BOOTP packet contains the MAC address, but no other address information.</p> <p>Server receives broadcast packet and sends BOOTP reply packet (provided the MAC address is in the bootptab file, or equivalent, on the BOOTP server). Reply packet contains server's IP address, IP address, IP subnet mask, and name of boot definition file.</p> <p>BOOTP response received and information stored from server in memory.</p> <p>During the boot process, each Ethernet segment is configured as an IP interface by default. This segment configuration has no relation to the configuration of the segments during run-time operation.</p>
BOOTDEF	<p>TFTP used to transfer boot definition file from server.</p> <p>Boot definition file parsed.</p> <p>While parsing boot definition file, names of image files and configuration file are obtained and stored in memory.</p>
IMAGE	<p>TFTP used to transfer image files from server, load them into memo, and execute them.</p>
CONFIG	<p>Name of the configuration file retrieved from memory.</p> <p>Interface (segment) received from BOOTP reply and interface configured for TFTP exchanges.</p> <p>TFTP used to transfer configuration file from server and save the file in memory.</p> <p>Interface de-configured.</p> <p>Configuration file executed.</p>
RUN-TIME	<p>Normal bridging and routing according to settings in configuration file.</p>

B.2.2 Cross Gateway--Boot Helper Service Used

The following table summarizes the netboot process used when the PowerHub and BOOTP server are on separate subnets and a boot helper service (such as IP Helper) is implemented. The intervening gateway that connects the segments can be another PowerHub or any other device that implements a boot helper service.

Table B.2 - Helper-Assisted Netbooting

Phase	Process
BOOTP	<p>PowerHub sends a BOOTP broadcast packet out each Ethernet segment. The BOOTP packet contains MAC address, but no other address information.</p> <p>BOOTP request is received by intervening gateway on a segment previously configured with an IP Helper address.</p> <p>IP Helper facility in intervening gateway forwards BOOTP packet to server.</p> <p>Server receives BOOTP request forwarded by intervening gateway and sends response packet to gateway. Response packet contains name of boot definition file, server IP address, PowerHub IP address, PowerHub IP subnet mask, and intervening gateway IP address.</p> <p>Gateway forwards response packet to client switch.</p> <p>Client switch receives BOOTP response and stores information from server in memory.</p> <p>During the boot process, each Ethernet segment is configured as an IP interface by default. This segment configuration has no relation to the configuration of the segments during run-time operation.</p>
BOOTDEF	Identical to point-to-point process.
IMAGE	Identical to point-to-point process.
CONFIG	Identical to point-to-point process.
RUN-TIME	Identical to point-to-point process.

B.2.3 Cross-Gateway--No Boot Helper Service Used

The following table summarizes the netboot process used for cross-gateway netbooting when the gateway does not have a boot helper service. If preferred, this method can be implemented even if the intervening gateway does contain a boot helper service.

Table B.3 - Cross-Gateway Netbooting — No Boot Helper Service

Phase	Process
BOOTP	<p>PowerHub uses boot parameters in NVRAM as substitute for BOOTP parameters. The following parameters can be specified in NVRAM: PowerHub IP address and subnet mask, gateway IP address, server IP address, name of the PowerHub boot definition (<code>bootdef</code>) file. The boot definition file contains the file names and pathnames of the software image files and configuration file.</p> <p>Unless all BOOTP parameters were supplied in NVRAM, PowerHub sends a BOOTP broadcast packet out each Ethernet segment configured as an IP interface. Parameters not configured in NVRAM are sought in the response from the BOOTP server.</p> <p>BOOTP request is received by intervening gateway. If boot parameters in NVRAM include information needed by gateway to forward the BOOTP packet, the packet is forwarded to the server. This information includes the PowerHub IP address and subnet mask and the server IP address.</p> <p>Server receives BOOTP request forwarded by intervening gateway and sends response packet through the gateway to the PowerHub.</p> <p>PowerHub receives BOOTP response and stores information from server, including name of boot definition file, in Boot PROM.</p> <p>During the boot process, each Ethernet segment is by default configured as an IP interface. This segment configuration has no relation to the configuration of the segments during run-time operation.</p>
BOOTDEF	Identical to point-to-point process.
IMAGE	Identical to point-to-point process.
CONFIG	Identical to point-to-point process.
RUN-TIME	Identical to point-to-point process.

B.3 Configuration Options

This section describes the configuration requirements for the BOOTP server, TFTP file server, and PowerHub for point-to-point netbooting. Implement this type of netbooting if the PowerHub, BOOTP server, and TFTP server are all attached to the same subnet. The subnet can be a single segment or multiple segments connected by bridges.

B.3.1 TFTP Server

Regardless of the netbooting method that is chosen, perform the following configuration tasks for the TFTP server (even if the BOOTP server and TFTP server are the same device):

- Install the system software image files.
- Edit and install the boot definition file(s). A separate boot definition file can be installed for each PowerHub or boot definition macros can be used to share a single boot definition file among multiple PowerHubs.
- Install a configuration file for each PowerHub.

These files can be installed in the TFTP home directory or subdirectories can be set up. If subdirectories are set up, make sure the appropriate pathnames are specified in the respective boot definition files.

B.3.2 BOOTP Server

Configure the same host device as both a BOOTP server and a TFTP server, or configure separate BOOTP and TFTP servers.



Although BOOTP and TFTP services can be provided by different hosts, using the same host results in faster booting because the PowerHub does not need to search across its interfaces multiple times for a server. In fact, some BOOTP servers do not support the file service from another host. In such cases a choice is not available.

Unless all the required values are configured in NVRAM, configure the BOOTP server to provide the following information to the PowerHub (even if the BOOTP server and TFTP server are the same device):

- Client switch IP address.
- Client switch subnet mask.
- Gateway IP address (if the client switch and server are on different subnets).
- TFTP server IP address.
- Name of the boot definition file (often called `bootdef`) used to boot the PowerHub. Install this file on the TFTP server, but specify the name on the BOOTP server in NVRAM. Note that the boot definition file is neither the image file (`7pe`) nor a configuration file (such as `cfg`).

The procedures for configuring the BOOTP server depend upon the BOOTP software being used. In some BOOTP software, a single database file contains the information items listed above for each client that uses the server. In some implementations, this file is called the `bootptab` file. See the applicable BOOTP software documentation for information.

B.3.3 Intervening Gateway

If a gateway separates the PowerHub from the server, do one of the following:

- If the gateway has a boot helper service, such as IP Helper, configure the helper service to help BOOTP packets sent from the PowerHub to reach the BOOTP server. If the gateway is another PowerHub, use the `ip add-helper` command.
- If the gateway does not have a boot helper service, configure the following values in NVRAM:
 - Client switch IP address.
 - Client switch subnet mask.
 - Gateway IP address (if on different subnets).
 - TFTP server IP address.
 - Name of the boot definition file.

B.3.4 Client PowerHub

To configure the PowerHub for netbooting:

- Specify the boot order in NVRAM. Do this regardless of the type of netbooting implemented.
- If needed, configure boot parameters in NVRAM. See the previous section.

A boot definition file contains instructions for loading the system software and configuration files used by the PowerHub when it boots. This section describes the boot definition file and boot definition macros, then explains how to edit and copy boot definition and configuration files onto a TFTP server.

The TFTP server must contain at least one boot definition file. Edit and install a separate boot definition file for each PowerHub, or share a single boot definition file among multiple PowerHubs. The installed PowerHub software contains a boot definition file called `bootdef`. This `bootdef` file supports booting from the Flash Memory Module or Compact Flash Card. It can be copied and modified for netbooting.

Following is an example of the `bootdef` file shipped with the PowerHub 7000¹:

```
%vstart 1
7pe2          m
%vend 1
```

To prepare a `bootdef` file for netbooting, copy the file shipped with the PowerHub onto the TFTP server, then modify the file as follows:

- Add or modify a line to load the configuration file. (If the configuration file has been saved before copying the `bootdef` file, the `bootdef` file already contains a line for loading the configuration file. This line needs to be modified.)
- Add the pathname and file name for the software image on the TFTP server.

Following is an example of a `bootdef` file that is modified for netbooting:

```
%vstart 1
fore/ph/configs/0000EF014A00.cfg      c
fore/ph/images/7-2.6.3.0/7pe         m
%vend 1
```

In this example, the MAC-layer hardware address of the PowerHub is used as the configuration file name. The pathnames for the configuration file and the software image file are included with the file names. Whether a pathname is specified depends on how the TFTP server is configured. When editing the `bootdef` file, make sure the pathnames entered are meaningful to the TFTP server.

¹ Some boot definition files might contain the lines `%vstart 0` and `%vend 0`. These lines are used for booting from a floppy diskette (PowerHub 7000 only) and do not work for netbooting. Ensure the boot definition files used for netbooting use `%vstart 1` and `%vend 1`.

B.3.5 Using the Same Boot Definition File with Multiple Switches

If only one PowerHub needs to be configured for netbooting, using the MAC-layer hardware address of the PowerHub to name the configuration file is a simple way to name the file. However, if more than one is being configured for network booting, do one of the following:

- Create a unique boot definition file for each PowerHub. If this method is chosen, use the `nvram set netbdfile` command to set the boot definition file in NVRAM of each PowerHub. Otherwise, each PowerHub attempts to use the default boot definition file name (`bootdef`).
- Use a single boot definition file, but use boot definition macro commands in place of the configuration file name. A boot definition macro *command* is a 2-character sequence consisting of a '\$' followed by a letter. The macro commands are expanded by the PowerHub Packet Engine Boot PROM when the boot definition file is read. Table B.4 lists the boot definition macro commands.

Table B.4 - Boot Definition Macro Commands

Macro Command	Process
\$E	ASCII representation of the MAC-layer hardware address; for example, "0000EF014A00." \$E always expands to 12 characters.*
\$e	ASCII representation of the three least significant octets of the MAC-layer hardware address, for example "014A00." \$e always expands to 6 characters.*
\$D	Directory part of the path name of the boot definition file.
\$B	Base name of the boot definition file (the directory part of the path name and anything following the rightmost dot of the file name are removed).
\$\$	Expands to a single '\$' character. Use this if the '\$' character with a boot definition macro is used.
*\$E and \$e expand hex digits A-F in uppercase	

Following is an example of a boot definition file that uses boot definition macro commands:

```
%vstart 1
%D/$E.cfg                                c
fore/ph/images/7-2.6.3.0/7pe            m
%vend 1
```

In this example, the `$D` expands into the pathname of the boot definition file. The `$E` expands into the MAC-layer hardware address. When that Packet Engine parses the boot definition file, it expands `$D` into `fore/ph/configs/` and `$E` into `0000EF014A00`.

B.3.6 Sharing Methods

If a common boot definition file is shared among multiple PowerHubs, decide on one of the following sharing methods:

where

MAC-address Each configuration file is named according to the MAC-layer hardware address of the PowerHub.

or

Link On boot servers that support symbolic links to files, give meaningful names to configuration files.

These methods are very similar. They differ only in that any name can be used for the configuration files if the link method is used. However, the MAC-layer hardware addresses must be used in the configuration file names if the MAC-address method is used.



To use the link method, the TFTP server must support symbolic links. Refer to the TFTP server documentation to determine if the TFTP server supports symbolic links.

The following sections contain examples of each sharing method.

B.3.6.1 MAC-Address Method

Here is an example of a TFTP server directory and file structure used to implement the MAC-address sharing method.

```
fore
  ph
    images
      7-2.6.4.0
        7pe
      7-2.6.4.1
        7pe
      FT_5.0
        7pe
    configs
      bootdef-7-2.6.4.0
      bootdef-7-2.6.4.1
      bootdef-FT_5.0
      014A00.cfg
      015AD0.cfg
      015AE0.cfg
      015AF0.cfg
      016AD0.cfg
```

This example shows TFTP subdirectories, but the files could just as easily be stored in the TFTP home directory. As shown in this example, the configs subdirectory contains a single boot definition and configuration files for multiple system software versions, but separate configuration files for each PowerHub using this TFTP server. Each configuration file is named after the last six hexadecimal digits of the MAC-layer hardware address of the respective PowerHub.

Recall that the boot definition file contains the name of the configuration and system software image files to be loaded. The BOOTP server tells the PowerHub which boot definition file to use. The boot definition macros are expanded to form the unique name of the configuration file for that PowerHub.

To name configuration files according to the MAC-address sharing method, use the following procedure for each PowerHub using the server.

For each configuration file, copy the file onto the server:

```
<MAC-addr>.cfg
```

where

<MAC-addr>.cfg Specify the MAC-layer hardware address of the PowerHub. The full hardware address (12 hex digits) or, for systems that do not support long file names, the last three octets (six hex digits) of the hardware address can be specified. Alphabetic hex characters **must** be in uppercase.

Netboot Options

When this procedure is complete, the TFTP server should contain separate configuration files for each PowerHub using the server to netboot. Following is an example of a TFTP server directory and file structure used to implement the link sharing method.

```
fore
  ph
    images
      7-2.6.4.0
        7pe
      7-2.6.4.1
        7pe
      FT_5.0
        7pe
    configs
      bootdef-7-2.6.4.0
      bootdef-7-2.6.4.1
      bootdef-FT_5.0
      ph-1.bd
      ph-2.bd
      ph-3.bd
      ph-1.cfg
      ph-2.cfg
      ph-3.cfg
```

This example shows a TFTP subdirectory structure, but the files could just as easily be stored in the TFTP home directory. As shown in this example, the configs subdirectory contains a single boot definition file for each version of runtime software, a configuration file for each client PowerHub using this TFTP server and a link to each configuration file. Each link has the same base name as the corresponding configuration file with a `.bd` extension. This base name is associated with a particular client switch by a bootptab file or other file containing IP information and other information needed by the BOOTP software. See the documentation for the file server to determine how to associate link names with configuration files.

To name the configuration files according to the link sharing method, use the following procedure for each client PowerHub using the server.

For each configuration file, copy the file into the `/fore/ph/configs` directory as shown below:

`<name>.bd`

where

<name>.bd Specify a meaningful name representing the file. Any legal file name can be used.

When this procedure is complete, the TFTP server should contain a separate configuration file for each client PowerHub using the server.

Index

Numerics

- 10/100 FEMA 7 - 16
 - values 7 - 17
- 802.1d 1 - 9

A

- AppleTalk routing 1 - 10
- ATM modules 1 - 3
- auto-negotiation
 - 10/100 FEMA 7 - 17

B

- bandwidth 7 - 17
- boot screen display 2 - 1
- bridge cache
 - description 1 - 9
- bridge commands 10 - 2
 - aging 10 - 3
 - allocate memory 10 - 12
 - bridge groups 10 - 13
 - bridge table 10 - 6
 - bridging 10 - 4
 - cache 10 - 9
 - configuration 10 - 10
 - IPX bridge translation 10 - 15
 - learning 10 - 18
 - relearn log 10 - 19
 - spanning tree 10 - 20
 - statistics 10 - 23
 - status 10 - 24

- bridge table and cache 1 - 8

C

- collisions 7 - 16
- comamnd syntax
 - using x
- commands
 - media operating-mode 7 - 17

D

- DECnet routing 1 - 10

E

- Ethernet modules 1 - 3, 7 - 16

F

- Fast Ethernet modules 1 - 4, 7 - 16
- FDDI commands 11 - 1
 - concentrator 11 - 2
 - dual attach concentrator 11 - 3
 - MIB variables 11 - 8
 - nonvolatile RAM 11 - 3
 - reset count 11 - 7
 - statistics 11 - 10
 - target token rotation time 11 - 5
 - time transmission variable 11 - 6
- FDDI modules 1 - 3
- ForeView 1 - 12
- full-duplex mode 7 - 16, 7 - 17

G

global commands	5 - 1
accessing	5 - 1
alias	5 - 1
checksum	5 - 2
copy	5 - 3
default-device	5 - 4
directory	5 - 4
format	5 - 5
help	5 - 6
history	5 - 7
history characters	5 - 7
logout	5 - 8
port number mode	5 - 9
read environment	5 - 10
remove	5 - 11
rename	5 - 11
return code prompt	5 - 9
save environment	5 - 12
set tty	5 - 13
set user	5 - 14
show configuration example	5 - 12
subsystems	5 - 14
timed command	5 - 15
type	5 - 16
unalias	5 - 16

H

half-duplex mode	7 - 16
host commands	9 - 2
config	9 - 3
keep alive delay	9 - 5
keep alive interval	9 - 6
kill	9 - 7
statistics	9 - 8
status	9 - 10

I

intelligent modules(INIMs)	1 - 2
intelligent packet switching	1 - 1
IP routing	1 - 9
IPX routing	1 - 10
IPX translation bridging	1 - 9

L

line speed	7 - 17
------------------	--------

M

media commands	7 - 1
config	7 - 2
configuring packet forwarding	7 - 22
Ethernet LED modes	7 - 5
inter-segment statistics	7 - 4
monitoring segments	7 - 14
operating mode	7 - 16
port level statistics	7 - 20
port monitoring	7 - 7
segment names	7 - 23
segment state detection	7 - 24
segment state detection threshold ..	7 - 28
statistics	7 - 31
status	7 - 30
UDP port receiver status	7 - 20
media operating-mode command	7 - 17

N

network interface modules	
description	1 - 2
network management features	1 - 12
ForeView	1 - 12
management information base	1 - 12
network management system	1 - 12

NVRAM commands	8 - 1
boot order	8 - 1
crash reboot	8 - 5
file server IP address	8 - 3
gateway IP address	8 - 4
my internet protocol address	8 - 2
my subnet mask	8 - 3
slot segments	8 - 5

O

on-line help	4 - 7
set	4 - 9
show	4 - 10
syntax	4 - 8
operating mode	7 - 16
OSPF	1 - 10

P

packet engine	
description	1 - 1
packet engine 1	1 - 2
packet engine 2	1 - 2
packet modifications	7 - 9
port monitoring	
description	7 - 7
performance considerations	7 - 8
port-level statistics	
displaying	7 - 21

R

RIP	1 - 10
-----	--------

S

segment state detection	
automatic	7 - 24
methods	7 - 25
100Base-FX	7 - 25
100Base-TX	7 - 25
10Base-FB	7 - 25
10Base-FL	7 - 25
ATM	7 - 25
AUI cable	7 - 25
FDDI	7 - 25
MAU	7 - 25
UTP	7 - 25
segment-state detection	
10Base-T	7 - 26
SNMP	
using SunNet manager	12 - 13
SNMP commands	12 - 2
community	12 - 3
configuration	12 - 7
manager	12 - 11
statistics	12 - 9
software features	1 - 5
automatic segment-state detection	1 - 7
boot sources	1 - 6
bridging and routing	1 - 8
command line interface	1 - 6
concurrent command line sessions	1 - 6
configuration files	1 - 6
file management system	1 - 6
multiprocessor optimization	1 - 5
parameter files	1 - 7
route protocol statistics	1 - 11
security filters	1 - 11
segment statistics	1 - 8
traffic monitoring	1 - 8
virtual local area networks	1 - 8

Index

- software subsystems
 - firmware 2 - 1
 - runtime software 2 - 4
- spanning-tree 1 - 9
- syntax 4 - 4
 - nouns 4 - 5
 - verbs 4 - 4
- system commands 6 - 1
 - accessing 6 - 2
 - baud 6 - 2
 - bootinfo 6 - 3
 - card swap 6 - 3
 - config 6 - 5
 - convert config 6 - 6
 - data carrier detect 6 - 7
 - date 6 - 6
 - Ethernet address 6 - 8
 - identification prom 6 - 8
 - memory 6 - 9
 - password 6 - 10
 - read configuration 6 - 11
 - reboot 6 - 11
 - save configuration 6 - 12
 - system location 6 - 12
 - system name 6 - 13
 - temperature 6 - 13
 - tty2 6 - 14
 - uptime 6 - 15
 - version 6 - 15
- system files
 - created files 3 - 4
 - other files 3 - 4
 - software 3 - 3
 - types 3 - 1
- T**
- Technical Assistance Center
 - contacting iv
- Telnet commands 14 - 1
 - close 14 - 3
 - open 14 - 2
 - status 14 - 4
- TFTP commands 13 - 6
 - get 13 - 7
 - put 13 - 9
 - read configuration 13 - 12
 - save configuration 13 - 13
 - server 13 - 6
- TFTP considerations 13 - 3
 - file naming conventions 13 - 4
 - pathnames 13 - 4
 - read/write permissions 13 - 3
 - remote file names 13 - 5
- Traps
 - Enterprise-Specific Traps 12 - 6
 - Standard Traps 12 - 4
- U**
- uration A - 1
- user interface 4 - 1
 - entering/editing 4 - 3
 - runtime prompt 4 - 1