



BBS on ATCA-F140 with SRstackware® for WR Linux

Programmer's Reference

P/N: 6806800N23X

August 2022



Legal Disclaimer*

SMART Embedded Computing, Inc. (SMART EC), dba Penguin Solutions™, assumes no responsibility for errors or omissions in these materials. **These materials are provided "AS IS" without warranty of any kind, either expressed or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.** SMART EC further does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within these materials. SMART EC shall not be liable for any special, indirect, incidental, or consequential damages, including without limitation, lost revenues or lost profits, which may result from the use of these materials. SMART EC may make changes to these materials, or to the products described therein, at any time without notice. SMART EC makes no commitment to update the information contained within these materials.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to a SMART EC website. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of SMART EC.

It is possible that this publication may contain reference to or information about SMART EC products, programming, or services that are not available in your country. Such references or information must not be construed to mean that SMART EC intends to announce such SMART EC products, programming, or services in your country.

Limited and Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by SMART EC.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data clause at DFARS 252.227-7013 (Nov. 1995) and of the Rights in Noncommercial Computer Software and Documentation clause at DFARS 252.227-7014 (Jun. 1995).

SMART Embedded Computing, Inc., dba Penguin Solutions

2900 S. Diablo Way, Suite 190

Tempe, Arizona 85282

USA

*For full legal terms and conditions, visit <https://www.penguinsolutions.com/edge/legal/>

Table of Contents

About this Manual	13
1 Introduction	21
1.1 Software Building Blocks	21
2 Installing Basic Blade Services Software	25
2.1 Package Information	25
2.2 Configuring a TFTP Server	27
2.2.1 Create /tftpboot Directory and Copy Target Files	27
2.3 Configuring ATCA-F140 for Network Booting	28
2.4 Booting the ATCA-F140 Using a RAM Disk	29
2.5 SATA and USB Disk Installation Procedure	30
2.5.1 Configuring the Boot Parameters for Network Installation	30
2.5.2 Installing BBS on a SATA or USB Device	31
2.5.3 Configuring ATCA-F140 to Boot From SATA or USB Drive	41
2.6 NFS Installation Procedure	42
2.6.1 Configuring the NFS Server	43
2.6.2 Copying and Installing the Root File System	43
2.6.3 Configuring the Boot Parameters for NFS	43
2.6.4 Installing the BBS RPMs	44
2.7 Upgrading the FPGA Firmware	45
2.8 Upgrading the BCM 8727/84740/84754 PHY Microcode	46
3 Licensed Software Modules	49
3.1 Installation	49
4 Firmware Upgrade Facility	55
4.1 Firmware Recovery Image Files	55
4.2 Backup Concept	55
4.3 fcu–Firmware Upgrade Command-Line Utility	56
4.4 Upgrading Firmware	60
4.4.1 Upgrading CPU Firmware	60
4.4.2 Upgrading the Device-Tree Blob	61
4.4.3 Upgrading the Kernel	61

Table of Contents

4.4.4	Activating the Backup Boot Bank	61
4.4.5	Upgrading IPMC Boot Loader Firmware	62
4.4.6	Upgrading IPMC Firmware.	63
5	Linux Distribution Description	65
5.1	Reliability	65
5.2	Login	65
5.3	Network Services Configuration	65
5.3.1	ATCA-F140 Ethernet Interfaces.	66
5.3.2	DHCP Server	67
5.3.3	TFTP Server.	67
5.3.4	SSH	67
5.4	Log File Management	68
5.4.1	Log Files.	68
6	Hardware Platform Management.	71
6.1	hpmagentd–HPM Agent Daemon	72
6.2	hpm–Start-Up Script	74
6.3	hpm–Shutdown and Reboot Scripts	74
6.4	hpmcmd–HPM Command Utility	75
6.4.1	Command Overview.	77
6.4.2	Supported Commands	79
6.4.2.1	bye	79
6.4.2.2	bootbankget	79
6.4.2.3	bootbankset	80
6.4.2.4	cmd	80
6.4.2.5	deviceid	81
6.4.2.6	ekeydownpath	81
6.4.2.7	ekeyuppath	81
6.4.2.8	exit	82
6.4.2.9	frudata	82
6.4.2.10	fruinfoget	82
6.4.2.11	fruinvent	83
6.4.2.12	fruread	84
6.4.2.13	fruwrite	84
6.4.2.14	help	85
6.4.2.15	ipmbaddress	85

6.4.2.16	ipmcdevice	85
6.4.2.17	ledget	85
6.4.2.18	ledprop	86
6.4.2.19	ledset	86
6.4.2.20	loglevelget	88
6.4.2.21	loglevelset	88
6.4.2.22	macaddress	89
6.4.2.23	motshelftype	89
6.4.2.24	muxctrl	89
6.4.2.25	partnumber	90
6.4.2.26	physlotnumber	90
6.4.2.27	portget	90
6.4.2.28	portset	91
6.4.2.29	postypeget	92
6.4.2.30	postypeset	93
6.4.2.31	quit	93
6.4.2.32	rebootpath	93
6.4.2.33	sdr	93
6.4.2.34	sdr_dump	94
6.4.2.35	sendcmd	95
6.4.2.36	sdrinfo	95
6.4.2.37	sendamc	96
6.4.2.38	shelfaddress	96
6.4.2.39	shelfslots	96
6.4.2.40	shutdownpath	97
6.4.2.41	slotmap	97
6.4.2.42	slotnumber	98
6.4.2.43	version	98
6.4.2.44	watchdog	98
7	Link Health Check	101
7.1	LHC MIB	101
7.1.1	Browsing the LHC MIB	101
7.1.2	Enabling SNMP Access to the LHC MIB	102
7.2	lhcd-LHC Daemon	102
7.3	lhc-lhcd Start-Stop Script	104
7.4	lhc-LHC Command Line Utility	105
7.4.1	Command Overview	105

Table of Contents

7.4.2	Command Usage for Common Parameters	106
7.4.3	Available Commands	108
7.4.3.1	exit q quit	108
7.4.3.2	get	108
7.4.3.3	help	108
7.4.3.4	index	109
7.4.3.5	loglevel	109
7.4.3.6	next	110
7.4.3.7	objects	110
7.4.3.8	set	110
7.4.3.9	tables	111
7.4.3.10	values	112
7.4.3.11	version	112
7.4.3.12	walk	112
7.5	LHC Configuration Script and Configuration File	113
7.6	LHC Fault Management	113
7.6.1	Determining to Change Active Network Plane	113
7.6.2	Controlling the Active Network Plane	113
7.6.3	Monitoring for Changes to the Active Network Plane	114
7.6.4	Fault Management Notifications	114
7.7	LHC Configuration Examples	116
7.7.1	Proctor Configuration Examples	117
7.7.2	Responder Configuration Examples	118
7.7.3	Baseboard Processor on Carrier Blade Configuration Example	119
7.7.4	AMC/PMC Module on Carrier Blade Configuration Example	121
7.7.5	Leaky Bucket Description and Configuration Example	121
7.8	LHC Configuration for ATCA-F140	122
8	Watchdog Behavior	125
8.1	BMC Watchdog	125
8.2	FPGA Watchdog	126
8.3	Boot Bank Switch on BMC Watchdog Timeout	126
9	Trident Reconfiguration Utility	129
9.1	Input Files	131
9.2	Customize Trident Configuration	135
9.2.1	Backups	137

9.2.1.1	Restoring a backup	137
9.3	SRS Initialization	137
10	Fabric Configuration	139
10.1	Fabric Interface Configuration Service	139
10.1.1	Usage	139
10.1.2	Logging	139
10.2	Configuration Files	140
10.2.1	slotconfig	140
10.2.2	fabconfig.rc	142
10.2.3	Blade Specific Configuration File	143
10.3	Supported Blades	144
10.3.1	ATCA-9405	144
10.3.2	ATCA-7470	144
11	SRstackware	145
11.1	Configuring SRstackware	145
11.2	Default Configuration	146
11.2.1	1440 and 14-100 SLOT1 Configuration	147
11.2.2	1440 and 14-100 SLOT2 Configuration	150
11.2.3	14-400 SLOT1 Configuration	153
11.2.4	14-400 SLOT2 Configuration	157
11.2.5	14-400 SLOT3 Configuration	160
11.2.6	14-400 SLOT4 Configuration	163
11.2.7	640 SLOT1 Configuration	166
11.2.8	640 SLOT2 Configuration	169
11.2.9	640-C04 SLOT1 Configuration	172
11.2.10	640-C04 SLOT2 Configuration	175
11.3	Additional Configuration	178
11.3.1	SLOT1 Configuration	178
11.3.2	SLOT2 Configuration	178
11.3.3	Mapping Physical RTM Interfaces to SRstackware Ports	179
11.3.4	ATCA-F140 Switch Configuration Examples	180
11.4	Custom Configuration	182
11.4.1	Enabling Protocols	183
11.4.2	Enabling RTM ports	183
11.4.3	Sample Configurations	183

Table of Contents

11.5	SNMP Usage Guidelines	204
11.6	SNMP Traps Usage Guidelines	205
11.6.1	Configuring Agent to Send SNMP Notifications	205
11.6.2	Configuring SNMP Manager to Receive SNMP Notifications	206
11.7	Configuring Logs	207
11.8	Switch Ports	207
11.8.1	Base Switch Port	207
11.8.2	Fabric Switch Port	208
12	HPI-B Software	209
13	SYSSTATMGR	211
13.1	Installation	212
13.2	Monitors	213
13.2.1	cpuMonitor	213
13.2.2	memMonitor	213
13.2.3	dfMonitor	214
13.2.4	fcuMonitor	214
13.2.5	hpmMonitor	214
13.2.6	fpgaMonitor	214
13.2.7	srsMonitor	215
13.2.8	tclkMonitor	215
A	MUXes Settings	217
A.1	Overview	217
A.2	FPGA MUX Control Registers	219
B	SFP Module Configuration	221
B.1	Overview	221
B.2	Tri-Speed Copper	221
B.3	Dual-rate SFP+ modules	221
C	Related Documentation	223
C.1	Penguin Solutions Documentation	223

List of Figures

Figure 1-1	BBS Architecture	21
Figure 6-1	Software Levels of the HPM Architecture	72
Figure 11-1	ATCA-F140 Switch Management Fabric Interface Bridge Configuration	180
Figure 11-2	ATCA-F140 Switch Management Base Interface Bridge Configuration	181
Figure 11-3	AXP14-400-ATCA-F140 Switch Management Fabric Interface Bridge Configuration	182
Figure 11-4	Base Switch Port Mapping	207
Figure 11-5	Fabric Switch Port Mapping	208

List of Figures

List of Tables

Table 2-1	BBS Distribution Files	25
Table 2-2	rootpartition U-Boot Environment Variable	42
Table 6-1	Command Overview	77
Table 7-1	ATCA-F140 Base Interface LHC Configuration	122
Table 7-2	ATCA-F140 Fabric Interface LHC Configuration	124
Table 8-1	Boot Bank Switch on BMC Watchdog Timeout	127
Table 9-1	Default Fabric Configuration-1440 Chassis	129
Table 9-2	Default Fabric Configuration-640 Chassis	130
Table 11-1	Mapping Physical RTM Interfaces to SRstackware Ports	179
Table 13-1	Penguin Edge OEM Sensor Type Codes	211
Table 13-2	Directory Structure of SYSSTAT Manager	212
Table A-1	Mux Control Signals	217
Table A-2	Mux Lock and SATA Mux Control Register (offset 0x08)	219
Table A-3	Ethernet Mux Control Register 2 (offset 0x0B)	219
Table C-1	Penguin Edge Documentation	223

List of Tables

About this Manual

Overview of Contents

This manual is divided into the following chapters and appendices.

Chapter 1, Introduction on page 21

Chapter 2, Installing Basic Blade Services Software on page 25

Chapter 3, Licensed Software Modules on page 49

Chapter 4, Firmware Upgrade Facility on page 55

Chapter 5, Linux Distribution Description on page 65

Chapter 6, Hardware Platform Management on page 71

Chapter 7, Link Health Check on page 101

Chapter 8, Watchdog Behavior on page 125

Chapter 9, Trident Reconfiguration Utility on page 129

Chapter 10, Fabric Configuration on page 139

Chapter 11, SRstackware on page 145

Chapter 12, HPI-B Software on page 209

Appendix A, MUXes Settings on page 217

Appendix B, SFP Module Configuration on page 221

Appendix C, Related Documentation on page 223

Abbreviations

This document uses the following abbreviations:

Abbreviation	Definition
AdvancedTCA, ATCA	Advanced Telecommunications Computing Architecture
AgentX	Agent Extensibility Protocol
AMC	Advanced Mezzanine Card
API	Application Programming Interface
ATIS	AdvancedTCA Interfaces Service
BBS	Basic Blade Services

About this Manual

Abbreviation	Definition
BC	Base Channel
BCM	Baseboard Management Controller
BIB	Board Information Block
BIOS	Basic Input Output System
BITS	Building Integrated Timing Supply
CGL	Carrier Grade Linux
CGM	Clock Generator Module
CMC	Common Mezzanine Card
CRC	Cyclic Redundancy Check
DAT	Domain Alarm Table
DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
ECC	Embedded Communications Computing
ER	Expected Responder
EST	Eastern Standard Time
EVQ	Event Queue
FC	Fabric Channel
FCU	FUF Command Line Utility
FM	Fault Management
FPGA	Field Programmable Gate Array
FRI	Firmware Recovery Image
FRU	Field Replaceable Unit
FUF	Firmware Upgrade Facility
FWH	Firmware Hub
GPIO	General Purpose Input/Output
HPI	Hardware Platform Interface
HPI-B	Hardware Platform Interface Version B

Abbreviation	Definition
HPM	Hardware Platform Management
IDE	Integrated Device Electronics
IPM	Intelligent Platform Management
IPMB	Intelligent Platform Management Bus
IPMC	Intelligent Platform Management Controller
IPMI	Intelligent Platform Management Interface
KCS	Keyboard Control Style
LCCB	Line Card Control Block
LHC	Link Health Check
LIUs	Line Interface Units
LSP	Linux Support Package
LT	Lower Threshold
LUN	Logic Unit Number
MAC	Media Access Control
MIB	Management Information Base
MMC	Module Management Controller
NTP	Network Time Protocol
OIDs	Object Identifiers
OEM	Original Equipment Manufacturer
OSDL	Open Source Development Labs
PBO	Phase Build Out
PCI	Peripheral Component Interconnect
PCIEX	PCI Express
PEM	Power Entry Module
PICMG	PCI Industrial Computers Manufacturers Group
PMC	PCI Mezzanine Card
PNE	Platform for Network Equipment

About this Manual

Abbreviation	Definition
PRC	Primary Reference Clock
PrPMC	Processor PMC
PTP	Precision Time Protocol
PXE	Preboot Execution Environment
RDR	Resource Data Record
RMCP	Remote Monitoring and Control Protocol
RPM	RedHat Package Manager
RTM	Rear Transition Module
SAF	Service Availability Forum
SAM	Shelf Manager and Alarm Module
SAS	Serial Attached SCSI
SATA	Serial ATA
SCSI	Small Computer System Interface
SCXB	System Controller Switching Blade
SDH	Synchronous Digital Hierarchy
SDK	Software Development Kit
SDR	Sensor Data Record
SIP	Serial Interface Protocol
SIT	Simple Internet Transition
SMI	Serial Management Interface
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Networking
SPI	Serial Peripheral Interface
SSH	Secure Shell
SSU	Synchronization Supply Unit
SyncE	Synchronous Ethernet
TAR	Tape Archiver








Abbreviation	Definition
TCM	Telco Clocking Module
TDM	Time Division Multiplex
TFTP	Trivial File Transfer Protocol
TTY	Teletypewriter
UC	Update Channel
UDP	User Datagram Protocol
UT	Upper Threshold
WDT	Watchdog Timer
X-Connect	Backplane cross connection between the Hub slots

Conventions

The following table describes the conventions used throughout this manual.

Notation	Description
0x00000000	Typical notation for hexadecimal numbers (digits are 0 through F), for example used for addresses and offsets
0b0000	Same for binary numbers (digits are 0 and 1)
bold	Used to emphasize a word
Screen	Used for on-screen output and code related elements or commands. Sample of Programming used in a table (9pt)
Courier + Bold	Used to characterize user input and to separate it from system output
<i>Reference</i>	Used for references and for table and figure descriptions
File > Exit	Notation for selecting a submenu
<text>	Notation for variables and keys
[text]	Notation for software buttons to click on the screen and parameter description
...	Repeated item for example node 1, node 2, ..., node 12

About this Manual

Notation	Description
.	Omission of information from example/command that is not necessary at the time
..	Ranges, for example: 0..4 means one of the integers 0,1,2,3, and 4 (used in registers)
	Logical OR
	Indicates a hazardous situation which, if not avoided, could result in death or serious injury
	Indicates a hazardous situation which, if not avoided, may result in minor or moderate injury
	Indicates a property damage message
	Indicates a hot surface that could result in moderate or serious injury
	Indicates an electrical situation that could result in moderate injury or death
<p data-bbox="272 1286 386 1338">Use ESD protection</p> 	Indicates that when working in an ESD environment care should be taken to use proper ESD practices
	No danger encountered, pay attention to important information

Summary of Changes

See the table below for manual revisions and changes.

Part Number	Date	Description
6806800N23X	August 2022	Rebrand to Penguin Solutions.
6806800N23W	November 2019	Rebrand to SMART Embedded Computing template.
6806800N23V	September 2017	Added AXP640-C04 chassis details in Input Files on page 158 and Default Configuration on page 176 and updated the sections 640-C04 SLOT1 Configuration on page 200 and 640-C04 SLOT2 Configuration on page 203.
6806800N23U	July 2017	Added registered trademark to SRstackware.
6806800N23T	June 2016	Re-branded to new template.
6806800N23S	July 2015	Updated Chapter 9, <i>Trident Reconfiguration Utility</i> , on page 153.
6806800N23R	May 2015	Updated <i>Default Configuration</i> , on page 170 and <i>Overview</i> , on page 57.
6806800N23P	February 2015	Added a new section <i>Log File Management</i> on page 78, updated the section <i>DHCP Server</i> on page 77, modified <i>Figure 11-1</i> on page 197, <i>Figure 11-2</i> on page 198 and <i>Figure 11-3</i> on page 199.
6806800N23N	December 2014	Updated <i>Upgrading IPMC Boot Loader Firmware</i> . Added figure <i>AXP14-400-ATCA-F140- Switch Management Fabric Interface Bridge Configuration</i> . Added <i>Licensed Software Module Chapter</i> and removed <i>License Management Chapter</i> . Updated the sections 5.3 Login and 5.4.3 TFTP server.
6806800N23M	May 2014	Updated Contact Address on page 2.
6806800N23L	March 2014	Updated <i>Chapter 9, Trident Reconfiguration Utility</i> , on page 402 and changed to Artesyn template.
6806800N23K	December 2013	Updated Chapter 12, ViewCheck, on page 459.
6806800N23J	November 2013	Updated Chapter 6, Link Health Check, on page 121 added Appendix C, Related Documentation.
6806800N23H	August 2013	Updated Chapter 2, Installing BBS on a SATA or USB Device and Configuring the ATCA-F140 to Boot from SATA or USB Drive.

About this Manual

Part Number	Date	Description
6806800N23G	June 2013	Updated Figure 1-1, Table 2-1, License Management, Table 13-1, Obtaining the Activation Record, Fabric Configuration, SRstackware, Appendix A, MUXes Settings, Chapter 15, SYSSTATMGR. Added Chapter 7, Watchdog Behavior, Chapter 10, Fabric Configuration, Chapter 12, ViewCheck, Chapter 15, SYSSTATMGR, on page 475.
6806800N23F	April 2013	Updated License Management, Obtaining the Activation Record, Fabric Configuration, SRstackware, Appendix A, MUXes Settings, Chapter 15, SYSSTATMGR. Added Chapter 7, Watchdog Behavior, Chapter 10, Fabric Configuration.
6806800N23E	February 2013	Updated Chapter 13, Obtaining the Activation Record, on page 464.
6806800N23D	December 2012	Updated Chapter 8, Telco Clocking Module, on page 157 and Chapter 13, Obtaining the Activation Record, on page 464.
6806800N23C	August 2012	Added Chapter 13, License Management, on page 465.
6806800N23B	June 2012	Updated Chapter 8, Telco Clocking Module, on page 157.
6806800N23A	February 2012	Initial Release

Introduction

This manual is applicable to the part number SA-BBS-WR30-F140.

The Basic Blades Services (BBS) software provides a set of services that support the blade on which the software is installed.

BBS includes:

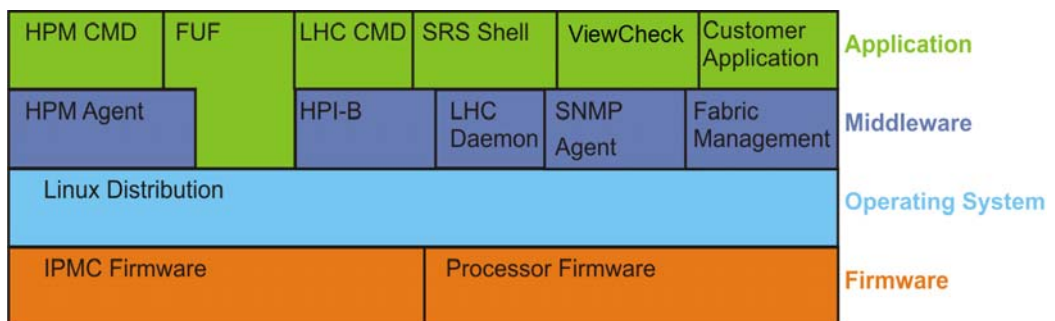
- Wind River Platform for Network Equipment, Linux Edition 3.0.
- Several custom hardware management functions for the unique hardware of the blade.
- A set of management routines for Linux and all hardware interfaces. Management access includes support for SNMP and a local console interface based on a standard Linux command shell.

1.1 Software Building Blocks

BBS services include a common set of functionality which is available for all AdvancedTCA blades and AMC modules, and a unique set of functionality which is tailored to a particular blade or module.

The figure below depicts the architecture of the BBS software.

Figure 1-1 BBS Architecture



HPM: Hardware Platform Management
 LHC: Link Health Check
 CMD: Command Line Tool
 FUF: Firmware Upgrade Facility
 HPI: Hardware Platform Interface

Introduction

BBS consists of the following software and services:

- **Firmware Upgrade Facility (FUF)**
The Firmware Upgrade Facility (FUF) provides a uniform way to upgrade firmware on Penguin Edge blades and AMC modules, regardless on which flash locations the firmware is stored. FUF upgrades the U-Boot firmware as well as the IPMC firmware (via HPM agent). The FUF currently consists of a Firmware Upgrade Command Line Utility (FCU), flash device drivers, and specially prepared firmware recovery image files. The FUF can be used on switch and node blades and on AMC modules.
- **Linux**
This BBS is based on Wind River Platform for Network Equipment, Linux Edition 3.0. Various Linux services (above the kernel) will be activated by the BBS installation scripts.
- **Hardware Platform Management (HPM)**
The HPM in AdvancedTCA systems is based on Intelligent Platform Management Interface specification (IPMI). IPMI commands can be complex and cumbersome. Using a certain set of commands, HPM facilitates the blade or module-level hardware management.
- **Link Health Check**
The Link Health Check (LHC) supports the configuration and operation of the LHC protocol. LHC performs the verification of Layer 2 connectivity and the distribution of the active network plane.
- **SRstackware®**
SRstackware package supports configuration and operation of switch devices through various protocols that it provides.

NOTE: Basic SRstackware features are provided with BBS distribution.

The following table lists the licensed functions of SRstackware.

Ordering Part Number	Product Name
SL-L3F140-01-001-STD	SRS Layer 3
SL-FPF140-04-001-STD	FlowPilot™

You need to purchase the paper based license for each function separately. These functions are released as separate RPMs, which can be installed on top of the ATCA-F140 BBS package which have SRstackware Base RPM.

To purchase the licenses, contact your local Penguin Solutions sales representative.

- HPI-B
This release contains an RPM which has all necessary files for developing HPI-B applications. For further information refer to the *System Management Interface Based on HPI-B (Centellis 31K/4100) User's Guide*.
- SNMP Agent
As each BBS blade or module is individually managed, the default installation script installs and initializes the "Net-SNMP" agent.
- ViewCheck
ViewCheck is a licensed software, which will intimate the user about the current state of devices through XML and CLI interfaces.

NOTE: ViewCheck is not part of basic BBS distribution.

The following table provides the ordering information for ViewCheck.

Ordering Part Number	Product Name
SL-DGF140-01-001-STD	ViewCheck™ for ATCA-F140

Installing Basic Blade Services Software

This chapter provides information on how to install the BBS software on ATCA-F140.

Consult your local Penguin Solutions sales representative on how to obtain BBS images.

2.1 Package Information

The BBS software is packaged with the Red Hat Package Manager (RPM) and it is installed as a part of the standard installation. In general, you will not need to install or upgrade an individual package.

The following table lists all files which constitute the BBS package. These files may vary from release to release.

Table 2-1 BBS Distribution Files

File Name	Description
kernel	Operating System Kernel (version 2.6.27)
Root File System	
nfs.tar.gz	Root file system
BBS RPM Files	
bbs-atcaf140-cpu-<version>.rpm	CPU firmware (U-Boot)
bbs-atcaf140-dtb-<version>.rpm	Device Tree Binary
bbs-atcaf140-ipmc-<version>.rpm	IPMC firmware
bbs-atcaf140-ipmc-boot-<version>.rpm	IPMC Boot Loader
bbs-atcaf140-kernel-<version>.rpm	Kernel image
bbs-fuf-atcaf140-<version>.rpm	Firmware upgrade facility
bbs-hpib-<version>.rpm	HPI-B package
bbs-hpib-clientsrc-<version>.rpm	HPI-B client package
bbs-hpib-daemon-<version>.rpm	HPI-B daemon package
bbs-hpib-debug-<version>.rpm	HPI-B debugger package
bbs-hpib-devel-<version>.rpm	HPI-B developer package

Installing Basic Blade Services Software

Table 2-1 BBS Distribution Files (continued)

File Name	Description
bbs-hpib-snmp-<version>.rpm	HPI-B SNMP package
bbs-hpmagentcmd-atcafl40-<version>.rpm	Hardware Platform Management
bbs-prom-utils-atcafl40-<version>.rpm	FPGA and PROM Upgrade utilities
bbs-tcm-atcafl40-<version>.rpm	Telco Clocking Module package
bbs-version-atcafl40-<version>.rpm	BBS Build Version information
bbs-atcafl40-bcm-ucode-<version>.rpm	BCM PHY microcode
bbs-atcafl40-fpga-<version>.rpm	Front panel FPGA image
bbs-atcafl40-fpga-rtm-<version>.rpm	RTM FPGA image
bbs-lhc-<version>.windriver.powerpc.rpm	Link Health Check
bbs-sfptool-atcafl40-<version>-windriver.rpm	SFP utility
bbs-uboot-utils-atcafl40-<version>-wrspe3.0.rpm	U-Boot environment variable editor
SRstackware RPM Files	
srstackware-config-f140-<version>.rpm	SRstackware configuration and script files
srstackware-demo-config-f140-<version>.rpm	SRstackware demonstration files
srstackware-f140-<version>.rpm	SRstackware basic features binaries
Miscellaneous files	
files.shalsum	Installation file
atca-f140.dtb	Device tree for installation
u-boot-<version>.bin	Raw U-Boot image for installation
atcafl40-initrd.gz.uboot	INITRD image for installation
ramdisk.image.gz	Ram disk root file system image

Table 2-1 BBS Distribution Files (continued)

File Name	Description
PrAMC721x-WR-3.0-BBS-RM01-prePINv.tar.bz2	Support file for AMC7211
kernel.tar.gz	Kernel source files
modules.tar.gz	Linux Modules
License RPMs (Not part of BBS)	
ssf_csim_rel.atcafl40-pne3.0-<version>.ppc.rpm, ssf_diagnosticsTLS_rel.atcafl40-pne3.0-<version>.ppc.rpm,ssf_main_rel.atcafl40-pne3.0-<version>.ppc.rpm	Bsf RPMs to provide XML and CLI interfaces for ViewCheck software
diagnostics-<version>.windriver3.0.linux.atcafl40.rpm	ViewCheck configuration, Tests, and Monitors
srstackware-fl40-fp-<version>.rpm	SRstackware FlowPilot RPM
srstackware-fl40-l3-<version>.rpm	SRstackware Layer 3 RPM

2.2 Configuring a TFTP Server

To install the BBS software over a network, you should prepare a Trivial File Transfer Protocol (TFTP) server on a host and create a directory structure for the boot images and installation files. The following section describes the procedure to configure the TFTP server.

2.2.1 Create /tftpboot Directory and Copy Target Files

You have to place TFTP files in the *tftpboot* directory. Regardless of the file system node you specify as the "root" for a TFTP service, the installation scripts expect such a directory structure when retrieving files.

Creating the /tftpboot Directory and Copying the Target Files

To create the expected directory structure and to copy the necessary files, follow these steps:

1. On the host, create a directory, if it does not exist. For example:
`mkdir /tftpboot`
2. Create a subdirectory for the blade. For example:
`mkdir /tftpboot/atca-f140`
3. Copy or move the installation files to the subdirectory.

2.3 Configuring ATCA-F140 for Network Booting

Configure U-Boot Environment Variables

Follow these steps to configure U-Boot environment variables in ATCA-F140 to download images from the network via front-panel Ethernet interface.

1. Power-up or reboot the blade.
2. Press **Ctrl-C** when you see the message `Hit Ctrl-C to stop autoboot:` on the console.
3. Type `printenv` to check the current U-Boot environment variables.
4. Set the `ethact` variable to use the front-panel Ethernet device.
`setenv ethact eTSEC2`
`saveenv`
5. Set the `loadaddr`, `ramdiskaddr`, and `blobaddr` variables.
`setenv loadaddr 1000000`
`setenv ramdiskaddr 2000000`
`setenv blobaddr a00000`
`saveenv`
6. Set the appropriate network environment variables for your network.
`setenv serverip <your TFTP server's IP address>`
`setenv ipaddr <your blade's IP address>`
`setenv gatewayip <your network gateway's IP address>`
`setenv netmask <your network's netmask>`
`setenv hostname <your blade's network name>`
`saveenv`

Alternatively, if there is a Dynamic Host Configuration Protocol (DHCP) server connected to the same network as the ATCA-F140, these variables can be loaded from the DHCP server via the `dhcp` U-Boot command.

```
dhcp
saveenv
```

2.4 Booting the ATCA-F140 Using a RAM Disk

This section describes how to network boot the ATCA-F140 from a TFTP server using a RAMDisk image.

Configuring the Boot Parameters for RAMDisk Booting

To configure ATCA-F140 to load the images from the network (via the front-panel Ethernet interface) and boot using a RAMDisk.

1. Power-up or reboot the blade.
2. Press **Ctrl-C** when you see the message `Hit Ctrl-C to stop autoboot:` on the console.
3. Type `printenv` to check the current U-Boot environment variables.
4. Update the appropriate environment variables.


```
setenv serverip <your TFTP server's IP address>
setenv ipaddr <your blade's IP address>
setenv gatewayip <your network gateway's IP address>
setenv netmask <your network's netmask>
saveenv
```
5. Set the `bootfile`, `blobfile`, and `ramdiskfile` variables.


```
setenv bootfile atca-f140/kernel
setenv blobfile atca-f140/atca-f140.dtb
setenv ramdiskfile atca-f140/ramdisk.image.gz
saveenv
```
6. Set or update the `ramdisksize` value (if required).


```
setenv ramdisksize <your ramdisksize value>
saveenv
```



Currently, U-Boot maintains a default `ramdisksize` value of 1048576 (1G).

7. Check whether the `rambootargs` script is configured, as required.


```
printenv rambootargs
rambootargs=setenv bootargs root=/dev/ram rw
ramdisk_size=$ramdisksize init=$lnxinit
ip=$ipaddr:$serverip:$gatewayip:$netmask:$hostname:$netdev:none
console=$consoledev,$baudrate mem=$mem $othbootargs;
```

Installing Basic Blade Services Software

8. Check whether the `ramboot` script is configured as required.

```
printenv ramboot
```

```
ramboot=run rambootargs && tftp $ramdiskaddr $ramdiskfile && tftp  
$loadaddr $bootfile && tftp $blobaddr $blobfile && bootm $loadaddr  
$ramdiskaddr $blobaddr
```

9. Set the `bootcmd` environment variable to invoke the `ramboot` script after a reset:

```
setenv bootcmd 'run ramboot'
```

```
saveenv
```

10. Reset the blade.

```
reset
```

Booting through the `ramdisk.image.gz` file installs all the BBS RPMs and launches all BBS daemons automatically. For upgrading firmware, refer the section [Upgrading Firmware on page 71](#).

2.5 SATA and USB Disk Installation Procedure

This section describes how to install BBS onto the SATA or USB device from a TFTP server.

2.5.1 Configuring the Boot Parameters for Network Installation

Configuring the Boot Parameters for Network Installation

To configure ATCA-F140 to network boot and install the U-boot, device-tree blob, kernel, root file system, and BBS RPMs:

1. Power-up or reboot the blade.
2. Press `Ctrl-C` when you see message `Hit Ctrl-C to stop autoboot:` on the console.
3. Type `printenv` to check the current U-Boot environment variables.
4. Set the `bootfile`, `blobfile`, and `ramdiskfile` variables.

```
setenv bootfile atca-f140/kernel
```

```
setenv blobfile atca-f140/atca-f140.dtb
```

```
setenv ramdiskfile atca-f140/atcaf140-initrd.gz.uboot
```

```
saveenv
```



The `ramdiskfile` used here is different from that of the `ramdiskfile` used in booting RAMDisk images.

5. Check whether the `rambootargs` script is configured, as required.

printenv rambootargs

```
rambootargs=setenv bootargs root=/dev/ram rw
ramdisk_size=$ramdisksize init=$lnxinit
ip=$ipaddr:$serverip:$gatewayip:$netmask:$hostname:$netdev:none
console=$consoledev,$baudrate mem=$mem $othbootargs;
```

6. Check whether the `ramboot` script is configured, as required.

printenv ramboot

```
ramboot=run rambootargs && tftp $ramdiskaddr $ramdiskfile && tftp
$loadaddr $bootfile && tftp $blobaddr $blobfile && bootm $loadaddr
$ramdiskaddr $blobaddr
```

7. To boot the blade and continue the installation process, execute the `run ramboot` command.

2.5.2 Installing BBS on a SATA or USB Device

Installing BBS on a SATA or USB Device

The boot process will invoke the `linuxrc` script which will guide you throughout the installation process.

1. The installation process begins by checking for necessary commands on the system. The output screen will be similar to as follows:

```
init started: BusyBox v1.11.1 (2010-05-03 09:25:54 MST)
starting pid 1023, tty '': '-/linuxrc'
```

```
*****
***** Installation Ramdisk *****
*****
```

Checking for necessary commands...

```
mount.....[ exists ]
umount.....[ exists ]
tar.....[ exists ]
gzip.....[ exists ]
mkdir.....[ exists ]
rmdir.....[ exists ]
```

Installing Basic Blade Services Software

```
rm.....[ exists ]
cp.....[ exists ]
mv.....[ exists ]
date.....[ exists ]
chmod.....[ exists ]
chown.....[ exists ]
grep.....[ exists ]
stty.....[ exists ]
ping.....[ exists ]
sed.....[ exists ]
fgrep.....[ exists ]
ifconfig.....[ exists ]
reboot.....[ exists ]
hwclock.....[ exists ]
fdisk.....[ exists ]
mkfs.ext3.....[ exists ]
cut.....[ exists ]
tftp.....[ exists ]
awk.....[ exists ]
seq.....[ exists ]
shasum.....[ exists ]
tr.....[ exists ]
chroot.....[ exists ]
flash_eraseall.....[ exists ]
flash_unlock.....[ exists ]
flashcp.....[ exists ]
```

Necessary commands found, safe to continue....

2. Start the installation by answering 'y' to the following query:
Do you wish to begin the installation? [y/n] **y**



There is no default value for this query. Choosing 'y' will begin the installation procedure. Choosing 'n' will abort the installation.

3. Choose static or dynamic IP configuration.
While using Dynamic Host Configuration Protocol (DHCP), answer 'n' to the following query ('n' is default):

You may choose to alter the default IP management scheme of DHCP here by answering 'yes' to the next question...

```
Do you wish to use static IP management? [y/N] n
```

You must ensure that the DHCP server is already configured to provide the ATCA-F140 with the `files.shalsum` file and the IP address of the TFTP server in which the installation files are downloaded (especially if the DHCP and TFTP servers are different, refer to the `next-server` and `filename` options in `dhcpd.conf` on the DHCP server). For example:

```
next-server <your TFTP server's IP address>;  
filename "atca-f140/files.shalsum";
```

If using DHCP, skip to [Step 5](#).

If choosing static IP configuration, you will need to provide input to the following queries:

Please enter the following information accurately.

```
Please enter the IP address [xxx.xxx.xxx.xxx]
```

```
Please enter the netmask [xxx.xxx.xxx.xxx]
```

```
Please enter the gateway [xxx.xxx.xxx.xxx]
```

```
Provide a Domain Name Server to use [xxx.xxx.xxx.xxx]
```

```
Do you wish to enter another DNS address? [y/N] n
```

```
Provide a domain name [domain.company.com]
```

```
Which TFTP server do you wish to use? [xxx.xxx.xxx.xxx]
```

4. Enter the TFTP server installation directory:
What is the installation files directory? [] **atca-f140**



Enter the directory name without leading or trailing slashes.

5. Downloading installation files.
During the file download process, the files are displayed as shown below:
Downloading 'atca-f140/files.shalsum' ... Done.
Downloading 'atca-f140/kernel' ... Done.
Downloading 'atca-f140/atca-f140.dtb' ... Done.
Downloading 'atca-f140/u-boot-<version>.bin' ... Done.

Installing Basic Blade Services Software

```
Downloading 'atca-f140/nfs.tar.gz' ... Done.
Downloading 'atca-f140/bbs-atca-f140-bcm-ucode-<version>.rpm' ...
Done.
Downloading 'atca-f140/bbs-atca-f140-cpu-<version>.rpm' ... Done.
Downloading 'atca-f140/bbs-atca-f140-dtb-<version>.rpm' ... Done.
Downloading 'atca-f140/bbs-atca-f140-fpga-<version>.rpm' ... Done.
Downloading 'atca-f140/bbs-atca-f140-fpga-rtm-<version>.rpm' ... Done.
Downloading 'atca-f140/bbs-atca-f140-ipmc-<version>.rpm' ... Done.
Downloading 'atca-f140/bbs-atca-f140-ipmc-boot-<version>.rpm' ...
Done.
Downloading 'atca-f140/bbs-atca-f140-kernel-<version>.rpm' ... Done.
Downloading 'atca-f140/bbs-atca-f140-sembltech-<version>.rpm' ... Done.
Downloading 'atca-f140/bbs-fuf-atca-f140-<version>-wrs-pne3.0.rpm' ...
Done.
Downloading 'atca-f140/bbs-hpib-<version>-wrs-pne3.0-linux.rpm' ...
Done.
Downloading 'atca-f140/bbs-hpib-clientsrc-<version>-wrs-pne3.0-
linux.rpm' ... Done.
Downloading 'atca-f140/bbs-hpib-daemon-<version>-wrs-pne3.0-
linux.rpm' ... Done.
Downloading 'atca-f140/bbs-hpib-debug-<version>-wrs-pne3.0-linux.rpm'
... Done.
Downloading 'atca-f140/bbs-hpib-devel-<version>-wrs-pne3.0-linux.rpm'
... Done.
Downloading 'atca-f140/bbs-hpib-snmp-<version>-wrs-pne3.0-linux.rpm'
... Done.
Downloading 'atca-f140/bbs-hpmagentcmd-atca-f140-<version>-
wrs-pne3.0.rpm' ... Done.
Downloading 'atca-f140/bbs-lhc-<version>.windriver.powerpc.rpm' ...
Done.
Downloading 'atca-f140/bbs-prom-utils-atca-f140-<version>-
wrs-pne3.0.rpm' ... Done.
Downloading 'atca-f140/bbs-sfptool-atca-f140-<version>-windriver.rpm'
... Done.
Downloading 'atca-f140/bbs-tcm-atca-f140-<version>-wrs-pne3.0.rpm' ...
Done.
Downloading 'atca-f140/bbs-uboot-utils-atca-f140-<version>-
wrs-pne3.0.rpm' ... Done.
```

```
Downloading 'atca-f140/bbs-version-atcaf140-<version>-wrspne3.0.rpm'
... Done.
Downloading 'atca-f140/srstackware-config-f140-
<version>.powerpc.rpm' ... Done.
Downloading 'atca-f140/srstackware-demo-config-f140-
<version>.powerpc.rpm' ... Done.
Downloading 'atca-f140/srstackware-f140-<version>.powerpc.rpm' ...
Done.
Downloading 'atca-f140/srstackware-enhanced-f140-
<version>.powerpc.rpm' ... Done.
```



The `linuxrc` script determines which files to download and install based on the contents of the files `.shalsum` file. Changing the contents of this file will change the installation process and possibly prevent the installation from succeeding.

6. Set the time.

Before installing the downloaded files, the time on the system has to be set correctly. If you wish to use Network Time Protocol (NTP) to set the system time, answer 'y' (default) to the following query:

If NTP servers were provided by the DHCP server, It will displays as follows:

```
Current date/clock :Fri Aug 6 17:20:27 UTC 2010
DHCP Server assigned NTP server(s) xxx.xxx.xxx.xxx
Do you wish to use NTP to set the current time? [Y/n] y
Using address xxx.xxx.xxx.xxx for NTP server.
```

If NTP servers were not provided by the DHCP server, an IP address will be requested as shown below:

```
Current date/clock :Fri Aug 6 17:20:27 UTC 2010
Do you wish to use NTP to set the current time? [Y/n] y
Please enter the NTP server address [xxx.xxx.xxx.xxx]
```

If you do not have a NTP server, or wish to set the time manually, answer 'n' to the query to use NTP to set the current time:

```
Current date/clock :Fri Aug 6 17:20:27 UTC 2010
Do you wish to use NTP to set the current time? [Y/n] n
```

To enter the date and time manually, answer 'y' to the query to manually set the date:

```
Do you wish to set the date manually? [Y/n] y
Enter date in 'MM/DD/YYYY' format. [] 08/06/2010
```

Note: Time has to be set in UTC/GMT format

Installing Basic Blade Services Software

Enter time in 'HH:MM' and 24-hour format. [] 17:20

7. Select the SATA or USB device as the target to install the root file system.

If the installation script detects both a SATA device and a USB device, the user will be prompted to select which device to use when installing the root file system.

There is a mux on the ATCA-F140 that controls access to the SATA drive (either to the P2020 or the AMC). If the mux is set to direct the SATA drive to the AMC, then only the USB device will be available to the installation process.



If installation is done with the USB as the only device and the SATA drive is later added by changing the mux setting, the U-boot "rootpartition" variable will need to be changed to /dev/sdb. For more details see [Section 2.5.3, Configuring ATCA-F140 to Boot From SATA or USB Drive on page 41](#).

If both devices are available, the SATA drive will always appear first in the list (as /dev/sda) and the USB drive will always appear second (as /dev/sdb). The example here is selecting the SATA device (#1 in the list) as the installation target.

The following disks are available for root filesystem installation:

```
1: Disk /dev/sda: 80.0 GB, 80026361856 bytes
2: Disk /dev/sdb: 2013 MB, 2013265920 bytes
```

```
Disk number to install the root filesystem? [] 1
```



The above disk selection query appears only if both SATA and USB devices are available.

If only one device is available, the installation script will not query the user for a target device. The device, whether it is a SATA drive or a USB drive, will always appear as /dev/sda. Also the installation script will warn the user that the disk will be reformatted as shown below.

```
This installation will reformat the target disk /dev/sda.
```

```
All data will be lost on the target disk.
```

```
Do you wish to proceed? [Y/n]
```



The above query does not occur if the disk has already two partitions.

8. Select the partition to install the root file system.

The following partitions are available for root filesystem installation:

```
1: /dev/sda1: 40.0 GB
```

```
2: /dev/sda2: 40.0 GB
```

```
Partition number to install the root filesystem? [] 1
```

9. Confirm the target selection and proceed.

At this point, the target devices will be displayed and the user is queried to proceed with the installation process.



The kernel, device-tree blob, and U-Boot will be automatically upgraded in the backup boot bank if the images are found in the `files.sha1sum` file during download.

Using the following MTD devices for the kernel, DTB, and U-Boot installation(s):

```
MTD device /dev/mtd5: (backup-bank-kernel partition)
```

```
MTD device /dev/mtd7: (backup-bank-dtb partition)
```

```
MTD device /dev/mtd9: (backup-bank-u-boot partition)
```

Using the following disk for root filesystem installation:

```
Disk /dev/sda1: 40.0 GB
```

```
Do you wish to proceed? [Y/n] y
```

10. Check SHA1 checksums and installation.

At the time of file downloading progress, their SHA1 checksums are calculated and checked against the checksum as indicated in the `files.sha1sum` file. If the checksums are matched, the file is installed accordingly. The kernel, device-tree blob, and U-Boot images are installed in the appropriate backup boot bank MTD partitions, all tar balls are uncompressed, and all RPMs are installed using the Red Hat Package Manager.

The progress of the installation is displayed in the console and will look like as follows:

```
Formatting root filesystem partition /dev/sda1 ... Done.
```

```
-----
```

```
Writing files to root filesystem partition /dev/sda1 ...
```

```
-----
```

```
Checking SHA1 Checksum for kernel ... Good.
```

```
Installing kernel on /dev/mtd5 ... Unlocking ... Erasing ... Copying ... Done.
```

```
Checking SHA1 Checksum for atca-f140.dtb ... Good.
```

Installing Basic Blade Services Software

```
Installing DTB on /dev/mtd7 ... Unlocking ... Erasing ... Copying
... Done.
Checking SHA1 Checksum for u-boot-<version>.bin ... Good.
Installing U-Boot on /dev/mtd9 ... Unlocking ... Erasing ... Copying
... Done.
Checking SHA1 Checksum for nfs.tar.gz ... Good.
Uncompressing using TAR ... Done.
Checking SHA1 Checksum for bbs-atcafl140-bcm-ucode-<version>.rpm ...
Good.
Preparing rpm wrapper script ... Done.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-atcafl140-cpu-<version>.rpm ... Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-atcafl140-dtb-<version>.rpm ... Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-atcafl140-fpga-<version>.rpm ... Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-atcafl140-fpga-rtm-<version>.rpm ...
Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-atcafl140-ipmc-<version>.rpm ... Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-atcafl140-ipmc-boot-<version>.rpm ...
Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-atcafl140-kernel-<version>.rpm ...
Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-atcafl140-semtech-<version>.rpm ...
Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-fuf-atcafl140-<version>-wrs pne3.0.rpm
... Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-hpib-<version>-wrs pne3.0-linux.rpm
... Good.
Installing using RPM ... Done.
```

Installing Basic Blade Services Software

```
Checking SHA1 Checksum for bbs-hpib-clientsrc-<version>-wrspne3.0-
linux.rpm ... Good.
  Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-hpib-daemon-<version>-wrspne3.0-
linux.rpm ... Good.
  Installing using RPM ...Executing bbs-hpib-daemon post-install
script:
Stopping HPI-B daemon.
HPI-B daemon runs on ATCA-M100 already - exiting.
Done.
Checking SHA1 Checksum for bbs-hpib-debug-<version>-wrspne3.0-
linux.rpm ... Good.
  Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-hpib-devel-<version>-wrspne3.0-
linux.rpm ... Good.
  Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-hpib-snmp-<version>-wrspne3.0-
linux.rpm ... Good.
  Installing using RPM ...Executing bbs-hpib-snmp post-install
script:
Stopping HPI-B SNMP subagent.
Starting HPI-B SNMP subagent.
Done.
Checking SHA1 Checksum for bbs-hpmagentcmd-atcafl40-<version>-
wrspne3.0.rpm ... Good.
  Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-lhc-<version>.windriver.powerpc.rpm
... Good.
  Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-prom-utils-atcafl40-<version>-
wrspne3.0.rpm ... Good.
  Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-sfptool-atcafl40-<version>-
windriver.rpm ... Good.
  Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-tcm-atcafl40-<version>-wrspne3.0.rpm
... Good.
```

Installing Basic Blade Services Software

```
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-uboot-utils-atcafl40-<version>-
wrs pne3.0.rpm ... Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for bbs-version-atcafl40-<version>-
wrs pne3.0.rpm ... Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for srstackware-config-f140-
<version>.powerpc.rpm ... Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for srstackware-demo-config-f140-
<version>.powerpc.rpm ... Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for srstackware-f140-<version>.powerpc.rpm
... Good.
Installing using RPM ... Done.
Checking SHA1 Checksum for srstackware-enhanced-f140-
<version>.powerpc.rpm ... Good.
Installing using RPM ... Done.
Disabling rpm wrapper script ... Done.
```

11. Final configuration.

Answer the following queries to set the host name and the root password for this system.

Beginning final configuration

Choose a hostname for this machine [] **atca-f140**

Set Root password

New UNIX password: <root password>

Retype new UNIX password: <root password again>

passwd: password updated successfully

Root password set.

After setting the root password, the initialization script will attempt to issue a boot bank switch to pick up the new installation at reboot.


```
-----  
Switching boot bank ... Done.  
Reboot will occur on boot bank 1.  
-----
```

After installation and final configuration, the blade will reboot automatically. The progress of the reboot is displayed in the console and will look similar to as follows:

```
Creating admin login...Done.  
Cleaning up root filesystem...Done.  
Unmounting local filesystems...Done.
```

```
Rebooting in 5 seconds... (Ctrl-C to break into shell)  
Rebooting...  
Restarting system.
```

12. If the installation script failed to switch the boot banks for using the newly installed U-Boot, kernel, and device-tree blob and if the installation process has installed the U-Boot, kernel, and the device-tree blob to the **backup boot bank**, in such case, to use the new images, the boot bank must be switched and the blade is to be reset. To switch the boot bank and to reset the blade, enter as follows:

```
bootset switch  
reset
```

The backup boot bank will become active and the blade will boot with the newly installed images. Proceed to [Section 2.5.3 on page 41](#) to configure U-Boot to boot the blade using the flash devices and the SATA or USB drive.

2.5.3 Configuring ATCA-F140 to Boot From SATA or USB Drive

This section describes how to configure U-Boot to boot the Linux kernel and device-tree blob stored in the flash and to mount the root file system from the SATA or USB drive. The procedure uses the `flashboot` U-Boot script.

Configuring the ATCA-F140 to Boot From SATA or USB Drive

To configure ATCA-F140 to boot from SATA or USB drive:

1. Power-up or reboot the blade.
2. Press `Ctrl-C` when you see the message `Hit Ctrl-C to stop autoboot:` on the console.
3. Check whether the `flashbootargs` script is configured, as required.

```
printenv flashbootargs
```

Installing Basic Blade Services Software

```
flashbootargs=setexpr num 1 + $bootbank && setenv rootpartition  
$root$num && setenv bootargs root=$rootpartition rw init=$lnxinit  
ip=$ipaddr:$serverip:$gatewayip:$netmask:$hostname:$netdev:$autoc  
onf console=$consoledev,$baudrate mem=$mem
```



The selection of the root file system device is based on the `$bootbank` U-Boot environment variable.

Table 2-2 *rootpartition U-Boot Environment Variable*

Root File System	Value (<code>\$root\$num</code>)
/dev/sda1	0x0801
/dev/sda2	0x0802
/dev/sdb1	0x0811
/dev/sdb2	0x0812

4. Check whether the `flashboot` script is configured, as required.

```
printenv flashboot
```

```
flashboot= run flashbootargs && bootm e0000000 - e1f00000
```

5. Set the `bootcmd` environment variable to invoke the `flashboot` script after a reset:

```
setenv bootcmd 'run flashboot'
```

```
saveenv
```

6. Reset the blade.

```
reset
```

The system boots using the U-Boot, kernel, and device-tree blob in the **active boot bank** and uses the root file system as specified in the `rootpartition` environment variable.

For upgrading firmware, refer to [Section 4.4, Upgrading Firmware on page 60](#).

2.6 NFS Installation Procedure

This section describes how to install BBS on a TFTP/NFS server, configure ATCA-F140 to boot using TFTP, and mount the root file system from the NFS server.

2.6.1 Configuring the NFS Server

Configure the NFS Server

Assume that you have a working knowledge of NFS and how to configure it. In the examples below, the directory `/export/atca-F140` is used for the ATCA-F140 root file system. Your configuration may also vary.

1. Log in as "root" on the NFS server.
2. Create a directory for ATCA-F140 root file system, if it does not exist.

```
# mkdir -p /export/atca-f140
```
3. All NFS server exports should be defined at `/etc/exports` location.

```
/export/atca-f140 *(rw, sync, no_root_squash, no_all_squash)
```
4. Restart NFS server.

```
# /etc/init.d/nfs restart
```

2.6.2 Copying and Installing the Root File System

Copying and Installing the Root File System

To copy and install the Root File System:

1. On the NFS server, log in as "root".
2. Change to the ATCA-F140 root file system mount point (created in the earlier section)

```
# cd /export/atca-f140
```
3. Copy or move the `rootfs.tar.gz` file into the exported directory.
4. Extract the root file system.

```
# tar -xzf rootfs.tar.gz
```

2.6.3 Configuring the Boot Parameters for NFS

Configuring the Boot Parameters for NFS

To configure ATCA-F140 to network boot the kernel using NFS mount point:



Following example assumes the TFTP server is also the NFS server and has the same IP address. Alternatively, you can set the `nfsroot=` variable used in `nfsbootargs` to reflect your NFS server's IP address.

1. Power-up or reboot the blade.

Installing Basic Blade Services Software

2. Press `ctrl-c` when you see the message `Hit Ctrl-C to stop autoboot:` on the console.
3. Type `printenv` to check the current U-Boot environment variables.
4. Set the `bootfile`, `blobfile` and `rootpath` environment variables.

```
setenv bootfile atca-f140/kernel
setenv blobfile atca-f140/atca-f140.dtb
saveenv
```

Using the NFS configuration examples:

```
setenv rootpath /export/atca-f140
```
5. Check for the `nfsbootargs` script setup.

```
print nfsbootargs
nfsbootargs= setenv bootargs root=/dev/nfs rw
nfsroot=$serverip:$rootpath init=$linuxinit
ip=$ipaddr:$serverip:$gatewayip:$netmask:$hostname:$netdev:none
console=$consoledev,$baudrate mem=$mem $othbootargs;
```
6. Check for the `nfsboot` script setup.

```
printenv nfsboot
nfsboot=run nfsbootargs && tftp $loadaddr $bootfile && tftp $blobaddr
$blobfile && bootm $loadaddr - $blobaddr
```
7. Set the `bootcmd` environment variable to invoke the `nfsboot` script after reboot:

```
setenv bootcmd 'run nfsboot'
saveenv
```
8. Reset the blade.

```
reset
```

2.6.4 Installing the BBS RPMs

Install the Remaining BBS Files

After the blade boots, BBS RPMs need to be installed. Reboot the blade after installing the RPMs.

1. Log onto ATCA-F140 as root.
2. Change directory to the location where the BBS RPMs exist.

```
# cd /opt/bladeservices/rpms
```
3. Install the BBS and SRstackware RPM files in the following sequence:

```
# rpm -ivh -nodeps bbs-atcaf140-bcm-ucode-<version>.rpm
# rpm -ivh -nodeps bbs-atcaf140-cpu-<version>.rpm
# rpm -ivh -nodeps bbs-atcaf140-dtb-<version>.rpm
# rpm -ivh -nodeps bbs-atcaf140-fpga-<version>.rpm
```

```
# rpm -ivh -nodeps bbs-atcaf140-fpga-rtm-<version>.rpm
# rpm -ivh -nodeps bbs-atcaf140-ipmc-<version>.rpm
# rpm -ivh -nodeps bbs-atcaf140-ipmc-boot-<version>.rpm
# rpm -ivh -nodeps bbs-atcaf140-kernel-<version>.rpm
# rpm -ivh -nodeps bbs-atcaf140-semtech-<version>.rpm
# rpm -ivh -nodeps bbs-fuf-atcaf140-<version>-wrspne3.0.rpm
# rpm -ivh -nodeps bbs-hpib-<version>-wrspne3.0-linux.rpm
# rpm -ivh -nodeps bbs-hpib-clientsrc-<version>-wrspne3.0-
linux.rpm
# rpm -ivh -nodeps bbs-hpib-daemon-<version>-wrspne3.0-linux.rpm
# rpm -ivh -nodeps bbs-hpib-devel-<version>-wrspne3.0-linux.rpm
# rpm -ivh -nodeps bbs-hpib-snmp-<version>-wrspne3.0-linux.rpm
# rpm -ivh -nodeps bbs-hpmagentcmd-atcaf140-<version>-
wrspne3.0.rpm
# rpm -ivh -nodeps bbs-prom-utils-atcaf140-<version>-wrspne3.0.rpm
# rpm -ivh -nodeps bbs-sfptool-atcaf140-<version>-windriver.rpm
# rpm -ivh -nodeps bbs-tcm-atcaf140-<version>-wrspne3.0.rpm
# rpm -ivh -nodeps bbs-uboot-utils-atcaf140-<version>-
wrspne3.0.rpm
# rpm -ivh -nodeps bbs-version-atcaf140-<version>-wrspne3.0.rpm
# rpm -ivh -nodeps srstackware-config-f140-<version>.powerpc.rpm
# rpm -ivh -nodeps srstackware-demo-config-f140-
<version>.powerpc.rpm
# rpm -ivh -nodeps srstackware-f140-<version>.powerpc.rpm
# rpm -ivh -nodeps srstackware-enhanced-f140-<version>.powerpc.rpm
# ldconfig
# depmod -ae
```

4. Reboot the blade to activate the BBS.

```
# reboot -f
```

For upgrading firmware, refer to [Section 4.4, Upgrading Firmware on page 60](#).

2.7 Upgrading the FPGA Firmware

To upgrade the FPGA firmware is a two-step procedure:

1. Upload the firmware image to the AT45 flash device.
2. Trigger the FPGA to upgrade with the latest firmware image.

Installing Basic Blade Services Software

The following table details the FPGA firmware upgrade command usage and its functionality.

Command	Description
<code># fpga_upg -v</code>	Retrieves the current FPGA version
<code># fpga_upg -u -f /opt/bladeservices/rom/atca-f140- fpga-<version>.bin</code>	Upgrades the front-blade FPGA firmware on ATCA-F140
<code># fpga_upg -u -r /opt/bladeservices/rom/atca-f140- rtm-fpga-<version>.bin</code>	Upgrades the RTM FPGA firmware on ATCA-F140.
<code># fpga_upg -c -f /opt/bladeservices/rom/atca-f140- fpga-<version>.bin</code>	Compares the front-blade FPGA firmware with the image
<code># fpga_upg -c -r /opt/bladeservices/rom/atca-f140- rtm-fpga-<version>.bin</code>	Compare the RTM FPGA firmware with the image.

The upgrade process automatically verifies whether the image loaded was written correctly to the flash device. The `fpga_upg` utility can also be used to verify the image currently populated on the flash device to an image stored in a file.

To trigger the upload from the flash device to the FPGA, the blade must undergo a power-cycle event. This can either be done via the shelf manager or the blade can be physically reset.



The `fpga_upg` utility shares the same SPI interface as the `prom_upg` and `SFP` utilities (`sfptool`). These utilities should be stopped (if running) prior to upgrading FPGA firmware. Otherwise, the `fpga_upg` utility may fail with "Device or resource busy".

2.8 Upgrading the BCM 8727/84740/84754 PHY Microcode

To upgrade the BCM 8727/84740/84754 PHY microcode:

1. Upload the microcode to the AT25 flash device.
2. Trigger the BCM 8727/84740/84754 PHYs to upgrade using the microcode.

The BCM 8727/84740/84754 PHY microcode does not have version information to identify itself. You should take care to know when to upload new images to the AT25 flash device.

Installing Basic Blade Services Software

The front-blade contains BCM 8727 and BCM 84740 devices. To upgrade the front-blade BCM 8727 and BCM 84740 PHY microcode, enter:

```
# prom_upg -u -f bcm8727 /opt/bladeservices/rom/bcm-8727-0511.bin
# prom_upg -u -f bcm84740 /opt/bladeservices/rom/bcm-84740-0106.bin
```

The RTM contains BCM 8727, BCM 84740, and BCM 84754 devices. To upgrade the RTM BCM 8727, BCM 84740, and BCM 84754 PHY microcode, enter:

```
# prom_upg -u -r bcm8727 /opt/bladeservices/rom/bcm-8727-0511.bin
# prom_upg -u -r bcm84740 /opt/bladeservices/rom/bcm-84740-0106.bin
# prom_upg -u -r bcm84754 /opt/bladeservices/rom/bcm-84754-0105.bin
```

The upgrade process automatically verifies that the image loaded is written correctly in to the flash device.

The `prom_upg` utility can also be used to verify the image currently populated on the flash device to an image stored in a file.

To compare the front-blade BCM 8727 and BCM 84740 PHY microcode to the image, enter:

```
# prom_upg -c -f /opt/bladeservices/rom/bcm-8727-0511.bin
# prom_upg -c -f /opt/bladeservices/rom/bcm-84740-0106.bin
```

To compare the RTM BCM 8727, BCM 84740, and BCM 84754 PHY microcode to the image, enter:

```
# prom_upg -c -r /opt/bladeservices/rom/bcm-8727-0511.bin
# prom_upg -c -r /opt/bladeservices/rom/bcm-84740-0106.bin
# prom_upg -c -r /opt/bladeservices/rom/bcm-84754-0105.bin
```

The BCM 8727, BCM 84740, and BCM 84754 PHYs are triggered to upload the image during reboot.

There are five flash devices corresponding to the five PHY devices. The target device is selected by the "-f" and "-r" arguments.

There is no check between the device being upgraded and the upload file. You should make sure that they are upgrading the proper device with the proper upgrade file. Upgrading the wrong device with the wrong image can render the PHY unusable.



The `prom_upg` utility shares the same SPI interface as the `fpga_upg` and `SFP` utilities (`sfptool`). These utilities should be stopped (if running) prior to upgrading BCM PHY microcode. Otherwise, the `prom_upg` utility may fail with "Device or resource busy".

Licensed Software Modules

In addition to the base BBS, there are several licensed software modules that you can purchase to perform specific tasks. Currently the licensed software modules for the ATCA-F140 include:

- SRS Layer 3 License for ATCA-F140 (SL-L3F140-01-001-STD)
SRstackware Layer3 Licensed Module is required to use Layer 3 protocols supported by SRstackware, such as, OSPFv2, VRRP, RIP etc. Base BBS distribution consists Layer2 features and static configuration of IPv4/IPv6. For more details see, *SRstackware Intelligent Network Software Layer 3 Configuration Guide*.
- FlowPilot License for ATCA-F140 (SL-FPF140-04-001-STD)
SRstackware FlowPilot licensed module offers Stateless Load Balancer as well as Equal Cost Multi Path (ECMP) Load Balancer features. Basic load-balancing feature is part of BBS distribution. For more details see, *FlowPilot Configuration Guide*.
- ViewCheck for ATCA-F140 (SL-DGF140-01-001-STD)
ViewCheck Licensed Module is required to use extensive diagnostics feature. For more details see, *ViewCheck for ATCA-F140 Installation and Use Guide*.

You need to buy a license for using the licensed software modules. For example, if you buy FlowPilot License for ATCA- F140, you will get FlowPilot software package access. This package can be installed on top of the ATCA-F140-BBS package, which contains only the unlicensed software, including Layer 2 SRstackware.

3.1 Installation

License modules are provided as RPMs, which can be installed manually on top of BBS with base SRstackware RPM. However, to facilitate installation of licensed software module RPMs on top of base BBS distribution, a script, `atcaf140_licenseRPMInstall.sh` provided as a reference, is packaged with base BBS, which is located at `/opt/bladeservices/scripts`. You can use this script for installing or packaging the licensed software module RPMs with flashboot or RAMDisk.

NOTE: This script is only for reference. You can customize it, as required for your installation process.

Currently, using this script, you can install only the following licensed modules:

- SRS Layer 3
- SRS FlowPilot
- ViewCheck

Licensed Software Modules

This script is provided to install licensed module RPMs. As of now, there are three license modules which are supported by this script, from input parameters point of view l3 represents SRS Layer3 License module, fp represents SRS FlowPilot License module and vc represents ViewCheck License module.

Usage

Following is the syntax to use the script, for installing the licensed software module RPMs for factory or flash or RAMDisk deployments:

```
sh atcaf140_licenseRPMInstall.sh
Usage: sh atcaf140_licenseRPMInstall.sh -t TARGET -f
LICENSED_MODULES [OPTION]...
```

Parameters

-t TARGET: It is used to specify the target location for installing the licensed module. It is a mandatory parameter. It can have either of the following values:

ftp_install: Get the licensed module RPMs from remote system through FTP and install RPMs on ATCA-F140. Prerequisite: FTP server.

flashboot_prep: Get the licensed module RPMs from local system, copy into rootfs location, and update files.sha1sum.

rpm_install: Get the licensed module RPMs from local system and install RPMs on ATCA-F140.

ramdisk: Get the licensed RPMs from local system and update the ramdisk. It retains the backup of original ramdisk.

verify: Verify if required licensed module RPMs are installed in the blade.

-f LICENSED_MODULES: It is used to specify the licensed module to be installed. Currently it can install only three modules - SRS Layer 3 (l3), SRS FlowPilot (fp), or ViewCheck (vc). It is a mandatory parameter.

This script can take following options:

-i IPADDR: It is used to specify the IP address of FTP server. It is applicable to FTP installation.

-u USER: It is used to specify user name for accessing the FTP server. It does not accept 'root' as user name. It is applicable only to the FTP installation.

-p PASSWORD: It is used to specify password for accessing the FTP server. It is applicable only to the FTP installation.

-L RPM_LOCATION: It is used to specify the location of licensed software module RPM in the FTP server. You can use -i option to specify the IP address. By default, the location is initialized to Present Working Directory (PWD).

`-r IMAGE_LOCATION`: It is used to specify location of rootfs, files.sha1sum, or ramdisk remote location for FTP installation, if IP address is provided.

By default, the location is initialized to Preset Working Directory (PWD).

`-l LOGFILE` It is used to redirect install steps into log file.

`-v` It is used to display version information.

`-h` It is used to display the command help.

This script can be used for:

- Licensed module installation from remote server
- License modules inclusion to Flashboot image
- Licensed Module RPM Installation
- License Module RPMs inclusion to RAMDisk

License Module Installation from Remote Server

To install the licensed software module RPMs from remote server, specify target as `ftp_install`.

Currently this script does not support `scp_install` and `tftp_install` options, due to their interactive nature.

For licensed module installation from remote server:

1. Set up a FTP server; with user name and password.
2. Get the licensed software module RPMs from SCML/SWORDS (as per licensed module order).
3. In FTP server, copy the required licensed software module RPMs as per licensed module order in a separate directory- do not keep any other RPMs in this location.
4. Execute the installation script with below inputs:
 - IP address of FTP Server
 - List of licensed modules as per licensed module order. For example, l3,fp,vc (separate by commas only)
 - Absolute path of licensed software module RPMs in the FTP server
 - User name and password to login to the FTP serverFor example, to install SRS Layer3 and ViewCheck software RPM through an FTP server with IP address 10.10.10.1, the RPMs are located at `/tftpboot/` directory.

```
sh atcaf140_licenseRPMInstall.sh -t ftp_install -f l3,vc -u $USER -p $PASSWORD -L /tftpboot/f140_rpms -l install_log -i 10.10.10.1
```

Licensed Software Modules

NOTE: The location should not have any other unnecessary files, as this process will copy all of the files available in that location.

5. Execute the script to verify if the RPMs are installed. Provide input parameter as verify along with the list of licensed software modules.

```
sh atcaf140_licenseRPMInstall.sh -t verify -f l3,vc
```

License modules inclusion to Flashboot image

You can include licensed software modules to the flashboot image by updating `files.sha1sum` file with the licensed module RPMs information.

In this case target type should be provided as `flashboot_prep`.

NOTE: It only prepares the image to use in next flashboot installation, it does not install RPMs.

To include licensed module software to flashboot installation:

1. Copy "`sh atcaf140_licenseRPMInstall.sh`" script to the server where flashboot image containing `files.sha1sum` file is present.
2. Copy the licensed module RPMs to the server.
3. Execute the script with the required inputs:
 - List of licensed modules as per licensed module order. For example, `l3,fp,vc` (separate by commas only)
 - Absolute path of licensed module RPMs (default path is PWD).
 - Absolute path of flashboot image with `files.sha1sum` file (default path is PWD)

For example, to prepare a flashboot image with SRS FlowPilot and ViewCheck software RPM.

```
sh atcaf140_licenseRPMInstall.sh -t flashboot_prep -f fp,vc -L /f140_rpms  
-r /rootfs_location
```

Licensed Module RPM Installation

To install license RPMs in an already installed system that has, base BBS, you have to just install the licensed module RPMs.

In this case target type should be provided as `rpm_install`.

To install licensed module RPMs:

1. Copy the licensed module RPMs to the blade.
2. Execute the script with required inputs:

- List of licensed modules as per licensed module order. For example, l3,fp,vc (separate by commas only)
 - Absolute path of licensed RPMs (default path is PWD).
For example, to install SRS FlowPilot and ViewCheck software RPM.

```
sh atcaf140_licenseRPMInstall.sh -t rpm_install -f fp,vc -L /f140_rpms
```
3. Execute script to verify if the RPMs are installed. Provide input parameter as verify along with the list of license modules.

```
sh atcaf140_licenseRPMInstall.sh -t verify -f l3,vc
```

License module RPMs inclusion to RAMDisk

To prepare RAMDisk image with licensed RPMs, specify target as ramdisk.

To add licensed module RPMs to RAMDisk:

1. Copy "sh atcaf140_licenseRPMInstall.sh" script to the server where RAMDisk image is present.
2. Copy licensed module RPMs to the server.
3. Execute the script with required inputs:
 - List of licensed modules as per licensed module order. For example, l3,fp,vc (separate by commas only)
 - Absolute path of licensed RPMs (default path is PWD).
 - Absolute path of ramdisk image file (default path is PWD)
For example, to prepare a ramdisk with SRS Layer3, FlowPilot and ViewCheck software RPM.

```
sh atcaf140_licenseRPMInstall.sh -t ramdisk -f l3,fp,vc -L /f140_rpms -r /ramdisk_location
```


Firmware Upgrade Facility

The Firmware Upgrade Facility (FUF) provides a uniform way to upgrade firmware on Penguin Edge hub blades, node blades, and AMC modules. It consists of a Firmware Upgrade Command-line Utility (FCU), flash device drivers, and specially prepared firmware recovery image files.

4.1 Firmware Recovery Image Files

FCU supports specially prepared Firmware Recovery Image (FRI) files as well as firmware images in the Hardware Platform Management (HPM.1) format. HPM.1 is a PICMG standard to upgrade IPMCs.

By default, the image files for the current hardware configurations are loaded as part of the BBS software at `/opt/bladeservices/rom` when the blade-specific firmware support packages are installed.

The image files mentioned in the following table are currently supported.

Filename	Description
<code>atca-f140-kernel.fri</code>	Kernel image for ATCA-F140
<code>atca-f140-cpu.fri</code>	U-boot firmware image for ATCA-F140
<code>atca-f140-dtb.fri</code>	Device Tree Binary for ATCA-F140
<code>atca-f140-ipmc-<version>.hpm</code>	IPMC firmware image for ATCA-F140
<code>atca-f140-ipmc-boot-<version>.hpm</code>	IPMC Boot Loader firmware image for ATCA-F140

4.2 Backup Concept

Firmware for the IPMC on the Penguin Edge ATCA-F140 switch is stored in an on-chip flash which has one active and one backup section. When upgrading the IPMC firmware, only the firmware in the backup section is updated. After the upgrade, the new firmware is checked and if it is functional and valid, the IPMC switches to the new, updated firmware. This happens without causing a payload reset.

This upgrade mechanism prevents both sections from having bad IPMC firmware and allows for the next firmware upgrade to be done easily.

When upgrading the U-Boot firmware, or Device Tree Binary image, or Linux kernel, FCU only writes into the currently stand-by bank. After the upgrade, the stand-by bank must be marked for next use, this means it will be executed after the next reboot.

4.3 fcu—Firmware Upgrade Command-Line Utility

Description

The Firmware Upgrade Command-line Utility (FCU) allows you to:

- Query the current versions of firmware installed on a blade and determine which firmware devices are active.
- Verify that a specified upgrade image is sound and compatible with the current hardware.
- Upgrade a firmware image.
- Mark a device to be used as the boot source on the next reset.
- Switch between the active and stand-by IPMC firmware banks without causing a payload reset.
- Show the version of a firmware image file and compare the version of a firmware image file with the version of the currently installed firmware.

By default, the FCU binary executable is installed at `/opt/bladeservices/bin`. This directory has been added to the PATH environment variable.

The FCU verify and upgrade operations require specially prepared FRI files or HPM files (see [Firmware Recovery Image Files on page 55](#)).

Synopsis

```
fcu --help [-t<slave address>]
fcu --version
fcu -q [-d <device-id>] [-t<slave address>]
fcu -v -f <filename> [-t<slave address>]
fcu -u -f <filename> [-t<slave address>]
fcu -a -f <filename>
fcu -m -b <bank-letter> -d <device-id> [-t<slave address>]
fcu -s -f <filename>
fcu -c -f <filename>
fcu --activate -b <bank-letter> -d <device-id>
```

Parameters

`-a`
`--full-upgrade`

This option is a shortcut for performing the verify, upgrade, and mark operations. The file option `-f` is required. This option should not be combined with other operations.

-r

--activate

This command is available for IPMC devices and images which conform to the HPM.1 standard.

HPM.1 compliant IPMCs store a redundant set of firmware images which may have the states operational, rollback or deferred. Using the `--activate` option, you can set the state of the addressed firmware bank to "operational". Unlike the `--mark` option, the `--activate` command does not affect the payload operation, this means you can set a previously "rollback" firmware bank to "operational" without rebooting or resetting the payload. The target IPMC is immediately functional after switching to the new firmware.

You can obtain the current states of the firmware banks by using the `-q` command. The state appears as part of the bank name, for example, "B - Rollback".

-b <bank-letter>

--bank=<bank-letter>

Specifies a flash bank (A or B, for example), where <bank-letter> is the letter designating a specific bank. This option can be used with the mark or activate operation. Use the query option `-q` to list available banks.

-c

--compare

Compares the image contained in the specified device with a specified file in the file system. This may be useful after an image upgrade to verify that the device actually contains a new and different image.

-d <device-id>

--device=<device-id>

Specifies a target firmware device, where <device-id> is the name of the device. This option is used with the mark or query operations. Device ID values vary by hardware. You can display supported devices on a given blade by using `fcu --help`. Currently supported values are listed in the following table.

Device ID	Description
atca-f140-cpu	U-Boot firmware device for ATCA-F140
atca-f140-kernel	Kernel device of ATCA-F140
atca-f140-hpm.1-ipmc	IPMC firmware device

Firmware Upgrade Facility

`-f <filename>`

`--file=<filename>`

Specifies the FRI file, where `<filename>` is the complete path and filename of the image file. This option is used with the verify and upgrade operations.

`--force`

This option allows the installation of images with non-matching part-number and part-revision FRU data fields. This option should be used with extreme caution only because installing an incompatible image on a device may render it inoperable.

`--help`

Displays a brief message describing command usage. It also displays a list of the devices supported on the blade. This option is exclusive and should not be used with other options.

`-m`

`--mark`

Informs FCU to set the boot select so that on the next boot the specified firmware bank will be active. When mark is combined with the upgrade operation, there is no need to specify a bank; the bank just upgraded will be marked. Otherwise, you must specify a bank and a device.

Note that on the ATCA-F140 the firmware image and the kernel image are coupled, this means when you mark a U-Boot image for the next reboot, the corresponding Linux kernel image is automatically marked as well. Refer also to the *ATCA-F140 Installation and Use* guide for details about the flash memory layout.

`-q`

`--query`

Informs FCU to return firmware information for a specific device (if used with `-d`) or information about all firmware devices. The query operation is exclusive and is not intended to be combined with other operations.

Some firmware images are stored in redundant flash devices which can be marked for use after the next reboot (using the `-m` command). For these firmware images, the `-q` command displays whether image is marked or not. For HMP.1 compliant firmware images, the `-q` command displays the state of the image, i.e., operational, rollback or deferred.

`-s`

`--show`

Shows detailed information about a specified file. The information shown includes for example image type, version, manufacturer name etc. This command may be useful before a firmware upgrade to determine the version of a new image file.

-t

--target

This option is needed to specify the IPMC address if the operation is to be done on a remote IPMC. If you do not specify this option, the software tries to access the local IPMC. The -t option provides a possibility to perform the firmware upgrade on a different blade.

-u

--upgrade

Tells FCU to upgrade the currently inactive bank of the device specified by the target FRI file. The file option -f is required. The upgrade operation may be combined with the verify and mark operations.

-v

--verify

Tells FCU to verify the image file specified by the required -f option. This operation verifies that the specified file is sound and compatible with the current hardware. The verify operation may be combined with the upgrade and mark operations.

--version

Displays version information for the utility. This option is exclusive and should not be used with other options.

Usage

Some FCU options can be combined. Some options are exclusive. The following list describes the valid option combinations:

- --compare --file=<filename>
- --full-upgrade --file=<filename>
- --full-upgrade --file=<filename> --target=xxxxxx
- --help
- --mark --bank=<bank-letter> --device=<device-id>
- --query
- --query --device=<device-id>
- --show --file=<filename>
- --upgrade --file=<filename>
- --upgrade --mark --file=<filename>
- --upgrade --file=<filename> --target=xxxxxx
- --verify --file=<filename>

Firmware Upgrade Facility

- `--verify --upgrade --file=<filename>`
- `--activate --bank=<bankletter> --device=<deviceID>`
- `--verify --upgrade --mark --file=<filename>`
- `--version`

Multi-character options may be abbreviated so long as they are unique. For example, `--full` is equivalent to `--full-upgrade`. Typing `--ver`, however, will not work since it matches both `--verify` and `--version`.

Single-character options may be combined without repeating the hyphen, as shown in the following examples:

- `fcu -vf /opt/bladeservices/rom/<filename>`
- `fcu -q -d <device-id>`
- `fcu -q -t 0x90 <device-id>`
- `fcu -mb a -d <device-id>`

Options are not case-sensitive. For example, `--help` is equivalent to `--HeLp`. However, option arguments, such as `filename` and `device ID` are case-sensitive.

When upgrading firmware, it is strongly recommended that you upgrade only one device at a time. While FCU performs many checks during upgrade to ensure success, if something goes wrong and both firmware banks become corrupted, the blade will be inoperable.

4.4 Upgrading Firmware

This section describes recommended procedures for upgrading firmware devices.

4.4.1 Upgrading CPU Firmware

Upgrading CPU Firmware

Follow these steps to upgrade the CPU firmware (U-Boot):

1. Query the current and backup U-Boot firmware on the ATCA-F140 blade.
`# fcu -qd atca-f140-cpu`
2. Show the version of the new U-Boot image.
`# fcu -s -f /opt/bladeservices/rom/atca-f140-cpu.fri`
3. If the U-Boot image is newer than the current version, upgrade the U-Boot firmware.
`# fcu -v -u -f /opt/bladeservices/rom/atca-f140-cpu.fri`
4. Query the device to ensure that the backup was upgraded.
`# fcu -qd atca-f140-cpu`

4.4.2 Upgrading the Device-Tree Blob

Upgrading the Device-Tree Blob

Follow these steps to upgrade the device-tree blob:

1. Query the current and backup device-tree blob images.
`# fcu -q -d atca-f140-dtb`
2. Show the version of the new device-tree blob.
`# fcu -s -f /opt/bladeservices/rom/atca-f140-dtb.fri`
3. If the device-tree-blob version in the release is newer than the current version installed, upgrade the device-tree blob.
`# fcu -v -u -f /opt/bladeservices/rom/atca-f140-dtb.fri`
4. Query the device to ensure whether the backup is upgraded.
`# fcu -q -d atca-f140-dtb`

4.4.3 Upgrading the Kernel

Upgrading the Kernel

Follow these steps to upgrade the kernel:

1. Query the current and backup kernel images.
`# fcu -q -d atca-f140-kernel`
2. Show the version of the new kernel.
`# fcu -s -f /opt/bladeservices/rom/atca-f140-kernel.fri`
3. If the kernel version in the release is newer than the current version installed, upgrade the kernel.
`# fcu -v -u -f /opt/bladeservices/rom/atca-f140-kernel.fri`
4. Query the device to ensure that the backup was upgraded.
`# fcu -q -d atca-f140-kernel`

4.4.4 Activating the Backup Boot Bank

After upgrading the backup boot bank with new U-Boot, device-tree blob, and kernel images, the backup boot bank must be marked as active for next boot and reboot the system. This can be done via FCU, or via U-Boot, or via the Shelf Manager.

Activating the Backup Boot Bank via FCU

1. To mark the backup boot bank, first query which bank is "Marked for next use"
`# fcu -q -d atca-f140-cpu`

Firmware Upgrade Facility

2. Mark the backup boot bank for next use after the next reboot.

If Bank A is "Marked for next use", mark Bank 'B' for the next reboot.

```
# fcu -mb B -d atca-f140-cpu
```

If Bank B is "Marked for next use", mark Bank 'A' for the next reboot.

```
# fcu -mb A -d atca-f140-cpu
```



The `-mb` option allows you to select which boot bank to use at the next boot and it is done through marking U-Boot. By extension, using the `-mb` option to mark U-boot also marks the device-tree blob and kernel as well. All three must be at the desired version in the target bank while booting that bank.

3. Reboot the blade.

```
# reboot -f
```

Activating the Backup Boot Bank via U-Boot

1. Power-up or reboot the blade.
2. Press `Ctrl-C` when you see the message `Hit Ctrl-C to stop autoboot:` on the console.
3. To switch the boot banks and to use the newly installed U-Boot, kernel, and device-tree blob, enter

```
bootsel switch
reset
```

The backup boot bank will become active after the restart.

Activating the Backup Boot Bank via the Shelf Manager

Refer to the Shelf Manager documentation for the IPMI command to initiate a boot bank switch. Reboot the blade for the boot bank switch to take effect.

4.4.5 Upgrading IPMC Boot Loader Firmware

Upgrading IPMC Boot Loader Firmware

Follow these steps to upgrade the IPMC boot loader firmware. The IPMC boot loader becomes active immediately after the successful upgrade.

1. Query the current IPMC boot loader on the ATCA-F140.

```
# fcu -q -d atca-f140-hpm.1-ipmc
```

2. Show the version of the new IPMC boot loader.
`# fcu -s -f /opt/bladeservices/rom/atca-f140-ipmc-boot-<version>.hpm`
3. If the IPMC boot loader version in the release is newer than the current version installed, upgrade the IPMC boot loader.
`# fcu -v -u -f /opt/bladeservices/rom/atca-f140-ipmc-boot-<version>.hpm`
4. Query the IPMC boot loader to ensure the new version is installed and activated.
`# fcu -qd atca-f140-hpm.1-ipmc`

NOTE: It is recommended that a delay of at least 120 seconds to be maintained between IPMC/IPMC boot loader upgrades.

4.4.6 Upgrading IPMC Firmware

Upgrading IPMC Firmware

The IPMC firmware becomes active immediately after the successful upgrade. To upgrade the IPMC firmware:

1. Query the current IPMC firmware on the ATCA-F140.
`# fcu -q -d atca-f140-hpm.1-ipmc`
2. Show the version of the new IPMC image.
`# fcu -s -f /opt/bladeservices/rom/atca-f140-ipmc-<version>.hpm`
3. If the IPMC version in the release is newer than the current version installed, upgrade the IPMC firmware.
`# fcu -v -u -f /opt/bladeservices/rom/atca-f140-ipmc-<version>.hpm`
4. Query the IPMC to ensure whether the new version is installed and activated.
`# fcu -q -d atca-f140-hpm.1-ipmc`

Linux Distribution Description

The BBS for the ATCA-F140 is based on Wind River Platform for Network Equipment, Linux Edition 3.0, built on Linux 2.6 kernel technology.

5.1 Reliability

The majority of errors caused by improper shutdown are fixed automatically during boot.

5.2 Login

A Linux shell can be accessed via the face plate serial port.

If you use a serial console or terminal emulator, the serial/RTM port settings are 9600 baud, no parity, eight data bits, and one stop bit.

If you use SSH, see [Network Services Configuration](#) below for default IP address assignments.

If you want to login as "root" via SSH, you need to first configure SSH using the console serial port. In the `/etc/ssh/sshd_config` file, set `PermitRootLogin` to `yes`. For this change to take effect, you must either reboot the blade/module or run the command `/etc/init.d/sshd restart`.

The following table lists available default login accounts, see [Section 11.2, Default Configuration on page 146](#).

Login Name	Password
root	root

5.3 Network Services Configuration

The following sections describe the default configuration for network services.



Network services configuration varies for ATCA-F140 blades depending on whether the blade is inserted into logical slot 1 or logical slot 2. Configuration scripts rely on HPM software to determine the slot location (see [Chapter 6, Hardware Platform Management on page 71](#)). If scripts are unable to determine the slot location, they default to a slot 1 configuration regardless of the actual slot. This may result in network conflicts.

5.3.1 ATCA-F140 Ethernet Interfaces

The following interfaces are configured at startup.

Interface	Description	VLAN	IP Address
Slot 1 Configuration			
vlan1.21	Base channel 1	21	192.168.21.1
vlan1.22	Base channel 2	22	192.168.22.1
vlan1.24	Base channel 3	24	192.168.24.1
vlan2.11	Fabric channel 1	11	192.168.11.1
vlan2.12	Fabric channel 2	12	192.168.12.1
eth0	Update Channel Ethernet	None	Configured in U-Boot
eth1	Front panel Ethernet	None	Configured in U-Boot
Slot 2 Configuration			
vlan1.21	Base channel 1	21	192.168.21.2
vlan1.22	Base channel 2	22	192.168.22.2
vlan1.24	Base channel 3	24	192.168.24.2
vlan2.11	Fabric channel 1	11	192.168.11.2
vlan2.12	Fabric channel 2	12	192.168.12.2
eth0	Update Channel Ethernet	None	Configured in U-Boot
eth1	Front panel Ethernet	None	Configured in U-Boot

The following interfaces remain as not configured:

Interface	Description
vlan1	Connection to base interface switch. vlan1.21, vlan1.22, and vlan1.24 are associated with tagged VLANs on this interface. BBS does not configure IP addresses on this interface and instead instantiates VLAN devices (vlan1.21, for example) on it with the <code>vconfig</code> command.
vlan2	Connection to fabric interface switch. vlan2.11 and vlan2.11 are associated with tagged VLANs on this interface. BBS does not configure IP addresses on this interface and instead instantiates VLAN devices (vlan2.11, for example) on it with the <code>vconfig</code> command.
eth0	Update Channel Ethernet

Interface	Description
eth1	Front panel Ethernet
sit0	SIT is a shortcut for Simple Internet Transition. This device is used to encapsulate IPv6 into an IPv4 tunnel. BBS does not configure this device.

For switch port configuration defaults, including VLAN assignments.

5.3.2 DHCP Server

A DHCP server on the ATCA-F140 starts at run levels 2–5. The server configuration varies depending on the slot location.

Slot Location	Listens On	Address Range
Slot 1	Base channel 1	192.168.21.100 to 192.168.21.125
	Fabric channel 1	192.168.11.100 to 192.168.11.125
Slot 2	Base channel 2	192.168.22.100 to 192.168.22.125
	Fabric channel 2	192.168.12.100 to 192.168.12.125
Slot 3	Fabric channel 3	192.168.13.100 to 192.168.13.125
Slot 4	Fabric channel 4	192.168.14.100 to 192.168.14.125

NOTE: Slot 3 and Slot 4 configurations mentioned in the above table are applicable to AXP14-400 chassis.

5.3.3 TFTP Server

The "xinetd" daemon starts running the TFTP server on the ATCA-F140 blade, when required. The "xinetd" daemon starts at run levels 3, 4, and 5.

5.3.4 SSH

A secure shell server starts at run levels 2–5 and listens on all Ethernet interfaces listed in [ATCA-F140 Ethernet Interfaces on page 66](#).

5.4 Log File Management

BBS, as well as Linux in general, generates various log files which are found in the `/var/log` directory. System administrators should take care that the log files do not grow too large, which may have a negative impact on their systems.

BBS provides the standard Linux *logrotate* package for managing the system log files as well as a set of configuration files for managing each log file individually or in groups. Configuration options allow system administrators to tailor the management of log files including, but not limited to, the following items:

- Maximum size of a log file before triggering *logrotate*
- Compression options
- Frequency of rotations
- Number of older log files to keep before being removed
- Notification options

The *logrotate* configuration files and scripts can be found in the `/etc/logrotate.d` directory. For more information on how to tailor these to your system's specific needs, refer to the Linux man page for *logrotate*.

5.4.1 Log Files

The following is the list of log files managed by *logrotate*:

- `/var/log/ha-debug`
- `/var/log/ha-log`
- `/var/log/hpibSubagent.log`
- `/var/log/named.log`
- `/var/log/ppp/connect-errors`
- `/var/log/sfptool.log`
- `/var/log/snmpd.log`
- `/var/log/srstackware.log`
- `/var/log/kern.log`
- `/var/log/messages`
- `/var/log/secure`
- `/var/log/maillog`
- `/var/log/spooler`
- `/var/log/boot.log`

- /var/log/cron
- /var/log/tcm/tclk_agent.log
- /var/log/vsftpd.log
- /var/log/wtmp

Hardware Platform Management

Hardware management in AdvancedTCA systems is based on the Intelligent Platform Management Interface (IPMI) specification. IPMI commands can be complex and cumbersome. To facilitate blade-level management, Penguin Solutions provides the Hardware Platform Management (HPM) package, which provides a set of commands that are based on IPMI commands. These commands are easier to use than the IPMI command itself. An HPM command can encapsulate a sequence of IPMI commands, for example upgrade the firmware or read the FRU data. An HPM command can be the unifier for OEM IPMI commands, which are different on different blade types, for example reading the CPU boot bank. For a catalogue of supported IPMI commands of the blade, refer to the respective IPMI manual.

The HPM package consists of:

- HPM daemon called `hpmagentd`
- Command line client called `hpmcmd`
- Script framework for managing shutdown and reboot events

The `hpmcmd` sends a given HPM command to the `hpmagentd` and displays the received response on the console. The `hpmagentd` executes the incoming HPM commands and returns the result to a `hpmcmd` client.

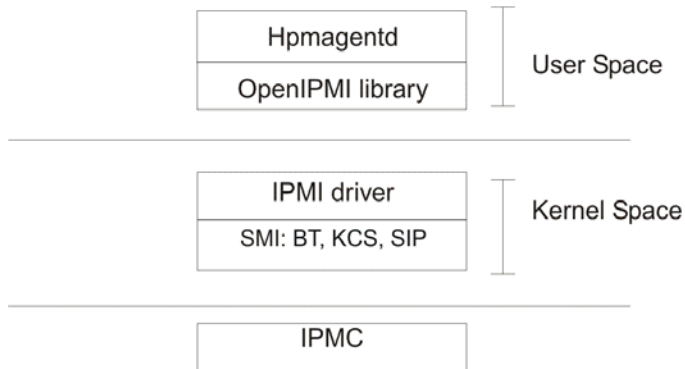
HPM commands include:

- Retrieving and modifying FRU data
- Reading and controlling status of IPMI-controlled LEDs
- Relaying e-key events to SRstackware on both hub and node blades
- Executing shutdown and reboot scripts in response to cold reset or graceful reboot requests
- Communicating local slot location information

Hardware Platform Management

The hpmagentd uses OpenIPMI to communicate with the local IPMC. OpenIPMI consists of two main parts: A device driver that goes into the Linux kernel, and a user-level library. The following picture shows the software levels that are involved in the HPM architecture.

Figure 6-1 Software Levels of the HPM Architecture



BT Block Transfer Interface
SIP Serial Interface Protocol
SMI System Management Interface
KCS Keyboard Control Style

The SMI (System Management Interface) driver provides the KCS interface to communicate with the IPMC.

If you need more information about the software aspects of the blade IPM controller, refer to the respective IPMI manual.

6.1 hpmagentd–HPM Agent Daemon

Description

The HPM agent daemon handles local communication to the intelligent platform management controller (IPMC) on a blade using the SMI. This SMI gets set up by the OpenIPMI driver.

By default, the hpmagentd binary executable is installed in `/opt/bladeservices/bin`. This directory has been added to the PATH environment variable.

This daemon has an Init script called `hpm`, which starts the daemon at run level 3 with the default settings.

When hpmagentd receives a graceful reboot or shutdown alert from the IPMC, it calls the respective script to run the reboot or shutdown sequence.

Synopsis

```
hpmagentd [-l log-level] [-r reboot-script] [-s shutdown-script]
hpmagentd {-i | -u | -h | -v}
```

Parameters

-l log-level

Specifies the level of message logging, where log-level is one of the standard syslog levels.

Log Level	Description
0	Emergency
1	Alert
2	Critical
3	Error
4	Warning
5	Notice (default)
6	Information
7	Debug

-r reboot-script

Specifies a graceful reboot script, which is called when a blade graceful reboot request is received by the IPMC, where reboot-script is the complete path and filename of the target script. The default script is `/opt/bladeservices/bin/hpmreboot` (see [hpm–Shutdown and Reboot Scripts on page 74](#)).

-s shutdown-script

Specifies a shutdown script that will be called when a blade shutdown request is received by the IPMC, where shutdown-script is the complete path and filename of the target script. The default script is `/opt/bladeservices/bin/hpmshutdown` (see [hpm–Shutdown and Reboot Scripts on page 74](#)).

-L

Disable LED management

-i

hpmagentd runs interactively, that is it will not run as daemon.

`-u | -h`

Displays a brief message about command usage.

`-v`

Displays the version of `hpmagentd` and the version of the OpenIPMI library it is linked against.

6.2 hpm–Start-Up Script

Description

An HPM agent init script, `hpm`, allows you to start, stop, and restart the HPM agent daemon using the agent's default option settings. By default, this script is installed in the `/opt/bladeservices/etc/init.d` directory during installation of the BBS software. It is also linked to `/etc/rc.d/rc3.d` to automatically start the HPM agent when the system boots.

Synopsis

```
hpm {start | stop | restart | force-reload}
```

Parameters

`start`

Starts the `hpm` agent daemon.

`stop`

Terminates the `hpm` agent daemon.

`restart`

Terminates and then starts the `hpm` agent daemon.

`force-reload`

Terminates and then starts the `hpm` agent daemon.

6.3 hpm–Shutdown and Reboot Scripts

Description

At any time during normal operation, a shelf manager may issue a shutdown (FRU Activation Deactivate) or graceful reboot (FRU Control Reboot) request to the IPMC on a given blade. The IPMC then forwards this information to the HPM agent. When the HPM agent receives such request, it calls the respective script to run the reboot or shutdown sequence. In case of a shutdown indication, all running processes should be notified about

the shutdown. In case of a reboot notification, the payload is responsible for invoking the reboot procedure. The IPMC is not involved in this process. This allows processes currently running on the blade to prepare for shutdown. After the notification, it takes roughly 30 seconds before the payload is powered off.

Two default scripts, `hpmshutdown` and `hpmreboot`, are installed by default at `/opt/bladeservices/bin` directory. Currently, these scripts print a banner indicating they have run and then issue `shutdown -h now` (`hpmshutdown` script) or `reboot` (`hpmreboot` script).

You can modify the default scripts to suit the needs of your system application or create new scripts. If you create new scripts, use the `-s` and `-r` options when starting `hpmagentd` to specify the new locations and names of the scripts. You may also need to update the `hpm` start up script at `/opt/bladeservices/etc/init.d/hpm`

Synopsis

```
hpmshutdown
hpmreboot
```

6.4 hpmcmd—HPM Command Utility

Description

The HPM command utility uses a socket to send commands to the HPM agent. The HPM agent takes care of translating the user-friendly commands into the elaborated IPMI commands that the IPMC is able to understand. Those IPMI commands are transferred to the local IPMC.

Only one HPM command can be outstanding with the HPM agent at any particular moment. This means that even though multiple instances of `hpmcmd` can be started, the HPM agent will handle only one command at a time. Once a command is sent, the `hpmcmd` program waits until the answer from the HPM agent is received or until a time-out occurs.

The HPM command utility can be started in interactive mode, where a prompt is displayed and the user enters commands; it can read in a file of commands; or it can process a single command.

By default, the `hpmcmd` binary executable is installed at `/opt/bladeservices/bin`. During installation of the BBS software, this directory is added to the `PATH` environment variable.

If you do not provide any options, you will see the prompt `hpmcmd>` once the program starts running.

From there you can start executing commands.

Hardware Platform Management

Synopsis

```
hpmcmd [-p new-prompt] [-o output] [-i input | -c command]
hpmcmd [--prompt new_prompt] [--output_file output] [--input_file
input | -cmd_line command]
```

Parameters

`-p new-prompt`

Specifies the prompt you would like to have for the `hpmcmd` interactive mode, where `new-prompt` is any string. The default prompt is `hpmcmd>`. This option should not be combined with the `-i` or `-c` options.

`-i input-file`

Specifies the name of a file with HPM commands, where `input-file` is the complete path and filename of the target file, a standard ASCII file with one command per line (comments are not supported). The default is Standard Input (stdin). This option should not be combined with the `-c` option.

Once it has executed all commands in the file, `hpmcmd` terminates.

`-o output-file`

Specifies the name of an output file, where `output-file` is the complete path and filename of the target file. The default is Standard Output (stdout).

`-c command`

This option executes a single command and terminates, where `command` is one of the supported commands. This allows you to use the arrow history functions supported in the base shell; a history is not available inside the `hpmcmd` program. This option should not be combined with the `-i` option.

If this option is combined with `-o`, `-c` should be last option entered, since all arguments that follow `-c` on the command line will be considered part of the command.

6.4.1 Command Overview

The following table lists all commands from the hpmcmd program available on the ATCA-F140. You can display this list and a short command description using the help command (see section [help on page 85](#)). A detailed description of the commands is given in section [Supported Commands on page 79](#).

Table 6-1 Command Overview

Command	Description
<i>bootbankget</i>	Gets the bootbank to boot from
<i>bootbankset</i>	Sets the bootbank to boot from
<i>bye</i>	Exits the hpmcmd program
<i>cmd</i>	Executes any IPMI command
<i>deviceid</i>	Gets the Device Id
<i>ekeydownpath</i>	Gets the location/path of a script which is invoked on ekeying port down events
<i>ekeyuppath</i>	Gets the location/path of a script which is invoked on ekeying port up events
<i>exit</i>	Exits the hpmcmd program
<i>frudata</i>	Gets FRU info in hex numbers
<i>fruinfoget</i>	Gets string fields from the FRU
<i>fruinv</i>	Gets the FRU size and addressable units
<i>fruread</i>	Gets x number of bytes from the FRU
<i>fruwrite</i>	Writes x number of bytes from the FRU
<i>help</i>	Gets list of hpmcmd commands
<i>ipmbaddress</i>	Gets the local board IPMB address
<i>ipmcdevice</i>	Gets the payload interface to the IPMC
<i>ledget</i>	Gets the state of a specific FRU LED
<i>ledprop</i>	Get the LED properties for this FRU
<i>ledset</i>	Controls the state of a specific FRU LED
<i>loglevelget</i>	Gets the hpmagentd log level
<i>loglevelset</i>	Sets the hpmagentd log level (0-7)

Hardware Platform Management

Table 6-1 Command Overview (continued)

Command	Description
<i>macaddress</i>	Gets the MAC addresses
<i>motshelftype</i>	Gets the Penguin Edge Shelf Type from the Shelf FRU (Board Product Name)
<i>muxctrl</i>	Sets the specified MUX control signal
<i>partnumber</i>	Gets the board part number
<i>physlotnumber</i>	Gets the board physical slot number
<i>portget</i>	Gets the current state E-Key governed intfs
<i>portset</i>	Enables/Disables ports in a channel
<i>posttypeget</i>	Gets the POST type from the IPMC
<i>posttypeset</i>	Sets the POST type to run at boot time
<i>quit</i>	Exits the hpmcmd program
<i>rebootpath</i>	Gets hpmagentd reboot script path
<i>sdr</i>	Gets the SDR records
<i>sdr_dump</i>	Gets the SDR records in raw format
<i>sdrinfo</i>	Gets SDR information
<i>sendamc</i>	Sends an IPMI command to a remote AMC or MMC of a remote IPMC
<i>sendcmd</i>	Sends an IPMI command
<i>shelfaddress</i>	Gets the Shelf Address String
<i>shelfslots</i>	Gets number of slots in the shelf
<i>shutdownpath</i>	Gets hpmagentd shutdown script path
<i>slotmap</i>	Gets the slotmap of the shelf
<i>slotnumber</i>	Gets the board logical slot number
<i>version</i>	Gets the hpmcmd version and the hpmagentd version
<i>watchdog</i>	Controls Payload WDT functionality

6.4.2 Supported Commands

This section lists the supported commands. All commands are case insensitive. The examples illustrate the use of `hpmcmd` in single-command mode (`-c`). If you start `hpmcmd` without the `-c` or `-i` options (that is, interactive mode), you simply enter these commands at the HPM command prompt.

Some of the `hpm` commands can be sent to a remote IPMC by specifying the `-t` option. This option is not mandatory. If it is not specified, the command is sent to the local IPMC. The following sections describe the available commands.

6.4.2.1 `bye`

Description

This command is for exiting the `hpmcmd` program when running in interactive mode.

Synopsis

```
bye
```

6.4.2.2 `bootbankget`

Description

This command retrieves the boot bank, which is currently marked as active for the CPU specified by `payload_cpu_selector`.

Firmware for the CPU on Penguin Edge AdvancedTCA blades is stored in redundant, persistent memory devices. This allows the firmware image in one bank to serve as a backup for the other bank. During normal operation, the CPU on a blade determines which bank to boot from based on a GPIO signal controlled by the IPMC. This bank is considered the active boot device.

Because you can change the “active” device with the `hpmcmd bootbankset` command, active status does not necessarily indicate which device was used on the last boot. It simply represents which device is set to be used on the next boot.

Synopsis

```
bootbankget <payload_cpu_selector>
```

Parameters

```
payload_cpu_selector
```

Is an integer between 0 and the number of CPU devices supported on the blade. On the PrAMC-7201, the only valid value for `payload_cpu_selector` is 0.

Example

```
hpmcmd -c bootbankget 0
```

6.4.2.3 bootbankset

Description

This command sets the boot bank for a particular CPU from which the blade is supposed to boot.

Synopsis

```
bootbankset <payload_cpu_selector> <newBootBank>
```

Parameters

`payload_cpu_selector`

Is an integer between 0 and the number of CPU devices supported on the blade.

`newBootBank`

Can be set to BANK0 or BANK1

Example

```
hpmcmd -c bootbankset 0 bank1
```

6.4.2.4 cmd

Description

This command allows you to enter commands understood by the IPMC. Commands are entered as a sequence of hexadecimal numbers as defined in the *IPMI 1.5 Specification*.

Synopsis

```
cmd <ipmi address> <netfn cmd> <cmd data>
```

Parameters

`ipmi address`

The IPMI address specifies the IPMC that receives the command, it can be the local IPMC or another IPMC on the IPMB. The IPMI address for the local IPMC consists of <f LUN> where f is the BMC channel number. The IPMI address for a remote IPMC consists of <0 SA LUN>.

`netfn cmd`

Identifies the command type.

`cmd data`

Specifies the message data associated with the command.

Example

GetDeviceId command to the local IMPC:

```
hpmcmd -c cmd f 0 6 1
```


GetDeviceId command to the remote IPMC on address 9a:

```
hpmcmd -c cmd 0 9a 0 6 1
```

GetDeviceId command to the remote IPMC on address 7a:

```
hpmcmd -c cmd 0 7a 0 6 1
```

6.4.2.5 deviceid

Description

This command retrieves the raw IPMI Get Device ID response and decodes the IPMI message.

Synopsis

```
deviceid [-t ipmbAddr[:mmcAddr]]
```

Parameters

-t

Sends the command to `ipmbAddr[:mmcAddr]`. `ipmbAddr` is the string `lc` if it is a local `mmcAddr`.

Example

```
hpmcmd -c deviceid
```



The least significant byte of the Auxiliary Revision indicates the build number inside the release.

6.4.2.6 ekeydownpath

Description

Gets the location/path of a script which is invoked whenever an interface port is to be closed. The script is used to implement an e-keying mechanism in software and will invoke `ifconfig` to close the port. You may want to do modifications to this script.

Synopsis

```
ekeydownpath
```

6.4.2.7 ekeyuppath

Description

Gets the location/path of a script which is invoked whenever an interface port is to be opened. The script is used to implement an e-keying mechanism in software and will invoke `ifconfig` to open the port. You may want to do modifications to this script.

Synopsis

ekeyuppath

6.4.2.8 exit

Description

This command is to exit the hpmcmd program when running in interactive mode.

Synopsis

exit

6.4.2.9 frudata

Description

This command dumps the content of the FRU data in hexadecimal format.

Synopsis

```
frudata <fruid> [-t ipmbAddr[:mmcAddr]]
```

Parameters

fruid

Is 0 for the main blade and 1 for the RTM.

-t

Sends the command to ipmbAddr[:mmcAddr]. ipmbAddr is the string lc if it is a local mmcAddr.

Example

```
hpmcmd -c frudata 0
```

6.4.2.10 fruinfoget

Description

This command retrieves information from the specified FRU.

Synopsis

```
fruinfoget <fruid> [field] [-v] [-t ipmbAddr[:mmcAddr]]
```

Parameters

fruid

Is 0 for the main blade and 1 for the RTM.

field

Is one of the following data fields. If no field is specified, it retrieves the whole fruinfo for that FRU.

Field	Description
bmanufacturer	Board manufacturer
bproductname	Board product name
bserialnumber	Board serial number
bpartnumber	Board part number
pmanufacturer	Product manufacturer
pproductname	Product product name
ppartnumber	Product part number
pversion	Product version number
pserialnumber	Product serial number
passetag	Product inventory asset identifier

-v

Verbose mode to get point-to-point connectivity information where no specific field is requested.

-t

Sends the command to `ipmbAddr[:mmcAddr]`. `ipmbAddr` is the string `lc` if it is a local `mmcAddr`.

Example

```
hpmcmd -c fruinfoget 1 bmanufacturer
```

The following example for `fruinfoget` is without fields and `-v` option.

```
hpmcmd -c fruinfoget 0
```

6.4.2.11 fruinv

Description

This command retrieves the FRU size and the addressable unit for the specified FRU.

Synopsis

```
fruinv <fruid> [-t ipmbAddr[:mmcAddr]]
```

Hardware Platform Management

Parameters

`fruid`

Is 0 for the main blade and 1 for the RTM (if supported).

`-t`

Sends the command to `ipmbAddr[:mmcAddr]`. `ipmbAddr` is the string `lc` if it is a local `mmcAddr`.

Example

```
hpmcmd -c fruinv 0
```

6.4.2.12 fruread

Description

This command gets a range of data from the specified FRU.

Synopsis

```
fruread <fruid> <startAddress> <nBytes> [-t ipmbAddr[:mmcAddr]]
```

Parameters

`fruid`

Is 0 for the main blade and 1 for the RTM (if supported).

`startAddress`

Is the starting address in decimal.

`nBytes`

Number of bytes to read in decimal; cannot exceed 16 because of IPMI message size limitations.

`-t`

Sends the command to `ipmbAddr[:mmcAddr]`. `ipmbAddr` is the string `lc` if it is a local `mmcAddr`.

Example

```
hpmcmd -c fruread 0 0 8
```

6.4.2.13 fruwrite

Description

This command allows to write `x` number of bytes to a FRU.

Synopsis

```
fruwrite <fruid> <startAddress> <nBytes> [-t ipmbAddr[:mmcAddr]]
```

Parameters

`fruid`

Is 0 for the main blade.

`startAddress`

Starting address in decimal numbers.

`nBytes`

is the number of bytes to write in decimal. `nBytes` cannot exceed 16 because of IPMI message size limitations.

`-t`

Sends the command to `ipmbAddr[:mmcAddr]`. `ipmbAddr` is the string `lc` if it is a local `mmcAddr`.

6.4.2.14 help

Description

This command lists the available commands from the `hpmcmd` program with a brief explanation about the command.

Synopsis

`help`

6.4.2.15 ipmbaddress

Description

This command retrieves the blade IPMB address.

Synopsis

`ipmbaddress`

6.4.2.16 ipmcdevice

Description

This command retrieves the payload tty device.

Synopsis

`ipmcdevice`

6.4.2.17 ledget

Description

This command gets information about a specified LED controlled by the IPMC.

Synopsis

```
ledget <fruid> <led> [-t ipmbAddr[:mmcAddr]]
```

Parameters

fruid

Is 0 for the main blade and 1 for the RTM (if supported).

led

Is BLUE for the hot swap LED or LEDN for FRU LED<n>. <n> is a number between 1 and the maximum FRU LEDs supported by the blade.

-t

Sends the command to ipmbAddr[:mmcAddr]. ipmbAddr is the string lc if it is a local mmcAddr.

Example

```
hpmcmd -c ledget 0 led1
```

6.4.2.18 ledprop

Description

This command displays the FRU LED properties under IPMC control.

Synopsis

```
ledprop <fruid>
```

Parameters

fruid

0 for the main board and 1 for the RTM.

Example

```
hpmcmd -c ledprop 0
```

```
FRU LEDs under IPMC control:
```

```
LED0 = BLUE
```

```
LED1 = RED or AMBER
```

```
LED2 = GREEN
```

6.4.2.19 ledset

Description

This command controls the override state of a specific FRU LED. The RTM FRU LEDs reflect the state of the main blade (FRU 0) LEDs. Therefore, overriding the state to something different than the main FRU LED state will not have any effect.

The blue LED is the only one that can be controlled separately.

Synopsis

```
ledset <fruid> <led> <operation> [offms] [onms] [color] [-t  
ipmbAddr[:mmcAddr]]
```

Parameters

fruid

Is 0 for the main blade and 1 for the RTM (if supported).

led

Is BLUE for the hot swap LED or LEDN for FRU LED<n>. <n> is a number between 1 and the maximum FRU LEDs supported by the blade

operation

ON = enable override state and turn LED on.

OFF = enable override state and turn LED off.

BLINK = enable override state and blink LED; off_duration and on_duration specify the blink duration; the default on and off duration is 300 ms.

LOCAL = cancel override state and restore LED control to the IPMC, that is, local state.

TEST = run lamp test for specified on_duration, then restore prior state.

offms

10–2500 in 10-millisecond increments; only valid if operation is BLINK

onms

Only valid if operation is BLINK or TEST:

If operation is BLINK, 10–2500 in 10-millisecond increments

If operation is TEST, 100-12800 in 100-millisecond increments

color

LED0 = BLUE

LED1 = RED

LED2 = GREEN

LED3 = AMBER

-t

Sends the command to ipmbAddr[:mmcAddr]. ipmbAddr is the string lc if it is a local mmcAddr.

Example

```
hpmcmd -c ledset 0 led1 on
```

Hardware Platform Management

6.4.2.20 loglevelget

Description

This command retrieves the current `hpmagentd` log level. See `loglevelset` for more detail.

Synopsis

```
loglevelget
```

Example

```
hpmcmd -c loglevelget
```

```
LogLevel 5 (NOTICE)
```

6.4.2.21 loglevelset

Description

This command sets the level of message logging for `hpmagentd`.

Synopsis

```
loglevelset <newLogLevel>
```

Parameters

`newLogLevel`

Is one of the standard syslog levels:

Level	Description
0	Emergency
1	Alert
2	Critical
3	Error
4	Warning
5	Notice
6	Information
7	Debug

Example

```
hpmcmd -c loglevelset 7
```


6.4.2.22 macaddress

Description

This command retrieves a list of available MAC addresses.

Synopsis

```
macaddress [-t ipmbAddr]
```

Parameters

-t

Sends the command to `ipmbAddr`.

Example

```
hpmcmd -c macaddress
```

```
BASE Interface Channel 0: 00:0E:0C:85:E9:91
```

```
BASE Interface Channel 1: 00:0E:0C:85:E9:90
```

6.4.2.23 motshelftype

Description

This command retrieves the shelf FRU (IPMB 20) Board Area Product Name (FRU 254).

Synopsis

```
motshelftype
```

Example

```
hpmcmd -c motshelftype
```

```
CHS1406
```

6.4.2.24 muxctrl

Description

This command sets the specified MUX control signal.

Synopsis

```
muxctrl mux value
```

Parameters

mux

mux1

mux2

...

mux15
value

 B1
 B2

For more information on setting different MUXes, see [Appendix A, MUXes Settings on page 217](#).

6.4.2.25 partnumber

Description

This command retrieves the part number of the main blade.

Synopsis

```
partnumber [-t ipmbAddr[:mmcAddr]]
```

Parameters

-t

Sends the command to `ipmbAddr[:mmcAddr]`. `ipmbAddr` is the string `lc` if it is a local `mmcAddr`.

Example

```
hpmcmd -c partnumber
```

6.4.2.26 physlotnumber

Description

This command retrieves the physical slot number in which the blade is plugged in.

Synopsis

```
physlotnumber
```

6.4.2.27 portget

Description

This command shows the current state of interfaces governed by e-keying. If no channel is specified, `portget` returns data for all channels in the specified interface. If neither interface nor channel are specified, `portget` will return data for all interfaces.

Synopsis

```
portget [interface] [channel] [-t ipmbAddr[:mmcAddr]]
```

Parameters

interface

Valid values are:

BASE | FABRIC | UPDATE

channel

an integer in the following range:

1–16 for Base

1–15 for Fabric

1 for Update

The value of channel must be valid for the blade. For example, node blades have only 2 channels for the base interface; using a value of 4 will return an error.

-t

Sends the command to `ipmbAddr[:mmcAddr]`. `ipmbAddr` is the string `lc` if it is a local `mmcAddr`.

Example

```
hpmcmd -c portget AMC 0
```

6.4.2.28 portset

Description

This command enables and disables ports in a channel. The following table lists the valid values for each parameter.

Synopsis

```
portset <intf> <chan> <grpId> <type> <typeX> <ports> <oper> [-t  
ipmbAddr[:mmcAddr]]
```

Parameters

intf

Valid values are:

BASE | FABRIC | UPDATE

chan

an integer in the following range:

1–16 for Base

1–15 for Fabric

1 for Update

The value of channel must be valid for the blade. For example, node blades have only 2 channels for the base interface; using a value of 4 will return an error.

grpId

Hardware Platform Management

Always 0 according to current shelf FRU information

type

Valid values are:

Valid Value	Description
BASE	for base interface
ETHER	for fabric interface
OEM	for the update interface, which is Penguin Edge specific

typeX

Always 0 in current implementation. Valid values are:

0 (for 1000Base-BX)

1 (for 10GBase-BX4)

2 (for FC-PI)

ports

A sequence of ports to act on.

For base and update channels, port is always 0.

For fabric channels, port can specify up to 4 ports as specified in PICMG 3.1:

Option 1: 0

Option 2: 01

Option 3: 0123

Option 7: 3

oper

Valid values are DISABLE or ENABLE.

-t

Sends the command to `ipmbAddr[:mmcAddr]`. `ipmbAddr` is the string `lc` if it is a local `mmcAddr`.

Example

```
hpmcmd -c portset base 1 0 base 0 0 enable
```

6.4.2.29 postypeget

Description

This command retrieves the postype to which the board is currently set to run at boot time.

Synopsis

```
postypeget
```

6.4.2.30 posttypeset

Description

This command sets the posttype to which the board is currently set to run at boot time.

Synopsis

```
posttypeset <newPostType>
```

Parameters

newPosttype

SHORT

LONG

OFF (disabled in system boot parameters)

6.4.2.31 quit

Description

This command is for exiting the hpmcmd program when running in interactive mode.

Synopsis

```
quit
```

6.4.2.32 rebootpath

Description

This command retrieves the path and filename of the current hpmagentd reboot script.

Synopsis

```
rebootpath
```

Example

```
hpmcmd -c rebootpath
```

```
/opt/bladeservices/bin/hpmreboot
```

6.4.2.33 sdr

Description

This command shows the SDR records.

Synopsis

```
sdr
```

Example

```
hpmcmd -c sdr
```

```
recID 1: full sensor record
    owner is IPMB 20 sensor num 10 on lun 00 channel 00
    logical entity: power module - instance 61
    SBC +1.05V Vtt : voltage : threshold
recID 2: full sensor record
    owner is IPMB 20 sensor num 11 on lun 00 channel 00
    logical entity: power module - instance 61
    SBC +1.1V : voltage : threshold
recID 3: full sensor record
    owner is IPMB 20 sensor num 12 on lun 00 channel 00
    logical entity: power module - instance 61
    SBC +1.2V SAS : voltage : threshold
.
.
.
recID 74: OEM sensor record
```

6.4.2.34 sdr_dump

Description

This command shows the SDR records in binary and hex format.

Synopsis

```
sdr_dump
```

Example

```
hpmcmd -c sdr_dump
```

```
SDR Records:
01 00 51 01 39 20 00 10 14 61 7f 69 02 01 04 22 "..Q.9 ...a.i..."
04 22 12 12 00 04 00 00 33 00 00 00 00 c0 07 cd ". ".....3.....Î"
d0 ca ff 00 00 d8 00 00 c2 00 01 01 00 00 00 ce "..... ".....Î"
53 42 43 20 2b 31 2e 30 35 56 20 56 74 74 "SBC +1.05V Vtt"
.
.
.
61 67 65 20 45 "age E"
```

6.4.2.35 sendcmd

Description

This command allows a user to send any of the commands supported in the IPMI spec to a remote IPMC.

Synopsis

```
sendcmd <IPMBaddress> <netfn> <cmd> <data0> ... <dataN>
```

Parameters

IPMBaddress

Destination IPMB address in hex digits.

netfn

IPMI request net function in hex digits.

cmd

IPMI request command in hex digits

data0 ... dataN

IPMI request data bytes. if any, in hex digits.

Example

```
hpmcmd -c sendcmd 90 06 59  
07 59 C1
```

6.4.2.36 sdrinfo

Description

This command shows the SDR information.

Synopsis

```
sdrinfo
```

Example

```
hpmcmd -c sdrinfo
```

SDR Information:

LUN 0 has 066 sensors; static sensor population

LUN 1 has 000 sensors; static sensor population

LUN 2 has 000 sensors; static sensor population

LUN 3 has 000 sensors; static sensor population

6.4.2.37 sendamc

Description

This command allows sending any of the commands supported in the IPMI spec to a remote AMC or MMC of a remote IPMC IPMB-L.

Synopsis

```
sendamc <IPMBaddress> <MMCaddress> <netfn> <cmd> <data0> ... <datan>
```

Parameters

IPMBaddress

Destination IPMB address in hex digits.

MMCaddress

Destination MMC address in hex digits.

netfn

IPMI request net function in hex digits.

cmd

IPMI request command in hex digits.

data0-datan

IPMI request data bytes. if any, in hex digits.

6.4.2.38 shelfaddress

Description

This command retrieves the shelf address string from the shelf FRU.

Synopsis

```
shelfaddress
```

Example

```
hpmcmd -c shelfaddress
```

```
01
```

6.4.2.39 shelfslots

Description

This command retrieves the total number of blade slots in the shelf.

Synopsis

```
shelfslots
```


Example

```
hpmcmd -c shelfslots
14 slots
```

6.4.2.40 shutdownpath

Description

This command retrieves the path and filename of the current hpmagentd shutdown script.

Synopsis

```
shutdownpath
```

Example

```
hpmcmd -c shutdownpath
/opt/bladeservices/bin/hpmshutdown
```

6.4.2.41 slotmap

Description

This command prints a slotmap table for the shelf the blade is installed in.

Synopsis

```
slotmap
```

Example

```
hpmcmd -c slotmap
```

```
-----
Physical Slot: 01 02 03 04 . 05 06 07 08 09 10 . 11 12 13 14
Logical Slot  : 01 03 05 07 . 09 11 13 04 06 08 . 10 12 14 02
IPMB Address  : 82 86 8A 8E . 92 96 9A 88 8C 90 . 94 98 9C 84
-----
```



The command output may vary based on the chassis in which the command is executed.

6.4.2.42 slotnumber

Description

This command retrieves the logical slot number of the slot where the blade is plugged in.

Synopsis

```
slotnumber
```

Example

```
hpmcmd -c slotnumber  
2
```

6.4.2.43 version

Description

This command retrieves the version of the hpmcmd software and sends a request to get the version of the hpmagent daemon that is running. Once the information is gathered, it is printed.

Synopsis

```
version
```

Example

```
hpmcmd -c version  
hpmcmd version bbs 3.1.0 build X  
hpmagentd version bbs 3.1.0 build X
```

6.4.2.44 watchdog

Description

This command is used handle the payload BMC watchdog.

Synopsis

```
watchdog set <tmr_use> <tmr_action> <pre_timeout> <flags> <lsb_val>  
<msb_val>  
watchdog set default  
watchdog get  
watchdog start  
watchdog stop  
watchdog reset
```

Parameters

set

Following are the possible values:

tmr_use	dont_stop stop
tmr_action	no_action hard_reset power_cycle power_down
pre_timeout	0-255
flags	clear dont_clear
lsb_val	0-255
msb_val	0-255

Link Health Check

The Link Health Check (LHC) package supports configuration and operation of the LHC protocol.

LHC performs two functions within a Centellis system:

- Verification of Layer2 connectivity
- Distribution of active network plane

The LHC daemon can be configured to manage one or more instances. Each instance has a role of either proctor or responder. In the proctor role, LHC periodically sends QUERY messages and expects RESPONSE messages. It also sends RESPONSE messages in response to QUERY messages received from other proctors. In the responder role, LHC sends RESPONSE messages in response to received QUERY messages.

Generally, the LHC daemon on hub blades will be configured with one instance with a proctor role. The LHC daemon on a payload blade with no internal switches will be configured with an instance with a responder role. The LHC daemon on carrier blades with an internal switch will be configured with two instances. One instance with a responder role with respect to the system network and the other with a proctor role with respect to the blade's internal network that connects to the AMCs/PMCs. The LHC daemon on an AMC/PMC will be configured with the instance with a responder role.

The LHC package includes:

- SNMP access to LHC
- Configuration and management daemon
- Command line utility program

7.1 LHC MIB

LHC makes its internal structures available as MIB tables and objects. To enable SNMP access, you must link to the enterprise MIB provided in the LHC package and enable SNMP access on the blade. You can also configure LHC by using the `lhcd` configuration script or via the `lhccmd` command line utility.

7.1.1 Browsing the LHC MIB

To support LHC, Penguin Solutions has created an enterprise MIB known as the LHC MIB, `/opt/bladeservices/mibs/LHC-MIB.txt`. This MIB defines the tables, objects, and MIB variables used to control and monitor LHC.

This MIB can be reviewed with any SNMPv2 compatible MIB browser.

7.1.2 Enabling SNMP Access to the LHC MIB

By default, the LHC MIB is installed at `/opt/bladeservices/mibs`. Add this directory to the `MIBDIRS` environment variable:

```
export MIBDIRS=${MIBDIRS}:/opt/bladeservices/mibs
```

SNMP is enabled automatically when SRS is installed. The configuration `/etc/opt/srstackware/config/snmpd.conf` creates the user (SNMP3) with user name, "admin" and password, "adminpwd123".

By default, SNMP is disabled when BBS is installed without SRS. In this case, to start `snmpd` at boot time, the user has to create a symbolic link under `/etc/rc.d/rc3.d/` directory pointing at `/etc/init.d/snmpd.bbs` and edit `/etc/snmp/snmp.conf`.

7.2 lhcd—LHC Daemon

Description

The Link Health Check (LHC) management daemon, `lhcd`, manages the LHC protocol. It serves several purposes:

- Provides through a command interface, the ability to customize operation of the LHC protocol.
- Collects operational statistics which allow monitoring of the LHC protocol.
- Accepts commands from fault management to change the active network plane.
- Notifies fault management of events of interest, such as responder not responding.

During normal operation, the LHC daemon process, `lhcd`, is created and runs as a system daemon. An `init` script, `lhc`, allows you to start and stop the LHC daemon. For further information, see [lhc—lhc Start-Stop Script on page 104](#).

Synopsis

```
lhcd [-c config_file] -f shelfnum -s slotnum -t sitenum  
[-l log-level][ -p pid-file]
```

```
lhcd -v
```

Parameters

`-h`

Displays a brief usage.

`-c config_file`

If supplied, LHC reads in the specified file (including path) that contains commands to configure LHC.

`-f shelfnum`

Blade is in shelf specified by shelfnum. This is a required option. The shelf number, slot number, site number, and instance number are used in MIB indices to uniquely identify each LHC instance. This is particularly significant in multishelf configurations.

`-s slotnum`

Blade is in slot specified by slotnum, where slotnum is the logical slot number where the blade is inserted. You can retrieve the slot number by entering `hpmcmd -c slotnumber`. This is a required option. The shelf number, slot number, site number, and instance number are used in MIB indices to uniquely identify each LHC instance.

`-t sitenum`

Blade is in site specified by sitenum, where sitenum is the logical site number where the blade is inserted. This is a required option. The shelf number, slot number, site number, and instance number are used in MIB indices to uniquely identify each LHC instance. Even though the site number is used to identify an AMC on a carrier blade, a site number must be supplied for LHC instances that are not running on an AMC/PMC. The recommended value for non-AMC/PMC is 255.

`-l log-level`

Sets log level to one of the standard syslog levels:

Log Level	Description
0	Emergency
1	Alert
2	Critical
3	Error
4	Warning
5	Notice (default)
6	Information
7	Debug

`-p pidfile`

Is the complete path and filename for the daemon's process ID file. This file is used by start-stop daemon to locate the correct lhcd process when stopping the daemon. If no PID file is specified, the file `lhcd.pid` is created in the current working directory (`pwd`) from which LHC was started.

-v

Displays the version information.

7.3 lhc–lhcd Start-Stop Script

Description

The lhc init script allows you to start, stop, and restart the LHC daemon. The script relies on the HPM agent to retrieve important blade-specific information.

By default, this script is installed in the `/opt/bladeservices/etc/init.d` directory during installation of the BBS software.

It is linked to `/etc/rc.d/rc2.d` so that LHC starts automatically in the run level 3.

You can use this script to start, stop, or restart the LHC daemon by changing to the `/opt/bladeservices/etc/init.d` directory and typing lhc with the appropriate option, for example:

```
./lhc start
```

Synopsis

```
lhc {start|debug_start|start_no_cfg|stop|restart|  
force-reload|generate_config}
```

Parameters

start

Starts lhcd.



Note that after starting LHC using the start parameter, the daemon automatically reads in a configuration file in the `/opt/bladeservices/etc/lhc` directory. The filename is of the form `lhc.<shelf>.<slot>.<site>.cfg`.

debug_start

Starts lhcd with debug-level logging (7) rather than the default (5).

start_no_cfg

Starts lhcd without reading in a configuration file.

stop

Terminates lhcd.

restart

Terminates lhcd, and then starts lhcd.

force-reload

Terminates lhcd, and then starts lhcd.

generate_config

Deletes the configuration file of the form, if it exists, and then regenerates it by executing the file `lhcConfig.gen` in the `/opt/bladesevices/etc/lhc` directory.

Environment Variables



You do not need to explicitly set the following variables. Under normal operating conditions the init script sets the environmental variables by executing `/etc/default/hpmvars`.

SHELF_NUMBER	Specifies the shelf number used with the <code>lhcd -f</code> option.
SLOT_NUMBER	Specifies the logical slot number of the blade used with the <code>lhcd -s</code> option.
SITE_NUMBER	Specifies the site number used with the <code>lhcd -t</code> option.

7.4 lhcd—LHC Command Line Utility

Description

The LHC command utility, `lhccmd`, provides shell-level access to the LHC daemon management component (`lhcd`) for LHC configuration and monitoring.

Synopsis

```
lhccmd [-v]
lhccmd command
```

Parameters

`-v`
Shows version of `lhccmd`. Also displays version of `lhcd` if it is running.

7.4.1 Command Overview

The following table lists the commands provided by the `lhccmd` utility. For complete command information, see [Available Commands on page 108](#).

Command	Description	Command Reference
<code>get objectName index</code>	MIB Get	get on page 108
<code>get tableName index</code>	MIB Getrow	get on page 108
<code>help ?</code>	Print usage information	help on page 108

Link Health Check

Command	Description	Command Reference
index tableName	Show MIB index information for a table	index on page 109
loglevel 0...7	Set/Show log level	loglevel on page 109
mem dump	Dump LEAP memory	
mem ignore	Ignore allocated LEAP memory	
mem unignore	Un-ignore all LEAP memory	
next objectName index	MIB Next	next on page 110
next tableName index	MIB Nextrow	next on page 110
objects tableName	Show MIB object IDs for table	objects on page 110
quit q exit	Exit	exit q quit on page 108
set objectName index value_keyword	MIB Enumerated Value Set	set on page 110
setc objectName index "value"	MIB Character String Set	set on page 110
seti objectName index value	MIB Integer Set	set on page 110
seto objectName index value	MIB Octet String Set	set on page 110
tables	Show MIB table IDs	tables on page 111
values objectName	Show values for MIB object	values on page 112
version	Display version information	values on page 112
walk tableName	MIB table Walk	walk on page 112

7.4.2 Command Usage for Common Parameters

Many commands support common parameters to specify MIB objects to operate on. The common command parameters include:

- `tablename`
Name of a MIB table. For specific values, use the `lhccmd tables` command. The `tableName` parameter is case-sensitive. Table names must be entered exactly as they appear in the MIB.

- **objectName**
Name of a MIB object within a table. For specific values, use the `lhccmd objects tableName` command, where `tableName` is the name of a specific table. The `objectName` parameter is case-sensitive. Object names must be entered exactly as they appear in the MIB.
- **index**
MIB index as a sequence of numbers (for example, 2.1.1.12). The format of the index parameter is a dot-separated group of fields. In general, each field of the index is a single 32-bit value. However, fields that denote a MAC address or an IP address have multiple dot-separated elements.

A MAC address has six elements; each element is eight bits of the 48-bit MAC address.

The MAC addresses are ordinarily expressed as a sequence of hexadecimal octets, for example: 00.0E.83.EB.33.6F. To indicate that an index element is a hexadecimal value, prefix it with "0x" as follows: 0x00.0x0E.0x83.0xEB.0x33.0x6F.

The fields that make up index vary for each type of object, as described in the following table. For more information about specific indices, refer to the LHC MIB `/opt/bladeservices/mibs/LHC-MIB.txt`.

MIB Table	Index Format
<code>lhcTable</code>	<code>shelf.slot.site</code>
<code>lhcComTable</code>	<code>shelf.slot.site.instance</code>
<code>lhcProctorTable</code>	<code>shelf.slot.site.instance</code>
<code>lhcResponderTable</code>	<code>shelf.slot.site.instance</code>
<code>lhcL2InterfaceTable</code>	<code>shelf.slot.site.instance.L2interfaceindex</code>
<code>lhcProctorL2InterfaceTable</code>	<code>shelf.slot.site.instance.L2interfaceindex</code>
<code>lhcProctorResponderGroupTable</code>	<code>shelf.slot.site.instance.L2interfaceindex.RGL2address</code>
<code>lhcProctorExpectedResponderTable</code>	<code>shelf.slot.site.instance.L2interfaceindex.RGL2address.ERshelf.Erslot.Ersite</code>
<code>lhcAuthorizedProctorTable</code>	<code>shelf.slot.site.instance.L2interfaceindex.RGL2address.APshelf.APslot.APsite</code>

- **value**
Value to set for a read-write object. For specific values, see the LHC MIB `/opt/bladeservices/mibs/LHC-MIB.txt`.

7.4.3 Available Commands

This section lists the supported commands.

7.4.3.1 exit | q | quit

Description

Terminates lhcd. MIB tables for status and control of LHC are destroyed.

Synopsis

```
exit  
q  
quit
```

7.4.3.2 get

Description

This command gets the specified MIB object or table row. Depending on whether `objectName` or `tableName` is used, the command corresponds to a MIB `get` or MIB `getrow` command.

Synopsis

```
get [objectName] [index]  
get [tableName] [index]
```

Parameters

`objectName`

See [Command Usage for Common Parameters on page 106](#).

`tableName`

See [Command Usage for Common Parameters on page 106](#).

`index`

See [Command Usage for Common Parameters on page 106](#).

7.4.3.3 help

Description

This command displays all available commands.

Synopsis

```
help <command>  
?
```

7.4.3.4 index

Description

This command displays the list of elements that make up the MIB index for the table name provided as an argument. For each index element, the number of 32-bit values required for the element appears in parentheses. For example, an IP address will use four 32-bit values (x.x.x.x) when appearing in a MIB index.

Synopsis

```
index [tableName]
```

Parameters

tableName

See [Command Usage for Common Parameters on page 106](#).

7.4.3.5 loglevel

Description

This command is used to set the LHC level.

Synopsis

```
loglevel [level]
```

Parameters

level

Is a valid syslog log level from 0 to 7.

All log messages with a level equal to or numerically lower than the selected level will appear in the system log file. A value of 0 indicates that only the most critical log messages will appear. A value of 7 indicates that all log messages will appear.

Log Level	Description
0	Emergency
1	Alert
2	Critical
3	Error
4	Warning
5	Notice (default)
6	Information
7	Debug

Link Health Check

7.4.3.6 next

Description

This command gets the next MIB object or table row after the specified object or table row. Depending on whether `objectName` or `tableName` is used, the command corresponds to a MIB `next` or MIB `nextrow` command.

Synopsis

```
next [objectName] [index]
next [tableName] [index]
```

Parameters

`objectName`

See [Command Usage for Common Parameters on page 106](#)

`index`

See [Command Usage for Common Parameters on page 106](#).

7.4.3.7 objects

Description

This command displays a list of MIB objects within `lhcd`.

Synopsis

```
objects [tableName]
```

Parameters

`tableName`

See [Command Usage for Common Parameters on page 106](#).

7.4.3.8 set

Description

This command sets the value of a specified object.

Synopsis

```
set [objectName] [index] [value_keyword]
seti [objectName] [index] [value]
setc [objectName] [index] [\"value\"]
seto [objectName] [index] [value]
```

Parameters

objectName

See [Command Usage for Common Parameters on page 106](#).

index

See [Command Usage for Common Parameters on page 106](#).

value

See [Command Usage for Common Parameters on page 106](#).

Command	Description
set	<p>Enumerated value set. This command sets the requested MIB object to the integer value associated with the value_keyword argument, where value_keyword is the textual representation of the integer value as defined in LHC-MIB. Allowable values are found in the SYNTAX section of a particular MIB object. Values for value_keyword are case-sensitive.</p> <p>For example, the SYNTAX of lhcRole is defined by the MIB as:</p> <pre>SYNTAX lhcRole</pre> <p>To set lhcRowStatus for index 2.2.255.1 to up, either of the following two commands may be used:</p> <pre>seti lhcRowStatus 2.2.255.1 1 (integer set) set lhcRowStatus 2.2.255.1 up (enumerated value set)</pre>
seti	<p>Integer set. Sets the object using an integer value, where value in this case is a specific integer. Value may be a decimal integer or hexadecimal integer (that is, preceded by 0x, for example, 0x100).</p>
setc	<p>Character string set. Sets the object using a character string, where value this case is a character string. Double quotes are a required part of the command syntax.</p> <p>When used with lhcmd from a shell prompt or shell script, the double quotes must be escaped by using backslashes before each double quote or by enclosing the quoted value in single quotes.</p>
seto	<p>Octet string set. Sets the object using an octet string, where value in this case is a dot separated hexadecimal string of the form hh.hh...hh, for example, FF.01.02.A1.</p>

7.4.3.9 tables

Description

This command displays a list of MIB tables supported by lhcd.

Synopsis

tables

Link Health Check

7.4.3.10 values

Description

When an object is defined in the MIB as having an enumerated list of integer values, this command will display both the text (which may be provided as the value argument in the `lhccmd set` command) and the associated integer value (which may be provided as the value argument in the `lhccmd` command).

Synopsis

```
values [objectName]
```

Parameters

objectName

See [Command Usage for Common Parameters on page 106](#).

7.4.3.11 version

Description

Displays the version of the loaded `lhcd` executable.

Synopsis

```
version
```

7.4.3.12 walk

Description

This command walks the specified MIB table, displaying every row.

Synopsis

```
walk [tableName]
```

Parameters

tableName

See [Command Usage for Common Parameters on page 106](#).

7.5 LHC Configuration Script and Configuration File

LHC ships with a configuration script called `lhcConfig.gen`. It is located at `/opt/bladeservices/etc/lhc/`. The first time the LHC start-stop script (see [lhc-lhcd Start-Stop Script on page 104](#)) is used to start the LHC daemon, the configuration script is executed and generates a configuration file with a name of the format `lhc.<shelf>.<slot>.<site>.cfg`. The start-stop script then starts the LHC daemon, passing the generated configuration file name as a command line argument. The LHC daemon reads in the generated configuration file in order to learn its initial configuration.



The `lhc` start-stop script checks for the existence of the generated configuration file. If a configuration file exists, the `lhc` start-stop script does not regenerate a configuration file from the configuration script.

7.6 LHC Fault Management

LHC exists as an aid to Fault Management (FM). LHC checks that the internal system network links are healthy, and reports failure and subsequent recovery of the links. Additionally, LHC is responsible for ensuring that all blades in a system use the same network plane. When FM (or some other entity) on a system controller determines that a switchover of the active network plane must be performed, LHC proctor is notified via its API. LHC then distributes the current active network plane to all blades in the system. On a ATCA-7107 or ATCA-7221 payload blade (responder), the LHC daemon receives the QUERY message indicating that the active plane has changed. LHC informs FM via an "active plane" FM notification. It is then up to the FM software to decide what to do.

7.6.1 Determining to Change Active Network Plane

When FM receives a "RESPONSE delinquent" or "QUERY delinquent" notification, it is an indication that there is a link failure to the blade identified in the notification. Either of these notifications could cause FM to decide to switchover the active network plane.

7.6.2 Controlling the Active Network Plane

If FM decides to switchover the active network plane, a MIB set communicates this fact to LHC. The following LHC command may be used to set the MIB variable, or an actual MIB set could be used: `lhccmd seti lhcProctorRequestedActivePlane 1.2.255.1 2`

This command would tell the (local) Proctor with indexing `<shelf 1, slot 2, site 255, instance 1>` to make plane 2 the active network plane.

7.6.3 Monitoring for Changes to the Active Network Plane

When LHC determines that an active network plane switchover has occurred, LHC generates an "Active Plane" FM notification. The same FM notification is generated whether the LHC role is Proctor or Responder.

7.6.4 Fault Management Notifications

When LHC detects a fault, LHC generates a Fault Management notification. LHC uses syslog to send the notifications to FM. Each FM notification begins with the standard syslog preamble including timestamp, followed by the string "LHC #<instance number>:" where <instance number> is the LHC instance number as used in the MIB indexing. The rest of the notification varies depending upon what fault is being notified. The format of each notification is shown in the following table.

Name	Definition/ Semantics	Syntax	Dependencies
Active Plane	Inform which plane is active	LHC #<inst_num>: active plane <plane_num>, originator sh <shelf> sl <slot> st <site> serial <ser_num>	-
Config file not found	The configuration file specified on the command line was not found.	LHC: Config file (<file_name>) not found, errno=<error_number>	-
Duplicate Response	RESPONSE message has been received with the same sequence number from the same source.	LHC #<inst_num>: RESPONSE duplicate <eth_name>/<shelf>/<slot>/<site>/<mcast_mac_address> seq_num=<sequence_num> src MAC=<source mac address>	-
Late or spurious response	RESPONSE message was received containing an unknown sequence number. The message either arrived late or was sent in error.	LHC #<inst_num>: RESPONSE late or spurious <eth_name>/<shelf>/<slot>/<site>/<mcast_mac_address> seq_num=<sequence_num> src MAC=<source mac address>	-
Layer 2 connection failed	The layer 2 connection failed to open. LHC will not be able to communicate on the indicated interface.	LHC #<inst_num>: L2 connection (<eth_name>) failed with error <error_number>	-

Name	Definition/ Semantics	Syntax	Dependencies
Layer 2 connection restored	The layer 2 connection that previously failed has now been successfully opened.	LHC #<inst_num>: L2 connection (<eth_name>) restored	-
QUERY Delinquent	QUERY message has not been received from an authorized proctor	LHC #<inst_num>: QUERY delinquent <eth_name>/<shelf>/<slot>/<site>/<mcast_mac_address>	For a particular Authorized Proctor, once this notification appears, it will not appear again until the "QUERY Received" notification appears
QUERY Received	QUERY message has been received from an authorized proctor	LHC #<inst_num>: QUERY received <eth_name>/<shelf>/<slot>/<site>/<mcast_mac_address>	For a particular Authorized Proctor, once this notification appears, it will not appear again until the "QUERY Delinquent" notification appears
QUERY from unauthorized proctor	QUERY message has been received from an unauthorized proctor	LHC #<inst_num>: QUERY from unauthorized proctor <eth_name>/<shelf>/<slot>/<site>/<mcast_mac_address>	-
RESPONSE Delinquent	RESPONSE message not received from an expected responder	LHC #<inst_num>: RESPONSE delinquent <eth_name>/<shelf>/<slot>/<site>/<mcast_mac_address>	For a particular expected responder, once this notification appears, it will not appear again until the "RESPONSE received" notification appears

Link Health Check

Name	Definition/ Semantics	Syntax	Dependencies
RESPONSE Received	RESPONSE message has been received from an expected responder	LHC #<inst_num>: RESPONSE received <eth_name>/<shelf>/<slot>/<site>/<mcast_mac_address>	For a particular expected responder, once this notification appears, it will not appear again until the "RESPONSE Delinquent" notification appears
RESPONSE from unexpected responder	RESPONSE message has been received from an unexpected responder	LHC #<inst_num>: RESPONSE from unexpected responder <eth_name>/<shelf>/<slot>/<site>	-
Src MAC address changed	The source MAC address in the QUERY message is different from the MAC address received in the previous QUERY. This could indicate a misconfiguration, i.e., two proctors are configured with the same shelf/slot/site.	LHC <inst_num>: src MAC changed since last QUERY <eth_name>/<shelf>/<slot>/<site>/<mcast_mac_address> new MAC=<new mac address> prev MAC=<previous mac address>	

7.7 LHC Configuration Examples

This section provides some examples on how to configure LHC. LHC configuration is done via the LHC MIB.

An L2 Interface entry represents an L2 interface, such as eth2, or a VLAN on an L2 interface, such as eth2.21. A Responder Group entry represents an LHC QUERY message sent to a configured MAC address (usually a multicast address) on a periodic basis, to which some number of expected Responders should respond with an LHC RESPONSE message. An Expected Responder entry represents an entity from whom an LHC RESPONSE message is expected to be received. An Authorized Proctor entry represents an entity from whom an LHC QUERY message is expected and allowed to be received.

The examples in this section are all based on the following single shelf system (shelf #1). Configuration of only the base interface is shown, whereas normally, the base and fabric interfaces are configured. The example single shelf system contains two switches (in logical slots 1 & 2) and two payload blades (in logical slots 4 & 5). Node blade #4 is a "standard" blade, while #5 is a carrier blade with two AMCs (in sites 7 & 8). On SCxB #1, VLAN eth0.21 connects to the node blades and SCxB #2 while VLAN eth0.22 connects to SCxB #2. On node blade #4, eth0 connects to SCxB #1 and eth1 connects to SCxB #2. On Node blade #5 baseboard processor, eth0.21 connects to SCxB #1, eth0.35 connects to AMC #7 and AMC #8. On the AMCs, eth0.35 connects to the baseboard processor. Note that the site number in MIB indexing for non-AMCs is 255.

7.7.1 Proctor Configuration Examples

Configuration for a proctor involves creating L2 Interfaces, responder groups, expected responders, and typically, authorized proctors.

LHC Instance #1 is created with role proctor. (shelf 1 slot 1 site 255 instance 1)

```
lhccmd set lhcRowStatu      1.1.255.1 createAndWait
lhccmd set lhcRole         1.1.255.1 proctor
lhccmd set lhcRowStatus    1.1.255.1 active
```

Two L2 interfaces are created within the LHC instance #1, interface #1 for eth0.21 and #2 for eth0.22. (shelf 1 slot 1 site 255 instance 1 index 1) and (shelf 1 slot 1 site 255 instance 1 index 2).

```
lhccmd set lhcL2InterfaceRowStatus 1.1.255.1.1 createAndWait
lhccmd setc lhcL2InterfaceName      1.1.255.1.1 \"eth0.21\"
lhccmd set lhcL2InterfaceRowStatus 1.1.255.1.1 active
lhccmd set lhcL2IntefaceRowStatus  1.1.255.1.2 createAndWait
lhccmd setc lhcL2InterfaceName      1.1.255.1.2 \"eth0.22\"
lhccmd set lhcL2InterfaceRowStatus 1.1.255.1.2 active
```

Within interface #1, create a single responder group #1 with destination MAC address 01.c0.f9.00.00.01. This responder group fields RESPONSE messages from the payload blades and SCxB #2. (shelf 1 slot 1 site 255 instance 1 index 1 MAC 01c0f9000001)

```
lhccmd set lhcProctorResponderGroupRowStatus
          1.1.255.1.1.0x01.0xc0.0xf9.0x00.0x00.0x01
createAndGo
```

Within that Responder Group, create three expected responders to represent the two payload blades and SCxB #2. (shelf 1 slot 1 site 255 instance 1 index 1 MAC 01c0f9000001 ERshelf 1 ERslot 2 ERsite 255) and (shelf 1 slot 1 site 255 instance 1 index 1 MAC 01c0f9000001 ERshelf 1 ERslot 4 ERsite 255) and (shelf 1 slot 1 site 255 instance 1 index 1 MAC 01c0f9000001 ERshelf 1 ERslot 5 ERsite 255)

Link Health Check

```
lhccmd set   lhcProctorExpectedResponderRowStatus
             1.1.255.1.1.0x01.0xc0.0xf9.0x00.0x00.0x01.1.2.255
createAndGo
lhccmd set   lhcProctorExpectedResonderRowStatus
             1.1.255.1.1.0x01.0xc0.0xf9.0x00.0x00.0x01.1.4.255
createAndGo
lhccmd set   lhcProctorExpectedResponderRowStatus
             1.1.255.1.1.0x01.0xc0.0xf9.0x00.0x00.0x01.1.5.255
createAndGo
```

Within interface #2, create an authorized proctor #1 with destination MAC address 01.c0.f9.00.00.01. This AP fields QUERY messages from SCxB #2. (shelf 1 slot 1 site 255 instance 1 index 2 MAC 01c0f9000001 APshelf 1 APslot 2 APsite 255)

```
lhccmd set   lhcAuthorizedProctorRowStatus
             1.1.255.1.2.0x01.0xc0.0xf9.0x00.0x00.0x01.1.2.255
createAndWait
lhccmd set   lhcAuthorizedProctorGuardTimerEnable
             1.1.255.1.2.0x01.0xc0.0xf9.0x00.0x00.0x01.1.2.255
true
lhccmd set   lhcProctorExpectedResponderRowStatus
             1.1.255.1.2.0x01.0xc0.0xf9.0x00.0x00.0x01.1.2.255
active
```

7.7.2 Responder Configuration Examples

Configuration for a responder involves creating L2 Interfaces and authorized proctors.

LHC Instance #1 is created with role responder. (shelf 1 slot 4 site 255 instance 1)

```
lhccmd set   lhcRowStatus      1.4.255.1   createAndWait
lhccmd set   lhcRole           1.4.255.1   responder
lhccmd set   lhcRowStatus      1.4.255.1   active
```

Two L2 interfaces are created within LHC instance#1, interface#1 for eth0 and #2 for eth1. (shelf1 slot 4 site 255 instance 1 index 1) and (shelf 1 slot 4 site 255 instance 1 index 2)

```
lhccmd set   lhcL2InterfaceRowStatus 1.4.255.1.1 createAndWait
lhccmd setc  lhcL2InterfaceName      1.4.255.1.1 \"eth0\"
lhccmd set   lhcL2InterfaceName      1.4.255.1.1 active
lhccmd set   lhcL2InterfaceRowStatus 1.4.255.1.2 createAndWait
lhccmd setc  lhcL2InterfaceName      1.4.255.1.2 \"eth1\"
lhccmd set   lhcL2InterfaceRowStatus 1.4.255.1.2 active
```

Within interface #1, create a single authorized proctor #1 with destination MAC address 01.c0.f9.00.00.1. This AP fields QUERY messages from SCxB #1. (shelf 1 slot 4 site 255 instance 1 index 1 MAC 01c0f9000001 APshelf 1 APslot 1 APsite 255)

```
lhccmd set   lhcAuthorizedProctorRowStatus
            1.4.255.1.1.0x01.0xc0.0xf9.0x00.0x00.0x01.1.1.255
createAndWait
lhccmd set   lhcAuthorizedProctorGuardTimerEnable
            1.4.255.1.1.0x01.0xc0.0xf9.0x00.0x00.0x01.1.1.255 true
lhccmd set   lhcAuthorizedProctorRowStatus
            1.4.255.1.1.0x01.0xc0.0xf9.0x00.0x00.0x01.1.1.255 active
```

Within interface #2, create a single authorized proctor #1 with destination MAC address 01.c0.f9.00.00.01. This AP fields QUERY messages from SCxB#2. (shelf 1 slot 4 site 255 instance 1 index 2 MAC 01c0f9000001 AP shelf 1 APslot 2 AP site 255)

Within interface #2, create a single authorized proctor #1 with destination MAC address 01.c0.f9.00.00.01. This AP fields QUERY messages from SCxB #2. (shelf 1 slot 4 site 255 instance 1 index 2 MAC 01c0f9000001 APshelf 1 APslot 2 APsite 255)

```
lhccmd set   lhcAuthorizedProctorRowStatus
            1.4.255.1.2.0x01.0xc0.0xf9.0x00.0x00.0x01.1.1.2.255
createAndWait
lhccmd set   lhcAuthorizedProctorGuardTimerEnable
            1.4.255.1.2.0x01.0xc0.0xf9.0x00.0x00.0x01.1.2.255 true
lhccmd set   lhcAuthorizedProctorRowStatus
            1.4.255.1.1.0x01.0xc0.0xf9.0x00.0x00.0x01.1.2.255 active
```

7.7.3 Baseboard Processor on Carrier Blade Configuration Example

Configuration on a carrier blade baseboard processor involves creating two LHC instances, one with role responder to handle the LHC conversation to the SCxBs and one with role proctor to handle the LHC conversation to the AMCs.

LHC Instance #1 is created with role proctor (shelf 1 slot 5 site 255 instance 1).

```
lhccmd set   lhcRowStatus 1.5.255.1. proctor
lhccmd set   lhcRole 1.5.255.1 proctor
lhccmd set   lhcRowStatus 1.5.255.1 active
```

One L2 Interface is created within LHC instance #1, interface #1 for eth0.35 (shelf 1 slot 5 site 255 instance 1 index 1).

```
lhccmd set   lhcL2InterfaceRowStatus 1.5.255.1.1 createAndWait
lhccmd setc  lhcL2InterfaceName1.5.255.1.1 \"eth0.35\"
```

Link Health Check

```
lhccmd set lhcL2InterfaceRowStatus 1.5.255.1.1 active
```

Within interface #1, create a single responder Group #1 with destination MAC address 01.c0.f9.00.00.01. This Responder Group fields RESPONSE messages from the AMCs. (shelf 1 slot 5 site 255 instance 1 index 1 MAC 01c0f9000001)

```
lhccmd set lhcProctorResponderGroupRowStatus
1.5.255.1.10x01.0xc0.0xf9.0x00.0x00.0x01 createAndGo
```

Within that responder group, create two expected responders to represent the two AMCs. (shelf 1 slot 5 site 255 instance 1 index 1 MAC 01c0f9000001 ERshelf 1 ERslot 5 ERsite 7) and (shelf 1 slot 5 site 255 instance 1 index 1 MAC 01c0f9000001 ERshelf 1 ERslot 5 ERsite 8)

```
lhccmd set lhcProtorExpectedResponderRowStatus
1.5.255.1.1.0x01.0xc0.0xf9.0x00.0x00.0x01.1.1.5.7
createAndGo
lhccmd set lhcProctorExpectedResponderRowStatus
1.5.255.1.1.0x01.0xc0.0xf9.0x00.0x00.0x01.1.1.5.8
createAndGo
```

LHC Instance #2 is created with role responder. (shelf 1 slot 5 site 255 instance 2)

```
lhccmd set lhcRowStatus 1.5.255.2 createAndWait
lhccmd set lhcRole 1.5.255.2 responder
lhccmd set lhcRowStatus 1.5.255.2 active
```

One L2 Interface is created within LHC instance #2, interface #1 for eth0.21. (shelf 1 slot 5 site 255 instance 2 index 1)

```
lhccmd set lhcL2InterfaceRowStatus 1.5.255.2.1 createAndWait
lhccmd setc lhcL2InterfaceName 1.5.255.2.1 \"eth0.21\"
lhccmd set lhcL2InterfaceRowStatus 1.5.255.2.1 active
```

Within interface #1, create an authorized proctor #1 with destination MAC address 01.c0.f9.00.00.01. This AP fields QUERY messages from SCxB #1 (shelf 1 slot 5 site 255 instance 2 index 1 APshelf 1 APslot 1 APsite 255).

```
lhccmd set lhcAuthorizedProctorRowStatus
1.5.255.2.1.0xc0.0xf9.0x00.0x00.0x01.1.1.255
createAndWait
lhccmd set lhcAuthorizedProctorGuardTimerEnable
1.5.255.2.1.0xc0.0xf9.0x00.0x00.0x01.1.1.255 true
lhccmd set lhcAuthorizedProctorRowStatus
1.5.255.2.1.0xc0.0xf9.0x00.0x00.0x01.1.1.255 active
```


7.7.4 AMC/PMC Module on Carrier Blade Configuration Example

The configuration on an AMC/PMC is similar to a standard payload blade except that there is only a single L2 interface for the base. The two AMC/PMC modules have duplicate configurations except for the site number, so only AMC/PMC #7 is shown here.

LHC Instance #1 is created with the role responder. (shelf 1 slot 5 site 7 instance 1)

```
lhccmd set  lhcTRowStatus      1.5.7.1      createAndWait
lhccmd set  lhcRole            1.5.7.1      responder
lhccmd set  lhcRowStatus      1.5.7.1      active
```

One L2 Interface is created within LHC instance #1, interface #1 for eth0.35. (shelf 1 slot 5 site 7 instance 1 index 1)

```
lhccmd set  lhcL2InterfaceRowStatus  1.5.7.1.1      createAndWait
lhccmd setc lhcL2InterfaceName      1.5.7.1.1      \"eth0.35\"
lhccmd set  lhcL2InterfaceRowStatus  1.5.7.1.1      active
```

Within interface #1, create a single authorized proctor #1 with destination MAC address 01.c0.f9.00.00.01. This AP fields QUERY messages from the baseboard processor. (shelf 1 slot 5 site 7 instance 1 index 1 MAC 01c0f9000001 APshelf 1 APslot 5 APsite 255)

```
lhccmd set  lhcAuthorizedProctorRowStatus
              1.5.7.1.1.0xc0.0xf9.0x00.0x00.0x01.1.5.255
createAndWait
lhccmd set  lhcAuthorizedProctorGuardTimerEnable
              1.5.7.1.1.0xc0.0xf9.0x00.0x00.0x01.1.5.255      true
lhccmd set  lhcAuthorizedProctorRowStatus
              1.5.7.1.1.0xc0.0xf9.0x00.0x00.0x01.1.5.255      active
```

7.7.5 Leaky Bucket Description and Configuration Example

LHC uses a 'leaky bucket' mechanism (described below) to give the user control over the frequency of certain FM notifications. The FM notifications that have this control are:

- RESPONSE Delinquent
- RESPONSE Received
- QUERY Delinquent
- QUERY Received

There are two independent leaky buckets, one that controls the RESPONSE pair of FM notifications and one that controls the QUERY pair of FM notifications.

Link Health Check

There are two writeable MIB objects to control each leaky bucket: upper threshold (UT) and lower threshold (LT). The UT and LT are controlled on a per-responder group and per-authorized proctor basis.

The following description applies to both the RESPONSE leaky bucket and the QUERY leaky bucket, but is described from the RESPONSE leaky bucket perspective.

When a proctor sends a QUERY message, a response timer is started. Each time that the response timer expires, if a RESPONSE message has not been received from an expected responder (ER), then a drop is added to the leaky bucket for that ER. If a RESPONSE message is received from an ER, then a drop is removed from the leaky bucket for that ER. The bucket size does not go below zero and does not get larger than the UT. LHC uses an internal variable to track whether the most recent FM notification was "Delinquent" or "Received". If upon adding a drop to the leaky bucket it equals UT and the most recent notification was "Received", then a "Delinquent" notification is generated. If upon removing a drop from the leaky bucket it equals LT and the most recent notification was "Delinquent" then a "Received" notification is generated.

By changing the UT and LT values, one can control the frequency of FM notifications such that not every dropped RESPONSE or QUERY is reported and that the reception of a single RESPONSE or QUERY is reported.

7.8 LHC Configuration for ATCA-F140

The following tables are excerpts from the ATCA-F140 LHC configuration, showing the rows for the Authorized Proctors, Responder Groups, and Expected Responders.

On the ATCA-F140, there is one LHC instance configured as a proctor.

In the following tables, the first five index elements in each row are shelf.slot.site.instance.interface and the next six elements are the multicast address. For this example, the ATCA-F140 is assumed to be in shelf 2 and logical slot 1.

Table 7-1 ATCA-F140 Base Interface LHC Configuration

Row Index	Row Type	Represents
2.1.255.1.2.1.192.249.0.0.1.2.2.255	Authorized Proctor	Queries from ATCA-F140 in logical slot 2
2.1.255.1.1.1.192.249.0.0.1	Responder Group	Queries to other ATCA-F140 and payload blades
2.1.255.1.1.1.192.249.0.0.1.2.2.255	Expected Responder	Response from logical slot 2 ATCA-F140
2.1.255.1.1.1.192.249.0.0.1.2.3.255	Expected Responder	Response from logical slot 3 payload blade

Table 7-1 ATCA-F140 Base Interface LHC Configuration (continued)

Row Index	Row Type	Represents
2.1.255.1.1.1.192.249.0.0.1.2.4.255	Expected Responder	Response from logical slot 4 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.5.255	Expected Responder	Response from logical slot 5 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.6.255	Expected Responder	Response from logical slot 6 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.7.255	Expected Responder	Response from logical slot 7 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.8.255	Expected Responder	Response from logical slot 8 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.9.255	Expected Responder	Response from logical slot 9 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.10.255	Expected Responder	Response from logical slot 10 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.11.255	Expected Responder	Response from logical slot 11 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.12.255	Expected Responder	Response from logical slot 12 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.13.255	Expected Responder	Response from logical slot 13 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.14.255	Expected Responder	Response from logical slot 14 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.15.255	Expected Responder	Response from logical slot 15 payload blade
2.1.255.1.1.1.192.249.0.0.1.2.16.255	Expected Responder	Response from logical slot 16 payload blade
2.1.255.1.3.1.192.249.0.0.1	Responder Group	Queries to shelf managers
2.1.255.1.3.1.192.249.0.0.1.2.101.255	Expected Responder	Response from logical slot 101 shelf manager
2.1.255.1.3.1.192.249.0.0.1.2.102.255	Expected Responder	Response from logical slot 102 shelf manager

Link Health Check

Table 7-2 ATCA-F140 Fabric Interface LHC Configuration

Row Index	Row Type	Represents
2.1.255.1.5.1.192.249.0.0.1.2.2.255	Authorized Proctor	Queries from ATCA-F140 in logical slot 2
2.1.255.1.4.1.192.249.0.0.1	Responder Group	Queries to other ATCA-F140 and payload blades
2.1.255.1.4.1.192.249.0.0.1.2.2.255	Expected Responder	Response from logical slot 2 ATCA-F140
2.1.255.1.4.1.192.249.0.0.1.2.3.255	Expected Responder	Response from logical slot 3 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.4.255	Expected Responder	Response from logical slot 4 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.5.255	Expected Responder	Response from logical slot 5 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.6.255	Expected Responder	Response from logical slot 6 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.7.255	Expected Responder	Response from logical slot 7 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.8.255	Expected Responder	Response from logical slot 8 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.9.255	Expected Responder	Response from logical slot 9 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.10.255	Expected Responder	Response from logical slot 10 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.11.255	Expected Responder	Response from logical slot 11 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.12.255	Expected Responder	Response from logical slot 12 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.13.255	Expected Responder	Response from logical slot 13 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.14.255	Expected Responder	Response from logical slot 14 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.15.255	Expected Responder	Response from logical slot 15 payload blade
2.1.255.1.4.1.192.249.0.0.1.2.16.255	Expected Responder	Response from logical slot 16 payload blade

Watchdog Behavior

This section describes the types of watchdogs that are available and configurable with ATCA-F140 and the way how they behave when used.

ATCA-F140 provides two programmable watchdogs: the IPMC (or BMC) watchdog and the FPGA watchdog. The system response varies based on the type of Watchdog triggered.

8.1 BMC Watchdog

The Baseboard Management Controller (BMC) watchdog protects against boot failures when transferring control from the IPMC to the system boot loader (U-Boot). Before transferring control to U-Boot, the IPMC starts the BMC watchdog with a timeout value of 20 seconds by default.

U-Boot uses the BMC watchdog timer when transferring control from the system boot loader to Linux to protect against boot hangs, until the system starts the Field Programmable Gate Array (FPGA) watchdog. By default, U-Boot sets the BMC watchdog with a value of 600 seconds. This is controllable via the U-Boot environment variable `oswatchdog`.

When the BMC watchdog timer triggers, the IPMC resets the blade. This is reflected by the SEL event log and is also reported by U-Boot as "IPMC" for the "Reset Cause" condition.

In addition, when the BMC watchdog triggers, the default behavior of the IPMC is to initiate a boot bank switch, if the Timer Use field is set to "BIOS FRB2" (001b), "BIOS/POST" (010b), or "OS Load" (011b). Both the IPMC and system boot loader set the Timer Use field such that a boot bank switch occur that makes the BMC watchdog to trigger. Refer [Boot Bank Switch on BMC Watchdog Timeout on page 126](#) for more information on bank switch behavior and how it can be controlled.

Refer the *ATCA-F140 Control via IPMI Programmer's Reference manual*, for further examples related to BMC watchdog timer commands.

Alternatively, see [Section 6.4 on page 75](#), for how to use the watchdog sub-command to Get and Set the BMC watchdog timer once Linux is booted.

NOTE: The `hpmcmd` does not set the Timer Use field to "BIOS FRB2" (001b), "BIOS/POST" (010b), or "OS Load" (011b) when managing the BMC watchdog, so no boot bank switch will occur if the BMC watchdog triggers after being set and started by `hpmcmd`.

8.2 FPGA Watchdog

The FPGA provides a software watchdog, which protects against system hangs after Linux has booted. This watchdog does not cause the IPMC to initiate a boot bank switch as it does not interact with the IPMC at all, but rather asserts the reset signal to the P2020 processor.

The FPGA watchdog is a two-stage watchdog that generates an interrupt to the P2020 processor when the first stage time outs (which is set to one-half the value of the timeout value). When the first stage time outs, an interrupt is sent to the P2020 processor which is registered in the P2020 system log (`/var/log/messages`) to indicate the system is in trouble. When the second stage time outs, a hard reset is sent to the P2020 processor to initiate a reboot. This is a hard reset to the processor which generates no log entries; however, this can be detected by U-Boot and is reported as "SwWatchdog" for the "Reset Cause" condition.

After the FPGA watchdog timer is started, the BMC watchdog timer is stopped. The BMC watchdog timer is then available for use by the user.

The default value of the FPGA watchdog is 60 seconds. A user can set a different value for the FPGA watchdog timeout by modifying the kernel command line via the U-Boot environment variable "bootargs". This can be done by appending the following string to the "bootargs" environment variable. The following example shows how to change the timeout value to 30 seconds.

```
atca_f140_wdt.timeout=30
```

Acceptable values for the FPGA watchdog are between 1 and 60 (inclusive).

NOTE: Lowering the timeout value of the FPGA watchdog (from its default value of 60) may require configuring the system watchdog daemon to reset the FPGA watchdog more frequently to prevent unwanted resets or interrupts (should the system be loaded). The default value of the reset interval is 10 seconds (see `/etc/watchdog.conf`).

8.3 Boot Bank Switch on BMC Watchdog Timeout

Under normal conditions, when the BMC Watchdog time outs, the IPMC will force a boot bank switch before resetting the processor and repeating the boot cycle. This protects against boot hangs when U-Boot or Linux fail to operate or otherwise fail to boot the system properly. This is not the case for the FPGA watchdog (that means no boot bank switch occurs when the FPGA watchdog triggers).

The default behavior is to switch boot banks when the BMC watchdog triggers [when the *Timer Use* field is set to "BIOS FRB2" (001b), "BIOS/POST" (010b), or "OS Load" (011b)], but this behavior can be controlled by the user through the use of the "Payload Upgrade Mode" and "Boot Bank Swap Control" flags.

For more information on how to Get and Set these flags and more detailed explanation about them, refer to the *ATCA-F140 Control via IPMI Programmer's Reference* manual.

The following table details the circumstances when a boot bank switch occurs, when any of the watchdogs is triggered.

Table 8-1 Boot Bank Switch on BMC Watchdog Timeout

Execution Phase		Governing Watchdog		Flags		Boot Bank Switch on Watchdog Timeout?
Current	Next	Source	Default Value	Boot Bank Swap	Upgrade Mode	
IPMC	U-Boot	IPMC	20 seconds	Enabled	No	Yes
				Enabled	Yes	Yes
				Disabled	No	No
				Disabled	Yes	Yes
U-Boot	Linux	IPMC	600 seconds	Enabled	No	Yes
				Enabled	Yes	Yes
				Disabled	No	No
				Disabled	Yes	Yes
Linux	n/a	FPGA	60 seconds	Don't Care	Don't Care	No

Trident Reconfiguration Utility

Trident Reconfiguration Utility allows for the easy reconfiguration of the ports connected to the Broadcom Trident 40G - fabric switch. This utility is called `trid-cfg` and is installed in the `/opt/bladervices/bin` directory. Use this utility only if you want to deviate from the two default configurations provided in [Table 9-1](#) and [Table 9-2](#). The default configurations are based on the two chassis types AXP1440, AXP-640, and AXP640-C04 in which the ATCA-F140 is currently supported.

Table 9-1 Default Fabric Configuration-1440 Chassis

Fabric Channel	Physical Slot	Configuration Type	Port Type
FC1		4x1G	
FC2	7	40G -KR4	Backplane
FC3	8	40G -KR4	Backplane
FC4	5	40G -KR4	Backplane
FC5	10	40G -KR4	Backplane
FC6	4	40G-KR4	Backplane
FC7	11	40G-KR4	Backplane
FC8	3	40G -KR4	Backplane
FC9	12	40G -KR4	Backplane
FC10	2	40G -KR4	Backplane
FC11	13	40G -KR4	Backplane
FC12	1	40G-KR4	Backplane
FC13	14	40G -KR4	Backplane
RTM1		40G -KR4	RTM QSFP RTM ETH7
RTM2		10G-4KR	RTM SFP+ ETH6, 5, 4, 3
FP1		40G-KR4	FP QSFP ETH1
FP2		40G -KR4	FP QSFP ETH2

Trident Reconfiguration Utility

Table 9-2 Default Fabric Configuration-640 Chassis

Fabric Channel	Physical Slot	Configuration Type	Port Type
FC1		4x1G	
FC2-Xlink		10G-KX4	Backplane
FC10	2	40G -KR4	Backplane
FC11	5	40G -KR4	Backplane
FC12	1	40G -KR4	Backplane
FC13	6	40G -KR4	Backplane
RTM 1		40G -KR4	RTM QSFP RTM ETH7
RTM 2		10G-4KR	RTM SFP+ ETH6, 5, 4, 3
FP 1		40G -KR4	FP QSFP ETH1
FP 2		40G -KR4	FP QSFP ETH2

NOTICE

The Broadcom Trident switch chip (fabric) can be configured for a total of 64 ports. The HW is divided into two parts, called the X and Y pipelines. The X pipeline encompasses WarpCore 0 - WarpCore 8. The Y pipeline encompasses WarpCore 9 - WarpCore 17. Besides the overall 64 port limitation, no pipeline (X or Y) can be configured for more than 32 ports. The “trid-cfg” utility has been developed to validate this HW design limitation.

The `trid-cfg` utility will run on the ATCA-F140 after logging in as the super user (root).

Command syntax:

```
trid-cfg [-i <inp_file>] [-r backup_ver] [-calyhv]
```

Where:

-h Help

-i Specify input file (full path)

default = `/opt/bladeservices/cfg/trid_config`

-c Display the current configuration

- a Activate the current configuration
- l List the available backup versions
- r Restore the specified backup version
- y Suppress the confirmation step
- v Verbose mode

Examples:

To display currently activated configuration:

```
trid-cfg -c
```

To validate and display a new configuration:

```
trid-cfg -i /opt/bladeservices/cfg/trid_config
```

To activate and install a new configuration (creates a new backup version):

```
trid-cfg -i /opt/bladeservices/cfg/trid_config -a
```

To list all available backup versions:

```
trid-cfg -l
```

To restore a specific backup version:

```
trid-cfg -r 20110712143010
```

NOTICE

It is required to execute `trid-cfg` with default configuration file, if the blade is moved from one chassis to another chassis or from one slot to another slot.

9.1 Input Files

Four input files are included in the `/opt/bladeservices/cfg` directory that match with the four default configurations:

"trid_1440_def":

```
#  
# Input file for Trident configuration on F140  
#  
#Chassis Type values = 1440, 640, 14400, 640c04  
  
CTYPE=1440  
# Fabric channel identifiers: FC2 - FC13
```

Trident Reconfiguration Utility

```
# Descriptors:
#   G - grouped mode, covering 40GBase-KR4, 10GBase-KX4, and
10GBase-BX4.
#   I - independent mode, covering 4x10GBase-KR, 4x1000Base-KX,
4x1000Base-BX,
#       2x10GBase-KR, 2x1000Base-KX, 2x1000Base-BX, 1x10GBase-KR,
1000Base-KX,
#       and 1000Base-BX.
#   N - not connected
FC2=G
FC3=G
FC4=G
FC5=G
FC6=G
FC7=G
FC8=G
FC9=G
FC10=G
FC11=G
FC12=G
FC13=G
# Fabric QSFP identifiers: QSFP1 - QSFP3
# QSFP1 is FP ETH1, QSFP2 is FP ETH2, QSFP3 is RTM ETH7
# Descriptors: G, I, N
QSFP1=G
QSFP2=G
QSFP3=G

"trid_1440_def":
#
# Input file for Trident configuration on F140
#
# Chassis Type values = 1440, 14400, 640
CTYPE=14400
# Fabric channel identifiers: FC2 - FC13
# Descriptors:
```

```
# G - grouped mode, covering 40GBase-KR4, 10GBase-KX4, and
10GBase-BX4.
# I - independent mode, covering 4x10GBase-KR, 4x1000Base-KX,
4x1000Base-BX,
#      2x10GBase-KR, 2x1000Base-KX, 2x1000Base-BX, 1x10GBase-KR,
1000Base-KX,
#      and 1000Base-BX.
# N - not connected
FC2=I
FC3=I
FC4=G
FC5=G
FC6=G
FC7=G
FC8=G
FC9=G
FC10=G
FC11=G
FC12=G
FC13=G
# Fabric QSFP identifiers: QSFP1 - QSFP3
# QSFP1 is FP ETH1, QSFP2 is FP ETH2, QSFP3 is RTM ETH7
# Descriptors: G, I, N
QSFP1=G
QSFP2=G
QSFP3=G

"trid_640_def":
#
# Input file for Trident configuration on F140
#
# Chassis Type values = 1440, 640, 14400, 640c04
CTYPE=640
# Fabric channel identifiers: FC2 - FC13
# Descriptors:
# G - grouped mode, covering 40GBase-KR4, 10GBase-KX4, and
10GBase-BX4.
```

Trident Reconfiguration Utility

```
# I - independent mode, covering 4x10GBase-KR, 4x1000Base-KX,
4x1000Base-BX,
#       2x10GBase-KR, 2x1000Base-KX, 2x1000Base-BX, 1x10GBase-KR,
1000Base-KX,
#       and 1000Base-BX.
# N - not connected
FC2=G
FC3=N
FC4=N
FC5=N
FC6=N
FC7=N
FC8=N
FC9=N
FC10=G
FC11=G
FC12=G
FC13=G
# Fabric QSFP identifiers: QSFP1 - QSFP3
# QSFP1 is FP ETH1, QSFP2 is FP ETH2, QSFP3 is RTM ETH7
# Descriptors: G, I, N
QSFP1=G
QSFP2=G
QSFP3=G
```

"trid_640c04_def":

```
#
# Input file for Trident configuration on F140
#
# Chassis Type values = 1440, 14400, 640, 640c04
CTYPE=640c04
# Fabric channel identifiers: FC2 - FC13
# Descriptors:
# G - grouped mode, covering 40GBase-KR4, 10GBase-KX4, and
10GBase-BX4.
```

```
# I - independent mode, covering 4x10GBase-KR, 4x1000Base-KX,  
4x1000Base-BX,  
#      2x10GBase-KR, 2x1000Base-KX, 2x1000Base-BX, 1x10GBase-KR,  
1000Base-KX,  
#      and 1000Base-BX.  
# N - not connected  
FC2=G  
FC3=G  
FC4=G  
FC5=G  
FC6=N  
FC7=N  
FC8=N  
FC9=N  
FC10=N  
FC11=N  
FC12=N  
FC13=N  
# Fabric QSFP identifiers: QSFP1 - QSFP3  
# QSFP1 is FP ETH1, QSFP2 is FP ETH2, QSFP3 is RTM ETH7  
# Descriptors: G, I, N  
QSFP1=G  
QSFP2=G  
QSFP3=G
```

9.2 Customize Trident Configuration

Based on the topology or on the system requirement, you can modify the modes of the ports either from group mode to independent mode or from independent mode to group mode.

To reconfigure the ports, perform the following steps:

Trident Reconfiguration Utility

1. Go to `/opt/bladeservices/cfg` directory,
`cd /opt/bladeservices/cfg`



The default trident configurations of various chassis' are available at `/opt/bladeservices/cfg/trid_<chassis>_def`

2. Create a copy of any of the default configuration files.
If in a 1440 chassis, copy the configuration using the following command:
`cp trid_1440_def trid_config`
If in a 14400 chassis, copy the configuration using the following command:
`cp trid_14400_def trid_config`
If in a 640 chassis, copy the configuration using the following command:
`cp trid_640_def trid_config`
If in a 640-C04 chassis, copy the configuration using the following command:
`cp trid_640c04_def trid_config`
3. Modify the `trid_config` file with new port/slot settings. You can modify the port modes from independent to group or from group to independent.
4. Validate the new configuration.
`trid-cfg`
5. Activate the new configuration, if no errors are displayed.
`trid-cfg -a`
6. Reboot the F140.

For example, if you want to change the mode of FC7 (Logical slot 8) from group mode to independent mode.

1. Create a copy of `trid_1440_def` configuration as `trid_config`
2. Change "`FC7=G`" to "`FC7=I`".
3. Execute `trid-cfg` as shown below to create configuration files. The `config.bcm` and `rc.soc` files are created, which are used for next reboot.
`trid-cfg -a -i /opt/bladeservices/cfg/trid_config`
4. After reboot, update the SRstackware port configuration of the ports, which are impacted due to mode changes. That means, in group mode, FC7 is represented as **1X40G port** (xe13) and in independent mode it is represented as **4X10G ports** (xe13, xe14, xe15, and xe16). In this case, xe14, xe15, and xe16 do not have any configuration and you have to manually configure all the ports as per topology requirement.

9.2.1 Backups

During the activate step, a backup is created for all the files being modified. These versions are stamped with date and time using the format = YYYYMMDDHHMMSS. The modified files are:

```
active_cfg
config.bcm
hpm_arrays
rc.f140.fabric2
rc.soc
```

9.2.1.1 Restoring a backup

1. List all backup versions.
`trid-cfg -l`
2. Restore the backup version.
`trid-cfg -r <backup-version>`
3. Reboot the F140.

9.3 SRS Initialization

When an ATCA-F140 is rebooted, the SRS initialization script performs the following logic with regard to the Trident fabric switch configuration:

```
Check for the "active_cfg" file in the '/opt/bladeservices/cfg' directory?
```

```
If present then initialize with this configuration.
```

```
If not present {
```

```
Check for the "trid_config" file in the /opt/bladeservices/cfg' directory?
```

```
If present {
```

```
Execute the "trid-cfg" utility with the "-a" and "-y" options.
```

```
}
```

```
If not present {
```

```
Determine the chassis type (1440 [default] or 640).
```

```
Install default files based on the chassis type.
```

```
}
```

```
}
```


Fabric Configuration

10.1 Fabric Interface Configuration Service

Some node blades require special configuration settings applied to the fabric interface, especially when operating at higher speeds (10G-BaseKR and 40G-BaseKR4).

In order to avoid performing these configuration steps manual, a service/tool "fabconfig" is provided. This tool determines the blade type in a specific slot and configures the F140 fabric interface as required for the blade.

10.1.1 Usage

```
fabconfig -c <fabric-channel>
```

Configures blade connected to a specific fabric channel.

```
fabconfig -s <logical slot>
```

Configures fabric interface for a blade in a specific logical slot.

```
fabconfig -i <srs-interface>
```

Configures fabric interface as named by SRStackware.

In addition "fabconfig" is called whenever an IPMI e-keying event is bringing a fabric port "up". This happens when a blade is inserted or restarted, so under normal circumstances it is not required to call "fabconfig" manually.

Create the following file to disable the background operating mode (no special content is required):

```
/opt/bladeservices/etc/fabconfig/ekey_disable
```

10.1.2 Logging

The fabconfig tool will log its actions via syslog (/var/log/messages file) using tag "fabconfig", and a severity of "local0.err" or "local0.info" for errors or information messages respectively.

Sample output when a fabric channel is brought up:

```
fabconfig[3030]: Configuring port 4 (Channel 6) for blade ATCA-9405[unknown] to [kr4] (Grouped mode), options [kr], tap [DriverCurrent=0x04 PreDriverCurrent=0x04 Preemphasis=0xCEC0]
```

10.2 Configuration Files

End users, blade or chassis developers/integrators have the possibility to configure and extend the fabconfig tool by modifying or adding scripts located in `/opt/bladeservices/etc/fabconfig`.

10.2.1 slotconfig

The slotconfig file (`/opt/bladeservices/etc/fabconfig/slotconfig`) can be edited by the user to select specific options for specific blades.

It contains a table which is searched from top to bottom, trying to match a specific chassis, blade name, blade revision and fabric channel. When a match is found, the specified options are applied to the corresponding fabric port(s) on the ATCA-F140.

The default entries are:

#	Chassis	Blade	Rev.	Channel	Options
#	-----	-----	-----	-----	-----
4440		ATCA-9405	all	all	kr
4440		ATCA-7470	all	all	auto

The first four columns are:

1. Chassis type.
Currently supported are "4440" for Penguin Edge™ AXP-1440/4440 and "640" for AXP 640. Additional chassis names can be provided via the "fabconfig.rc" file.
2. Blade name.
This name is retrieved via HPM/IPMI from the FRU data of the node blade. FRU contents can be mapped to blade names in the "fabconfig.rc" file.
3. Blade revision.
This information is retrieved via HPM/IPMI from the FRU data of a node blade.
4. Channel.
The fabric channel number (which is the logical slot number of a blade minus 1).

The keyword "all" acts as wildcard matching everything.

All keywords beginning with the 5th column are treated as configuration options. The following options are currently defined (additional options might be implemented in future versions of the tool, or implemented by blade specific configuration files):

""auto"

Enables auto-negotiation on the port

""kr"

Disables auto-negotiation and configures the fabric interface to

- 10GBase-KR mode if the channel is configured to "Independent" mode by trid-cfg, or
- 40GBase-KR4 mode if the channel is configured to "Grouped" mode.

""kx"

Disables auto-negotiation and configures the fabric interface to

- 1000BaseX mode if the channel is configured to "Independent" mode by trid-cfg, or
- 10GBase-KX4 (XGMII) mode if the channel is configured to "Grouped" mode.

""fec"

Enables forward error correction "FEC" for KR/KR4 mode. By default it is disabled if auto-negotiation has been disabled.

""ignore"

Ignores the blade, no changes to the current port setup are done.

""no_txswap"

This option is specific to the ATCA-9405, which requires swapping the SerDes polarity for some tx lines. It disables the swapping and leaves the SerDes lanes in default mode.

If no entry is found for a node blade the port is set to "auto" mode with no further options.

Example for configuring:

- KR/KR4 mode and FEC for a ATCA-9405 blade connected via fabric channel 5 (logical slot 6).
- KR/KR4 mode without FEC for all other ATCA-9405 blades.
- auto-negotiation for all ATCA-7470 blades.
- No modifications applied when an unknown blade is present/inserted.

# Chassis	Blade	Rev.	Channel	Options
4440	ATCA-9405	all	5	kr fec
4440	ATCA-9405	all	all	kr
4440	ATCA-7470	all	all	auto
all	all	all	all	ignore

10.2.2 fabconfig.rc

This configuration file can be edited to add support for additional blades and chassis types.

It contains two shell functions:

1. `chassis_names`

This function prints a table with two columns:

- A regular expression string which is compared against the chassis name as reported by the "hpmcmd -c shelftype" command, with all spaces removed. A simple substring match is performed for values without special regular expression characters.
- The identifier used throughout the tool for this chassis (for example, Column 1 in the "slotconfig" table).

2. `blade_names`

This function maps a blade name as it is stored in the blade's FRU data to an identifier used throughout the tool.

It prints a table with two columns:

- A regular expression string which is compared against the blade's FRU content (product name in the FRU data's product info area as reported by "hpmcmd -c fruinfoget 0 pproductname"). A simple substring match is performed for values without special regular expression characters.
- The identifier used throughout the tool for this blade (for example, Column 2 in the "slotconfig" table).

Examples:

Example to add a shelf "myshef" if hpmcmd reports a string containing "foo-bar":

```
chassis_names()  
{  
    echo AXP-1440 4440  
    echo AXP-4440 4440  
    echo foo-bar myshef  
}
```

"myshef" can now be used as chassis identifier in the slotmap file.

Example to map a blade to the identifier "ATCA-0815" if the fru-data contains the sub-string "ATCA-0815" or "Somevendor-ATCA-0815-Rev1.99".

The default entries map FRU names containing the string "ATCA-9405" and "ATCA-7470" to the same blade identifier:

```
blade_names()
{
    echo ATCA-9405 ATCA-9405
    echo ATCA-7470 ATCA-7470
    echo ATCA-0815 ATCA-0815
    echo Somevendor-ATCA-0815-Rev1.99 ATCA-0815
}
```

The identifier ATCA-0815 can now be used as blade identifier in the slotmap file.

10.2.3 Blade Specific Configuration File

By default, the "fabconfig" tool supports two Penguin Edge node blades (ATCA-9405 and ATCA-7470). If required, support for additional blades can be provided by adding an additional blade configuration file named

```
/opt/bladeservices/etc/fabconfig/fabconfig.<blade-name> [.<blade-  
revision>]
```

<blade-name> is the name resolved via the `blade_names()` method in `fabconfig.rc` (see previous section). <blade-rev> is the blade revision as stored in the blades FRU data (Product version field in the Product info area, "hpmcmd -c fruinfoget 0 pversion").

The tool first searches for a file with the revision suffix. If none is found, a file without revision index is searched.

As final fallback, the file "fabconfig.unknown-blade" is used.

The blade specific configuration file must provide two shell functions named "board_get_txtap" and "board_get_workaround". Refer to the "fabconfig.unknown-blade" file for more details (this file can be used as template for new blades).

10.3 Supported Blades

10.3.1 ATCA-9405

The ATCA-9405 packet processing blade has the following restrictions which are covered by the fabconfig tool:

- Auto-negotiation is not supported on the fabric interface.
- "KR/KR4 mode requires slot specific configuration settings for the SerDes transmitters (pre/main/post tap values).
- "The transmit polarity needs to be swapped for specific SerDes lanes (can be disabled with the "no_txswap" option).

10.3.2 ATCA-7470

The ATCA-7470 blade requires the following settings which are covered by the fabconfig tool:

- The Driver/PreDriver current of the SerDes transmitters needs to be adjusted.

SRstackware

The Switching and Routing stackware (SRstackware®) is a protocol suite bundled along with BBS as an installable package on the ATCA blade systems. SRstackware Release 2.1 provides features to perform ATCA-F140 switch configuration and checks the statistics in addition to the protocol stack support. SRstackware Release 2.1 supports the following features and protocols:

- IEEE 802.1p and IEEE 802.1q VLAN and bridge configuration, IEEE 802.1d - Spanning Tree Protocol (STP)
- IEEE 802.1w - Rapid Spanning Tree Protocol (RSTP)
- IEEE 802.1s - Multiple Spanning Tree Protocol (MSTP)
- IEEE 802.3ad Link Aggregation Control Protocol (LACP)
- Class of Service (CoS) and Quality of Service (QoS)
- IGMP Snooping, IGMPv3 Protocol (IGMP)
- Open Shortest Path First version 2 (OSPFv2)
- Static routing
- IEEE 802.1q GVRP, GMRP, GARP protocols
- Q-in-Q or VLAN Stacking
- Virtual Router Redundancy Protocol (VRRP)
- Router Information Protocol (RIPv2)
- Bidirectional Forwarding Detection (BFD)
- Optical RTM
- Simple Network Management Protocol (SNMP)
- Linux OS version Wind River PNE 3.0 support with Broadcom BCM SDK 5.10.2
- ATCA-F140 switching blade

11.1 Configuring SRstackware

SRstackware, after installation, comes up with the default configuration. You can update the configuration using CLI. Refer [Appendix C, Related Documentation](#), for the list of CLI documents.

You can reconfigure the ports connected to the Broadcom Trident 40G - fabric switch using the Trident Reconfiguration Utility. For more information, see [Chapter 9, Trident Reconfiguration Utility on page 129](#).



Use the Trident Reconfiguration Utility (`trid-cfg`), only if you want to deviate from the default fabric configurations on the 1440, 14-100, 14-400, 640, and 640-C04 chassis.

11.2 Default Configuration

The SRstackware default configuration creates two bridges, one for the base network and another for the fabric network. The switch ports are named as `ge#` for the 1G switch ports and `xe#` for the 10G/40G switch ports. These switch ports are accessible using the SRstackware CLI.

The base channel and fabric channel are configured as two bridges, with the corresponding VLAN ports associated to each bridge. You can modify the configuration using the SRstackware CLI and store it as default configuration. The default configuration file is saved at `/etc/opt/srstackware/config/srs.<slot#>.cfg`. During first boot based on the chassis type, slot number, `active_cfg` file, SRstackware `init` scripts will pick the default configuration file.

`/etc/opt/srstackware/config/srs.<chassis>.<slot#>.cfg` and copy in place of `/etc/opt/srstackware/config/srs.<slot#>.cfg`.

Chassis types:

14400: 14 slots chassis with 4 hub blades.

14100: 14 slots chassis with 2 hub blades.

1440: 14 slot chassis (1440,4440), with 2 hub blades.

640: 6 slots chassis (640), with 2 hub blades.

640-C04: 6 slots chassis (640-C04), with 2 hub blades. With Different Physical Logical Slot Mapping

By default, the SRstackware does not enable any protocol on the ports.

The following sub sections lists the switch ports and associated VLANs. The VLAN configuration depends on whether ATCA-F140 is placed in slot 1 or 2.



The default configuration of SRstackware is similar to the configuration of the Centellis BBS release, with minor differences.

11.2.1 1440 and 14-100 SLOT1 Configuration

Base Channel

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge1	ge0	1000Base-T	BIX - Backplane BC3	21(u)	NO
ge2	ge1	1000Base-T	BIX - Backplane BC4	21(u)	NO
ge3	ge2	1000Base-T	BIX - Backplane BC5	21(u)	NO
ge4	ge3	1000Base-T	BIX - Backplane BC6	21(u)	NO
ge5	ge4	1000Base-T	BIX - Backplane BC7	21(u)	NO
ge6	ge5	1000Base-T	BIX - Backplane BC8	21(u)	NO
ge7	ge6	1000Base-T	BIX - Backplane BC9	21(u)	NO
ge8	ge7	1000Base-T	BIX - Backplane BC10	21(u)	NO
ge9	ge8	1000Base-T	BIX - Backplane BC11	21(u)	NO
ge10	ge9	1000Base-T	BIX - Backplane BC12	21(u)	NO
ge11	ge10	1000Base-T	BIX - Backplane BC13	21(u)	NO
ge12	ge11	1000Base-T	BIX - Backplane BC14	21(u)	NO
ge13	ge12	1000Base-T	BIX - Backplane BC15	21(u)	NO
ge14	ge13	1000Base-T	BIX - Backplane BC16	21(u)	NO
ge15	ge14	10/100B-T	BIX - BC 1A	24(u)	YES
ge16	ge15	10/100B-T	BIX - BC 1B	24(u)	YES
ge17	ge16	1000Base-T	BIX - X-Connect BC2	21(t), 22(t)	NO
ge18	ge17	SERDES	BIX - PrAMC/P0	21 (u)	YES
ge19	ge18	SGMII	BIX - UC-0/1 X-Connect PrAMC	21 (u)	YES
ge20	ge19	SGMII	Clock TopSync or RTM	21 (u)	YES
ge21	ge20	SGMII	BIX - RTM	93(u)	NO
ge22	ge21	SGMII	BIX - RTM	93(u)	NO
ge23	ge22	SGMII	BIX - RTM	93(u)	NO
ge24	ge23	SGMII	BIX - RTM	93(u)	NO

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
xe1	xe0	XAUI	BIX - RTM	93(u)	NO
xe2	xe1	XAUI	BIX - RTM	93(u)	NO
xe3	xe2	XAUI	BIX - FP SFP+	93(u)	NO
xe4	xe3	XAUI	BIX - FP SFP+	93(u)	NO

Fabric Channel

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
xe5	xe0	40GBase-KR4	FIX - Backplane FC2	11(u)	NO
xe9	xe1	40GBase-KR4	FIX - Backplane FC3	11(u)	NO
xe13	xe2	40GBase-KR4	FIX - Backplane FC4	11(u)	NO
xe17	xe3	40GBase-KR4	FIX - Backplane FC5	11(u)	NO
xe21	xe4	40GBase-KR4	FIX - Backplane FC6	11(u)	NO
xe25	xe5	40GBase-KR4	FIX - Backplane FC7	11(u)	NO
xe29	xe6	40GBase-KR4	FIX - Backplane FC8	11(u)	NO
xe33	xe7	40GBase-KR4	FIX - Backplane FC9	11(u)	NO
xe37	xe8	40GBase-KR4	FIX - Backplane FC10	11(u)	NO
xe41	xe9	40GBase-KR4	FIX - Backplane FC11	11(u)	NO
xe45	xe10	40GBase-KR4	FIX - Backplane FC12	11(u)	NO
xe49	xe11	40GBase-KR4	FIX - Backplane FC13	11(u)	NO
xe53	xe12	XLAUI	FIX-RTM QSFP+	91(u)	NO
xe57	xe13	XFI	FIX-RTM SFP+	91(u)	NO
xe58	xe14	XFI	FIX-RTM SFP+	91(u)	NO
xe59	xe15	XFI	FIX-RTM SFP+	91(u)	NO
xe60	xe16	XFI	FIX-RTM SFP+	91(u)	NO
xe61	xe17	XLAUI	FIX-RTM QSFP+	91(u)	NO

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
xe65	xe18	XLAUI	FIX-RTM QSFP+	91(u)	NO
ge25	ge0	1G-SGMII	FIX - Logical Slot 15 or TopSync	11(u)	NO
ge26	ge1	1G-SGMIII	FIX- X-Connect FC1	11(t), 12(t)	NO
ge27	ge2	1G-SGMIII	FIX-UC [3/4] X-Connect PrAMC	11(u)	YES
ge28	ge3	SerDes	FIX - AMC Port 1	11(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
11	Fabric Interface-1	ge26	ge25, ge27, ge28, xe5, xe9, xe13, xe17, xe21, xe25, xe29, xe33, xe37, xe41, xe45, xe49
12	Fabric Interface-2	ge26	none
91	Fabric RTM 1	none	xe53, xe57, xe58, xe59, xe60, xe61, xe65
21	Base Interface-1	ge17	ge1, ge2, ge3, ge4, ge5, ge6, ge7, ge8, ge9, ge10, ge11, ge12, ge13, ge14, ge18, ge19, ge20
22	Base Interface-2	ge17	none
24	Base Interface-3	none	ge15, ge16
93	Base RTM 1	none	ge21, ge22, ge23, ge24, xe1, xe2, xe3, xe4

11.2.2 1440 and 14-100 SLOT2 Configuration

Base Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge1	ge0	1000Base-T	BIX - Backplane BC3	22(u)	NO
ge2	ge1	1000Base-T	BIX - Backplane BC4	22(u)	NO
ge3	ge2	1000Base-T	BIX - Backplane BC5	22(u)	NO
ge4	ge3	1000Base-T	BIX - Backplane BC6	22(u)	NO
ge5	ge4	1000Base-T	BIX - Backplane BC7	22(u)	NO
ge6	ge5	1000Base-T	BIX - Backplane BC8	22(u)	NO
ge7	ge6	1000Base-T	BIX - Backplane BC9	22(u)	NO
ge8	ge7	1000Base-T	BIX - Backplane BC10	22(u)	NO
ge9	ge8	1000Base-T	BIX - Backplane BC11	22(u)	NO
ge10	ge9	1000Base-T	BIX - Backplane BC12	22(u)	NO
ge11	ge10	1000Base-T	BIX - Backplane BC13	22(u)	NO
ge12	ge11	1000Base-T	BIX - Backplane BC14	22(u)	NO
ge13	ge12	1000Base-T	BIX - Backplane BC15	22(u)	NO
ge14	ge13	1000Base-T	BIX - Backplane BC16	22(u)	NO
ge15	ge14	10/100B-T	BIX - BC 1A	24(u)	YES
ge16	ge15	10/100B-T	BIX - BC 1B	24(u)	YES
ge17	ge16	1000Base-T	BIX - X-Connect BC 2	21(t), 22(t)	NO
ge18	ge17	SERDES	BIX - PrAMC/P0	22(u)	YES
ge19	ge18	SGMII	BIX - UC-0/1 X-Connect PrAMC	22(u)	YES
ge20	ge19	SGMII	Clock TopSync or RTM	22(u)	YES
ge21	ge20	SGMII	BIX-RTM	94(u)	NO
ge22	ge21	SGMII	BIX-RTM	94(u)	NO
ge23	ge22	SGMII	BIX-RTM	94(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge24	ge23	SGMII	BIX-RTM	94(u)	NO
xe1	xe0	XAUI	BIX-RTM	94(u)	NO
xe2	xe1	XAUI	BIX-RTM	94(u)	NO
xe3	xe2	XAUI	BIX- FP SFP+	94(u)	NO
xe4	xe3	XAUI	BIX- FP SFP+	94(u)	NO

Fabric Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
xe5	xe0	40GBase-KR4	FIX - Backplane FC2	12(u)	NO
xe9	xe1	40GBase-KR4	FIX - Backplane FC3	12(u)	NO
xe13	xe2	40GBase-KR4	FIX - Backplane FC4	12(u)	NO
xe17	xe3	40GBase-KR4	FIX - Backplane FC5	12(u)	NO
xe21	xe4	40GBase-KR4	FIX - Backplane FC6	12(u)	NO
xe25	xe5	40GBase-KR4	FIX - Backplane FC7	12(u)	NO
xe29	xe6	40GBase-KR4	FIX - Backplane FC8	12(t)	NO
xe33	xe7	40GBase-KR4	FIX - Backplane FC9	12(u)	NO
xe37	xe8	40GBase-KR4	FIX - Backplane FC10	12(u)	NO
xe41	xe9	40GBase-KR4	FIX - Backplane FC11	12(u)	NO
xe45	xe10	40GBase-KR4	FIX - Backplane FC12	12(u)	NO
xe49	xe11	40GBase-KR4	FIX - Backplane FC13	12(u)	NO
xe53	xe12	XLAUI	FIX-RTM QSFP+	92(u)	NO
xe57	xe13	XFI	FIX-RTM SFP+	92(u)	NO
xe58	xe14	XFI	FIX-RTM SFP+	92(u)	NO
xe59	xe15	XFI	FIX-RTM SFP+	92(u)	NO
xe60	xe16	XFI	FIX-RTM SFP+	92(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
xe61	xe17	XLAUI	FIX- FP QSFP+	92(u)	NO
xe65	xe18	XLAUI	FIX- FP QSFP+	92(u)	NO
ge25	ge0	1G-SGMII	FIX - Logical Slot 15 or TopSync	12(u)	NO
ge26	ge1	1G-SGMII	FIX- X-Connect FC1	11(t), 12(t)	NO
ge27	ge2	1G-SGMII	FIX-UC [3/4] X-Connect PrAMC	12(u)	YES
ge28	ge3	SerDes	FIX - AMC Port 1	12(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
11	Fabric Interface-1	ge26	none
12	Fabric Interface-2	ge26	ge25, ge26, ge27, ge28, xe5, xe9, xe13, xe17, xe21, xe25, xe29, xe33, xe37, xe41, xe45, xe49
92	Fabric RTM 1	none	xe53, xe57, xe58, xe59, xe60, xe61, xe65
21	Base Interface-1	ge17	none
22	Base Interface-2	ge17	ge1, ge2, ge3, ge4, ge5, ge6, ge7,ge8, ge9, ge10, ge11, ge12, ge13,ge14, ge18, ge19,ge20
24	Base Interface-3	none	ge15, ge16
94	Base RTM 1	none	ge21, ge22, ge23, ge24, xe1, xe2, xe3, xe4

11.2.3 14-400 SLOT1 Configuration

Base Channel

Interface name	BCM interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
ge1	ge0	1000Base-T	BIX - Backplane BC3	21(u)	NO
ge2	ge1	1000Base-T	BIX - Backplane BC4	21(u)	NO
ge3	ge2	1000Base-T	BIX - Backplane BC5	21(u)	NO
ge4	ge3	1000Base-T	BIX - Backplane BC6	21(u)	NO
ge5	ge4	1000Base-T	BIX - Backplane BC7	21(u)	NO
ge6	ge5	1000Base-T	BIX - Backplane BC8	21(u)	NO
ge7	ge6	1000Base-T	BIX - Backplane BC9	21(u)	NO
ge8	ge7	1000Base-T	BIX - Backplane BC10	21(u)	NO
ge9	ge8	1000Base-T	BIX - Backplane BC11	21(u)	NO
ge10	ge9	1000Base-T	BIX - Backplane BC12	21(u)	NO
ge11	ge10	1000Base-T	BIX - Backplane BC13	21(u)	NO
ge12	ge11	1000Base-T	BIX - Backplane BC14	21(u)	NO
ge13	ge12	1000Base-T	BIX - Backplane BC15	21(u)	NO
ge14	ge13	1000Base-T	BIX - Backplane BC16	21(u)	NO
ge15	ge14	1000Base-T	BIX - BC 1A	24(u)	YES
ge16	ge15	1000Base-T	BIX - BC 1B	24(u)	YES
ge17	ge16	10/100B-T	BIX - X-Connect BC 2	21(t), 22(t)	NO
ge18	ge17	1000Base-T	BIX-PrAMC/P0	21(u)	YES
ge19	ge18	SERDES	BIX - UC-0/1 X-Connect PrAMC	21(u)	YES
ge20	ge19	SGMII	Clock TopSync or RTM	21(u)	YES
ge21	ge20	SGMII	BIX-RTM	93(u)	NO
ge22	ge21	SGMII	BIX-RTM	93(u)	NO
ge23	ge22	SGMII	BIX-RTM	93(u)	NO

Interface name	BCM interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
ge24	ge23	SGMII	BIX-RTM	93(u)	NO
xe1	xe0	XAU1	BIX-RTM	93(u)	NO
xe2	xe1	XAU1	BIX-RTM	93(u)	NO
xe3	xe2	XAU1	BIX- FP SFP+	93(u)	NO
xe4	xe3	XAU1	BIX- FP SFP+	93(u)	NO

Fabric Configuration

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
xe5	xe0	1GBase-KX	FIX - Backplane FC2	11(t) 13(t)	NO
xe6	xe1	Unused port	-	-	-
xe7	xe2	Unused port	-	-	-
xe8	xe3	Unused port	-	-	-
xe9	xe4	1GBase-KX	FIX - Backplane FC3	11(t) 14(t)	NO
xe10	xe5	Unused port	-	-	-
xe11	xe6	Unused port	-	-	-
xe12	xe7	Unused port	-	-	-
xe13	xe8	40GBase-KR4	FIX - Backplane FC4	11(u)	NO
xe17	xe9	40GBase-KR4	FIX - Backplane FC5	11(u)	NO
xe21	xe10	40GBase-KR4	FIX - Backplane FC6	11(u)	NO
xe25	xe11	40GBase-KR4	FIX - Backplane FC7	11(u)	NO
xe29	xe12	40GBase-KR4	FIX - Backplane FC8	11(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
xe33	xe13	40GBase-KR4	FIX - Backplane FC9	11(u)	NO
xe37	xe14	40GBase-KR4	FIX - Backplane FC10	11(u)	NO
xe41	xe15	40GBase-KR4	FIX - Backplane FC11	11(u)	NO
xe45	xe16	40GBase-KR4	FIX - Backplane FC12	11(u)	NO
xe49	xe17	40GBase-KR4	FIX - Backplane FC13	11(u)	NO
xe53	xe18	XLAUI	FIX-RTM QSFP+	91(u)	NO
xe57	xe19	XFI	FIX-RTM SFP+	91(u)	NO
xe58	xe20	XFI	FIX-RTM SFP+	91(u)	NO
xe59	xe21	XFI	FIX-RTM SFP+	91(u)	NO
xe60	xe22	XFI	FIX-RTM SFP+	91(u)	NO
xe61	xe23	XLAUI	FIX-FP QSFP+	91(u)	NO
xe65	xe24	XLAUI	FIX-FP QSFP+	91(u)	NO
ge25	ge0	1G-SGMII	FIX - Logical Slot 15 or TopSync	11(u)	NO
ge26	ge1	1G-SGMII	FIX- X-Connect FC1	11(t), 12(t)	NO
ge27	ge2	1G-SGMII	FIX-UC [3/4] X-Connect PrAMC	11(u)	NO
ge28	ge3	SerDes	FIX - AMC Port 1	11(u)	NO

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
11	Fabric Interface-1	ge26, xe5, xe9	ge25, ge27, ge28, xe13, xe17, xe21, xe25, xe29, xe33, xe37, xe41, xe45, xe49
12	Fabric Interface-2	ge26	None

SRstackware

VLAN ID	Description	Tagged ports	Untagged ports
13	Fabric Interface-3	xe5	None
14	Fabric Interface-4	xe9	None
91	Fabric RTM 1	none	xe53, xe57, xe58, xe59, xe60, xe61, xe65
21	Base Interface-1	ge17	ge1, ge2, ge3, ge4, ge5, ge6, ge7,ge8, ge9, ge10, ge11, ge12, ge13,ge14, ge18, ge19,ge20
22	Base Interface-2	ge17	None
24	Base Interface-3	none	ge15, ge16
93	Base RTM 1	none	ge21, ge22, ge23, ge24, xe1, xe2, xe3, xe4

11.2.4 14-400 SLOT2 Configuration

Base Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
ge1	ge0	1000Base-T	BIX - Backplane BC3	22(u)	NO
ge2	ge1	1000Base-T	BIX - Backplane BC4	22(u)	NO
ge3	ge2	1000Base-T	BIX - Backplane BC5	22(u)	NO
ge4	ge3	1000Base-T	BIX - Backplane BC6	22(u)	NO
ge5	ge4	1000Base-T	BIX - Backplane BC7	22(u)	NO
ge6	ge5	1000Base-T	BIX - Backplane BC8	22(u)	NO
ge7	ge6	1000Base-T	BIX - Backplane BC9	22(u)	NO
ge8	ge7	1000Base-T	BIX - Backplane BC10	22(u)	NO
ge9	ge8	1000Base-T	BIX - Backplane BC11	22(u)	NO
ge10	ge9	1000Base-T	BIX - Backplane BC12	22(u)	NO
ge11	ge10	1000Base-T	BIX - Backplane BC13	22(u)	NO
ge12	ge11	1000Base-T	BIX - Backplane BC14	22(u)	NO
ge13	ge12	1000Base-T	BIX - Backplane BC15	22(u)	NO
ge14	ge13	1000Base-T	BIX - Backplane BC16	22(u)	NO
ge15	ge14	1000Base-T	BIX - BC 1A	24(u)	YES
ge16	ge15	1000Base-T	BIX - BC 1B	24(u)	YES
ge17	ge16	10/100B-T	BIX -X-Connect BC 2	21(t), 22(t)	NO
ge18	ge17	1000Base-T	BIX-PrAMC/P0	22(u)	YES
ge19	ge18	SERDES	BIX - UC-0/1 X-Connect PrAMC	22(u)	YES
ge20	ge19	SGMII	Clock TopSync or RTM	22(u)	YES
ge21	ge20	SGMII	BIX-RTM	94(u)	NO
ge22	ge21	SGMII	BIX-RTM	94(u)	NO
ge23	ge22	SGMII	BIX-RTM	94(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
ge24	ge23	SGMII	BIX-RTM	94(u)	NO
xe1	xe0	XAUI	BIX-RTM	94(u)	NO
xe2	xe1	XAUI	BIX-RTM	94(u)	NO
xe3	xe2	XAUI	BIX- FP SFP+	94(u)	NO
xe4	xe3	XAUI	BIX- FP SFP+	94(u)	NO

Fabric Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
xe5	xe0	40GBase-KR4	FIX - Backplane FC2	12(t) 13(t)	NO
xe9	xe1	40GBase-KR4	FIX - Backplane FC3	12(t) 14(t)	NO
xe13	xe2	40GBase-KR4	FIX - Backplane FC4	12(u)	NO
xe17	xe3	40GBase-KR4	FIX - Backplane FC5	12(u)	NO
xe21	xe4	40GBase-KR4	FIX - Backplane FC6	12(u)	NO
xe25	xe5	40GBase-KR4	FIX - Backplane FC7	12(u)	NO
xe29	xe6	40GBase-KR4	FIX - Backplane FC8	12(u)	NO
xe33	xe7	40GBase-KR4	FIX - Backplane FC9	12(u)	NO
xe37	xe8	40GBase-KR4	FIX - Backplane FC10	12(u)	NO
xe41	xe9	40GBase-KR4	FIX - Backplane FC11	12(u)	NO
xe45	xe10	40GBase-KR4	FIX - Backplane FC12	12(u)	NO
xe49	xe11	40GBase-KR4	FIX - Backplane FC13	12(u)	NO
xe53	xe12	XLAUI	FIX-RTM QSFP+	92(u)	NO
xe57	xe13	XFI	FIX-RTM SFP+	92(u)	NO
xe58	xe14	XFI	FIX-RTM SFP+	92(u)	NO
xe59	xe15	XFI	FIX-RTM SFP+	92(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
xe60	xe16	XFI	FIX-RTM SFP+	92(u)	NO
xe61	xe17	XLAUI	FIX-FP QSFP+	92(u)	NO
xe65	xe18	XLAUI	FIX-FP QSFP+	92(u)	NO
ge25	ge0	1G-SGMII	FIX - Logical Slot 15 or TopSync	12(u)	NO
ge26	ge1	1G-SGMII	FIX- X-Connect FC1	11(t), 12(t)	NO
ge27	ge2	1G-SGMII	FIX-UC [3/4] X-Connect PrAMC	12(u)	YES
ge28	ge3	SerDes	FIX - AMC Port 1	12(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
12	Fabric Interface-1	ge26, xe5, xe9	ge25, ge26, ge27, ge28, xe13, xe17, xe21, xe25, xe29, xe33, xe37, xe41, xe45, xe49
11	Fabric Interface-2	ge26	None
13	Fabric Interface-3	xe5	None
14	Fabric Interface-4	xe9	None
92	Fabric RTM 1	none	xe53, xe57, xe58, xe59, xe60, xe61, xe65
21	Base Interface-1	ge17	ge1, ge2, ge3, ge4, ge5, ge6, ge7, ge8, ge9, ge10, ge11, ge12, ge13, ge14, ge18, ge19, ge20
22	Base Interface-2	ge17	None
24	Base Interface-3	none	ge15, ge16
94	Base RTM 1	none	ge21, ge22, ge23, ge24, xe1, xe2, xe3, xe4

11.2.5 14-400 SLOT3 Configuration

Base Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
ge1	ge0	1000Base-T	Not Connected	-	-
ge2	ge1	1000Base-T	Not Connected	-	-
ge3	ge2	1000Base-T	Not Connected	-	-
ge4	ge3	1000Base-T	Not Connected	-	-
ge5	ge4	1000Base-T	Not Connected	-	-
ge6	ge5	1000Base-T	Not Connected	-	-
ge7	ge6	1000Base-T	Not Connected	-	-
ge8	ge7	1000Base-T	Not Connected	-	-
ge9	ge8	1000Base-T	Not Connected	-	-
ge10	ge9	1000Base-T	Not Connected	-	-
ge11	ge10	1000Base-T	Not Connected	-	-
ge12	ge11	1000Base-T	Not Connected	-	-
ge13	ge12	1000Base-T	Not Connected	-	-
ge14	ge13	1000Base-T	Not Connected	-	-
ge15	ge14	10/100B-T	BIX - BC1	21(u)	YES
ge16	ge15	10/100B-T	Not Connected	-	-
ge17	ge16	1000Base-T	BIX-BC 2	22(u)	NO
ge18	ge17	SERDES	Base PrAMC	21(u)	-
ge19	ge18	SGMII	BIX - UC-0/1 X-Connect PrAMC	21(u)	-
ge20	ge19	SGMII	Not Connected	-	-
ge21	ge20	SGMII	BIX-RTM	93(u)	NO
ge22	ge21	SGMII	BIX-RTM	93(u)	NO
ge23	ge22	SGMII	BIX-RTM	93(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
ge24	ge23	SGMII	BIX-RTM	93(u)	NO
xe1	xe0	XAUI	BIX-RTM	93(u)	NO
xe2	xe1	XAUI	BIX-RTM	93(u)	NO
xe3	xe2	XAUI	BIX- FP SFP+	93(u)	NO
xe4	xe3	XAUI	BIX- FP SFP+	93(u)	NO

Fabric Configuration

Interface name	BCM Interface name	Interface description	Destination	VLAN (s) Assigned	Default Enabled
xe5	xe0	40GBase-KR4	FIX - Backplane FC2	13(t) 12(t)	NO
xe9	xe1	40GBase-KR4	FIX - Backplane FC3	14(t) 13(t)	NO
xe13	xe2	40GBase-KR4	FIX - Backplane FC4	13(u)	NO
xe17	xe3	40GBase-KR4	FIX - Backplane FC5	13(u)	NO
xe21	xe4	40GBase-KR4	FIX - Backplane FC6	13(u)	NO
xe25	xe5	40GBase-KR4	FIX - Backplane FC7	13(u)	NO
xe29	xe6	40GBase-KR4	FIX - Backplane FC8	13(u)	NO
xe33	xe7	40GBase-KR4	FIX - Backplane FC9	13(u)	NO
xe37	xe8	40GBase-KR4	FIX - Backplane FC10	13(u)	NO
xe41	xe9	40GBase-KR4	FIX - Backplane FC11	13(u)	NO
xe45	xe10	40GBase-KR4	FIX - Backplane FC12	13(u)	NO
xe49	xe11	40GBase-KR4	FIX - Backplane FC13	13(u)	NO
xe53	xe12	XLAUI	FIX-RTM QSFP+	92(u)	NO
xe57	xe13	XFI	FIX-RTM SFP+	92(u)	NO
xe58	xe14	XFI	FIX-RTM SFP+	92(u)	NO
xe59	xe15	XFI	FIX-RTM SFP+	92(u)	NO
xe60	xe16	XFI	FIX-RTM SFP+	92(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN (s) Assigned	Default Enabled
xe61	xe17	XLAUI	FIX-FP QSFP+	92(u)	NO
xe65	xe18	XLAUI	FIX-FP QSFP+	92(u)	NO
ge25	ge0	1G-SGMII	FIX - Logical Slot 15 or TopSync	13(u)	NO
ge26	ge1	1G-SGMII	FIX- X-Connect FC1	13(t), 11(t)	NO
ge27	ge2	1G-SGMII	FIX-UC [3/4] X-Connect PrAMC	13(u)	YES
ge28	ge3	SerDes	FIX - AMC Port 1	13(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
13	Fabric Interface	ge26, xe5, xe9	ge25, ge27, ge28, xe13, xe17, xe21, xe25, xe29, xe33, xe37, xe41, xe45, xe49
11	Fabric Interface-1	ge26	None
12	Fabric Interface-2	xe5	None
14	Fabric Interface-4	xe9	None
92	Fabric RTM 1	none	xe53, xe57, xe58, xe59, xe60, xe61, xe65
21	Base Interface-1	None	ge15
22	Base Interface-2	none	ge17
94	Base RTM 1	none	ge21, ge22, ge23, ge24, xe1, xe2, xe3, xe4

11.2.6 14-400 SLOT4 Configuration

Base Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
ge1	ge0	1000Base-T	Not Connected	-	-
ge2	ge1	1000Base-T	Not Connected	-	-
ge3	ge2	1000Base-T	Not Connected	-	-
ge4	ge3	1000Base-T	Not Connected	-	-
ge5	ge4	1000Base-T	Not Connected	-	-
ge6	ge5	1000Base-T	Not Connected	-	-
ge7	ge6	1000Base-T	Not Connected	-	-
ge8	ge7	1000Base-T	Not Connected	-	-
ge9	ge8	1000Base-T	Not Connected	-	-
ge10	ge9	1000Base-T	Not Connected	-	-
ge11	ge10	1000Base-T	Not Connected	-	-
ge12	ge11	1000Base-T	Not Connected	-	-
ge13	ge12	1000Base-T	Not Connected	-	-
ge14	ge13	1000Base-T	Not Connected	-	-
ge15	ge14	10/100B-T	BIX - BC1	21(u)	YES
ge16	ge15	10/100B-T	Not Connected	-	-
ge17	ge16	1000Base-T	BIX-BC 2	22(u)	NO
ge18	ge17	SERDES	Base PrAMC	22(u)	-
ge19	ge18	SGMII	BIX - UC-0/1 X-Connect PrAMC	22(u)	-
ge20	ge19	SGMII	Not Connected	-	-
ge21	ge20	SGMII	BIX-RTM	93(u)	NO
ge22	ge21	SGMII	BIX-RTM	93(u)	NO
ge23	ge22	SGMII	BIX-RTM	93(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) Assigned	Default Enabled
ge24	ge23	SGMII	BIX-RTM	93(u)	NO
xe1	xe0	XAUI	BIX-RTM	93(u)	NO
xe2	xe1	XAUI	BIX-RTM	93(u)	NO
xe3	xe2	XAUI	BIX- FP SFP+	93(u)	NO
xe4	xe3	XAUI	BIX- FP SFP+	93(u)	NO

Fabric Configuration

Interface name	BCM Interface name	Interface description	Destination	VLAN (s) Assigned	Default Enabled
xe5	xe0	40GBase-KR4	FIX - Backplane FC2	14(t) 12(t)	NO
xe9	xe1	40GBase-KR4	FIX - Backplane FC3	14(t) 13(t)	NO
xe13	xe2	40GBase-KR4	FIX - Backplane FC4	14(u)	NO
xe17	xe3	40GBase-KR4	FIX - Backplane FC5	14(u)	NO
xe21	xe4	40GBase-KR4	FIX - Backplane FC6	14(u)	NO
xe25	xe5	40GBase-KR4	FIX - Backplane FC7	14(u)	NO
xe29	xe6	40GBase-KR4	FIX - Backplane FC8	14(u)	NO
xe33	xe7	40GBase-KR4	FIX - Backplane FC9	14(u)	NO
xe37	xe8	40GBase-KR4	FIX - Backplane FC10	14(u)	NO
xe41	xe9	40GBase-KR4	FIX - Backplane FC11	14(u)	NO
xe45	xe10	40GBase-KR4	FIX - Backplane FC12	14(u)	NO
xe49	xe11	40GBase-KR4	FIX - Backplane FC13	14(u)	NO
xe53	xe12	XLAUI	FIX-RTM QSFP+	92(u)	NO
xe57	xe13	XFI	FIX-RTM SFP+	92(u)	NO
xe58	xe14	XFI	FIX-RTM SFP+	92(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN (s) Assigned	Default Enabled
xe59	xe15	XFI	FIX-RTM SFP+	92(u)	NO
xe60	xe16	XFI	FIX-RTM SFP+	92(u)	NO
xe61	xe17	XLAUI	FIX-FP QSFP+	92(u)	NO
xe65	xe18	XLAUI	FIX-FP QSFP+	92(u)	NO
ge25	ge0	1G-SGMII	FIX - Logical Slot 15 or TopSync	14(u)	NO
ge26	ge1	1G-SGMII	FIX- X-Connect FC1	14(t),11(t)	NO
ge27	ge2	1G-SGMII	FIX-UC [3/4] X-Connect PrAMC	14(u)	YES
ge28	ge3	SerDes	FIX - AMC Port 1	14(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
14	Fabric Interface	ge26, xe5, xe9	ge25, ge27, ge28, xe13, xe17, xe21, xe25, xe29, xe33, xe37, xe41, xe45, xe49
11	Fabric Interface-1	ge26	None
12	Fabric Interface-2	xe5	None
13	Fabric Interface-4	xe9	None
92	Fabric RTM 1	None	xe53, xe57, xe58, xe59, xe60, xe61, xe65
21	Base Interface-1	None	ge15
22	Base Interface-2	None	ge17
94	Base RTM 1	None	ge21, ge22, ge23, ge24, xe1, xe2, xe3, xe4

11.2.7 640 SLOT1 Configuration

NOTICE					
<p>On a 640 chassis with single ATCA-F140, it is recommended to place the Hub-blade in SLOT2. In such configuration, SLOT1 can be used for placing a payload blade.</p>					

Base Channel

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge1	ge0	1000Base-T	Not Connected	-	-
ge2	ge1	1000Base-T	Not Connected	-	-
ge3	ge2	1000Base-T	Not Connected	-	-
ge4	ge3	1000Base-T	Not Connected	-	-
ge5	ge4	1000Base-T	Not Connected	-	-
ge6	ge5	1000Base-T	Not Connected	-	-
ge7	ge6	1000Base-T	Not Connected	-	-
ge8	ge7	1000Base-T	Not Connected	-	-
ge9	ge8	1000Base-T	BIX - Backplane BC11	21(u)	NO
ge10	ge9	1000Base-T	BIX - Backplane BC12	21(u)	NO
ge11	ge10	1000Base-T	BIX - Backplane BC13	21(u)	NO
ge12	ge11	1000Base-T	BIX - Backplane BC14	21(u)	NO
ge13	ge12	1000Base-T	Not Connected	-	-
ge14	ge13	1000Base-T	Not Connected	-	-
ge15	ge14	10/100B-T	BIX - BC 1A	24(u)	YES
ge16	ge15	10/100B-T	BIX - BC 1B	24(u)	YES
ge17	ge16	1000Base-T	BIX - X-Connect BC2	21(t), 22(t)	NO
ge18	ge17	SERDES	BIX - PrAMC/P0	21 (u)	YES
ge19	ge18	SGMII	BIX - UC-0/1 X-Connect PrAMC	21 (u)	YES
ge20	ge19	SGMII	Clock TopSync or RTM	21 (u)	YES

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge21	ge20	SGMII	BIX - RTM	93(u)	NO
ge22	ge21	SGMII	BIX - RTM	93(u)	NO
ge23	ge22	SGMII	BIX - RTM	93(u)	NO
ge24	ge23	SGMII	BIX - RTM	93(u)	NO
xe1	xe0	XAUI	BIX - RTM	93(u)	NO
xe2	xe1	XAUI	BIX - RTM	93(u)	NO
xe3	xe2	XAUI	BIX - FP SFP+	93(u)	NO
xe4	xe3	XAUI	BIX - FP SFP+	93(u)	NO

Fabric Channel

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
xe5	xe0	40GBase-KR4	FIX - X-Connect FC2	11(t), 12(t)	NO
xe9	xe1	40GBase-KR4	FIX - Backplane FC10	11(u)	NO
xe13	xe2	40GBase-KR4	FIX - Backplane FC11	11(u)	NO
xe17	xe3	40GBase-KR4	FIX - Backplane FC12	11(u)	NO
xe21	xe4	40GBase-KR4	FIX - Backplane FC13	11(u)	NO
xe25	xe5	XLAUI	FIX-RTM QSFP+	91(u)	NO
xe29	xe6	XFI	FIX-RTM SFP+	91(u)	NO
xe30	xe7	XFI	FIX-RTM SFP+	91(u)	NO
xe31	xe8	XFI	FIX-RTM SFP+	91(u)	NO
xe32	xe9	XFI	FIX-RTM SFP+	91(u)	NO
xe33	xe10	XLAUI	FIX-FP QSFP+	91(u)	NO
xe37	xe11	XLAUI	FIX-FP QSFP+	91(u)	NO
ge25	ge0	1G-SGMII	FIX - Logical Slot 15 or TopSync	11(u)	NO

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge26	ge1	1G-SGMIII	FIX - Logical Slot 16 or X-Connect FC1	11(t), 12(t)	NO
ge27	ge2	1G-SGMIII	FIX-UC [3/4]	11(u)	YES
ge28	ge3	SerDes	FIX - AMC Port 1	11(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
11	Fabric Interface-1	ge26, xe5	ge25, ge27, ge28, xe9, xe13, xe17, xe21
12	Fabric Interface-2	ge26, xe5	none
91	Fabric RTM 1	none	xe25, xe29, xe30, xe31, xe32, xe33, xe37
21	Base Interface-1	ge17	ge9, ge10, ge11, ge12, ge18, ge19,ge20
22	Base Interface-2	ge17	none
24	Base Interface-3	none	ge15, ge16
93	Base RTM 1	none	ge21, ge22, ge23, ge24, xe1, xe2, xe3, xe4

11.2.8 640 SLOT2 Configuration

Base Channel

NOTICE

In a 640 chassis with a single ATCA-F140 in SLOT2 and a payload blade in SLOT1, BC2 and FC2 of the payload will be connected to the X-Connects BC2 (ge17) and FC2 (xe5) of ATCA-F140 respectively.

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge1	ge0	1000Base-T	Not Connected	-	-
ge2	ge1	1000Base-T	Not Connected	-	-
ge3	ge2	1000Base-T	Not Connected	-	-
ge4	ge3	1000Base-T	Not Connected	-	-
ge5	ge4	1000Base-T	Not Connected	-	-
ge6	ge5	1000Base-T	Not Connected	-	-
ge7	ge6	1000Base-T	Not Connected	-	-
ge8	ge7	1000Base-T	Not Connected	-	-
ge9	ge8	1000Base-T	BIX - Backplane BC11	22(u)	NO
ge10	ge9	1000Base-T	BIX - Backplane BC12	22(u)	NO
ge11	ge10	1000Base-T	BIX - Backplane BC13	22(u)	NO
ge12	ge11	1000Base-T	BIX - Backplane BC14	22(u)	NO
ge13	ge12	1000Base-T	Not Connected	-	-
ge14	ge13	1000Base-T	Not Connected	-	-
ge15	ge14	10/100B-T	BIX - BC 1A	24(u)	YES
ge16	ge15	10/100B-T	BIX - BC 1B	24(u)	YES
ge17	ge16	1000Base-T	BIX - X-Connect BC 2	21(t), 22(t)	NO
ge18	ge17	SERDES	BIX - PrAMC/P0	22(u)	YES
ge19	ge18	SGMII	BIX - UC-0/1 X-Connect PrAMC	22(u)	YES

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge20	ge19	SGMII	Clock TopSync or RTM	22(u)	YES
ge21	ge20	SGMII	BIX-RTM	94(u)	NO
ge22	ge21	SGMII	BIX-RTM	94(u)	NO
ge23	ge22	SGMII	BIX-RTM	94(u)	NO
ge24	ge23	SGMII	BIX-RTM	94(u)	NO
xe1	xe0	XAUI	BIX-RTM	94(u)	NO
xe2	xe1	XAUI	BIX-RTM	94(u)	NO
xe3	xe2	XAUI	BIX- FP SFP+	94(u)	NO
xe4	xe3	XAUI	BIX- FP SFP+	94(u)	NO

Fabric Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
xe5	xe0	40GBase-KR4	FIX - X-Connect FC2	11(t), 12(t)	NO
xe9	xe1	40GBase-KR4	FIX - Backplane FC10	12(u)	NO
xe13	xe2	40GBase-KR4	FIX - Backplane FC11	12(u)	NO
xe17	xe3	40GBase-KR4	FIX - Backplane FC12	12(u)	NO
xe21	xe4	40GBase-KR4	FIX - Backplane FC13	12(u)	NO
xe25	xe5	XLAUI	FIX-RTM QSFP+	92(u)	NO
xe29	xe6	XFI	FIX-RTM SFP+	92(u)	NO
xe30	xe7	XFI	FIX-RTM SFP+	92(u)	NO
xe31	xe8	XFI	FIX-RTM SFP+	92(u)	NO
xe32	xe9	XFI	FIX-RTM SFP+	92(u)	NO
xe33	xe10	XLAUI	FIX-FP QSFP+	92(u)	NO
xe37	xe11	XLAUI	FIX-FP QSFP+	92(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge25	ge0	1G-SGMII	FIX - Logical Slot 15 or TopSync	12(u)	NO
ge26	ge1	1G-SGMII	FIX - Logical Slot 16 or X-connect FC1	11(t), 12(t)	NO
ge27	ge2	1G-SGMII	FIX-UC [3/4]	12(u)	YES
ge28	ge3	SerDes	FIX - AMC Port 1	12(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
11	Fabric Interface-2	ge26, xe5	none
12	Fabric Interface-1	ge26, xe5	xe9, xe13, xe17, xe21, ge25, ge27, ge28
92	Fabric RTM 1	none	xe25, xe29, xe30, xe31, xe32, xe33, xe37
21	Base Interface-1	ge17	none
22	Base Interface-2	ge17	ge9, ge10, ge11, ge12, ge18, ge19,ge20
24	Base Interface-3	none	ge15, ge16
94	Base RTM 1	none	ge21, ge22, ge23, ge24, xe1, xe2, xe3, xe4

11.2.9 640-C04 SLOT1 Configuration

Base Channel

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge1	ge0	1000Base-T	BIX - Backplane BC3	21(u)	NO
ge2	ge1	1000Base-T	BIX - Backplane BC4	21(u)	NO
ge3	ge2	1000Base-T	BIX - Backplane BC5	21(u)	NO
ge4	ge3	1000Base-T	BIX - Backplane BC6	21(u)	NO
ge5	ge4	1000Base-T	Not Connected	-	-
ge6	ge5	1000Base-T	Not Connected	-	-
ge7	ge6	1000Base-T	Not Connected	-	-
ge8	ge7	1000Base-T	Not Connected	-	-
ge9	ge8	1000Base-T	Not Connected	-	-
ge10	ge9	1000Base-T	Not Connected	-	-
ge11	ge10	1000Base-T	Not Connected	-	-
ge12	ge11	1000Base-T	Not Connected	-	-
ge13	ge12	1000Base-T	Not Connected	-	-
ge14	ge13	1000Base-T	Not Connected	-	-
ge15	ge14	10/100B-T	BIX - BC 1A	24(u)	YES
ge16	ge15	10/100B-T	BIX - BC 1B	24(u)	YES
ge17	ge16	1000Base-T	BIX - X-Connect BC2	21(t), 22(t)	NO
ge18	ge17	SERDES	BIX - PrAMC/P0	21 (u)	YES
ge19	ge18	SGMII	BIX - UC-0/1 X-Connect PrAMC	21 (u)	YES
ge20	ge19	SGMII	Clock TopSync or RTM	21 (u)	YES
ge21	ge20	SGMII	BIX - RTM	93(u)	NO
ge22	ge21	SGMII	BIX - RTM	93(u)	NO
ge23	ge22	SGMII	BIX - RTM	93(u)	NO

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge24	ge23	SGMII	BIX - RTM	93(u)	NO
xe1	xe0	XAUI	BIX - RTM	93(u)	NO
xe2	xe1	XAUI	BIX - RTM	93(u)	NO
xe3	xe2	XAUI	BIX - FP SFP+	93(u)	NO
xe4	xe3	XAUI	BIX - FP SFP+	93(u)	NO

Fabric Channel

Interface name	BCM interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
xe5	xe0	40GBase-KR4	FIX - Backplane FC2	11(u)	NO
xe9	xe1	40GBase-KR4	FIX - Backplane FC3	11(u)	NO
xe13	xe2	40GBase-KR4	FIX - Backplane FC4	11(u)	NO
xe17	xe3	40GBase-KR4	FIX - Backplane FC5	11(u)	NO
xe21	xe4	XLAUI	FIX-RTM QSFP+ ETH7	91(u)	NO
xe25	xe5	XFI	FIX-RTM SFP+ ETH6	91(u)	NO
xe26	xe6	XFI	FIX-RTM SFP+ ETH5	91(u)	NO
xe27	xe7	XFI	FIX-RTM SFP+ ETH4	91(u)	NO
xe28	xe8	XFI	FIX-RTM SFP+ ETH3	91(u)	NO
xe29	xe9	XLAUI	FIX-Fabric Front ETH1 QSFP+	91(u)	NO
xe33	xe10	XLAUI	FIX-Fabric Front ETH2 QSFP+	91(u)	NO
ge25	ge0	1G-SGMII	FIX - Logical Slot 15 or TopSync	11(u)	NO
ge26	ge1	1G-SGMIII	FIX - Logical Slot 16 or X-Connect FC1	11(t), 12(t)	NO
ge27	ge2	1G-SGMIII	FIX-UC [3/4]	11(u)	YES
ge28	ge3	SerDes	FIX - AMC Port 1	11(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
11	Fabric Interface-1	ge26	ge25, ge27, ge28, xe5, xe9, xe13, xe17
12	Fabric Interface-2	ge26	none
91	Fabric RTM 1	none	xe21, xe25, xe26, xe27, xe28, xe29, xe33
21	Base Interface-1	ge17	ge1, ge2, ge3, ge4, ge18, ge19,ge20
22	Base Interface-2	ge17	none
24	Base Interface-3	none	ge15, ge16
93	Base RTM 1	none	ge21, ge22, ge23, ge24, xe1, xe2, xe3, xe4

11.2.10 640-C04 SLOT2 Configuration

Base Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge1	ge0	1000Base-T	BIX - Backplane BC3	22(u)	NO
ge2	ge1	1000Base-T	BIX - Backplane BC4	22(u)	NO
ge3	ge2	1000Base-T	BIX - Backplane BC5	22(u)	NO
ge4	ge3	1000Base-T	BIX - Backplane BC6	22(u)	NO
ge5	ge4	1000Base-T	Not Connected	-	-
ge6	ge5	1000Base-T	Not Connected	-	-
ge7	ge6	1000Base-T	Not Connected	-	-
ge8	ge7	1000Base-T	Not Connected	-	-
ge9	ge8	1000Base-T	Not Connected	-	-
ge10	ge9	1000Base-T	Not Connected	-	-
ge11	ge10	1000Base-T	Not Connected	-	-
ge12	ge11	1000Base-T	Not Connected	-	-
ge13	ge12	1000Base-T	Not Connected	-	-
ge14	ge13	1000Base-T	Not Connected	-	-
ge15	ge14	10/100B-T	BIX - BC 1A	24(u)	YES
ge16	ge15	10/100B-T	BIX - BC 1B	24(u)	YES
ge17	ge16	1000Base-T	BIX - X-Connect BC 2	21(t), 22(t)	NO
ge18	ge17	SERDES	BIX - PrAMC/P0	22(u)	YES
ge19	ge18	SGMII	BIX - UC-0/1 X-Connect PrAMC	22(u)	YES
ge20	ge19	SGMII	Clock TopSync or RTM	22(u)	YES
ge21	ge20	SGMII	BIX-RTM	94(u)	NO
ge22	ge21	SGMII	BIX-RTM	94(u)	NO
ge23	ge22	SGMII	BIX-RTM	94(u)	NO

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge24	ge23	SGMII	BIX-RTM	94(u)	NO
xe1	xe0	XAUI	BIX-RTM	94(u)	NO
xe2	xe1	XAUI	BIX-RTM	94(u)	NO
xe3	xe2	XAUI	BIX- FP SFP+	94(u)	NO
xe4	xe3	XAUI	BIX- FP SFP+	94(u)	NO

Fabric Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
xe5	xe0	40GBase-KR4	FIX -Backplane FC2	12(u)	NO
xe9	xe1	40GBase-KR4	FIX - Backplane FC3	12(u)	NO
xe13	xe2	40GBase-KR4	FIX - Backplane FC4	12(u)	NO
xe17	xe3	40GBase-KR4	FIX - Backplane FC5	12(u)	NO
xe21	xe4	XLAUI	FIX-RTM QSFP+ ETH7	92(u)	NO
xe25	xe5	XFI	FIX-RTM SFP+ ETH6	92(u)	NO
xe26	xe6	XFI	FIX-RTM SFP+ ETH5	92(u)	NO
xe27	xe7	XFI	FIX-RTM SFP+ ETH4	92(u)	NO
xe28	xe8	XFI	FIX-RTM SFP+ ETH3	92(u)	NO
xe29	xe9	XLAUI	FIX-Fabric Front ETH1 QSFP+	92(u)	NO
xe33	xe10	XLAUI	FIX-Fabric Front ETH2 QSFP+	92(u)	NO
ge25	ge0	1G-SGMII	FIX - Logical Slot 15 or TopSync	12(u)	NO
ge26	ge1	1G-SGMII	FIX - Logical Slot 16 or X-connect FC1	11(t), 12(t)	NO
ge27	ge2	1G-SGMII	FIX-UC [3/4]	12(u)	YES
ge28	ge3	SerDes	FIX - AMC Port 1	12(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
11	Fabric Interface-2	ge26	none
12	Fabric Interface-1	ge26	xe5, xe9, xe13, xe17, ge25, ge27, ge28
92	Fabric RTM 1	none	xe21, xe25, xe26, xe27, xe28, xe29, xe33
21	Base Interface-1	ge17	none
22	Base Interface-2	ge17	ge1, ge2, ge3, ge4, ge18, ge19,ge20
24	Base Interface-3	none	ge15, ge16
94	Base RTM 1	none	ge21, ge22, ge23, ge24, xe1, xe2, xe3, xe4

11.3 Additional Configuration

The chassis with the SHM1500 shelf manager needs additional configuration, along with the generic default configuration of VLANs for ATCA-F140. This configuration is automatically done after identifying the chassis during startup.

The following sub sections detail the additional configuration information.

11.3.1 SLOT1 Configuration

Base Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN (s) assigned	Default Enabled
ge15	ge0	10/100B-T	BIX - BC 1A	24(u)	YES
ge16	ge1	10/100B-T	BIX - BC 1A	24(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
24	Shelf Management		ge15, ge16

Also vlan1.24 is assigned with the IP address 192.168.24.1/24

11.3.2 SLOT2 Configuration

Base Channel

Interface name	BCM Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge15	ge0	10/100B-T	BIX - BC 1A	24(u)	YES
ge16	ge1	10/100B-T	BIX - BC 1A	24(u)	YES

VLAN Configuration

VLAN ID	Description	Tagged ports	Untagged ports
24	Shelf Management		ge15, ge16

Also Vlan1.24 is assigned with the IP address 192.168.24.2/24.

11.3.3 Mapping Physical RTM Interfaces to SRstackware Ports

Default Switch Configuration Examples.

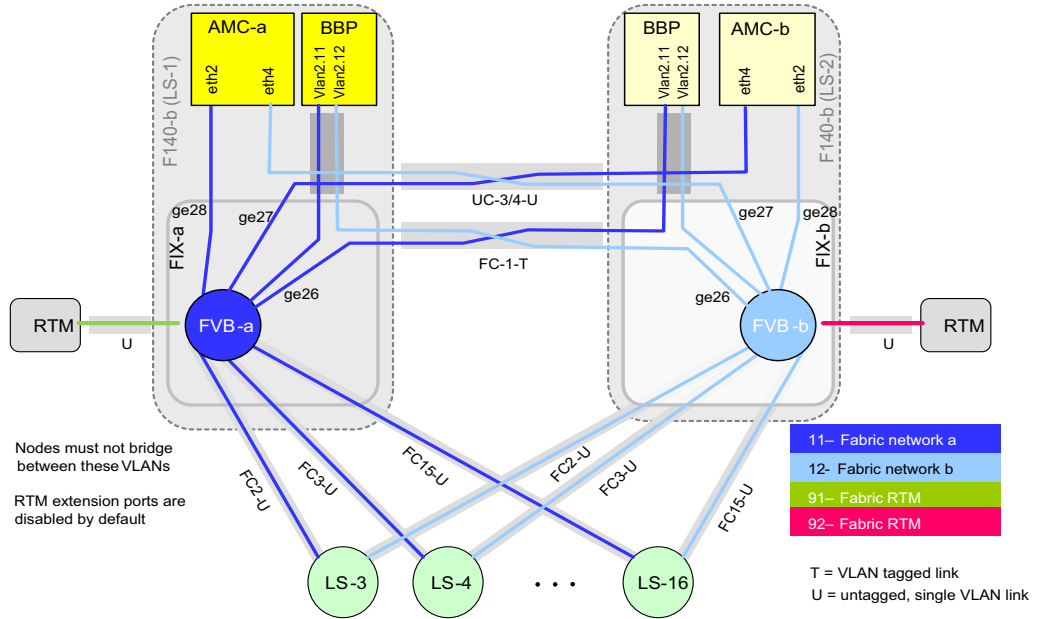
Table 11-1 Mapping Physical RTM Interfaces to SRstackware Ports

Physical Interface	SRstackware Port
ETH8	ge21
ETH9	ge22
ETH10	ge23
ETH11	ge24
ETH1	xe1
ETH2	xe2
ETH7	xe53
ETH6	xe57
ETH5	xe58
ETH4	xe59
ETH3	xe60

11.3.4 ATCA-F140 Switch Configuration Examples

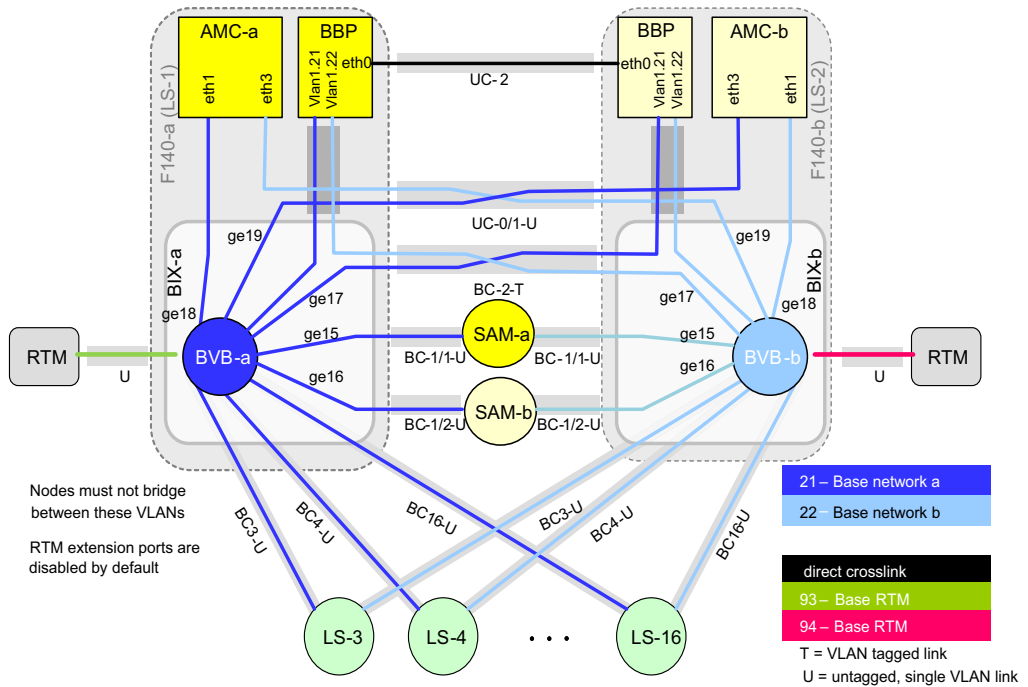
The following figures show the VLAN configuration of the blade.

Figure 11-1 ATCA-F140 Switch Management Fabric Interface Bridge Configuration



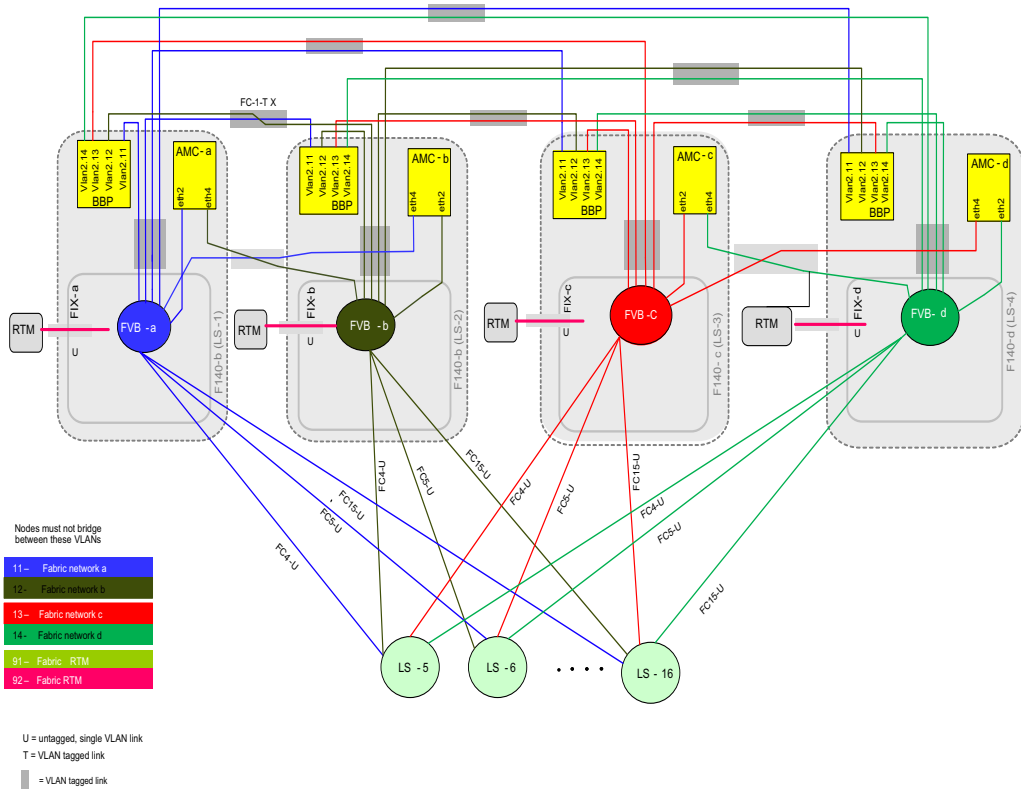
NOTE: The Fabric interface name of PrAMC-7311 varies with the type of Linux flavor installed. For Ubuntu it is eth3 and for Wind River and RedHat/CentOS it is eth2.

Figure 11-2 ATCA-F140 Switch Management Base Interface Bridge Configuration



NOTE: The Base2 interface name of PrAMC-7311 varies with the type of Linux flavor installed. For Ubuntu it is eth2 and for Wind River and RedHat/CentOS it is eth3. This configuration is applicable for AMC7311.

Figure 11-3 AXP14-400-ATCA-F140 Switch Management Fabric Interface Bridge Configuration



NOTE: The Fabric interface name of PrAMC-7311 varies with the type of Linux flavor installed. For Ubuntu it is eth3 and for Wind River and RedHat/CentOS it is eth2.

11.4 Custom Configuration

You can customize SRstackware after starting the SRstackware CLI using the following steps:

1. Login as `root` at the ATCA-F140 console or using the SSH session.
2. Run `/opt/srstackware/bin/imish`.

A sample output of the above steps is shown below.

```
root@Slot-1_f140:/opt/srstackware/bin> ./imish
ZebOS version 7.6.1.e0 IPIRouter 07/14/08 22:33:20
Slot-1_f140>enable
Slot-1_f140#show interface
```

11.4.1 Enabling Protocols

You can enable protocols on the base or fabric switch port of ATCA-F140. By default, the protocols are disabled. The base and fabric switch ports are associated with two bridges, bridge 1 and bridge 2 that are configured by default. You can enable any protocol on these bridges and enable the respective interfaces for the protocol to function properly on the bridge. Even though STP is enabled on these bridges they are administratively down on the interfaces.

In case a bridge is removed from an interface, all the default configurations associated to that interface are also removed. You need to enable the desired VLANs and protocols on those interfaces.

It is not advisable to connect the base and fabric channel ports.

11.4.2 Enabling RTM ports

The SRstackware default configuration disables the base and fabric RTM ports, with STP protocol disabled. You can enable the desired ports and configure to avoid any looping.

11.4.3 Sample Configurations

This section provides a quick reference to the sequence of CLI commands for enabling protocols on the RTM ports or back plane ports.

- To check if the link at ge17 interface is up or down:

```
root@Slot-1_f140:/opt/srstackware/bin> ./imish
ZebOS version 7.6.1 IPIRouter 11/09/10 07:31:33
Slot-1_f140>enable
Slot-1_f140#show interface ge17
Interface ge17
  Hardware is Ethernet
  Current HW addr: 00fd.96bf.0013
  Physical:00fd.96bf.0013
  Description: BC2-XLink
```

```
index 5017 metric 1 mtu 1500 duplex-full arp ageing timeout 0
<UP,BROADCAST,RUNNING,MULTICAST>
VRF Binding: Not bound
Bandwidth 1g
VRRP Master of : VRRP is not configured on this interface.
input packets 075743, bytes 04850346, dropped 03, multicast
packets 014
output packets 011, bytes 02266, multicast packets 06 broadcast
packets 075732
```

In the above sample output, UP in the following line indicates that the interface is administratively up, whereas RUNNING indicates that the interface link is operationally up.

```
<UP,BROADCAST,RUNNING,MULTICAST>
```

In case the link is operationally down, the sample output contains the following line.

```
<UP,BROADCAST,MULTICAST>
```

- To show all the interfaces (switch ports):

```
root@Slot-1_f140:/opt/srstackware/bin> ./imish
ZebOS version 7.6.1 IPIRouter 11/09/10 07:31:33
Slot-1_f140>enable
Slot-1_f140#show interface
```

You can use include and begin options with this command to display the modified output.

- To configure a bridge with STP enabled on it:

```
root@Slot-1_f140:/opt/srstackware/bin> ./imish
ZebOS version 7.6.1 IPIRouter 11/09/10 07:31:33
Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#bridge 2 protocol ieee
Slot-1_f140(config)#
```

- To configure a VLAN and associate it to a bridge:

```
root@Slot-1_f140:/opt/srstackware/bin> ./imish
ZebOS version 7.6.1 IPIRouter 11/09/10 07:31:33
```



```
Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#bridge 2 protocol ieee vlan-bridge
Slot-1_f140(config)#vlan database
Slot-1_f140(config-vlan)#vlan 23 bridge 1 name VLAN23 state enable
You can use the following command to configure the VLAN, with intervlan routing
enabled:
Slot-1_f140(config-vlan)#vlan 23 bridge 1 name VLAN23 state enable
intervlan-route enable
```

- To associate the VLAN 23 (mentioned in the above sample) to a switch port:

```
root@Slot-1_f140:/opt/srstackware/bin> ./imish
ZebOS version 7.6.1 IPIRouter 11/09/10 07:31:33
Slot-1_f140>enable
Slot-1_f140#
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#interface gel7
Slot-1_f140(config-if)#switchport mode access
Slot-1_f140(config-if)#switchport access vlan 23
```

- To enable STP on Base Port:

Bridge level configuration

- Create a bridge, if required. (By default, the bridges are created as a part of the default configuration.) In case, the interface has to be associated to the VLAN, tag it with `vlan-bridge` (as mentioned in the code below), while creating the bridge.
- Enable the L2 protocol, such as STP, RSTP, or MSTP on the bridge.

Interface level configuration

- Associate the interface with the bridge, it automatically enables the corresponding protocol of the bridge.
- Associate the interface to the VLAN.

```
root@Slot-1_f140:/opt/srstackware/bin> ./imish
ZebOS version 7.6.1 IPIRouter 11/09/10 07:31:33
Slot-1_f140>enable
```

```
Slot-1_f140#
Slot-1_f140#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.

/** Associate STP to bridge-1 */
Slot-1_f140(config)#bridge 1 protocol ieee vlan-bridge

/* Go to interface mode */
Slot-1_f140(config)#interface ge21

/* Remove bridge so that STP can be enabled */
Slot-1_f140(config-if)# no bridge-group 1

/* Enables STP on the interface */
Slot-1_f140(config-if)# bridge-group 1

/* Re-configure the necessary VLANs; because VLANs disappear if bridge is
removed */
Slot-1_f140(config-if)# switchport mode hybrid
Slot-1_f140(config-if)# switchport hybrid vlan 93
Slot-1_f140(config-if)# switchport mode hybrid acceptable-frame-
type all
Slot-1_f140(config-if)# switchport hybrid allowed vlan add 93
egress-tagged disable

/* Enable the interface */
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
```

- To enable RSTP on the base RTM port 1:
Configuring RSTP is similar to configuring STP as mentioned in the above section, *To enable STP on the base RTM port 1*. To configure RSTP replace *ieee* with *rstp*, as in the following code.

```
/* Associate RSTP to bridge-1 */
Slot-1_f140(config)# bridge 1 protocol rstp vlan-bridge
```

- To enable MSTP on the fabric RTM port 1:

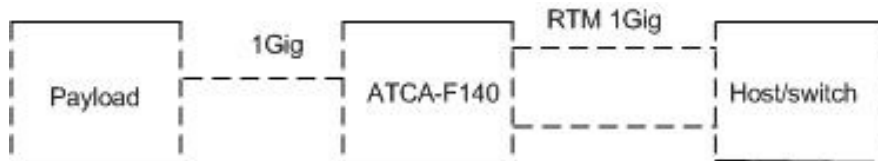
```

root@Slot-1_f140:/opt/srstackware/bin> ./imish
ZebOS version 7.6.1 IPIRouter 11/09/10 07:31:33
Slot-1_f140>enable
Slot-1_f140#
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#bridge 2 protocol mstp
Slot-1_f140(config)#interface xe21
Slot-1_f140(config-if) # bridge-group 2
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#

```

- LACP related configuration

Enable LACP on two base RTM ports as per the following topology.



Only bandwidth, duplex value, vlan, and bridge configuration are propagated to the aggregator port from the first member port. All other desired configuration should be done on the aggregator port (po#/sa#) itself.

- Following are the assumptions while configuring channel-group or static-channel-group on a set of ports in different modes:
 - LACP (channel-group) in auto-negotiation mode

When the first port is up while adding to aggregator, the aggregator assumes that these are the master port properties. While adding second port onwards to the aggregator, the bandwidth and duplex have to be same as the first port added to aggregator. This requires that the ports have to be linked up and auto-negotiated properly to match the first port values.

To avoid the above situation, all the ports should be added to aggregator in shutdown state. After adding all the ports, bring up the aggregator port using no shutdown CLI.

- LACP (channel-group) in manually configured duplex mode
All the member ports should match the first port's manual configuration.
- Static-channel-group in manually configured duplex mode or in auto-negotiation mode
All the member ports should be administratively up and in operationally running state and all the member ports should match the first port's configuration.
The following example explains how to configure VLAN 93 in the trunk mode for the interfaces `ge21` and `ge22`, aggregate them, and configure source IP based load balancing on the aggregator.

This example assumes that VLAN 93 is configured on the interface connected to the payload. Note that the payload side interface settings are not mentioned here.

```
Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#interface ge21
Slot-1_f140(config-if)#switchport mode trunk
Slot-1_f140(config-if)#switchport trunk allowed vlan add 93
Slot-1_f140(config-if)#switchport trunk native vlan 93
Slot-1_f140(config-if)#channel-group 2 mode active
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface ge22
Slot-1_f140(config-if)#switchport mode trunk
Slot-1_f140(config-if)#switchport trunk allowed vlan add 93
Slot-1_f140(config-if)#switchport trunk native vlan 93
Slot-1_f140(config-if)#channel-group 2 mode active
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface po2
Slot-1_f140(config-if)#port-channel load-balance src-ip
Slot-1_f140(config-if)#
```

In the above code, interface `po2` is the aggregated interface of `ge21` and `ge22`. You can use multiple traffic streams to test the load balancing.



The configuration of normal or enhanced load balanced algorithm can be performed only at chipset level.



Save the aggregator configuration only when the aggregator is functionally up, in sync and running. This is very important to have the configuration to be persistent.

- To set the IP address to an interface:


```
Slot-1_f140(config)#interface ge21
Slot-1_f140 (config-if)# no switchport
Slot-1_f140(config-if)#ip address 11.11.11.1/24
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface ge22
Slot-1_f140 (config-if)# no switchport
Slot-1_f140(config-if)#ip address 12.12.12.1/24
Slot-1_f140(config-if)#exit
```
- QoS related configuration:

To enable QoS with strict priority:

```
Slot-1_f140>
Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#mls qos 0 0 0 1 0 2 0 3 0 4 0 5 0 6 0 7
```

To assign the value 7 to DSCP in the IP header, for source IP address 11.11.11.11 going through ge17.

```
Slot-1_f140(config)#ip-access-list 1 permit 11.11.11.11
0.0.0.255
Slot-1_f140(config)#class-map cmap1
Slot-1_f140(config-cmap)#match access-group 1
Slot-1_f140(config-cmap)#exit
Slot-1_f140(config)#policy-map pmap1
Slot-1_f140(config-pmap)#class cmap1
Slot-1_f140(config-pmap-c)#set ip-dscp 7
Slot-1_f140(config-pmap-c)#exit
Slot-1_f140(config-pmap)#exit
```

```
Slot-1_f140(config)#interface ge21
Slot-1_f140(config-if)#service-policy input pmap1
Slot-1_f140(config-if)#exit
```

To block the packets with source IP address 11.11.11.11 going through ge21:

```
Slot-1_f140(config)#ip-access-list 1 deny 11.11.11.11 0.0.0.255
Slot-1_f140(config)#class-map cmap1
Slot-1_f140(config-cmap)#match access-group 1
Slot-1_f140(config-cmap)#exit
Slot-1_f140(config)#policy-map pmap1
Slot-1_f140(config-pmap)#class cmap1
Slot-1_f140(config-pmap-c)#exit
Slot-1_f140(config-pmap)#exit
Slot-1_f140(config)#interface ge21
Slot-1_f140(config-if)#service-policy input pmap1
Slot-1_f140(config-if)#exit
```

To set the COS value for an ingress packet:

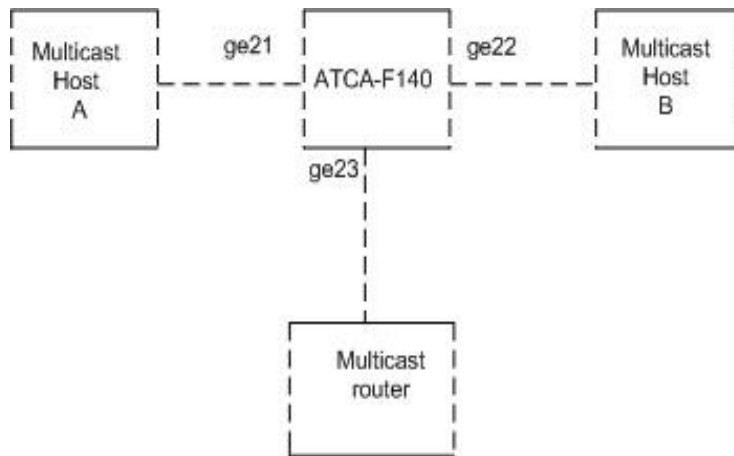
```
SLOT2-f140>
SLOT2-f140>enable
SLOT2-f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SLOT2-f140(config)#bridge 1 protocol ieee vlan-bridge
SLOT2-f140(config)#vlan database
SLOT2-f140(config-vlan)#vlan 2 bridge 1 state enable
SLOT2-f140(config-vlan)#exit
SLOT2-f140(config)#interface ge21
SLOT2-f140(config-if)#shutdown
SLOT2-f140(config-if)#bridge-group 1 spanning-tree disable
SLOT2-f140(config-if)#switchport mode trunk
SLOT2-f140(config-if)#switchport trunk allowed vlan add 2
SLOT2-f140(config-if)#no shutdown
SLOT2-f140(config-if)#exit
SLOT2-f140(config)#mls qos 0 0 0 1 0 2 0 3 0 4 0 5 0 6 0 7
SLOT2-f140(config)#class-map cmap1
SLOT2-f140(config-cmap)#match vlan 2
```

```
SLOT2-f140(config-cmap)#exit
SLOT2-f140(config)#policy-map pmap1
SLOT2-f140(config-pmap)#class cmap1
SLOT2-f140(config-pmap-c)#set cos 6
SLOT2-f140(config-pmap-c)#exit
SLOT2-f140(config-pmap)#exit
SLOT2-f140(config)#interface ge21
SLOT2-f140(config-if)#service-policy input pmap1
SLOT2-f140(config-if)#
```

To block the packets with source MAC address 00:04:5A:46:98:FF going through ge21.

```
Slot-1_f140(config)#mac-access-list 2002 deny 0004.5a46.98ff
0000.0000.ffff any 2
Slot-1_f140(config)#class-map cmap1
Slot-1_f140(config-cmap)#match access-group 2002
Slot-1_f140(config-cmap)#exit
Slot-1_f140(config)#policy-map pmap1
Slot-1_f140(config-pmap)#class cmap1
Slot-1_f140(config-pmap-c)#exit
Slot-1_f140(config-pmap)#exit
Slot-1_f140(config)#interface ge21
Slot-1_f140(config-if)#service-policy input pmap1
Slot-1_f140(config-if)#exit
```

- IGMP related configuration:
Enable IGMP snooping on two base RTM ports using the following topology.



To enable IGMP snooping on the `ge21` and `ge22` interfaces:

```
Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#bridge 1 protocol ieee vlan-bridge
Slot-1_f140(config)#interface ge21
Slot-1_f140(config-if)#shutdown
Slot-1_f140(config-if)#bridge-group 1
Slot-1_f140(config-if)#switchport mode access
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface ge22
Slot-1_f140(config-if)#shutdown
Slot-1_f140(config-if)#bridge-group 1
Slot-1_f140(config-if)#switchport mode access
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface ge23
Slot-1_f140(config-if)#shutdown
Slot-1_f140(config-if)#bridge-group 1
```

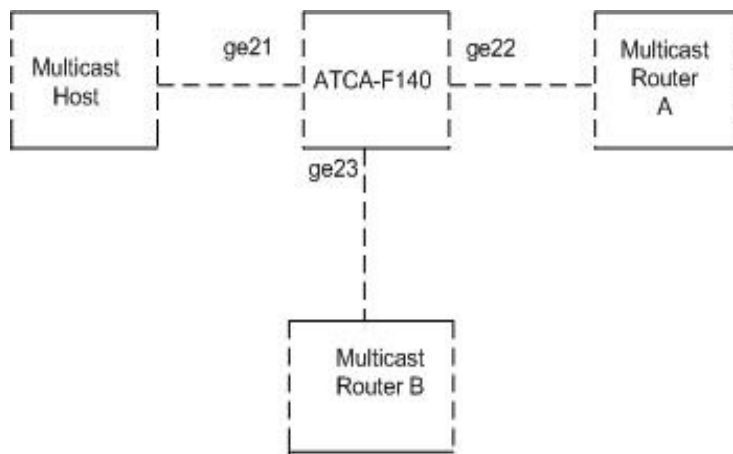


```

Slot-1_f140(config-if)#switchport mode access
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#ip igmp snooping
Slot-1_f140(config)#interface vlan1.1
Slot-1_f140(config-if)#ip igmp snooping mrouter interface ge23

```

To enable IGMP proxy on the base RTM ports with the following topology.



To enable IGMP proxy on ge21, ge22, and ge23 interfaces.

```

Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#interface ge21
Slot-1_f140(config-if)#shutdown
Slot-1_f140(config-if)#no switchport
Slot-1_f140(config-if)#ip address 192.168.3.1/24
Slot-1_f140(config-if)#ip igmp proxy-service
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface ge22
Slot-1_f140(config-if)#shutdown
Slot-1_f140(config-if)#no switchport

```

```
Slot-1_f140(config-if)#ip address 192.168.4.1/24
Slot-1_f140(config-if)#ip igmp mroute-proxy ge21
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface ge23
Slot-1_f140(config-if)#shutdown
Slot-1_f140(config-if)#no switchport
Slot-1_f140(config-if)#ip address 192.168.5.1/24
Slot-1_f140(config-if)#ip igmp mroute-proxy ge21
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
```

To enable static routing within the following topology.



On the payload blade:

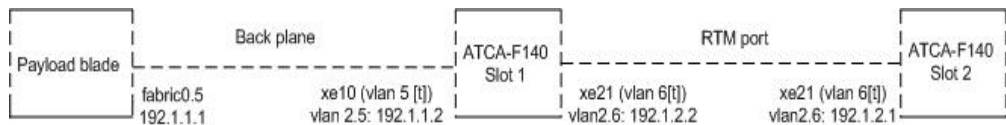
```
ifconfig fabric0 192.1.1.1/24
route add -net 192.1.2.0/24 gw 192.1.1.2 dev fabric0.5
Slot-1_f140>en
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#interface xe10
Slot-1_f140(config-if)#no switchport
Slot-1_f140(config-if)#ip address 192.1.1.2/24
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface xe21
Slot-1_f140(config-if)#no switchport
Slot-1_f140(config-if)#ip address 192.1.2.2/24
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#ip forwarding
Slot-2_f140>en
```

```

Slot-2_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-2_f140(config)#interface xe21
Slot-2_f140(config-if)#no switchport
Slot-2_f140(config-if)#ip address 192.1.2.1/24
Slot-1_f140(config-if)#no shutdown
Slot-2_f140(config-if)#exit
Slot-2_f140(config)#ip route 192.1.1.0/24 192.1.2.2

```

To enable static routing through VLAN interfaces within the following topology.



On the payload blade:

```

vconfig add fabric0 5
ifconfig fabric0.5 192.1.1.1/24
route add -net 192.1.2.0/24 gw 192.1.1.2 dev fabric0.5

```

On Slot 7:

Create VLAN 5 and 6, with intervlan routing enabled, and associate it to Bridge 2.

```

/** If static routing through VLANs has to be done through ports
of two fabric chip sets, then these VLANs should be added on hi-
gig using "vlan <vlan-id> hi-gig". */

```

```

Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#interface xe10
Slot-1_f140(config-if)#bridge-group 2 spanning-tree disable
Slot-1_f140(config-if)#switchport mode trunk
Slot-1_f140(config-if)#switchport trunk allowed vlan add 5
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface vlan2.5
Slot-1_f140(config-if)#ip address 192.1.1.2/24
Slot-1_f140(config-if)#no shutdown

```

```
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface xe21
Slot-1_f140(config-if)#bridge-group 2 spanning-tree disable
Slot-1_f140(config-if)#switchport mode trunk
Slot-1_f140(config-if)#switchport trunk allowed vlan add 6
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface vlan2.6
Slot-1_f140(config-if)#ip address 192.1.2.2/24
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#ip forwarding
```

On Slot 8:

Create VLAN 6, with intervlan routing enabled, and associate it to Bridge 2.

```
Slot-2_f140>enable
Slot-2_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-2_f140(config)#interface xe21
Slot-2_f140(config-if)#bridge-group 2 spanning-tree disable
Slot-2_f140(config-if)#switchport mode trunk
Slot-2_f140(config-if)#switchport trunk allowed vlan add 6
Slot-2_f140(config-if)#no shutdown
Slot-2_f140(config-if)#exit
Slot-2_f140(config)#interface vlan2.6
Slot-2_f140(config-if)#ip address 192.1.2.1/24
Slot-1_f140(config-if)#no shutdown
Slot-2_f140(config-if)#exit
Slot-2_f140(config)#ip route 192.1.1.0/24 192.1.2.2
```

To enable Q-in-Q (VLAN stacking) on the ATCA-F140 with the following topology.

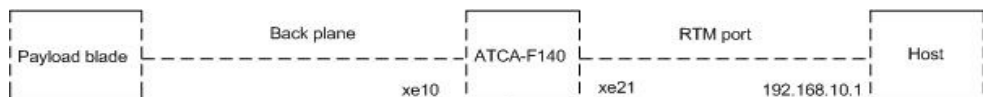


```

Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#bridge 2 protocol ieee vlan-bridge
Slot-1_f140(config)#vlan database
Slot-1_f140(config-vlan)#vlan 5 bridge 2 state enable
Slot-1_f140(config-vlan)#exit
Slot-1_f140(config)#interface xe21
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#bridge-group 2
Slot-1_f140(config-if)#switchport mode access
Slot-1_f140(config-if)#switchport vlan-stacking customer-edge-
port
Slot-1_f140(config-if)#switchport access vlan 5
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface xe22
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#bridge-group 2
Slot-1_f140(config-if)#switchport mode trunk
Slot-1_f140(config-if)#switchport mode trunk allowed vlan add 5
Slot-1_f140(config-if)#switchport vlan-stacking provider-port
Slot-1_f140(config-if)#exit

```

To enable VLAN Classification based on Source IP network with the following topology.



```

Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#bridge 2 protocol ieee vlan-bridge
Slot-1_f140(config)#vlan database
Slot-1_f140(config-vlan)#vlan 5 bridge 2 state enable

```

```
Slot-1_f140(config-vlan)#exit
Slot-1_f140(config)# vlan classifier rule 2 ipv4 192.168.10.0/24
vlan 5
Slot-1_f140(config)# vlan classifier group 10 add rule 2
Slot-1_f140(config)#interface xe10
Slot-1_f140(config-if)#bridge-group 2
Slot-1_f140(config-if)#switchport mode trunk
Slot-1_f140(config-if)#switchport trunk allowed vlan add 5
Slot-1_f140(config-if)# vlan classifier rule 2 ipv4
192.168.10.0/24 vlan 5
Slot-1_f140(config-if)#no shutdown
```

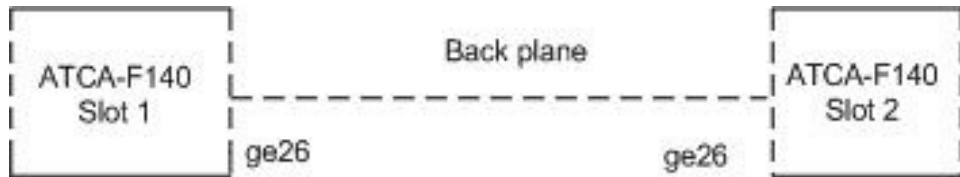
- OSPF related configuration:
Enable OSPF on the two ATCA-F140s using the following topology.



```
Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#interface ge21
Slot-1_f140(config-if)#no switchport
Slot-1_f140(config-if)#ip address 192.1.1.1/24
Slot-1_f140(config-if)#exit
Slot-1-f140(config)#router ospf 1
Slot-1-f140(config-router)#network 192.1.1.0/24 area 1.1.1.1
Slot-2_f140>enable
Slot-2_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-2_f140(config)#interface ge21
Slot-2_f140(config-if)#no switchport
```

```
Slot-2_f140(config-if)#ip address 192.1.1.2/24
Slot-2_f140(config-if)#exit
Slot-2-f140(config)#router ospf 1
Slot-2-f140(config-router)#network 192.1.1.0/24 area 1.1.1.1
```

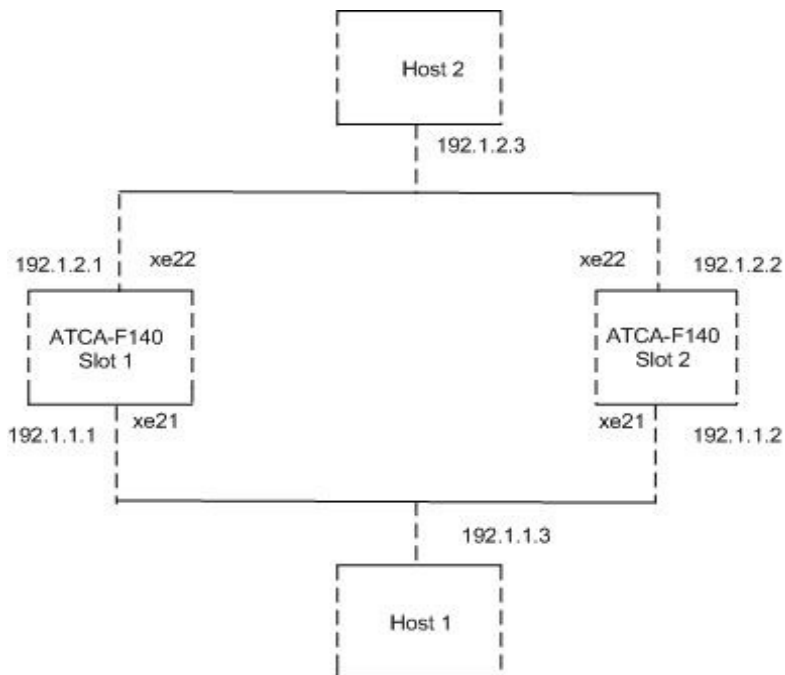
- RIP related configuration:
To enable RIP on the two ATCA-F140s with the following topology.



```
Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#interface ge26
Slot-1_f140(config-if)#no switchport
Slot-1_f140(config-if)#ip address 192.1.1.1/24
Slot-1_f140(config-if)#exit
Slot-1-f140(config)#router rip
Slot-1-f140(config-router)#network 192.1.1.0/24
Slot-2_f140>enable
Slot-2_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-2_f140(config)#interface ge26
Slot-2_f140(config-if)#no switchport
Slot-2_f140(config-if)#ip address 192.1.1.2/24
Slot-2_f140(config-if)#exit
Slot-2-f140(config)#router rip
Slot-2-f140(config-router)#network 192.1.1.0/24
```

SRstackware

- VRRP related configuration
To enable VRRP on the following topology.



On Host 1:

```
ifconfig eth0 192.1.1.3/24
route add -net 192.1.2.0/24 gw 192.1.1.1 dev eth0
```

On Host 2:

```
ifconfig eth0 192.1.2.3/24
route add -net 192.1.1.0/24 gw 192.1.2.1 dev eth0
```

On ATCA-F140 Slot 7:

```
Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#interface xe21
Slot-1_f140(config-if)#no switchport
Slot-1_f140(config-if)#ip address 192.1.1.1/24
```



```
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#interface xe22
Slot-1_f140(config-if)#no switchport
Slot-1_f140(config-if)#ip address 192.1.2.1/24
Slot-1_f140(config-if)#no shutdown
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#router vrrp 1 xe21
Slot-1_f140(config-router)# virtual-ip 192.1.1.1 master
Slot-1_f140(config-router)# enable
Slot-1_f140(config-router)# exit
Slot-1_f140(config)#ip forwarding
```

On ATCA-F140 Slot 8:

```
Slot-2_f140>enable
Slot-2_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-2_f140(config)#interface xe21
Slot-2_f140(config-if)#no switchport
Slot-2_f140(config-if)#ip address 192.1.1.2/24
Slot-2_f140(config-if)#no shutdown
Slot-2_f140(config-if)#exit
Slot-2_f140(config)#interface xe22
Slot-2_f140(config-if)#no switchport
Slot-2_f140(config-if)#ip address 192.1.2.2/24
Slot-2_f140(config-if)#no shutdown
Slot-2_f140(config-if)#exit
Slot-2_f140(config)#router vrrp 1 xe21
Slot-2_f140(config-router)# virtual-ip 192.1.1.1 backup
Slot-2_f140(config-router)# enable
Slot-2_f140(config-router)# exit
Slot-2_f140(config)#ip forwarding
```

With the above configuration:

- the xe21 interface of ATCA-F140 Slot 7 should be elected as master.

- the xe21 interface of ATCA-F140 Slot 8 should recognize itself as backup.
- Host 1 should be able to ping Host 2 through xe21, xe22 of ATCA-F140 Slot 7.

You can shut down xe21 of ATCA-F140 Slot 7 to verify whether the VRRP backup router is taking up the role of Master, when the actual is down.

```
Slot-1_f140(config)#interface xe21
Slot-1_f140(config-if)#shutdown
Slot-1_f140(config-if)#exit
```

Change the route to network 192.1.1.0/24 at Host 2, to go via 192.1.2.2 gateway, instead of 192.1.2.1 gateway.

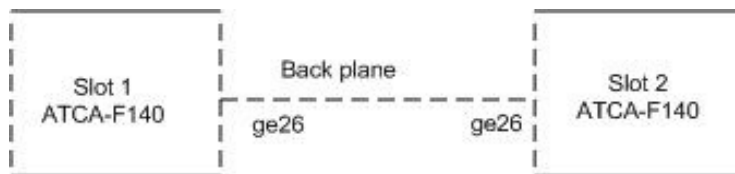
```
route add -net 192.1.1.0/24 gw 192.1.2.2 dev eth0
```

With this configuration:

- the link of xe21 on ATCA-F140 Slot 7, should be down.
- the xe21 interface of ATCA-F140 Slot 8, should elect itself as Master now.
- Host 1 should be able to ping Host 2 through xe21, xe22 of ATCA-F140 Slot 8.

- RIPng related configuration:

To enable RIPng on the two ATCA-F140s with the following topology:



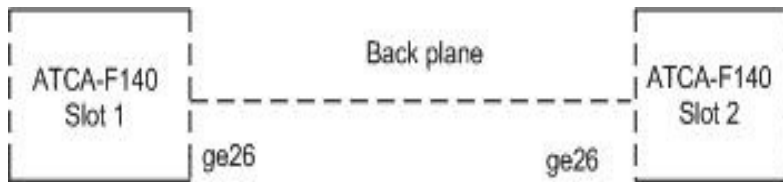
```
Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#interface ge26
Slot-1_f140(config-if)#no switchport
Slot-1_f140(config-if)#ipv6 address fec0:0:0:c400::21/54
Slot-1_f140(config-if)#ipv6 router rip
Slot-1_f140(config-if)#exit
Slot-1_f140(config)#router ipv6 rip
Slot-2_f140>enable
```

```

Slot-2_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-2_f140(config)#interface ge26
Slot-2_f140(config-if)#no switchport
Slot-2_f140(config-if)#ipv6 address fec0:0:0:c400::22/54
Slot-2_f140(config-if)#ipv6 router rip
Slot-2_f140(config-if)#exit
Slot-2_f140(config)#router ipv6 rip

```

- BFD related configuration:
To enable BFD on the two ATCA-F140s with the following topology:



```

Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140(config)#interface ge26
Slot-1_f140(config-if)#no switchport
Slot-1_f140(config-if)#ip address 192.1.1.1/24
Slot-1_f140(config-if)#bfd interval 1000 minRx 1000 multiplier 3
sipv4 192.1.1.1 dipv4 192.1.1.2
Slot-2_f140>enable
Slot-2_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-2_f140(config)#interface ge26
Slot-2_f140(config-if)#no switchport
Slot-2_f140(config-if)#ip address 192.1.1.2/24
Slot-2_f140(config-if)#bfd interval 1000 minRx 1000 multiplier 3
sipv4 192.1.1.2 dipv4 192.1.1.1
Slot-2_f140(config-if)#end
Slot-2_f140#bfd show

```

11.5 SNMP Usage Guidelines

For Bridge-based MIBs, the context parameter of the `snmp` commands (`snmpget/snmpset/snmpwalk`) is used to specify the `bridge_number` and the details of the corresponding bridge are processed.



The string `Bridge` to be specified with the context parameter is case-sensitive.

The exact string to be used in the command can be obtained by doing a `snmpwalk` on the object `"srsBridgeName"` which is a proprietary mib-object provided by SRstackware.

In case of F140, "Bridge1" represents Base switch and "Bridge2" represents Fabric switch.

Example of an `snmp` command:

```
snmpget -v3 -u admin -n "Bridge1" -l noAuthNoPriv -a MD5 -A adminpwd123 localhost dot1dBaseBridgeAddress.0
```

Here, the option 'n' is used to specify the context. The above command will return the MAC address used by this bridge when it must be referred to in a unique fashion.

For the mibs that do not distinguish based on the bridge, needs to be run with null context (that is `-n ""`) or the context option can be entirely removed from the `snmp` command.

- BRIDGE MIB
- P-BRIDGE MIB
- Q-BRIDGE MIB
- RSTP MIB

The following are the mibs, currently supported, that do not need any context to be specified:

- OSPF MIB
- RIP MIB
- IF MIB
- IEEE8023-LAG-MIB
- SRS MIB (Penguin Edge Proprietary MIB)

11.6 SNMP Traps Usage Guidelines

SNMP traps and SNMP Inform requests are the two SNMP notifications supported with SRstackware package. The following sections detail the configurations to be made at SNMP agent and manager in order to facilitate these notifications.

11.6.1 Configuring Agent to Send SNMP Notifications

Description

In order to configure SRstackware to send SNMP traps and Inform requests, a shell command has been provided as part of SRstackware package.

This shell command has the following syntax.

Synopsis

```
#srs_trap_config <IPv4 Address> <NONE/TRAP/INFORM>
```

Parameters

NONE

Neither SNMP traps nor Inform requests are sent to the SNMP manager running on the machine with IPv4 Address.

TRAP

Trap messages are sent to SNMP manager running on the machine with IPv4 Address.

INFORM

SNMP Inform requests are sent to SNMP manager running on the machine with IPv4 Address.

Example

In order to send traps to a system with IP address 10.10.10.10, run the command at the agent as follows:

```
#srs_trap_config 10.10.10.10 TRAP
```

Please note that the parameters `NONE`, `TRAP`, and `INFORM` are case sensitive.

11.6.2 Configuring SNMP Manager to Receive SNMP Notifications

Description

The `snmptrapd` daemon needs to be run at the receiver to receive the SNMP notifications sent by the agent. Also `snmptrapd` needs a configuration file with proper configuration in order to receive the traps properly. The `snmptrapd` configuration file must have the below syntax.

Synopsis

```
createUser -e <Agent's Engine-ID> admin MD5 adminpwd123 DES
authUser log,execute admin
```

Parameters

Agent's Engine-ID

The Engine-ID with which SNMP Agent is running.

To obtain the Agent's Engine-ID, type the following command:

```
snmpget -v3 -u admin -A adminpwd123 <Agent's IP address>
snmpEngineID.0
```

The result of this command will be in hexadecimal format and needs to be passed as a single string in the configuration file.

Example

```
snmpget -v3 -u admin -A adminpwd123 10.10.10.20 snmpEngineID.0
```

The output for the above example will be similar to as follows:

```
SNMP-FRAMEWORK-MIB::snmpEngineID.0 = Hex-STRING: 80 00 1F 88 04 65 6D 65
72 73 6F 6E
```

Finally, the `snmptrapd` configuration file will look as follows:

```
createUser -e 0x80001F8804656D6572736F6E admin MD5 adminpwd123 DES
authUser log,execute admin
```

The `snmptrapd` daemon may be run at the receiver (SNMP manager) as follows:

```
/usr/sbin/snmptrapd -x tcp:localhost:705 -c
"<snmptrapd_configuration_file>" -C -f -Le
```



The formatting options can be modified as desired by the user, please refer the man pages of `snmptrapd` for more details.

11.7 Configuring Logs

You can configure SRstackware to store the logs in a specified file or `syslog`, using the `log` command. As a general practice the logs are stored in `/var/log/messages`. As ATCA-F140 has limited flash space, it is recommended to configure ATCA-F140 to store logs on a remote syslog server.

```
Slot-1_f140>enable
Slot-1_f140#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Slot-1_f140#log file log_file1
```

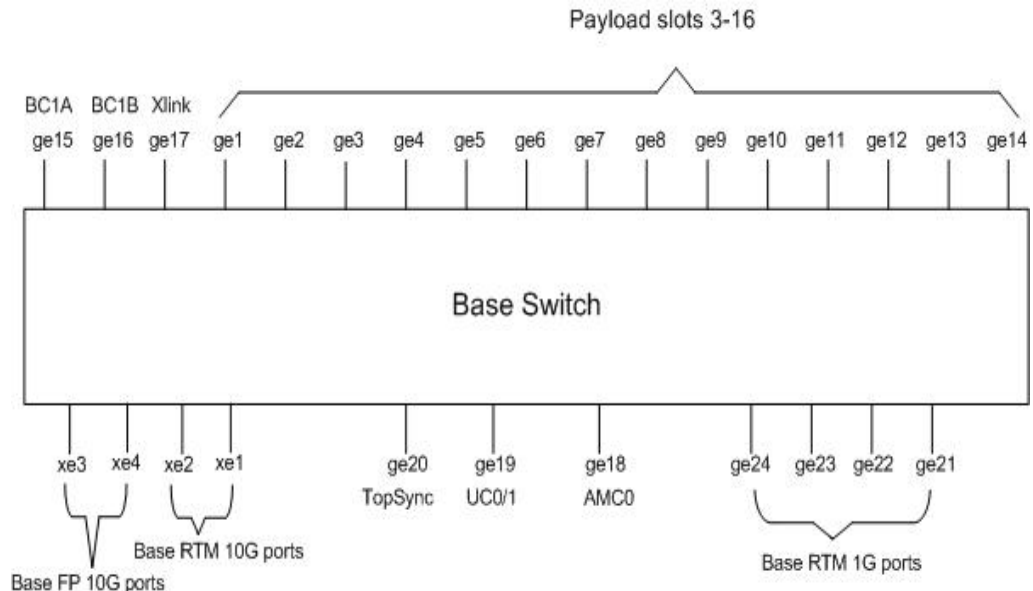
By default, logs are stored in `var/log/srstackware.log` file.

11.8 Switch Ports

11.8.1 Base Switch Port

The following figure illustrates the mapping of payload slots to the corresponding base switch ports.

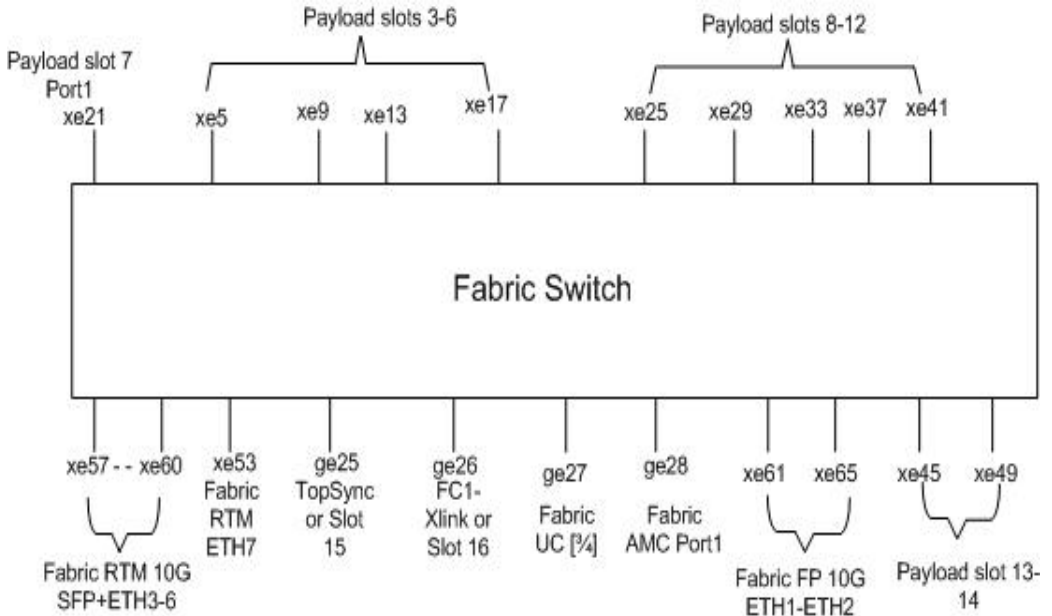
Figure 11-4 Base Switch Port Mapping



11.8.2 Fabric Switch Port

The following figure illustrates the mapping of payload slots to the corresponding fabric switch ports.

Figure 11-5 Fabric Switch Port Mapping



HPI-B Software

To help implementation of highly available systems with off-the-shelf building blocks, the Service Availability Forum (SA Forum) Hardware Platform Interface (HPI) specification HPI-B defines a set of platform-independent programming interfaces to monitor and control systems, such as AdvancedTCA systems, designed to provide high availability. HPI provides applications and middleware a consistent, standardized interface for managing hardware components.

This BBS release contains an HPI-B library package. For more information on Penguin Solutions's HPI-B implementation, refer to *System Management Interface Based on HPI-B User's Guide*.

SYSSTATMGR

The SYSSTATMGR software package includes a daemon and a number of system status monitors. The daemon provides the infrastructures to start the monitors, log the status reported by the monitors to the syslog as well as the IPMI event system, and perform the pre-configured actions for corresponding status.

SYSTAT Sensor

Table 13-1 *Penguin Edge OEM Sensor Type Codes*

Sensor Type	Sensor Type Code	Sensor-specific Offset	Event Data 2	Event Data 3
SYSSTAT	CEh	<p>No Sensor specific offsets. Instead, use the standard severity event states offsets. That is, the sensor Event / Reading Type code is 07h. For more details, refer to IPMI V1.5.</p> <ul style="list-style-type: none"> ● 00h Transition to OK. ● 01h Transition to Non-Critical from OK. ● 02h Transition to Critical from less severe. ● 03h Transition to Non-Recoverable from less severe. ● 04h Transition to Non-Critical from more severe. ● 05h Transition to Critical from Non-Recoverable. ● 06h Transition to Non-Recoverable. Not used. ● 07h Monitor. Indicating the board is under recovering, and monitoring is required to get updated status. ● 08h Informational. 	<p>Monitor Identifier. Configuration defined. FFh if not used.</p>	<p>Status Modifier: This byte provides extra information to help diagnostic. Monitor implementation specific. FFh if not used. FFh if not used or unspecified error. FEh reserved, time out of the monitor. FDh reserved, unknown monitor, if Event Data 2 is FFh.</p>

13.1 Installation

This section describes installation of SYSSTAT Manager on ATCA-F140.

RPM	Description
<code>rpm -i sysstatmgr-atca-f140-0.0.0-1-windriver.rpm</code>	SYSSTAT Manager RPM for installing on ATCA-F140.

- To install SYSSTAT Manager, run the following command:
`rpm -i sysstatmgr-atca-f140-0.0.0-1-windriver.rpm`

After installation, the directory structure listed in the following table is created at `/opt/sysstatmgr`.

Table 13-2 Directory Structure of SYSSTAT Manager

Folder	Description	Files
<code>/opt/sysstatmgr/bin</code>	Contains executable files.	<code>sysstatmgr/bin:</code> <code>sysstatagentd sysstatcmd</code>
<code>/opt/sysstatmgr/etc</code>	Contains configuration files of <code>sysstatmgr</code> .	<code>sysstatmgr/etc:</code> <code>sysstat.conf</code> <code>sysstat_factory.conf</code>
<code>/opt/sysstatmgr/lib</code>	Contains library files	<code>sysstatmgr/lib:</code> <code>libssmapi.a</code>
<code>/opt/sysstatmgr/monitors</code>	Contains scripts/configurations of monitors.	<code>sysstatmgr/monitors:</code> <code>cpuMonitor.conf</code> <code>fcuMonitor.sh</code> <code>memMonitor.conf</code> <code>tclkMonitor.conf</code> <code>cpuMonitor.sh</code> <code>fpgaMonitor.conf</code> <code>memMonitor.sh</code> <code>tclkMonitor.sh</code> <code>dfMonitor.conf</code> <code>fpgaMonitor.sh</code> <code>monitors_config.sh</code> <code>dfMonitor.sh</code> <code>hpmMonitor.conf</code> <code>srsMonitor.conf</code> <code>fcuMonitor.conf</code> <code>hpmMonitor.sh</code> <code>srsMonitor.sh</code>

- To start the daemon, run the following command:
`/opt/sysstatmgr/bin/sysstatagentd`
- To get the status of all monitors, run the following command:
`/opt/sysstatmgr/bin/sysstatcmd list`

13.2 Monitors

The SYSSTAT Manager provides following monitoring commands to check health of the ATCA-F140 blade and its components.

- `cpuMonitor`
- `memMonitor`
- `dfMonitor`
- `fcuMonitor`
- `hpmMonitor`
- `fpgaMonitor`
- `srsMonitor`
- `tclkMonitor`

NOTICE

All these monitoring scripts/commands are located at `/opt/sysstatmgr/monitors`.

13.2.1 cpuMonitor

`cpuMonitor` scripts monitor the percentage of the idle CPU using the `top` command for every 10 minutes. Following are its threshold values defined in

`/opt/sysstatmgr/monitors/cpuMonitor.sh`

- `HIGH_CPU_WARNING` (idle <10%)
- `HIGH_CPU_CRITICAL` (idle <5%)
- `HIGH_CPU_ABS_HIGH` (idle <2%)

13.2.2 memMonitor

`memMonitor` scripts monitor the amount of free memory space reported by `/proc/meminfo` for every 5 minutes. Following are its threshold values defined in

`/opt/sysstatmgr/monitors/memMonitor.sh`

- LOW_MEM_WARNING (MemFree < 32MB)
- LOW_MEM_CRITICAL (MemFree < 8MB)
- LOW_MEM_ABS_LOW (MemFree < 2MB)

13.2.3 dfMonitor

dfMonitor scripts monitor the available spaces in percentage from rootfs for every 10 minutes. Following are its threshold values defined in

`/opt/sysstatmgr/monitors/dfMonitor.sh`

- DISK_SPACE_WARNING (free space < 10%)
- DISK_SPACE_CRITICAL (free space < 5%)
- DISK_SPACE_ABS_LOW (free space < 2%)

Devices to be monitored are listed in

`/opt/sysstatmgr/monitors/monitors_config.sh`.

13.2.4 fcuMonitor

fcuMonitor scripts verify whether FCU queries each device successfully for every 4 hours.

- FCU_NOT_FOUND (“fcu” utility not found)
- FCU_CPU_ERROR (query “atca-f140-cpu” failed)
- FCU_DTB_ERROR (query “atca-f140-dtb” failed)
- FCU_KERNEL_ERROR (query “atca-f140-kernel” failed)
- FCU_IPMC_ERROR (query “atca-f140-hpm.1-ipmc” failed)

13.2.5 hpmMonitor

hpmMonitor scripts verify the hpmagentd and hpmcmd for every 4 hours.

- HPMAGENTD_NO_PS (“hpmagentd” daemon not running)
- HPMAGENTD_NO_BIN (“hpmagentd” file not found)
- HPMAGENTD_NO_UTIL (“hpmcmd” utility not found)
- HPMAGENTD_OOS (IPMC not respond to “hpmcmd”)

13.2.6 fpgaMonitor

fpgaMonitor scripts verify FPGA access and version for every 4 hours.

- FPGA_TOOL_NOT_FOUND (“fpga_upg” utility not found)

- FPGA_QUERY_ERROR (failed to read FPGA version)
- FPGA_VERSION_ERROR (unexpected FPGA version)
- FPGA_RTM_QUERY_ERROR (failed to read RTM FPGA version)
- FPGA_RTM_VERSION_ERROR (unexpected RTM FPGA version)

The FPGA versions are defined in `/opt/sysstatmgr/monitors/monitors_config.sh`

13.2.7 srsMonitor

`srsMonitor` scripts monitor a set of ATCA-F140 SRS utility for every 1 hour.

- SRS_NSM_NOT_FOUND (“nsm” utility not found)
- SRS_NSM_NO_PS (“nsm” process not found)
- SRS_MSTPD_NOT_FOUND (“mstpd” utility not found)
- SRS_MSTPD_NO_PS (“mstpd” process not found)
- SRS_IMI_NOT_FOUND (“imi” utility not found)
- SRS_IMI_NO_PS (“imi” process not found)
- SRS_LACPD_NOT_FOUND (“lacpd” utility not found)
- SRS_LACPD_NO_PS (“lacpd” process not found)
- SRS OSPFD_NOT_FOUND (“ospfd” utility not found)
- SRS OSPFD_NO_PS (“ospfd” process not found)
- SRS_RIPD_NOT_FOUND (“ripd” utility not found)
- SRS_RIPD_NO_PS (“ripd” process not found)
- SRS_RIPNGD_NOT_FOUND (“ripngd” utility not found)
- SRS_RIPNGD_NO_PS (“ripngd” process not found)

13.2.8 tclkMonitor

`tclkMonitor` scripts verifies the tclk agent and tclk firmware version for every 4 hours.

- SNMPD_NO_SNMPGET (“snmpget” utility not found)
- SNMPD_OFFLINE (“snmpd” process not found)
- SNMPD_TIMEOUT (“snmpget” timeout)
- TCLK_NOT_FOUND (“tclk_agent” utility not found)
- TCLK_NO_PS (“tclk_agent” process not found)
- TCLK_SNMPD_TIMEOUT (tclk fw query timeout)

SYSSTATMGR

- `TCLK_NOT_AVAIL` (tclk fw query not available)
- `TCLK_UNEXPECTED_VERSION` (unexpected tclk fw version)
- `TCLK_UNKNOWN_VERSION` (unrecognized tclk fw version)

The tclk firmware versions are defined in
`/opt/sysstatmgr/monitors/monitors_config.sh`

MUXes Settings

A.1 Overview

This section provides information about setting the different MUXes.

Table A-1 Mux Control Signals

Default Config	Signal Path		MUX# and Select Setting [B1: A<=>B1 B2: A<=>B2]						
	From	To	1	15	12	13	11	14	
default	FIX WC0-0	Topsync	B2						B2
	FIX WC0-1	Fabric channel 1 (Switch X-Connect)							
	FIX WC0-0	Fabric Interface logical slot 15	B1						
	FIX WC0-1	Fabric Interface logical slot 16							
default	HDD SATA	P2020 SATA		B1					
	HDD SATA	AMC Port 2		B2					
default	BIX_GE_P18	Base X Connect (UC0:1)			B1				
	BIX_GE_P18	Zone 3 connector			B2				
default	BIX_GE_P19	Topsync				B1		B1	
	BIX_GE_P19	Zone 3 connector				B2			
default	BIX_GE_P12	Base Interface logical slot 15					B1		
	BIX_GE_P13	Base Interface logical slot 16							
	BIX_GE_P12	Zone 3 connector					B2		
	BIX_GE_P13	Zone 3 connector							
			FIX_GEO_GE1_MUX_SEL	SATA_MUX_SEL	BIX_P19_MUX_SEL	BIX_P20_MUX_SEL	BIX_P13_P14_MUX_SEL	TOPSYNC_MUX_SEL	

MUXes Settings

To program a Mux for the B1 path, write 0 to the Ethernet Mux control register bit corresponding to the Mux control signal as shown in the [Table A-1 on page 217](#). The default state of the Ethernet Mux control register bits is 0, except for Mux 1, which has a default value of 1. To program the Mux for the B2 path, write 1 to the corresponding Ethernet Mux control register bit. Some signal paths require two Mux control bits to be programmed. For example, to change the Topsynd routing from the default base switch port BIX_GE_P19 to fabric switch port FIX_WC0-0, perform the following steps:

- Write 1 to FIX_GE0_GE1_MUX_SEL bit
- Write 1 to TOPSYNC_MUX_SEL bit

The Mux settings can be changed in Linux using the `hpmcmd` command. For example, to change the default direction of `SATA_MUX_SEL` (MUX #15) from the P2020 (B1) to the AMC (B2), use the following syntax:

```
hpmcmd -c muxctrl mux15 B2
```

The same Mux change can be accomplished in U-boot by writing 0x03 to FPGA register address 0x8:

```
mw.b ffd0008 03 1
```

For more information, see [FPGA MUX Control Registers on page 219](#).

For a Mux setting that is desired to be made on every power-up, `env` parameters can be set up in `uboot`. The following example shows `uboot` `env` parameter settings that can be used to set the SATA Mux to connect the on board SATA drive to the AMC and then `netboot` the ATCA-F140:

```
setenv ipaddr 192.168.1.105
setenv serverip 192.168.1.199
setenv bootargs 'root=/dev/ram0 rw ramdisk_size=1048576
console=ttyS0,9600 IMG='
setenv tftpkernel 'tftp 1000000 ATCA-F140/ATCAF140.kernel'
setenv tftpramdisk 'tftp 2000000 ATCA-F140/ATCAF140.initrd'
setenv tftpdtb 'tftp a00000 ATCA-F140/ATCAF140.dtb'
setenv amcsatamux 'mw.b ffd0008 3 1'
setenv bootcmd 'run amcsatamux && run tftpkernel && run tftpdtb && run
tftpramdisk && bootm 1000000 2000000 a00000'
```

A.2 FPGA MUX Control Registers

The following table shows the Mux Lock and SATA Mux Control Register (offset 0x08).

Table A-2 Mux Lock and SATA Mux Control Register (offset 0x08)

Register	Bit	Type	Function
0x08	7	RW	Lock access to Mux settings
0x08	3	RO	Current SATA Mux setting
0x08	2	RO	IPMC SATA Mux setting
0x08	1	RW	SATA Mux control
0x08	0	RW	SATA Mux control source (0 = IPMC, 1 = register)

Bit 7 of this register can only be written by the IPMC, but can be read by both the IPMC and service processor. When bit 7 is set, the service processor is prevented from writing to any of the mux control registers.

Bits 0 and 1 control how the SATA Mux select is driven. If bit 0 is clear, the SATA mux is controlled by a dedicated hardware signal directly from the IPMC. If bit 0 is set, the SATA mux control is driven by bit 1 of this register. In all cases, bit 2 reflects the state of the hard control signal from the IPMC and bit 3 reflects the actual state of the SATA mux (which depends on the hardware control and bits 0 and 1).

Table A-3 Ethernet Mux Control Register 2 (offset 0x0B)

Register	Bit	Type	Function
0x0B	6	RW	BIX_P20_MUX_SEL
0x0B	3	RW	BIX_P19_MUX_SEL
0x0B	2	RW	FIX_GE0_GE1_MUX_SEL
0x0B	1	RW	BIX_P13_P14_MUX_SEL
0x0B	0	RW	TOPSYNC_MUX_SEL

The read/write register controls the Ethernet multiplexers. *Table A-3*, shows which bits correspond to which physical signals. Bits not listed are reserved. Refer to the hardware specification for details of the function that each signal switches.

These registers are not reset to default values by a payload reset, but are reset to defaults by an FPGA reset from the IPMC or by a power cycle.

SFP Module Configuration

B.1 Overview

Certain SFP and SFP+ modules require reconfiguration in order to take advantage of their multifunction capabilities.

B.2 Tri-Speed Copper

Some copper SFP modules support a tri-mode speed capability where they will auto-negotiate and automatically link at 10, 100, or 1000 Mbps.

For example, the Finisar FCLF8522P2BTL. These modules do not power-up in this mode and therefore must be reconfigured to do so.

This support is only available on the 1G RTM ports. For both ATCA-F125 and ATCA-F140, these are the ports numbered between 4-7. A new `sfptool` option (`-m`) is used to enable this feature. There is no `sfptool` option to disable this feature. To restore the module to the default configuration, the user can remove and reinsert it or reboot/power-cycle the switch itself.

Example:

```
sfptool -p4 -m
```

Enable tri-mode support for the SFP attached to port number 4.

B.3 Dual-rate SFP+ modules

Some SFP+ modules support a dual-rate capability and they link at 1G or 10G.

For example, the Finisar FTLX8571D3BCV. By default, these modules will be configured for the 1G or low speed mode. A new `sfptool` option “-R1” allows the user to reconfigure the module for high speed mode or “-R0” for low rate mode. The high speed mode does not survive a switch reboot or power-cycle and must be specified on every boot.

Examples:

```
sfptool -p10 -R1
```

Enable high speed mode for the SFP+ attached to port number 10.

```
sfptool -p0 -R0
```

Enable low speed mode for the SFP+ attached to port number 0.

SFP Module Configuration

NOTE: The “-R” option does not interface to SRS. So, the user must also make sure that the desired rate matches the SRS configuration of the port. This is typically done using the SRS `imish` command or by modifying the SRS configuration file directly.

Related Documentation

C.1 Penguin Solutions Documentation

Technical documentation can be found by using the Documentation Search at <https://www.penguinolutions.com/edge/support/> or you can obtain electronic copies of documentation by contacting your local sales representative.

Table C-1 Penguin Edge Documentation

Document Title	Document Number
PrAMC-7311 Data Sheet	PrAMC-7311-DS
ATCA-7470 Data Sheet	ATCA-7470-DS
ATCA-F140 Series Data Sheet	ATCA-F140-DS
ATCA-9405 Data Sheet	ATCA-9405-DS
PrAMC-7311 Installation and Use	6806800P34
ATCA-7470 Installation and Use	6806800P15
ATCA-F140 Installation and Use	6806800M67
ATCA-9405 Installation and Use	6806800M71
ViewCheck on ATCA-F140 Installation and Use	6806800S50
ViewCheck on ATCA-7370 Installation and Use	6800800R81
BBS on ATCA-F140 with SRstackware® for WR Linux Command Line Interface Usage	6806800S59
SRstackware Intelligent Network Software	6806800L88
SRstackware Intelligent Network Software Protocol Demo Guide	6806800L35
SRstackware Intelligent Network Software VRRP Command Reference	6806800L36
SRstackware Intelligent Network Software RIP Command Reference	6806800L37
SRstackware Intelligent Network Software Layer 2 Configuration Guide	6806800L39
SRstackware Intelligent Network Software OSPF Command Reference	6806800L40
SRstackware Intelligent Network Software Layer 2 Command Reference	6806800L41
SRstackware Intelligent Network Software Layer 3 Configuration Guide	6806800L42
SRstackware Intelligent Network Software Troubleshooting guide	6806800L34
SRstackware FAQ	6806800L44

Table C-1 Penguin Edge Documentation

Document Title	Document Number
SRstackware Intelligent Network Software Layer3 Command Reference	6806800L87
SRstackware Application Programming Interface Developer Guide	6806800L43
ATCA-F140 Telco Clocking Module Reference Guide	6806800T38

PENGUIN[™]

SOLUTIONS 

Penguin Solutions is a trade name used by SMART Embedded Computing, Inc., a wholly owned subsidiary of SMART Global Holdings, Inc. Penguin Edge is a trademark owned by Penguin Computing, Inc., a wholly owned subsidiary of SMART Global Holdings, Inc. All other logos, trade names, and trademarks are the property of their respective owners. ©2022 SMART Embedded Computing, Inc.