COMPUTING

Blade Services Software on ATCA-9305

Programmer's Reference

P/N: 6806800J36K

July 2017

ARTESYN™

EMBEDDED TECHNOLOGIES

Trademarks

Notice

Limited and Restricted Rights Legend

Contact Address

Artesyn Embedded Technologies

Marketing Communications

2900 S. Diablo Way, Suite 190

Tempe, Arizona 85282

Contents

About this Manual 11

1 Introduction 19

1.1 Overview 19
1.2 Software Building Blocks 19

2 Installing the Blade Services Software 23

2.1 Overview 23
2.1.1 Package Information 23
2.1.2 Installation Overview 23
2.2 Configuring a TFTP Server 24
2.2.1 Create /tftpboot Directory and Copy Target Files 24
2.2.2 Configuring the TFTP Server 27
2.3 WindRiver PNE 2.0 Layer 27
2.4 Loading Ramdisk on Management Processor 29
2.5 Booting NFS-mounted Root File System on Management Processor 30
2.6 Installation into the Flash Device 31
2.6.1 Installation of the root filesystem 32
2.6.2 Boot Linux from the flash device 34
2.7 Booting OCTEON from Linux on Management Processor 35
2.7.1 Running Linux on OCTEONS 35
2.7.2 Running the Example Forwarder SE-S Application 36
2.7.3 Running Multiple Applications on Multiple OCTEON Cores 38

3 Firmware Upgrade Facility 39

3.1 Overview 39
3.2 Firmware Recovery Image Files 39
3.3 Backup Concept 39
3.4 fcu—Firmware Upgrade Command-Line Utility 40
3.5 Upgrading Firmware Image 45
3.5.1 U-Boot Firmware Upgrade 46
3.5.2 IPMC Upgrade 50
3.5.3 Management Processor Linux Kernel Upgrade 52
3.5.4 Management Processor Device Tree Block (DTB) Upgrade 55

4 Linux Distribution Description 57

4.1 Distribution Description 57
4.2 Reliability 57
4.3 Login 57
4.4 Network Services Configuration 58
4.4.1 ATCA-9305 Ethernet Interfaces 58
4.4.1.1 8548 Management Ethernet Interface eth0 58
4.4.1.2 Switch Ethernet Interface eth1 58
4.4.2 SSH 58
4.5 U-Boot Parameter 59

5 Hardware Platform Management 61

5.1 Overview 61
5.2 hpmagentd—HPM Agent Daemon 62
5.3 hpm—Start-Up Script 64
5.4 hpm—Shutdown and Reboot Scripts 65
5.5 hpmcmd—HPM Command Utility 66
5.5.1 Command Overview. 68
5.5.2 Supported Commands. 70
5.5.2.1 bootbankget 70
5.5.2.2 bootbankset 71
5.5.2.3 bootparamerase 72
5.5.2.4 bootparamget 73
5.5.2.5 bootparamset 73
5.5.2.6 bye 74
5.5.2.7 cmd 74
5.5.2.8 deviceid 75
5.5.2.9 ekeydownpath 76
5.5.2.10 ekeyuppath 76
5.5.2.11 exit 76
5.5.2.12 frudata 77
5.5.2.13 fruinfoget 77
5.5.2.14 fruinfoset 82
5.5.2.15 fruinv 83
5.5.2.16 fruread 83

5.5.2.17 fruwrite 84
5.5.2.18 fwprogevent 85
5.5.2.19 help 85
5.5.2.20 ipmbaddress 86
5.5.2.21 ipmcdevice 86
5.5.2.22 ipmcstatus 86
5.5.2.23 ledget 86
5.5.2.24 ledprop 87
5.5.2.25 ledset 88
5.5.2.26 loglevelget 89
5.5.2.27 loglevelset 90
5.5.2.28 macaddress 90
5.5.2.29 motshelftype 91
5.5.2.30 partnumber 91
5.5.2.31 physlotnumber 92
5.5.2.32 portget 92
5.5.2.33 portset 93
5.5.2.34 posttypeget 94
5.5.2.35 posttypeset 95
5.5.2.36 quit 95
5.5.2.37 rebootpath 95
5.5.2.38 sdr 96
5.5.2.39 sdr_dump 97
5.5.2.40 sdrinfo 97
5.5.2.41 sendamc 98
5.5.2.42 sendcmd 98
5.5.2.43 shelfaddress 99
5.5.2.44 shelfslots 100
5.5.2.45 shutdownpath 100
5.5.2.46 slotmap 100
5.5.2.47 slotnumber 101
5.5.2.48 updmacaddress 101
5.5.2.49 version 102
5.5.2.50 watchdog 102

6 SRstackware 105

6.1 Overview 105
6.2 Configuring SRstackware 106
6.2.1 Default Configuration 106
6.2.2 Custom Configuration 107
6.2.2.1 Enabling Protocols 108
6.2.2.2 Sample Configurations 108
6.2.3 SNMP Usage Guidelines 116
6.2.4 SNMP Traps Usage Guidelines 117
6.2.4.1 Configuring Agent to Send SNMP Notifications 117
6.2.4.2 Configuring SNMP Manager to Receive SNMP Notifications 118
6.2.5 Configuring Logs 119
6.3 Standards Supported 119
6.4 2x10G/12x1G RTM Ports 121
6.5 18x1G RTM Ports 124
6.6 6X10G RTM Ports 125

A Related Documentation 127

A.1 Artesyn Embedded Technologies - Embedded Computing Documentation 127
A.2 Related Specifications 128

List of Tables

Table 2-1 Software Packages for Management Processor 8548 25
Table 2-2 Software Packages for OCTEON CN58XX 25
Table 5-1 Command Overview 68
Table 6-1 Standards Supported 119
Table 6-2 ARTM-9305 2x10G/12x1G 121
Table 6-3 ARTM-9305 18x1G Ports 124
Table 6-4 ARTM-9305 6x10G Ports 125
Table A-1 Artesyn Embedded Technologies - Embedded Computing Publications 127
Table A-2 Related Specifications 128

List of Figures

Figure 1-1 Blade Services Architecture 20
Figure 5-1 Software Levels of the HPM Architecture 62
Figure 6-1 ATCA-9305 BCM56802 switch with 1G IO RTM ports 123
Figure 6-2 6X10G RTM Ports 125

Overview of Contents

This manual is divided into the following chapters and appendices.

- Chapter 1, *Introduction, on page 19*
- Chapter 2, *Installing the Blade Services Software, on page 23*
- Chapter 3, *Firmware Upgrade Facility, on page 39*
- Chapter 4, *Linux Distribution Description, on page 57*
- Chapter 5, *Hardware Platform Management, on page 61*
- Chapter 6, *SRstackware, on page 105*
- Appendix A, *Related Documentation, on page 127*

Abbreviations

This document uses the following abbreviations:

Abbreviation	Definition
AMC	Advanced Mezzanine Card
API	Application Programming Interface
AdvancedTCA	Advanced Telecommunications Computing Architecture
ATCA	Advanced Telecommunications Computing Architecture
ATIS	AdvancedTCA Interfaces Service
BBS	Basic Blade Services
BC	Base Channel
BIB	Board Information Block
BIOS	Basic Input Output System
BT	Block Transfer
CGL	Carrier Grade Linux
CMC	Common Mezzanine Card
CMD	Command Line Tool
CPU	Central Processing Unit

Abbreviation	Definition
DAT	Domain Alarm Table
DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
ECC	Embedded Communications Computing
ER	Expected Responder
EST	Eastern Standard Time
ETH	Ethernet
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
HPM	Hardware Platform Management
I/O	Input Output
IDE	Integrated Device Electronics
IP	Internet Protocol
IPM	Intelligent Platform Management
IPMB	Intelligent Platform Management Bus
IPMC	Intelligent Platform Management Controller
IPMI	Intelligent Platform Management Interface
KCS	Keyboard Control Style

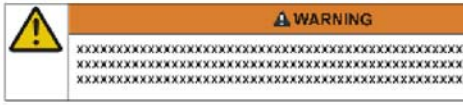
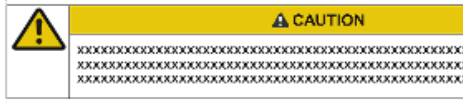
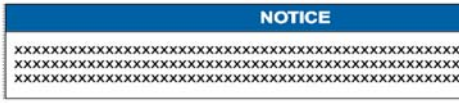
Abbreviation	Definition
LED	Light Emitting Diode
LHC	Link Health Check
LSP	Linux Support Package
LT	Lower Threshold
LUN	Logic Unit Number
MAC	Media Access Control
MIB	Management Information Base
MMC	Module Management Controller
NTP	Network Time Protocol
OEM	Original Equipment Manufacturer
OSDL	Open Source Development Labs
PC	Personal Computer
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
PrPMC	Processor PMC
PXE	Preboot Execution Environment
RAM	Random Access Memory
RDR	Resource Data Record
RMCP	Remote Monitoring and Control Protocol
ROM	Read Only Memory
RPM	RedHat Package Manager
RTM	Rear Transition Module
SAF	Service Availability Forum
SAM	Shelf Manager and Alarm Module

Abbreviation	Definition
SAS	Serial Attached SCSI
SATA	Serial ATA
SCSI	Small Computer System Interface
SCXB	System Controller Switching Blade
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
SSH	Secure Shell
SSU	Synchronization Supply Unit
TAR	Tape Archiver
TBD	To Be Defined
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TTY	Teletypewriter
UC	Update Channel
UDP	User Datagram Protocol
USB	Universal Serial Bus
UT	Upper Threshold
VLAN	Virtual Local Area Network
WDT	Watchdog Timer

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 in body text
`Courier + Bold`	Used to characterize user input and to separate it from system output
Reference	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
. . .	Omission of information from example/command that is not necessary at the time being
..	Ranges, for example: 0..4 means one of the integers 0,1,2,3, and 4 (used in registers)
|	Logical OR

Notation	Description
	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
	No danger encountered. Pay attention to important information

Summary of Changes

See the table below for manual revisions and changes.

Part Number	Date	Description
6806800J36K	July 2017	Added registered trademark to SRstackware.
6806800J36J	May 2014	Re-branded to Artesyn template.
6806800J36H	July 2011	Corrected variable **`tffs_tffs_format`** to **`tffs.tffs_format`** in section *Installation of the root filesystem* on page 32.
6806800J36G	June 2011	Added *SNMP Usage Guidelines* on page 116. Updated *Loading Ramdisk on Management Processor* on page 29.
6806800J36F	October 2010	Updated *6X10G RTM Ports* on page 125.

Part Number	Date	Description
6806800J36E	August 2010	Chapter 1, updated *Overview* on page 19. Chapter 2, updated *Package Information* on page 23, *Booting OCTEON from Linux on Management Processor* on page 35, *Running Linux on OCTEONS* on page 35, *Running the Example Forwarder SE-S Application* on page 36, *Running Multiple Applications on Multiple OCTEON Cores* on page 38. Added *WindRiver PNE 2.0 Layer* on page 27, *Installation into the Flash Device* on page 31. Chapter 3, updated *U-Boot Firmware Upgrade* on page 46 , *IPMC Upgrade* on page 50. Added *Management Processor Linux Kernel Upgrade* on page 52, *Management Processor Device Tree Block (DTB) Upgrade* on page 55. Chapter 4, added *U-Boot Parameter* on page 59. Updated *ATCA-9305 Ethernet Interfaces* on page 58 and *Reliability* on page 57. Chapter 5, added *macaddress* on page 90, *updmacaddress* on page 101. Chapter 6, updated *Standards Supported* on page 119, *2x10G/12x1G RTM Ports* on page 121, *18x1G RTM Ports* on page 124, *6X10G RTM Ports* on page 125, *Default Configuration* on page 106.
6806800J36D	April 2010	Added *SNMP Usage Guidelines* on page 116.
6806800J36C	December 2009	Updated 6.2.2.2 Sample Configurations and 6.3 Switch with 1G IO RTM Ports sections.
6806800J36B	October 2009	Second edition
6806800J36A	July 2009	Early Access

1.1 Overview

The blade services software for the ATCA-9305 includes a common set of functionality which is available for all AdvancedTCA blades and AMC modules, and a unique set of functionality which is tailored specifically to the ATCA-9305. It is designed to provide a unified software operating environment to an application and provides a standard management interface to the rest of the AdvancedTCA system environment.

The blade services software provides a set of services that supports the blade on which the software is installed. The blade services include:

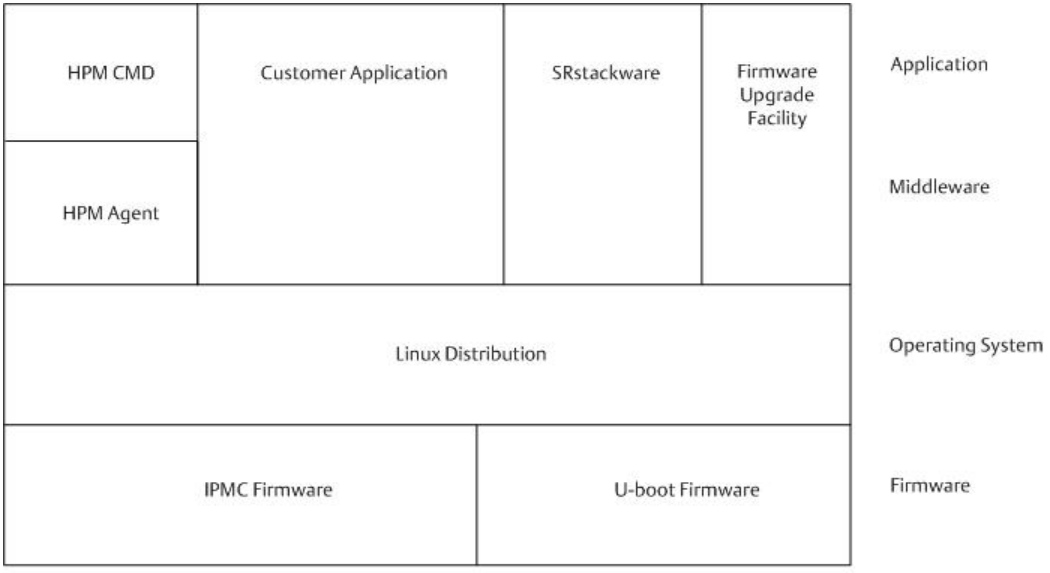
- Wind River Platform for Network Equipment (PNE) Linux edition 2.0.
- Several custom hardware management functions for the unique hardware of the blade.
- A set of management routines for Linux and all hardware interfaces. Management access includes support for SNMP and a local console interface based on the standard Linux command shell.
- SRstackware; a switch management software with protocols support.

1.2 Software Building Blocks

The blade services include a common set of functionality that is available for all ATCA blades and AMC modules. The blade services also include a unique set of functionality that is tailored to a particular blade or module.

Figure 1-1 illustrates the blade services software architecture.

Figure 1-1 Blade Services Architecture



HPM CMD – Hardware Platform Manager Command line tool

IPMC – Intelligent Platform Management Controller

Blade Services consists of the following software and services:

- Firmware Upgrade Facility
  The Firmware Upgrade Facility (FUF) provides a uniform way to upgrade firmware on Artesyn blades and AMC modules, regardless on which flash location the firmware is stored. FUF upgrades the BIOS 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
  Blade services is based on Wind River Platform for Network Equipment (PNE) Linux edition 2.0. Linux services (above the kernel) are activated by the blade services installation scripts.
- Hardware Platform Management
  The Hardware Platform Management (HPM) in ATCA systems is based on Intelligent Platform Management Interface (IPMI) specification. IPMI commands can be complex and cumbersome. Using a certain set of commands, HPM facilitates the blade or module-level hardware management.
- SRstackware®
  SRstackware is a switching and routing stackware which supports basic switch configuration along with Layer 2 and Layer 3 protocols support.
- SNMP Agent
  As each blade services blade/module is individually managed, the default installation script installs and initializes the Net-SNMP agent.

2.1 Overview

This section describes how to install the blade services software for the ATCA-9305. Artesyn provides software images, including software updates, to its licensed customers. To obtain the latest blade services release, please contact your local sales representative.

2.1.1 Package Information

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

The following rpm commands are useful to review package information.

Command	Description
`rpm -qa`	List all installed packages. Use **`rpm -qa | grep hpi`** to list only HPI packages.
`rpm -qi -p <RPM file name>`	List information about a package, including the package name. This may help to avoid confusion between RPM file name and package name.
`rpm -ql package-name`	List the content of a package, where `package-name` is the name of a specific package, for example, `rpm -ql openhpi`.
`rpm -qi package-name`	List information about a package, where `package-name` is the name of a specific package, for example, `rpm -qi openhpi`.
`rpm -qf <path to file>`	Finds out which RPM a file belongs to.

For more information about the rpm command, see its `man` page.

For more information, refer ATCA-9305 Release Notes.

2.1.2 Installation Overview

The installation process boots a custom ramdisk kernel from the network. Current blade services release does not support installation from CD.

The blade must have access to a TFTP server to retrieve the files.

The procedures in this chapter assume that you are installing from a network following these basic steps that are described in the following subsections:

1. Configure TFTP server
2. Prepare ATCA-9305 to boot from network
3. Boot from the network
4. Install and configure the software

2.2 Configuring a TFTP Server

It is necessary to prepare a TFTP server on a host prior to installing the blade services software over a network. This requires that you create the expected directory structure for boot images and installation files. It also requires that you appropriately configure the TFTP server as described in the following sections.

2.2.1 Create /tftpboot Directory and Copy Target Files

It is customary to place TFTP files in a tftpboot directory. Regardless of the file system node you specify as the root for your TFTP service, the installation scripts expect a certain directory structure when retrieving files.

Creating the /tftpboot Directory and Copying the Target Files

To create the expected directory structure and copy the required files:

1. On the host create `/tftpboot` directory, if it does not already exist.
   `mkdir /tftpboot`
2. Create a sub-directory for your blade, for example:
   `mkdir /tftpboot/atca-9305`

3. Copy or move the installation files to the sub-directory.

The name and number of installation files depends on the particular blade services release. Refer the respective release notes for exact file name. The following table is an example of possible blade services installation files.

Table 2-1 Software Packages for Management Processor 8548

File	Description
`emerson_atca9305_8548-System.map-WR2.0-<version>_standard`	Linux Kernel system map.
`emerson_atca9305_8548-default_kernel_image-WR2.0-<version>_standard`	Linux Kernel for U-Boot.
`emerson_atca9305_8548-linux-modules-WR2.0-<version>_standard.tar.bz2`	Linux Kernel modules.
`emerson_atca9305_8548.dtb`	Compiled Device Tree.
`emerson_atca9305_8548-glibc_cgl-standard-dist.tar.bz2`	Linux rootfs with hpm, fcu, srs, and octpcitools.
`emerson_atca9305_8548-initrd.gz`	Linux ramdisk includes rootfs, hpm, fcu, srs, IPMI firmware, Linux kernel for OCTEON, forwarder, and octpcitools.
`emerson_atca9305_8548-initrd.gz.uboot`	Linux ramdisk for U-Boot includes rootfs, hpm, fcu, srs, IPMI firmware, Linux kernel for OCTEON, forwarder, and octpcitools.

Table 2-2 Software Packages for OCTEON CN58XX

File	Description
`emerson_atca9305_cn58xx-System.map-WR2.0-<version>_standard`	Linux Kernel system map.

Table 2-2 Software Packages for OCTEON CN58XX (continued)

File	Description
`emerson_atca9305_cn58xx-default_kernel_image-WR2.0-<version>_standard`	Linux Kernel for U-Boot.
`emerson_atca9305_cn58xx-linux-modules-WR2.0-<version>_standard.tar.bz2`	Linux Kernel modules.
`emerson_atca9305_cn58xx-glibc_small-standard-dist.tar.bz2`	Linux rootfs based on Busybox including octlinuxtools.
`emerson_atca9305_cn58xx-initrd.gz`	Linux ramdisk with rootfs based on Busybox and octlinuxtools.
`emerson_atca9305_cn58xx-initrd.gz.uboot`	Linux Ramdisk for U-Boot including rootfs based on Busybox and octlinuxtools.
`emerson_atca9305_cn58xx-vmlinux-with-initramfs`	Linux Kernel with embedded ramdisk. This image is also part of the 8548 ramdisk.
`emerson_atca9305_cn58xx-sysroot.tgz`	PNE2.0 toolchain to build 64-bit applications for the OCTEON.
`octlinuxtools-<version>.atca9305_cn58xx_64-linux_seum.rpm`	Linux tools from OCTEON SDK.

NOTICE

For latest software packages, refer release notes.

To ensure that the downloaded files are correct, sha1 checksums are used. The checksum of all installation files is stored in `files.sha1sum` file. If the sha1 checksums are not correct, an error message is displayed during the installation process. If you make changes to any of the files, you need to update the `files.sha1sum` file as well. If you still get an error message during the installation, it is likely that one or more of the files have not been copied successfully. Copy all the files to the `tftpboot` directory again and restart the installation.

2.2.2 Configuring the TFTP Server

The instructions in this section can be used to configure standard TFTP servers, BSD compatible, that are under the control of `xinetd`.

Configuring the TFTP Server

To configure TFTP as root on the host, complete the following steps:

1. Create (or edit) `/etc/inetd.d/tftp` file. Use the following format:

```
#/etc/inetd.d/tftp
service tftp
{
   socket_type = dgram
   wait = yes
   user = root
   log_on_success += USERID
   log_on_failure += USERID
   server = /bin/in.tftpd
   server_args = -r blksize /tftpboot
   disable = no
   protocol = udp
}
```

2. If there are any TFTP daemons that have not timed out, you need to stop them using the following command:
   `killall in.tftpd`

3. Enter the following command to allow `inetd` re-read the configuration file:
   `/etc/rc.d/init.d/inetd restart`

The TFTP server is now configured.

2.3 WindRiver PNE 2.0 Layer

The `atca9305-layers.tgz` file contains a WindRiver PNE 2.0 layer for building the Linux kernel and root filesystem for the Management Processor and the Octeon.

Using the layer

The following steps describe how to extract and build the blade services software using the ATCA-9305 WindRiver PNE 2.0 layer. For detailed description on layers, refer the WindRiver PNE 2.0 Manuals.

1. Extract the `atca9305-layers.tgz` tar file to any directory. It is not necessary that the layer file is extracted within the PNE 2.0 installation directory. The only requirement is that the PNE 2.0 installation directory (for example, `/opt/windriver`) should be accessible. An example to install it under the home directory of user foo is
   ```
   $ cd /home/foo
   $ tar xfz atca9305-layers.tgz
   ```
2. Change the current directory to the layer
   ```
   $ cd atca9305
   ```
3. Copy the example Management Processor environment settings file `env-ars068-8546.sh` to your environment file if you want to build the software for the Management Processor
   ```
   $ cp env-ars068-8548.sh env-foo-8548.sh
   ```
4. Copy the example Octeon environment settings file `env-ars068-cn58xx.sh` to your environment file if you want to build the software for the Octeon
   ```
   $ cp env-ars068-cn58xx.sh env-foo-8548.sh
   ```
5. Edit your environment file(s) and update the settings to be suitable for your environment. Normally only `WIND_HOME` and `INSTALL_DIR` needs to be changed.
6. Source your environment file e.g
   ```
   $ . env-foo-8548
   ```
7. Build the Kernel and root filesystem
   ```
   $ make
   ```

Note that the sourcing of the correct environment file influences the build. If you source the environment file for Management Processor, a call of make builds software for the Management Processor and if you source the environment file for the Octeon, a make will build software for the Octeon.

2.4 Loading Ramdisk on Management Processor

You can load ramdisk on the management processor MPC8548. To load the ramdisk:

If you are upgrading ATCA-9305 on top of an older blade services version then the SRstackware software uses the old configuration files. If you want the default configurations of the current version then you need to move or delete the following files in flash before loading the new software:

- /mnt/flash/srstackware/config/srs.cfg
- /mnt/flash/srstackware/config/srsinit_active.conf
- /mnt/flash/srstackware/config/snmpd_active.conf

1. Set the environment variables from U-Boot, as per your environment.

   ```
   =>setenv initrd_high 0x10000000

   =>setenv bootargs root=/dev/ram0 rw ramdisk_size=600000
   console=ttyS0,9600
   ```

2. Save the environment variables.

   ```
   => save
   ```

3. Load the kernel.

   ```
   =>tftpboot 800000 emerson_atca9305_8548-default_kernel_image-WR2.0-
   <version>_standard
   ```

4. Load the device tree.

   ```
   =>tftpboot 1000000 emerson_atca9305_8548.dtb
   ```

5. Load the ramdisk.

   ```
   =>tftpboot 2000000 emerson_atca9305_8548-initrd.gz.uboot
   ```

6. Boot the kernel, ramdisk, and device tree.

   ```
   =>bootm 800000 2000000 1000000
   ```

7. Login in as `root` with password as `root`.

2.5 Booting NFS-mounted Root File System on Management Processor

You can install NFS-mounted root file system on the management processor MPC8548. To install NFS you need to extract `emerson_atca9305_8548-glibc_cgl-standard-dist.tar.bz2` and `emerson_atca9305_8548-linux-modules-WR2.0-<version>_standard.tar.bz2` files to a location that is exported via NFS. Each ATCA-9305 blade needs a separate copy of the file system. You can copy firmware images and OCTOEN™ Linux kernel into the file system.

To boot NFS on the management processor MPC8548:

1. Set the environment variables from U-Boot, as per your environment.

   ```
   =>setenv bootargs console=ttyS0,9600 root=/dev/nfs rw
   nfsroot=192.168.20.1:/9305nfs
   ip=192.168.20.2:192.168.20.1:192.168.20.1:255.255.255.0::eth0:
   ```

   In this example, IP address of the NFS server is `192.168.20.1` and name of the NFS is `9305nfs`.

2. Save the environment variables.

   ```
   => save
   ```

3. Load the kernel.

   ```
   =>tftpboot 800000 emerson_atca9305_8548-default_kernel_image-WR2.0-
   <version>_standard
   ```

4. Load the device tree.

   ```
   =>tftpboot 1000000 emerson_atca9305_8548.dtb
   ```

5. Boot the kernel and device tree.

   ```
   =>bootm 800000 - 1000000
   ```

6. Login in as `root` with password as `root`.

2.6 Installation into the Flash Device

The blade services software for the Management Processor can be installed into the flash device. However, the root filesystem can only be installed if the ATCA-9305 frontboard has a DOC (Disk-On-Chip) device or if you use the ARTM 6x10GE.

The installation is divided into the following:

1. Installation/upgrade of the Linux kernel using FCU, see *Management Processor Linux Kernel Upgrade* on page 52.
2. Installation/upgrade of the DTB using FCU, see *Management Processor Device Tree Block (DTB) Upgrade* on page 55.
3. Installation of the root filesystem using the `instrootfs` script, see *Installation of the root filesystem* on page 32.

The section below describes the root filesystem installation and how to boot from the flash device. The installation/upgrade of the Linux kernel and DTB are described in Chapter 3, *Firmware Upgrade Facility, on page 39*.

2.6.1 Installation of the root filesystem

The following steps are necessary to install the root filesystem:

1. Boot Linux via initrd or NFS as described in *Loading Ramdisk on Management Processor* on page 29 and *Booting NFS-mounted Root File System on Management Processor* on page 30.

If you plan to install the root filesystem on the DOC device and this is the first usage of the device then it needs to be low-level formatted first. This can be done by adding 'tffs.tffs_format=0' in the u-boot bootargs environment setting.

For example, if your bootargs setting contains the value

```
=> printenv bootargs

=> bootargs=root=/dev/ram0 rw ramdisk_size=600000
console=ttyS0,9600
```

then set the bootargs to:

```
=> setenv bootargs 'root=/dev/ram0 rw ramdisk_size=600000
console=ttyS0,9600 tffs_tffs_format=0'
```

Do not store this bootargs, otherwise the DOC device will be formatted on every Linux boot.

2. Install the root filesystem using the `instrootfs` script. The script downloads the blade services dist tarball (`emerson_atca9305_8548-glibc_cgl-standard-dist.tar.bz2`) from a TFTP server specified. The following options are supported:

```
-d DEVICE   block device to install to [default=/dev/tffsa]
-i IPADDR   IP address of TFTP server [default=127.0.0.1]
-t FILE     location of the dist tar file on the TFTP server
            or local location if '-a' option is used
[default=emerson_atca9305_8548-glibc_cgl-standard-dist.tar.bz2]
-a          do not download dist tar from TFTP server but use
local one
-l FILE    output install steps into logfile [default=/dev/null]
-f          do not ask whether its okay to install
-V          print version information and exit
```

```
-h          display this help and exit
```

The default device option is `/dev/tffsa`, which is the correct one for installation on the DOC device. For installation on the flash device of the ARTM 6x10GE you have to set it to `/dev/sda`.

The following example uses the dist tarball specified with the `-t` option on the TFTP server 172.16.166.55 to install the root filesystem on the DOC device. The installation details are logged into the file `instrootfs.log`.

```
root@ATCA-9305-9-255:/> instrootfs -i 172.16.166.55 -t
emerson_atca9305_8548-glibc_cgl-standard-dist.tar.bz2 -l
instrootfs.log
block device:        /dev/tffsa
tftp server:         172.16.166.55
dist tar file:       emerson_atca9305_8548-glibc_cgl-standard-
dist.tar.bz2
local dist tar file: false
log file:            instrootfs.log

WARNING: All data on device '/dev/tffsa' will be lost!
Continue (y/n)

Installation started - this may take a while.
Please do not abort this script!

*) removing any existing partitions on device '/dev/tffsa' [OK]
*) creating new primary partition '/dev/tffsa1' [OK]
*) creating an ext3 fs on '/dev/tffsa1' [OK]
*) mounting '/dev/tffsa1' on '/mnt.instrootfs.tffsa1' [OK]
*) downloading 'emerson_atca9305_8548-glibc_cgl-standard-
dist.tar.bz2' from tftp server '172.16.166.55' to
'/mnt.instrootfs.tffsa1/emerson_atca9305_8548-glibc_cgl-
standard-dist.tar.bz2' [OK]
*) extracting '/mnt.instrootfs.tffsa1/emerson_atca9305_8548-
glibc_cgl-standard-dist.tar.bz2' to '/mnt.instrootfs.tffsa1'
[OK]
*) extracting '/dev' of
'/mnt.instrootfs.tffsa1/emerson_atca9305_8548-glibc_cgl-
standard-dist.tar.bz2' to '/mnt.instrootfs.tffsa1'
```

```
*) removing local dist tarball
'/mnt.instrootfs.tffsa1/emerson_atca9305_8548-glibc_cgl-
standard-dist.tar.bz2' [OK]
*) unmounting '/mnt.instrootfs.tffsa1' [OK]

Installation completed successfully.
```

Please make sure the 'root' kernel parameter is set to 'root=/dev/tffsa1' if you want to boot the installed rootfs.

The installation re-partitions and re-formats the whole flash device. Any data or previous installation will be lost.

2.6.2 Boot Linux from the flash device

The following steps are necessary to boot Linux from the flash device. Make sure that you have already upgraded kernel and DTB as described in *Management Processor Linux Kernel Upgrade* on page 52 and *Management Processor Device Tree Block (DTB) Upgrade* on page 55.

1. Reboot into u-boot
2. Update your bootargs setting as follows:
   - If using the DOC device
     ```
     => setenv bootargs 'root=/dev/tffsa1 rw console=ttyS0,9600
     ```
   - If using the ARTM 6x10GE
     ```
     => setenv bootargs 'root=/dev/sda1 rw console=ttyS0,9600
     ```
3. Store the flash boot command into a u-boot variable
   ```
   => setenv flashboot 'cp.b f6800000 1000000 fffff; bootm f6000000
   - 1000000'
   ```
4. Optionally turn on autoboot
   ```
   => setenv bootcmd 'run flashboot'
   ```
5. Optionally save the settings to make it persistent
   ```
   => save
   ```

6. Boot Linux from flash
   `=> run flashboot`

2.7 Booting OCTEON from Linux on Management Processor

The Octeon Processors run u-boot as bootloader. The NOR flash at the Management Processor contains partitions to store the active and backup u-boot image for both Octeons. On booting the Management Processor copies the currently active u-boot image to the Octeons. The Octeons stay in u-boot after startup.

With the help of the oct-pci tools available on the Management Processor, the Octeon Processor can be managed. The oct-pci tools are derived from an older Octeon SDK version and adapted to run on the ATCA-9305. All oct-pci tools on the ATCA-9305 require an additional parameter to specify the Processor number (1 for the first, 2 for the second Octeon). Refer to the Octeon SDK documentation for details about the oct-pci tools.

2.7.1 Running Linux on OCTEONS

The provided Linux image `/opt/motorola/images/emerson_atca9305_cn58xx-vmlinux-with-initramfs` combines the Linux kernel and a WindRiver PNE 2.0 root filesystem in one initramfs image file.

The PNE 2.0 root filesystem uses the glibc_small filesystem type which is based on busybox. After startup of the Linux kernel, a bash shell is started at the console. The SPI network interfaces spi0 and spi1 are initialized and ready-to-use but still down with no IP address configured.

This image should be treated as example image. The customer can build their own initramfs for the Octeon using the provided layer.

The ramdisk of the management processor MPC8548 contains a Linux image that can be downloaded to the OCTEON. The Linux image is stored in `/opt/motorola/images` directory.

To load the Linux image into the memory of both the OCTEONs:

```
oct-pci-load 1 0x20000000 emerson_atca9305_cn58xx-vmlinux-with-initramfs
```

```
oct-pci-load 2 0x20000000 emerson_atca9305_cn58xx-vmlinux-with-initramfs
```

After loading the image, you can use the following commands to boot Linux on both the OCTEONs. The `coremask` options can be modified to run Linux on a specific or multiple core(s).

```
oct-pci-cmd 1 "bootoctlinux 0x20000000 coremask=0x0001 endbootargs
console=ttyS0,115200"

oct-pci-cmd 2 "bootoctlinux 0x20000000 coremask=0x0001 endbootargs
console=ttyS0,115200"
```

The previous commands uses the serial console of the Octeon. If the PCI console should be used, the following command line argument should be changed:

```
'console=ttyS0,115200' to 'console=pci'
```

After that the tool oct-pci-console can be used to access the console. The following example accesses the console over PCI and defines the escape character 1 which is '^A' (means pressing ^A will exit the console).

```
oct-pci-console -escape 1 <cpu_num> 0
```

2.7.2 Running the Example Forwarder SE-S Application

The ramdisk of the management processor MPC8548 contains the forwarder SE-S application in `/opt/motorola/images` directory. This directory also contains the shell script `runforwarder.sh` to load and run the forwarder on four cores of the second OCTEON.

To load the forwarder application in the memory of the first OCTEON:

```
oct-pci-load 1 0x2000000 forwarder-atca9305_cn58xx_64-ses
```

After loading the image, you can use the following commands to boot the forwarder. The `coremask` options can be modified to run the forwarder on a specific or multiple core(s).

```
oct-pci-cmd 1 "bootoct 0x20000000 coremask=0x000f"
```

It is up to the SE-S application whether the console is opened on the serial interface or on PCI.

In this case the forwarder checks how the console is configured in u-boot and then opens its console on the currently active interface. Thus if you want to use the PCI console, you have to set it under u-boot first.

To use u-boot over PCI you can use the oct-pci-console tool along with the '-uboot' parameter. The tool internally reconfigures u-boot to use PCI. With the following example you can use u-boot over PCI. If you run the forwarder later on its in/output will also use this console. The escape character is defined to 1 which is '^A' (means pressing ^A will exit the console).

```
oct-pci-console -uboot -escape 1 <cpu_num> 0
```

Where:

`cpu_num` - selects the Octeon processor and the value can be 1 or 2.

Command Line Options

Configuration and operation of the application is done with the command line options. The command line options are as follows:

- Static L2 Entries: Add destination MAC addresses/egress port-list pairs to override default forwarding rules.
- Loopback Mode: Received packets are sent back on the same SPI-4.2 Channel.
- HW Pass-through Mode: Configures the Octeon's Hardware Pass-through mode using the compiled default forwarding rules.
- Dedicated Linux Filter mode: All received frames are delivered to Linux Cavium Ethernet Driver.
- Mixed Linux Filter / Normal L2 Forwarding operation: Add Static L2 entries for each MAC destinations within Linux. Any received broadcasts or frames with MAC address matches will be delivered to Linux. Other frames (including all broadcasts) follow normal forwarding rules.

Forwarder Usage

```
bootoct 100000 coremask=0xabcd endbootargs <help> <options>
<mac/port list>
```

where parameters are:

`help` - displays this usage

`filter -` delivers all received packets to linux, no forwarding

`qcount -` use specified value for the number of PKO queues per port

in form qcount number, where valid number values are 0 through 7

0 results in runtime allocation of PKO queues.

`loopback -` loopback spi channels - no stats

`passthru -` hw passthrough spi channels - config only

`nowait -` When set requests for new work, return immediately with new work or NULL

MAC address list of up to 64 MAC Address/egress port list pairs in form xx:xx:xx:xx:xx:xx portlist, where port list is a comma separated list of egress ports. Note that there must be no spaces in the port list.

On MPU-A valid egress ports are:

0 for spi0, 16 for spi1, and 40 for linux

On MPU-B valid egress ports are:

0 for spi0, 16 for spi1, and 40 for linux

When the forwarder is running standalone (without Linux) and running on more than one core, core 0 shows a prompt as follows:

```
Enter s to dump stats, c to clear stats, m to dump config, p to dump
pow/pko status, r for registers, h for boot help or x to reboot
```

2.7.3 Running Multiple Applications on Multiple OCTEON Cores

If multiple applications, such as Linux and SE-S or multiple SE-S run on an OCTEON then the OCTEON is held in reset mode until core 0 is booted. This means that before booting an application on core 0 all the applications must be downloaded and all other applications must have been booted.

Multiple applications can write to the console but the output may be mixed. Only the application on core 0 can read from the console.

3.1 Overview

The Firmware Upgrade Facility (FUF) provides a uniform way to upgrade firmware on Artesyn 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.

3.2 Firmware Recovery Image Files

FCU supports specially prepared firmware recovery image (FRI) files as well as firmware images in the 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 blade services software in `/opt/motorola/rom` when the blade-specific firmware support packages are installed.

The following image files are currently supported.

Filename	Description
atca-9305-8548-cpu-<version>.fri	U-Boot for the PPC
atca-9305-cn58xx-cpu-<version>.fri	U-Boot for the OCTEONS
atca-9305-hpm.1-all.img	IPMC firmware and IPMC Bootloader
atca-9305-hpm.1-boot.img	IPMC Bootloader
atca-9305-hpm.1-ipmc.img	IPMC firmware
atca9305-fruinfo-<version>.fri	IPMC FRU Inventory

3.3 Backup Concept

Firmware for IPMC on the Artesyn ATCA-9305 switch is stored in an on-chip flash, which has one active and one backup section. While 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 having bad IPMC firmware on both the sections and allows the next firmware upgrade to be done easily.

When upgrading the U-Boot firmware, 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.

3.4 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 payload reset.
- Show the version of a firmware image file and compare it with the version of the currently installed firmware.

By default, the FCU binary executable is installed in `/opt/motorola/bin`. This directory is specified in 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 39.

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 state 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, 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-9305-8548-cpu	U-Boot firmware device of the PPC
atca-9305-cn58xx-cpu1	U-Boot firmware device of the first OCTEON
atca-9305-cn58xx-cpu2	U-Boot firmware device of the second OCTEON
atca-9305-hpm.1-ipmc	IPMC firmware device
artm-9305-IO-1Gx12-hpm.1-ipmc	MMC firmware device of RTM 10x1G
artm-9305-IO-1Gx18-hpm.1-ipmc	MMC firmware device of RTM 18x1G

`-f <filename>`
`--file=<filename>`

Specifies the FRI file, where `<filename>` is the complete path and name 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 devices supported on the blade. This option is exclusive and should not be used with other options.

`-m`
`--mark`

Tells FCU to set the boot select so that on the next boot the specified firmware bank is 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: On ATCA-9305 the firmware image and 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.

`-q`
`--query`

Tells 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 the image is marked or not. For HMP.1 compliant firmware images, the `-q` command displays the state of the image as operational, rollback, or deferred.

`-s`
`--show`

Displays detailed information about a specified file. The displayed information includes for example image type, version, and manufacturer name. This command can 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, whereas 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>
- --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 in these examples:

- **fcu –vf /opt/motorola/rom/<filename>**
- **fcu –q –d <device-id>**
- **fcu –q –d-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.

3.5 Upgrading Firmware Image

This section describes recommended procedures for upgrading firmware devices.

3.5.1 U-Boot Firmware Upgrade

U-Boot Firmware Upgrade of the Management Processor

Follow these steps for U-Boot firmware upgrade of the management processor. The shown file names and paths are only meant as an example and should be replaced with file names and paths applicable to your configuration.

1. Query the current U-Boot firmware images on the blade.

```
root@ATCA-9305-5-255:/opt/motorola/rom# fcu -q -d atca-9305-8548-cpu
********************[[[[[REPORT BEGIN]]]]]********************
OPERATION : Query
RESULT    : SUCCESS
MESSAGE   : Device              : atca-9305-8548-cpu
            Part number         : 11000234-00-0
            Part revision       :
                BANK                : A
                Firmware Name       : U-Boot
                Firmware Version    : U-Boot 1.1.4 (Jun 22 2009 -
13:35:31)1.1.3
                Marked for next use : yes
                BANK                : B
                Firmware Name       : U-Boot
                Firmware Version    : U-Boot 1.1.4 (Apr  6 2009 -
16:03:10)0.9k.1
                Marked for next use : no
                BANK                : C
                Firmware Name       : U-Boot
```

```
                Firmware Version    : EMPTY BANK

                Marked for next use : no

********************[[[[[ REPORT END ]]]]]********************
```

In this example, Bank A and Bank B are in the NOR flash of the PPC. However, Bank C is the socketed flash.

2. Upgrade the U-boot firmware to version 1.1.3.

```
root@ATCA-9305-5-255:/opt/motorola/rom# fcu -uf atca-9305-8548-cpu-
1.1.3.fri

region 000000-00FFFF: ... OK

region 010000-01FFFF: ... OK

region 020000-02FFFF: ... OK

region 030000-03FFFF: ... OK

region 040000-04FFFF: ... OK

region 050000-05FFFF: ... OK

region 060000-06FFFF: ... OK

region 070000-07FFFF: ... OK

********************[[[[[REPORT BEGIN]]]]]********************

OPERATION : Verify

RESULT    : NOTICE

MESSAGE   : forcing partrevision

OPERATION : Verify

RESULT    : SUCCESS : compatible image found

MESSAGE   : OEM:EMERSON PROD:atca-9305 PN: 11000234-00-0 REV:  VER:U-
Boot 1.1.4 (Jun 22 2009 - 13:35:31)1.1.3 NAME:U-BOOT

OPERATION : Write
```

```
RESULT    : SUCCESS UPGRADING

MESSAGE   : !atca-9305-8548-cpu:EMERSON:atca-9305:11000234-00-0:*:U-
BOOT:U-Boot 1.1.4 (Jun 22 2009 - 13:35:31)1.1.3

********************[[[[[ REPORT END ]]]]]********************
```

To protect the latest running U-Boot version in the active bank, FCU writes to the backup bank. The active bank is independent of the bank, that is marked for next use. Ideally the active bank should not be same as the bank marked for next use. Although it is possible that the two banks are same.

3. Mark the bank where FCU loaded the new firmware.

   ```
   fcu -mbA -d atca-9305-8548-cpu
   ```

   In this example, U-Boot version 1.1.3 is in bank A.

4. Reboot the Linux.

U-Boot firmware upgrade of the OCTEONS

To upgrade the U-Boot on the OCTEONS it is recommended to have U-Boot version 1.1.3 on the Management Processor, otherwise the mark operation does not work as expected.

You run fcu on the Management Processor. fcu is not available on an OCTEON.

The Management Processor has four partitions in the NOR flash to hold the U-Boot images for the OCTEONS. Each OCTEON has its own active and backup bank.

At startup, the U-Boot on the Management Processor transfers the U-Boot image to the OCTEONS via the PCI bus.

Show the OCTEONS banks using the `fcu -q` command.

```
MESSAGE   : Device              : atca-9305-cn58xx-cpu1

            Part number         : 11000234-00-0

            Part revision       :

                BANK                : A
```

```
                Firmware Name       : U-Boot
                Firmware Version    : U-Boot 1.1.1 (Apr  6 2009 -
17:49:00)0.9b.1
                Marked for next use : yes
                BANK                : B
                Firmware Name       : U-Boot
            Firmware Version    : U-Boot 1.1.1 (May 19 2009 - 11:55:11)1.0.1
                Marked for next use : no
OPERATION : Query
RESULT    : SUCCESS
MESSAGE   : Device              : atca-9305-cn58xx-cpu2
            Part number         : 11000234-00-0
            Part revision       :
                BANK                : A
                Firmware Name       : U-Boot
            Firmware Version    : U-Boot 1.1.1 (May 19 2009 - 11:55:11)1.0.1
                Marked for next use : no
                BANK                : B
                Firmware Name       : U-Boot
                Firmware Version    : U-Boot 1.1.1 (Apr  6 2009 -
17:49:00)0.9b.1
                Marked for next use : yes
```

You can query a single OCTEON only with "fcu -q -d atca-9305-cn58xx-cpu1" or "fcu -q -d atca-9305-cn58xx-cpu2".

To upgrade both the OCTEONS with the same version you just enter: "fcu -uf atca-9305-cn58xx-cpu-1.0.1.fri". To upgrade a single OCTEON only you also specify the device name: "fcu -uf <image> -d atca-9305-cn58xx-cpu1" or "fcu -uf <image> -d atca-9305-cn58xx-cpu2". fcu does not protect the active bank of the OCTEON, it just writes into the bank which is not marked for next use. After the upgrade you can mark the updated bank and reboot the Management Processor.

The u-boot version number as printed in the banner consists of the u-boot release version, a date code, and the Artesyn assigned version number. Only the Artesyn assigned version number is relevant.

3.5.2 IPMC Upgrade

Upgrading IPMC for ATCA-9305

The IPMC firmware can be upgraded using FCU from the Management Processor. If you are upgrading from firmware version 1.0.0 to 1.51.056, the processor will probably reset after the new firmware is activated. It is also possible that Linux hangs-up and you need to reset the board from the shelf manager.

Follow these steps to upgrade the IPMC. The file names, paths, and command outputs used in this section are only meant as an example and may vary depending on your configuration.

The upgrade of the MMC image on an RTM is similar, except for the device ID which must be `artm-9305-IO-1Gx12-hpm.1-ipmc` or `artm-9305-IO-1Gx18-hpm.1-ipmc`.

1. Query the current IPMC firmware images on the blade.
   `fcu -qd atca-9305-hpm.1-ipmc`
2. Show the version of the new IPMC file; to verify that it has actually a newer version than the already installed IPMC file.
   `fcu --show -f /opt/motorola/rom/atca-9305-hpm.1-all.img`
3. Upgrade the firmware image.
   `fcu -uf /opt/motorola/rom/atca-9305-hpm.1-all.img`
   If the upgrade was successful, the IPMC switches to the new, upgraded firmware (without resetting or rebooting the payload).

4. Query the new image to ensure that the version information is correct,
   `fcu -q`

```
********************[[[[[REPORT BEGIN]]]]]********************
OPERATION : Query
RESULT    : SUCCESS
MESSAGE   : Device              : atca-9305-hpm.1-ipmc
            Part number         : 10000000-00-0
            Part revision       :

                BANK                : A - Operational
                Firmware Name       : ATCA9305 F/W
                Firmware Version    : 1.51.00000056


                BANK                : B - Rollback
                Firmware Name       : ATCA9305 F/W
                Firmware Version    : 1.51.00000055

                BANK                : D - Operational

                Firmware Name       : ATCA9305 B/L

                Firmware Version    : 1.51.00000056

********************[[[[[ REPORT END ]]]]]********************
```

A and B are the Firmware.

D is the Boot Loader.

5. Upgrade the FRU inventory data.

```
fcu -uf /opt/motorola/rom/atca9305-fruinfo-<version>.fri

current fru info saved to /tmp/atca-9305-hpm.1-ipmc-00-00.fru.

********************[[[[[REPORT BEGIN]]]]]********************

OPERATION : Verify

RESULT    : NOTICE

MESSAGE   : forcing partrevision

OPERATION : Verify

RESULT    : SUCCESS : compatible image found

MESSAGE   : OEM:EMERSON PROD:atca-9305 PN: 11000234-00-0 REV:  VER:1.0.0
NAME:fruinfo-bin

OPERATION : Write

RESULT    : SUCCESS UPGRADING

MESSAGE   : atca-9305-hpm.1-ipmc:EMERSON:atca-9305:11000234-00-
0:*:fruinfo-bin:1.0.0

********************[[[[[ REPORT END ]]]]]********************
```

FCU saves the previous FRU inventory data to the `/tmp/atca-9305-hpm.1-ipmc-00-00.fru` FRU file.

3.5.3 Management Processor Linux Kernel Upgrade

The FCU supports installation/upgrade of the Management Processors Linux kernel on the NOR flash. See also *Installation into the Flash Device* on page 31, where the root filesystem upgrade is explained. The Linux kernel has only one bank to update. Since you can always download another kernel via network under u-boot this is no security issue.

The following shows an example how fcu can be used to query and upgrade the Linux kernel:

Query the current Linux kernel in the flash

```
# fcu -q -d atca-9305-8548-kernel
********************[[[[[REPORT BEGIN]]]]]********************
OPERATION : Query
RESULT    : SUCCESS
MESSAGE   : Device              : atca-9305-8548-kernel
            Part number         : 10010654-00-B
            Part revision       :

                BANK                : A
                Firmware Name       : Linux Kernel
                Firmware Version    : 2.6.21.7-hrt1-WR2.0-
2.0_RC8_standard

********************[[[[[ REPORT END ]]]]]********************
```

Upgrade the Linux kernel to version 2.0 RC9

```
# fcu -uf atca-9305-8548-kernel-2.0_RC9_standard.fri
region 000000-01FFFF: ... OK
region 020000-03FFFF: ... OK
region 040000-05FFFF: ... OK
region 060000-07FFFF: ... OK
region 080000-09FFFF: ... OK
region 0A0000-0BFFFF: ... OK
region 0C0000-0DFFFF: ... OK
```

```
region 0E0000-0FFFFF: ... OK

region 100000-11FFFF: ... OK

region 120000-13FFFF: ... OK

region 140000-15FFFF: ... OK

region 160000-17FFFF: ... OK

region 180000-19FFFF: ... OK

region 1A0000-1BFFFF: ... OK

region 1C0000-1DFFFF: ... OK

region 1E0000-1FFFFF: ... OK

region 200000-208F8E: ... OK

********************[[[[[REPORT BEGIN]]]]]********************

OPERATION : Verify

RESULT    : NOTICE

MESSAGE   : forcing partrevision

OPERATION : Verify

RESULT    : SUCCESS : compatible image found

MESSAGE   : OEM:EMERSON PROD:atca-9305 PN: 10010654-00-B REV:
VER:2.6.21.7-hrt1-WR2.0-2.0_RC9_standard NAME:Linux Kernel

OPERATION : Write

RESULT    : SUCCESS UPGRADING

MESSAGE   : !atca-9305-8548-kernel:EMERSON:atca-9305:10010654-00-
B:*:Linux Kernel:2.6.21.7-hrt1-WR2.0-2.0_RC9_standard

********************[[[[[ REPORT END ]]]]]********************
```

3.5.4 Management Processor Device Tree Block (DTB) Upgrade

The FCU supports installation/upgrade of the Management Processors Device Tree Block (DTB) on the NOR flash. See also, *Installation into the Flash Device* on page 31, where the root filesystem upgrade is explained. The DTB has only one bank to update. Since you can always download another DTB via network under u-boot this is no security issue.

The following shows an example how fcu can be used to query and upgrade the DTB:

Query the current DTB in the flash

```
# fcu -q -d atca-9305-8548-dtb
********************[[[[[REPORT BEGIN]]]]]********************
OPERATION : Query
RESULT    : SUCCESS
MESSAGE   : Device              : atca-9305-8548-dtb
            Part number         : 10010654-00-B
            Part revision       :

                BANK                : A
                Firmware Name       : Device Tree Blob
               Firmware Version    : emerson_atca9305_8548-2.0_RC5

********************[[[[[ REPORT END ]]]]]********************
```

Upgrade the DTB to version 2.0 RC5

```
# fcu -uf atca-9305-8548-dtb-2.0_RC5.fri
region 000000-0012D3: ... OK
********************[[[[[REPORT BEGIN]]]]]********************
```

```
OPERATION : Verify

RESULT    : NOTICE

MESSAGE   : forcing partrevision

OPERATION : Verify

RESULT    : SUCCESS : compatible image found

MESSAGE   : OEM:EMERSON PROD:atca-9305 PN: 10010654-00-B REV:
VER:emerson_atca9305_8548-2.0_RC5 NAME:Device Tree Blob

OPERATION : Write

RESULT    : SUCCESS UPGRADING

MESSAGE   : !atca-9305-8548-dtb:EMERSON:atca-9305:10010654-00-
B:*:Device Tree Blob:emerson_atca9305_8548-2.0_RC5

********************[[[[[ REPORT END ]]]]]********************
```

4.1 Distribution Description

The blade services for the ATCA-9305 is based on Wind River Platform for Network Equipment (PNE) 2.0 Linux edition on Linux 2.6.21, which is a Linux distribution built on Linux 2.6 kernel technology,

4.2 Reliability

This distribution is configured to use the journaling file system `jffs2`. The majority of errors caused by improper shutdown are fixed automatically during boot.

This is only applicable in case of blade services software being installed in local Flash memory. It is not applicable when the software is installed on a remote server.

4.3 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* on page 58 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. Set `PermitRootLogin` in the `/etc/ssh/sshd_config` file to `yes`. For this to take effect you must either reboot the blade/module or run `/etc/init.d/ssh restart` command.

The following table lists available default login accounts.

Login Name	Password
root	root

4.4 Network Services Configuration

The following sections describe the default configuration for network services.

4.4.1 ATCA-9305 Ethernet Interfaces

4.4.1.1 8548 Management Ethernet Interface eth0

On startup, the eth0 interface is not configured.

If required the eth0 interface can easily be configured in u-boot by supplying the 'ip' parameter to the bootargs string, for example,

```
setenv bootargs root=/dev/ram0 rw ramdisk_size=600000
console=ttyS0,9600 ip=192.168.20.2:::255.255.255.0::eth0:
```

This sets the IP address 192.168.20.2 with netmask 255.255.255.0 on interface eth0. For more information, refer *Booting NFS-mounted Root File System on Management Processor* on page 30.

This setup allows to synchronize the IP address used in u-boot and in Linux. Alternatively the network configuration can be set under Linux in file `/etc/sysconfig/network-scripts/ifcfg-eth0` which needs to be created.

4.4.1.2 Switch Ethernet Interface eth1

On startup, the eth1 interface connected to the Broadcom switch is configured for DHCP for VLANs with VID 11, 12, 21, 22.

The configuration files for eth1 address management are under the following location: `/etc/sysconfig/network-scripts/ifcfg-eth1.<vid>`

For switch port configuration defaults, including VLAN assignments, see *Configuring SRstackware* on page 106.

4.4.2 SSH

A secure shell server starts in run levels 2–5 and listens on all the Ethernet interfaces listed in *ATCA-9305 Ethernet Interfaces* on page 58.

4.5 U-Boot Parameter

The u-boot parameter `bmc_wd_timeout` defines whether the IPMI watchdog shall be enabled when the operating system is loaded, and the time-out:

- If the parameter is undefined, or if it is set to -1, the IPMI watchdog is disabled before u-boot enters the command prompt or executes the boot command.
- If set to a value > 0, the IPMI watchdog is configured to the specified number of seconds before the boot prompt is executed. If no automatic boot is selected (that is, the command prompt is entered) the IPMI watchdog will not be enabled.

Under Linux, enable the IPMI watchdog by starting the watchdog daemon. This can be done by linking the `/etc/init.d/watchdog` script to your appropriate runlevel directory. You can modify the watchdog timeout in the sysfs entry `/sys/module/ipmi_watchdog/parameters/timeout`. Adjust the interval how often the watchdog is reset by the daemon in `/etc/watchdog.conf`. Restart the watchdog daemon.

For more information on watchdog, see *watchdog* on page 102.

5.1 Overview

Hardware management in ATCA systems is based on the Intelligent Platform Management Interface (IPMI) specification. IPMI commands can be complex and cumbersome. To facilitate blade-level management, Artesyn provides the Hardware Platform Management (HPM) package that provides a set of commands that are based on IPMI commands but which 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 that 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, hpmagentd
- Command line client, hpmcmd
- Script framework for managing shutdown and reboot events

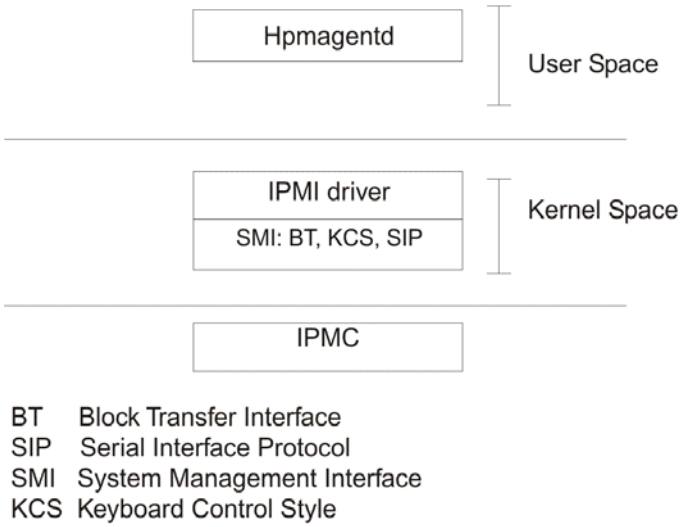
The hpmcmd sends a given HPM command to the hpmagentd and displays the received response on the console. The hpmagentd executes the incoming HPM commands and returns the result to a hpmcmd client.

HPM commands include:

- Retrieving and modifying FRU data
- Reading and controlling status of IPMI-controlled LEDs
- Relaying e-key events to Switch Agent software 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

The hpmagentd makes use of OpenIPMI driver to talk to the local IPMC. The following picture shows the software levels that are involved in the HPM architecture:

Figure 5-1 Software Levels of the HPM Architecture



The System Management Interface (SMI) driver provides the low level interface for talking to the IPMC and could be a KCS driver or Block Transfer (BT) driver or other.

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

5.2 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/motorola/bin/`. This directory has been added to the PATH environment variable.

This daemon has an init script called hpm that starts the daemon in 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 that 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 is `/opt/motorola/bin/hpmreboot` (see *hpm—Shutdown and Reboot Scripts* on page 65).

`-s shutdown-script`

Specifies a shutdown script that is 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 is `/opt/motorola/bin/hpmshutdown` (see *hpm—Shutdown and Reboot Scripts* on page 65).

`-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.

5.3 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/motorola/etc/init.d` directory during installation of the blade services software. It is also linked to `/etc/rec.d/recS.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.

5.4 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. The HPM agent listens for such requests from the IPMC. When it receives a 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 in the `/opt/motorola/bin` directory. Currently, these scripts simply print a banner indicating they have run and then issue `shutdown -h now` (hpmshutdown script) or `reboot` (hpmreboot script).

You may 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 in `/opt/motorola/etc/init.d/hpm`.

Synopsis

`hpmshutdown`

`hpmreboot`

5.5 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 in `/opt/motorola/bin` directory. During installation of the blade services software, this directory is added to the PATH environment variable.

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

From there you can start executing commands.

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` then, `-c` should be last option entered, since all arguments that follow `-c` on the command line will be considered part of the command.

5.5.1 Command Overview

The following table lists all commands from the hpmcmd program available on ATCA-9305. 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 70.

Table 5-1 Command Overview

Command	Description
bootbankget	Gets the bootbank to boot from
bootbankset	Sets the bootbank to boot from
bootparamerase	Erase boot parameter value
bootparamget	Get boot parameter value
bootparamset	Set a boot parameter value
bye	Exits the hpmcmd program
cmd	Executes any IPMI command
deviceid	Gets the Device Id
ekeydownpath	Gets the location/path of a script which is invoked on ekeying port down events
ekeyuppath	Gets the location/path of a script which is invoked on ekeying port up events
exit	Exits the hpmcmd program
frudata	Gets FRU info in hex numbers
fruinfoget	Gets string fields from the FRU
fruinfoset	Sets string fields of the FRU
fruinv	Gets the FRU size and addressable units
fruread	Reads x number of bytes from the FRU
fruwrite	Writes x number of bytes from the FRU
fwprogevent	Sends a Firmware Progress Sensor Event
help	Gets list of hpmcmd commands
ipmbaddress	Gets the local board IPMB address

Table 5-1 Command Overview (continued)

Command	Description
ipmcdevice	Gets the payload interface to the IPMC
ipmcstatus	Gets the IPMC status
ledget	Gets the state of a specific FRU LED
ledprop	Get the LED properties for this FRU
ledset	Controls the state of a specific FRU LED
loglevelget	Gets the hpmagentd log level
loglevelset	Sets the hpmagentd log level (0-7)
motshelftype	Gets the Artesyn Shelf Type from the Shelf FRU (Board Product Name)
partnumber	Gets the board part number
physlotnumber	Gets the board physical slot number
portget	Gets the current state E-Key governed intfs
portset	Enables/Disables ports in a channel
posttypeget	Gets the posttype to run at boot
posttypeset	Sets the posttype to run at boot
quit	Exits the hpmcmd program
rebootpath	Gets hpmagentd reboot script path
sdr	Gets the SDR records
sdr_dump	Gets the SDR records in raw format
sdrinfo	Gets SDR information
sendamc	Sends a IPMI request to a MMC behind a remote IPMC
sendcmd	Sends an IPMI request to a IPMB address IPMC
shelfaddress	Gets the Shelf Address String
shelfslots	Gets number of slots in the shelf
shutdownpath	Gets hpmagentd shutdown script path

Table 5-1 Command Overview (continued)

Command	Description
slotmap	Gets the slotmap of the shelf
slotnumber	Gets the board logical slot number
version	Gets the hpmcmd version and the hpmagentd version
watchdog	Controls Payload WDT functionality

5.5.2 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 "`-t 0`" and "`-t lc`" does the same.

To send the command to an AMC attached on the blade which runs hpmagentd you can use:

```
-t0:c0  or -tlc:c0
```

To send the command to another IPMC on the IPMB you can use:

```
-t 92
```

To send the command to an MMC which is attached on another blade on the IPMB you can use:

```
-t 92:c0
```

5.5.2.1 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 Artesyn ATCA 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.

This command works only for CPU = 0, which is the Management Controllers CPU. It is not supported on the OCTEONS.

Example

```
hpmcmd -c bootbankget 0
Bank1 - first bank in NOR flash
Bank2 - second bank in NOR flash
Socket - socketed flash
```

5.5.2.2 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 `BANK1`, `BANK2`, or `SOCKET`.

You should not set the boot bank to socket, if the socketed flash is empty.

Example

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

5.5.2.3 bootparamerase

Description

This command erases the boot parameter.

Synopsis

```
bootparamerase section [name] [-t ipmbAddr[:mmcAddr]]
```

Parameters

`section`

Can have value as `USER`, `DEFAULT`, `TEST`, or `OS_PARAM`.

`name`

Specifies name of the parameter.

`t`

Sends the command to `ipmbAddr:mmcAddr`.

5.5.2.4 bootparamget

Description

This command gets the boot parameter value.

Synopsis

```
bootparamget section [name] [-t ipmbAddr[:mmcAddr]]
```

Parameters

`section`

Can have value as `USER`, `DEFAULT`, `TEST`, or `OS_PARAM`.

`name`

Specifies name of the parameter.

`t`

Sends the command to `ipmbAddr:mmcAddr`.

5.5.2.5 bootparamset

Description

This command sets the boot parameter value.

Synopsis

```
bootparamset section name=value [-t ipmbAddr[:mmcAddr]]
```

Parameters

`section`

Can have value as `USER, TEST`, or `OS_PARAM`.

`name`

Specifies name of the parameter.

`t`

Sends the command to `ipmbAddr:mmcAddr`.

5.5.2.6 bye

Description

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

Synopsis

```
bye
```

5.5.2.7 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 command`

The ipmi command specifies the sequence of hexadecimal bytes as entered using the ipmicmd tool from the OpenIPMI library. The ipmi command can have value, such as:

```
0f 00 XX ZZ W1 W2 ... Wn
```

In this example:

XX specifies netfunct in hexadecimal.

ZZ specifies the command number, as stated in the IPMI/PICMG spec.

W1 to Wn specifies the data bytes according to the command supports.

`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>, where SA is the slave address.

`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 AMC attached on this blade. MMC address of the AMC is 7a:
`hpmcmd -c cmd 7 7a 0 6 1`

5.5.2.8 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 `ipmbAddre:mmcAddr`.

Example

```
hpmcmd -c deviceid
```

5.5.2.9 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
```

5.5.2.10 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
```

5.5.2.11 exit

Description

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

Synopsis

```
exit
```

5.5.2.12 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 rear transition module (RTM) or AMC.

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

Example

```
hpmcmd -c frudata 0
```

5.5.2.13 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 rear transition module (RTM) or AMC.

`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 area manufacturer
bproductname	Board area product name
bserialnumber	Board area serial number
bpartnumber	Board area part number
pmanufacturer	Product area manufacturer
pproductname	Product area product name
ppartnumber	Product area part number
pversion	Product area version
pserialnumber	Product area serial number
passettag	Product area asset tag

`-v`

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

`-t`

Sends the command to `ipmbAddr:mmcAddr.`

Example

```
hpmcmd -c fruinfoget 0 -v

Common Header:

    Format Version = 1
```

```
Internal Use Area:

    Version = 1

Board Info Area:

    Version   = 1

    Language Code            = 25

   Mfg Date/Time           = Apr 17 12:00:00 2008 (6466320 minutes
since 1996)

    Board Manufacturer       = Emerson Network Power - Embedded
Computing, Inc.

    Board Product Name       = ATCA-9305

    Board Serial Number      = 0123456789

    Board Part Number        = 10000234-00-0

    FRU Programmer File ID   = fru-info.inf


Product Info Area:

    Version   = 1

    Language Code            = 25

    Manufacturer Name        = Emerson Network Power - Embedded
Computing, Inc.

    Product Name             = ATCA-9305

    Product Part / Model#    = 11000234-00-0

    Product Version          =

    Product Serial Number    = 730-1234

    Asset Tag                =
```

```
    FRU Programmer File ID   = fru-info.inf


Multi Record Area:


    PICMG Board Point-to-Point Connectivity Record (ID=0x14)
        Version = 0
    OEM GUID Count          = 0
    Link Descriptor:
       Link Grouping ID     =  0x00
      Link Type           =  0x01  PICMG3.0 Base 10/100/1000 Base-T
     Link Type Extension  =  0x0 10/100/1000BASE-T Link (four-pair)
       Link Designator      =  0x101 Channel1/BaseInterface/Ports0
    Link Descriptor:
       Link Grouping ID     =  0x00
      Link Type           =  0x01  PICMG3.0 Base 10/100/1000 Base-T
     Link Type Extension  =  0x0 10/100/1000BASE-T Link (four-pair)
       Link Designator      =  0x102 Channel2/BaseInterface/Ports0
    Link Descriptor:
       Link Grouping ID     =  0x00
       Link Type            =  0x02  PICMG3.1 Ethernet Fabric
       Link Type Extension  =  0x1
       Link Designator      =  0xF41
Channel1/FabricInterface/Ports0123
```

```
    Link Descriptor:
       Link Grouping ID     =  0x00
       Link Type            =  0x02  PICMG3.1 Ethernet Fabric
       Link Type Extension  =  0x0
      Link Designator     =  0x141 Channel1/FabricInterface/Ports0
    Link Descriptor:
       Link Grouping ID     =  0x00
       Link Type            =  0x02  PICMG3.1 Ethernet Fabric
       Link Type Extension  =  0x1
       Link Designator      =  0xF42
Channel2/FabricInterface/Ports0123
    Link Descriptor:
       Link Grouping ID     =  0x00
       Link Type            =  0x02  PICMG3.1 Ethernet Fabric
       Link Type Extension  =  0x0
      Link Designator     =  0x142 Channel2/FabricInterface/Ports0


    AMC Carrier Information Table Record (ID=0x1a)
        Version = 0
    Site Count = 1
    Site Numbers: 0x01
```

```
    AMC Carrier Activation and Current Management Record (ID=0x17)
        Version = 0
    Maximum Internal Current Draw = 12.5 Amps at 12V Payload
    Allowance for Module Activation Readiness = 5 seconds
    Module Activation and Power Descriptor count = 1
      Local IPMB-L 0x72, Max Module Current 6.7 Amps, Config Value:
0xff
```

5.5.2.14 fruinfoset

Description

This command sets some individual field in a FRU or replaces the whole content of the FRU from a file.

Synopsis

```
fruinfoset <fruid> -f <frufilepath> [-t ipmbAddr[:mmcAddr]]
fruinfoset <fruid> <field> <newvalue> [-t ipmbAddr[:mmcAddr]]
```

Parameters

`fruid`

Is 0 for the main blade and other for the RTM or AMC.

`frufilepath`

Specifies full path of the FRU info binary file, when using a file.

`field`

Is one of the following data fields. If no field is specified, it retrieves the whole fruinfo for that FRU. Refer *fruinfoget* on page 77, for the list of data fields.

`newvalue`

Specifies the new value to be set. It should be less than 16 bytes.

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

5.5.2.15 fruinv

Description

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

Synopsis

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

Parameters

`fruid`

Is 0 for the main blade and 1 for the rear transition module (if present) or AMC.

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

Example

```
hpmcmd -c fruinv 0
```

5.5.2.16 fruread

Description

This command gets nBytes of fruid from the startAddress of the specified FRU.

Synopsis

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

Parameters

`fruid`

Is 0 for the main blade and 1 for the rear transition module (if present) or AMC.

`startAddress`

Is the starting address for reading the fruid.

`nBytes`

Number of bytes to read in decimal.

`-t`

Sends the command to `ipmbAddr:mmcAddr`.

Example

```
hpmcmd -c fruread 0 0 280
```

5.5.2.17 fruwrite

Description

This command allows to write hexadecimal byte values to `fruid` starting at `startAddr`.

Synopsis

```
fruwrite <fruid> <startAddress> <hexval1> [hexval2] [...]
[hexval16] [-t ipmbAddr[:mmcAddr]]
```

Parameters

`fruid`

Is 0 for the main blade and other for the RTM or AMC.

`startAddress`

Starting address for writing.

`hexval1 .. hexvalN`

Is the hexadecimal value to write.

`-t`

Sends the command to `ipmbAddr:mmcAddr.`

5.5.2.18 fwprogevent

Description

This command sends a Firmware Progress Sensor Event to the Shelf Manager SEL. Refer IPMI specifications for details on values.

Synopsis

```
fwprogevent <data1> <data2> <data3>
```

Parameters

`data1`

Stores hexadecimal value as; 00 for Error, 01for Hang, and 02 for Progress.

`data2`

Stores hexadecimal value as; 00-0D for Error, 00-19 for Hang or Progress.

`data3`

Stores hexadecimal value as; FF unless an OEM data2 is specified.

5.5.2.19 help

Description

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

Synopsis

```
help
```

5.5.2.20 ipmbaddress

Description

This command retrieves the blade IPMB address.

Synopsis

```
ipmbaddress
```

5.5.2.21 ipmcdevice

Description

This command shows the payload interface to the IPMC.

Synopsis

```
ipmcdevice
```

5.5.2.22 ipmcstatus

Description

This command retrieves the status of given IPMC.

Synopsis

```
ipmcstatus [-t ipmbAddr]
```

Parameters

`-t`

Specifies the target with `ipmbAddr`.

Example

```
hpmcmd -c ipmcstatus
```

5.5.2.23 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 rear transition module (if present).

`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.`

Example

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

5.5.2.24 ledprop

Description

This command displays the FRU LED properties under IPMC control.

Synopsis

```
ledprop <fruid>
```

Parameters

`fruid`

Is `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
```

5.5.2.25 ledset

Description

This command controls the override state of a specific FRU LED.

Synopsis

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

Parameters

`fruid`

Is `0` for the main blade and `1` for the rear transition module (if present) or AMC.

`led`

Is BLUE for the hot swap LED or LEDN for FRU LED<n>. <n> is a number between 0 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. The default duration is 5000ms.

`offms`

Specifies OFF duration in milliseconds. It can have value from 10ms to 2500ms in the 10ms increments. It is valid only if the operation is BLINK.

`onms`

Specifies ON duration in milliseconds. It can have value from 10ms to 2500ms in the 10ms increments. It is valid only if the operation is BLINK.

`color`

LED0 = BLUE
LED1 = RED or AMBER
LED2 = GREEN (if supported by IPMC)
LED3 = AMBER (if supported by IPMC)

`-t ipmbAddr`

Sends the command to `ipmbAddr:mmcAddr`.

Example

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

5.5.2.26 loglevelget

Description

This command retrieves the current `hpmagentd` log level. Refer *loglevelset* on page 90 for more details.

Synopsis

```
loglevelget
```

Example

```
hpmcmd -c loglevelget

Loglevel 5 (NOTICE)
```

5.5.2.27 loglevelset

Description

This command sets the hpmagentd log level to a desired level.

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
```

5.5.2.28 macaddress

Description

This command retrieves a list of available MAC addresses.

Synopsis

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

Parameter

`-t` - sends the command to ipmbAddr

5.5.2.29 motshelftype

Description

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

Synopsis

```
motshelftype
```

Example

```
hpmcmd -c motshelftype
CHS1406
```

5.5.2.30 partnumber

Description

This command retrieves the part number (FRU 0) of the main blade.

Synopsis

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

Parameters

`-t ipmbAddr`

Sends the command to `ipmbAddr:mmcAddr`.

Example

```
hpmcmd -c partnumber
```

5.5.2.31 physlotnumber

Description

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

Synopsis

```
physlotnumber
```

5.5.2.32 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] [devid] [-t ipmbAddr[:mmcAddr]]
```

Parameters

`interface`

Valid values are:

```
BASE | FABRIC | UPDATE | AMC
```

`channel`

It is a number from 1 to the maximum number of channels supported for the interface. Node blades usually support 2 Base and 2 Fabric channels, and switch blades support 16 Base, 15 Fabric, and 1 Update channels.

`devid`

For AMC only: it is an on-Carrier device ID that identifies the on-Carrier device to which the desired channel is connected.

`-t ipmbAddr`

Sends the command to `ipmbAddr:mmcAddr`.

Example

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

5.5.2.33 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> [devid]
[-t ipmbAddr[:mmcAddr]]
```

Parameters

`intf`

Valid values are:

```
BASE | FABRIC | UPDATE | AMC
```

`chan`

It is a number from 1 to the maximum number of channels supported for the interface. Node blades usually support 2 Base and 2 Fabric channels, and switch blades support 16 Base, 15 Fabric, and 1 Update channels.

`grpid`

Specifies the group id. Always 0 according to the current shelf FRU information.

`type`

Valid values are:

```
BASE | ETHER | EXPRESS | INFINI | STAR | OEM
```

`typeX`

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 (for port 0)
Option 2: 01 (for ports 0,1)
Option 3: 0123 (for ports 0,1,2,3)
Option 7: 3 (for port 3)

`oper`

Valid values are `DISABLE` or `ENABLE`.

`devid`

For AMC only: it is an on-Carrier device ID that identifies the on-Carrier device to which the desired channel is connected.

`-t ipmbAddr`

Sends the command to `ipmbAddr:mmcAddr`.

Example

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

5.5.2.34 posttypeget

Description

This command retrieves the postType to which the board is currently set to run at boot time, for the specified CPU.

Synopsis

```
posttypeget <payload_cpu_selector>
```

Parameters

`payload_cpu_selector`

The specified CPU is set to postType to run.

5.5.2.35 posttypeset

Description

This command sets the postType to which the board is currently set to run at boot time, for the specified CPU.

Synopsis

```
posttypeset <payload_cpu_selector> <newPostType>
```

Parameters

`payload_cpu_selector`

It is an integer between 0 and number of CPU devices supported per board.

`newPostType`

Valid values are: `SHORT | LONG`.

5.5.2.36 quit

Description

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

Synopsis

```
quit
```

5.5.2.37 rebootpath

Description

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

Synopsis

```
rebootpath
```

Example

```
hpmcmd -c rebootpath
/opt/motorola/bin/hpmreboot
```

5.5.2.38 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
```

5.5.2.39 sdr_dump

Description

This command shows the SDR records in binary and hexadecimal 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......Í"
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"
```

5.5.2.40 sdrinfo

Description

This command shows the SDR information.

Synopsis

```
sdrinfo
```

Example

```
hpmcmd -c sdrinfo

SDR Information:
LUN 0 has 031 sensors; static sensor population
LUN 1 has 000 sensors;
LUN 2 has 000 sensors;
LUN 3 has 000 sensors;
```

5.5.2.41 sendamc

Description

This command allows to send any of the commands supported in the IPMI specifications 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.

5.5.2.42 sendcmd

Description

This command allows a user to send any of the commands supported in the IPMI specifications 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
```

5.5.2.43 shelfaddress

Description

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

Synopsis

```
shelfaddress
```

Example

```
hpmcmd -c shelfaddress
01
```

5.5.2.44 shelfslots

Description

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

Synopsis

```
shelfslots
```

Example

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

5.5.2.45 shutdownpath

Description

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

Synopsis

```
shutdownpath
```

Example

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

5.5.2.46 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
------------------------------------------------------------
```

5.5.2.47 slotnumber

Description

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

Synopsis

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

Parameters

`-t ipmbAddr`

Sends the command to `ipmbAddr:mmcAddr`.

Example

```
hpmcmd -c slotnumber
2
```

5.5.2.48 updmacaddress

Description

This command updates the mac addresses in the FRU info.

Synopsis

```
updmacaddress [base address]
```

Parameter

`base address` - optional specify the base address

5.5.2.49 version

Description

This command retrieves the version of the hpm 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
hpmagentcmd version bbs 1.3.9 build 5.windriver
```

5.5.2.50 watchdog

Description

This command is used to 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`

Possible values are

Value	Description
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

6.1 Overview

The Switching and Routing stackware (SRstackware®) is a protocol suite that is installed as a part of blade services installation on the ATCA blade. SRstackware provides features to perform ATCA-9305 switch configuration and check the statistics in addition to the protocol stack support. SRstackware 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)
- Router Information Protocol next generation (RIPng)
- SNMP support
- Linux OS version Wind River PNE 2.0 support with Broadcom BCM SDK 5.5.5
- ATCA-9305 switching blade

Refer Appendix A, *Related Documentation, on page 127*, for the documents and sections to be referred for using the protocols and features defined in the current SRstackware release.

6.2 Configuring SRstackware

SRstackware, after installation, comes up with the default configuration. You can update the configuration using CLI.

6.2.1 Default Configuration

The ATCA-9305 default configuration creates one bridge and multiple VLANs. All the VLAN ports are associated to the bridge. You can configure additional logical bridges and VLANs, if required. The switch ports are named as ge# for the 1G ports and xe# for the 10G ports. These switch ports are accessible using the ATCA-9305 CLI.

You can modify the default configuration using the ATCA-9305 CLI. The default configuration file is saved at `/mnt/flash/srstackware/config/srs.cfg`. This file is persistent across reboots and SRstackware picks up this file while bootup to restore its configuration. If the administrator issues `write file` command through imish it saves the configuration to this file. It is advised to save a copy of the original configuration file for a later use.

By default the ATCA-9305 does not enable any protocol on the ports.

The following table lists the switch ports and associated VLANs.

Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
ge1	SGMII	base0 interface	21(u)	YES
ge2	SGMII	base1 interface	22(u)	YES
ge3	SGMII	Front Panel Switch Port	1(u)	YES
ge4	SGMII	BBP	11(t), 12(t), 21(t), 22(t)	YES
xe1	XAUI	Octeon 1 SPI-4.2 IF2	2(u)	YES
xe2	XAUI	Octeon 1 SPI-4.2 IF1	11(u)	YES
xe3	XAUI	fabric1 interface	12(u)	YES
xe4	XAUI	fabric0 interface	11(u)	YES
xe5	XAUI	2x10G RTM-Not Connected / 6x10G RTM-ETH6	2(u)	NO

Interface name	Interface description	Destination	VLAN(s) assigned	Default Enabled
xe6	XAUI	2x10G RTM-Not Connected / 6x10G RTM-ETH5	2(u)	NO
xe7	XAUI	Octeon 2 SPI-4.2 IF1	12(u)	YES
xe8	XAUI	Octeon 2 SPI-4.2 IF2	3(u)	YES
xe9	XAUI	2x10G RTM-SFP+ Port1 / 6X10G RTM-ETH2	3(u)	NO
xe10	XAUI	2x10G RTM-SFP+ Port2 / 6X10G RTM-ETH4	3(u)	NO
xe11	XAUI	2x10G RTM-Not Connected / 6X10G RTM-ETH1	3(u)	NO
xe12	XAUI	2x10G RTM-Not Connected / 6X10G RTM-ETH3	3(u)	NO

When a VLAN-aware bridge is created and assigned to a switch port interface; SR stackware adds the interface to the Linux kernel as a network interface. The interface is named as `vlanx.y`, where x specifies the bridge name and y specifies the VLAN id.

6.2.2 Custom Configuration

You can customize the ATCA-9305 after starting the ATCA-9305 CLI using the following steps:

1. Login as `root` at the ATCA-9305.
2. Run `/opt/srstackware/bin/imish`.

A sample output of the above steps is shown below.

```
root@Slot-1_9305:/opt/srstackware/bin>./imish

ZebOS version 7.6.1.e0 IPIRouter 05/11/09 22:33:20

Slot-1_9305>enable

Slot-1_9305#show interface
```

6.2.2.1 Enabling Protocols

You can enable protocols on switch port of ATCA-9305. By default, the protocols are disabled. For proper functionality of the bridge, you can enable any protocol and the respective interfaces, on the bridge.

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.

6.2.2.2 Sample Configurations

This section provides a quick reference to the sequence of CLI commands for enabling protocols on the RTM ports.

- To check if the link at ge1 interface is up or down:

```
root@Slot-1_9305:/opt/srstackware/bin>./imish

ZebOS version 7.6.1.e0 IPIRouter 05/11/09 22:33:20

Slot-7_9305>en
Slot-7_9305#show interface ge1

Interface ge1

  Hardware is Ethernet, address is 00f7.2214.0003 (bia 00f7.2214.0003)

  Description: BC1

  index 5001 metric 1 mtu 28836 duplex-full arp ageing timeout 0

  <UP,BROADCAST,RUNNING,MULTICAST>

  VRF Binding: Not bound

  Bandwidth 1g

   input packets 012480, bytes 0799356, dropped 00, multicast packets
044515

    output packets 032106, bytes 02054932, multicast packets 032106
broadcast packets 066
```

In this sample output, RUNNING in the following line indicates that the interface link is up.

```
<UP,BROADCAST,RUNNING,MULTICAST>
```

In case the link is down the sample output contains the following line.

```
 <UP,BROADCAST,MULTICAST>
```

- To show all the interfaces (switch ports):

```
root@Slot-1_9305:/opt/srstackware/bin>./imish

ZebOS version 7.6.1.e0 IPIRouter 05/11/09 22:33:20

Slot-1_9305>en
Slot-1_9305#show interface
```

You can use `include` and `begin` options with this command to modify the output and display it.

- To configure a bridge with STP enabled on it:

```
root@Slot-1_9305:/opt/srstackware/bin>./imish

ZebOS version 7.6.1.e0 IPIRouter 05/11/09 22:33:20

Slot-1_9305>en
Slot-1_9305#conf t

Enter configuration commands, one per line.  End with CNTL/Z.

Slot-1_9305(config)#bridge 1 protocol ieee
Slot-1_9305(config)#
```

- To configure a VLAN and associate it to a bridge:

```
root@Slot-1_9305:/opt/srstackware/bin>./imish

ZebOS version 7.6.1.e0 IPIRouter 05/11/09 22:33:20

Slot-1_9305>en
Slot-1_9305#conf t

Enter configuration commands, one per line.  End with CNTL/Z.

Slot-1_9305(config)#bridge 1 protocol ieee vlan-bridge
Slot-1_9305(config)#vlan database
Slot-1_9305(config-vlan)#vlan 23 bridge 1 name VLAN23 state
enable
Slot-1_9305(config-vlan)#
```

- To configure the VLAN 23 (mentioned in the above sample) to a switch port:

```
root@Slot-1_9305:/opt/srstackware/bin>./imish

ZebOS version 7.6.1.e0 IPIRouter 05/11/09 22:33:20

Slot-1_9305>en
Slot-1_9305#
Slot-1_9305#conf t

Enter configuration commands, one per line.  End with CNTL/Z.

Slot-1_9305(config)#interface ge1
Slot-1_9305(config-if)#switchport mode access
Slot-1_9305(config-if)#switchport access vlan 23
```

- To enable STP on the base RTM port 1:
  To configure an L2 protocol, such as STP, RSTP, or MSTP on a switch 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 VLAN configuration, tag it with `vlan-bridge` (as mentioned in the code below).
  - 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.
  - Create the VLANs.

```
root@Slot-1_9305:/opt/srstackware/bin>./imish

ZebOS version 7.6.1.e0 IPIRouter 05/11/09 22:33:20

Slot-1_9305>enable
Slot-1_9305#
Slot-1_9305#configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

/*** Associate STP to bridge-1 as part of Step (2) ***/
Slot-1_9305(config)#bridge 1 protocol ieee vlan-bridge
```

```
/* Go to interface mode */
Slot-1_9305(config)#interface ge1

/* Remove bridge so that STP can be enabled  */
Slot-1_9305(config-if)# no bridge-group 1

/* Enables STP on the interface as part of Step (3) */
Slot-1_9305(config-if)#bridge-group 1

/* Re-configure the necessary VLANs; because VLANs disappear if
bridge is removed */
Slot-1_9305(config-if)#switchport mode hybrid

/* As a part of Step (4) */
Slot-1_9305(config-if)#switchport hybrid vlan 93

Slot-1_9305(config-if)#switchport mode hybrid acceptable-frame-
type all
Slot-1_9305(config-if)#switchport hybrid allowed vlan add 93
egress-tagged disable
/* Enable the interface */
Slot-1_9305(config-if)#no shutdown

Slot-1_9305(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.

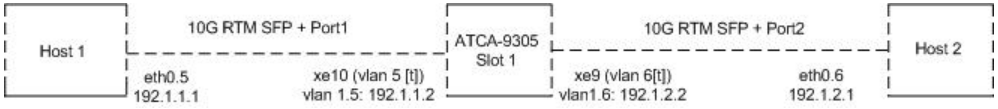
```
/* Associate RSTP to bridge-1 */
Slot-1_9305(config)# bridge 1 protocol rstp vlan-bridge
```

- To enable MSTP on the fabric RTM port 1:

```
root@Slot-1_9305:/opt/srstackware/bin>./imish

ZebOS version 7.6.1.e0 IPIRouter 05/11/09 22:33:20

Slot-1_9305>en
Slot-1_9305#
```

```
Slot-1_9305#conf t

Enter configuration commands, one per line.  End with CNTL/Z.

Slot-1_9305(config)#bridge 1 protocol mstp vlan-bridge
Slot-1_9305(config)#interface xe4
SLOT-1_9305(config-if)# no bridge-group 1
SLOT-1_9305(config-if)# bridge-group 1
Slot-1_9305(config-if)#no shutdown
Slot-1_9305(config-if)#
```

- To enable static routing through VLAN interfaces within the following topology.



On Host 1:

```
vconfig add eth0 5

ifconfig eth0.5 192.1.1.1/24

route add -net 192.1.2.0/24 gw 192.1.1.2 dev eth0.5
```

On ATCA-9305 Slot 1:
Create VLAN 5 and 6, with intervlan routing enabled, and associate it to Bridge 1.

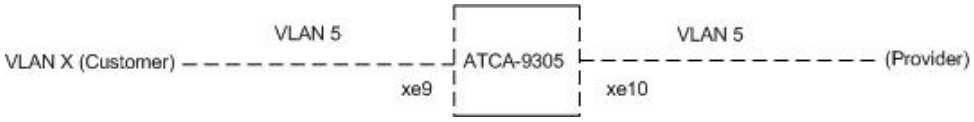
```
/** If static routing through VLANs has to be done through ports of two
fabric chip sets, then these VLANs should be added on hi-gig using "vlan
<vlan-id> hi-gig". **/

Slot-1_9305>en

Slot-1_9305#conf t

Enter configuration commands, one per line.  End with CNTL/Z.

Slot-1_9305(config)#interface xe10

Slot-1_9305(config-if)#bridge-group 1 spanning-tree disable

Slot-1_9305(config-if)#switchport mode trunk
```

```
Slot-1_9305(config-if)#switchport trunk allowed vlan add 5
Slot-1_9305(config-if)#no shutdown
Slot-1_9305(config-if)#exit
Slot-1_9305(config)#interface vlan1.5
Slot-1_9305(config-if)#ip address 192.1.1.2/24
Slot-1_9305(config-if)#no shutdown
Slot-1_9305(config-if)#exit

Slot-1_9305(config)#interface xe9
Slot-1_9305(config-if)#bridge-group 1 spanning-tree disable
Slot-1_9305(config-if)#switchport mode trunk
Slot-1_9305(config-if)#switchport trunk allowed vlan add 6
Slot-1_9305(config-if)#no shutdown
Slot-1_9305(config-if)#exit
Slot-1_9305(config)#interface vlan1.6
Slot-1_9305(config-if)#ip address 192.1.2.2/24
Slot-1_9305(config-if)#no shutdown
Slot-1_9305(config-if)#exit
Slot-1_9305(config)#ip forwarding
```

On Host 2:

```
vconfig add eth0 6
ifconfig eth0.6 192.1.2.1/24
route add -net 192.1.1.0/24 gw 192.1.2.2 dev eth0.6
```

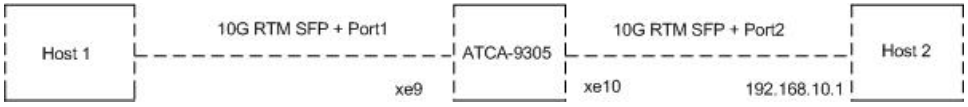
- To enable Q-in-Q (VLAN stacking) on the ATCA-9305 with the following topology.




```
Slot-1_9305>en
Slot-1_9305#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Slot-1_9305(config)#bridge 1 protocol ieee vlan-bridge
Slot-1_9305(config)#vlan database
Slot-1_9305(config-vlan)#vlan 5 bridge 1 state enable
Slot-1_9305(config-vlan)#exit

Slot-1_9305(config)#interface xe9
Slot-1_9305(config-if)#no shutdown
Slot-1_9305(config-if)#bridge-group 1
Slot-1_9305(config-if)#switchport mode access
Slot-1_9305(config-if)#switchport vlan-stacking customer-edge-port
Slot-1_9305(config-if)#switchport access vlan 5
Slot-1_9305(config-if)#exit

Slot-1_9305(config)#interface xe10
Slot-1_9305(config-if)#no shutdown
Slot-1_9305(config-if)#bridge-group 1
Slot-1_9305(config-if)#switchport mode trunk
```

```
Slot-1_9305(config-if)#switchport mode trunk allowed vlan add 5

Slot-1_9305(config-if)#switchport vlan-stacking provider-port

Slot-1_9305(config-if)#exit
```

- To enable VLAN Classification based on Source IP network with the following topology.




```
Slot-1_9305>en

Slot-1_9305#conf t

Enter configuration commands, one per line.  End with CNTL/Z.

Slot-1_9305(config)#bridge 1 protocol ieee vlan-bridge

Slot-1_9305(config)#vlan database

Slot-1_9305(config-vlan)#vlan 5 bridge 1 state enable

Slot-1_9305(config-vlan)#exit


Slot-1_9305(config)# vlan classifier rule 2 ipv4 192.168.10.0/24 vlan 5

Slot-1_9305(config)# vlan classifier group 10 add rule 2


Slot-1_9305(config)#interface xe9

Slot-1_9305(config-if)#bridge-group 2

Slot-1_9305(config-if)#switchport mode trunk

Slot-1_9305(config-if)#switchport trunk allowed vlan add 5

Slot-1_9305(config-if)# vlan classifier rule 2 ipv4 192.168.10.0/24
vlan 5
```

```
Slot-1_9305(config-if)#no shutdown

Slot-1_9305(config-if)#exit
```

6.2.3 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 ATCA-9305.

In case of 9305, by default only Bridge1 is created.

Example of a `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.

The following are the mibs, among the currently supported mibs for which context needs to be specified as `Bridge_number`:

- 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 (Artesyn Proprietory MIB)

6.2.4 SNMP Traps Usage Guidelines

SNMP traps and SNMP Inform requests are the two SNMP notifications supported with the SRstackware package. The following sections detail the configurations to be made at SNMP agent and manager in order to facilitate these notifications.

6.2.4.1 Configuring Agent to Send SNMP Notifications

In order to configure SRstackware to send SNMP traps/Inform requests, a shell command has been provided as part of SRstackware package.

This shell command has the following syntax:

```
#srs_trap_config <IPv4 Address> <NONE/TRAP/INFORM>
```

`NONE`: Neither SNMP traps nor Inform requests are sent to the SNMP manager running on machine with IPv4 Address.

`TRAP`: Trap messages are sent to SNMP manager running on machine with IPv4 Address.

`INFORM`: SNMP Inform requests are sent to SNMP manager running on machine with IPv4 Address.

Example usage of the command:

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
```

The keywords NONE/TRAP/INFORM are case sensitive.

6.2.4.2 Configuring SNMP Manager to Receive SNMP Notifications

The snmptrapd daemon needs to be run at the receiver to receive the SNMP notifications sent by the agent. snmptrapd needs a configuration file to be supplied with proper configuration in order to receive the traps properly. snmptrapd configuration file must have the following two lines:

```
createUser -e <Agent's Engine-ID> admin MD5 adminpwd123 DES
```

```
authUser log,execute admin
```

`Agent's Engine-ID`: The Engine-ID with which SNMP Agent is running.

To obtain the Agent's Engine-ID, issue the following command:

```
snmpget -v3 -u admin -A adminpwd123 <Agent's IP address>
snmpEngineID.0
```

The result of this command will be in Hex-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
```

Output:

```
SNMP-FRAMEWORK-MIB::snmpEngineID.0 = Hex-STRING: 80 00 1F 88 04 65
6D 65 72 73 6F 6E
```

Now, snmptrapd configuration file will look as follows:

```
createUser -e 0x80001F8804656D6572736F6E admin MD5 adminpwd123 DES
```

```
authUser log,execute admin
```

snmptrapd may be run at the receiver (SNMP manager) as follows (the formatting options can be modified as you desire, refer the man pages of snmptrapd for details):

```
/usr/sbin/snmptrapd -x tcp:localhost:705 -c
"<snmptrapd_configuration_file>" -C -f -Le
```

6.2.5 Configuring Logs

You can configure ATCA-9305 to store the logs in a specified file or `syslog`, using the `log` command. The log file is rotated for 1MB of file size, as a general practice the logs are stored in `/mnt/flash/log/messages`. As ATCA-9305 has limited flash space, it is recommended to configure ATCA-9305 to store logs on a remote syslog server.

6.3 Standards Supported

Table 6-1 Standards Supported

Protocol	Standard
STP	802.1d
RSTP	802.1w
MSTP	802.1s
GVRP	802.1Q
GMRP	802.1Q
IGMP Snooping	RFC 4541
LACP	802.3ad
OSPFv2	RFC 2328
OSPF NSSA Option	RFC 3101
OSPF Graceful Restart	RFC 3623
OSPF Restart Signaling	draft-nguyen-ospf-restart-05

Table 6-1 Standards Supported (continued)

Protocol	Standard
OSPF Opaque LSA Option	RFC 2370
OSPF Alternative Implementation of OSPF ABR	RFC 3509
IGMPv3	RFC 3376 and draft-ietf-magma-igmp-proxy-06
QoS	802.1p
RIP	RFC 2453
RIPng	RFC 2080
VRRP	RFC 3768
SNMP MIBS	
Bridge-MIB	RFC 4188
P-Bridge-MIB	RFC 4363
Q-Bridge-MIB	RFC 4363
IF-MIB	RFC 2863
OSPF-MIB	RFC 1850
IEEE8023-LAG-MIB	IEEE 802.3ad MIB
RSTP-MIB	RFC 4318
RIP MIB	RFC 1724
VRRP MIB	draft-ietf-vrrp-unified-mib-06

6.4 2x10G/12x1G RTM Ports

Table 6-2 ARTM-9305 2x10G/12x1G

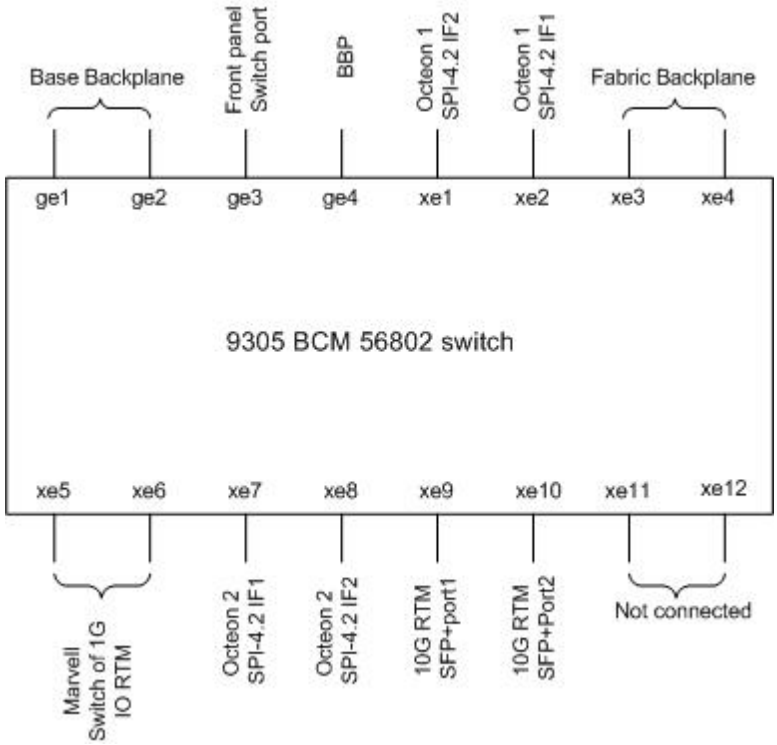
ATCA-9305				ARTM-9305 2x10G, 12x1G			
Description	BCM56802 Port	SRS Port	U-boot Port	Description	Marvell Port	SRS Port	U-boot Port
Base	1	ge1	0				
Base	2	ge2	1				
Front Panel RJ45	3	ge3	2				
MPC8548	4	ge4	3				
Octeon 1 SP I-1	5	xe1	4				
Octeon 1 SP I-0	6	xe2	5				
Fabric1	7	xe3	6				
Fabric0	8	xe4	7				
not connected on BCM56803	9						
not connected on BCM56803	10						

Table 6-2 ARTM-9305 2x10G/12x1G (continued)

ATCA-9305				ARTM-9305 2x10G, 12x1G			
Description	BCM56802 Port	SRS Port	U-boot Port	Description	Marvell Port	SRS Port	U-boot Port
Zone 3 (J30)	11	xe5	8	Zone 3 (J30)	24	n/a	24
Zone 3 (J30)	12	xe6	9	Zone 3 (J30)	25	n/a	25
Octeon 2 SP I-0	13	xe7	10				
Octeon 2 SP I-1	14	xe8	11				
Zone 3 (J31)	15	xe9	12	SFP+ 1	-		
Zone 3 (J31)	16	xe10	13	SFP+ 2	-		
Zone 3 (J31)	17	xe11		-			
Zone 3 (J31)	18	xe12		-			
				SFP 1	1	n/a	0
				SFP 2	2	n/a	1
				SFP 3	3	n/a	2
				SFP 4	4	n/a	3
				SFP 5	5	n/a	4
				SFP 6	6	n/a	5
				SFP 7	7	n/a	8
				SFP 8	8	n/a	9
				SFP 9	9	n/a	10
				SFP 10	10	n/a	11
				SFP 11	11	n/a	12
				SFP 12	12	n/a	13

Figure 6-1, illustrates the ATCA-9305 BCM56802 switch with 1G IO RTM ports.

Figure 6-1 ATCA-9305 BCM56802 switch with 1G IO RTM ports



Due to hardware limitation; **xe9** and **xe10** ports supports only 10G bandwidth. The SFP modules are not supported on these ports.

6.5 18x1G RTM Ports

Table 6-3 ARTM-9305 18x1G Ports

ATCA-9305				ARTM-9305 18x1G			
Description	BCM56802 Port	SRS Port	U-boot Port	Description	Marvell Port	SRS Port	U-boot Port
Zone 3 (J30)	11	Xe5	8	Zone 3 (J30)	24	n/a	24
Zone 3 (J30)	12	Xe6	9	Zone 3 (J30)	25	n/a	25
				SFP 1	1	n/a	0
				SFP 2	2	n/a	1
				SFP 3	3	n/a	2
				SFP 4	4	n/a	3
				SFP 5	5	n/a	4
				SFP 6	6	n/a	5
				SFP 9	7	n/a	8
				SFP 10	8	n/a	9
				SFP 11	9	n/a	10
				SFP 12	10	n/a	11
				SFP 13	11	n/a	12
				SFP 14	12	n/a	13
				SFP 15	13	n/a	14
				SFP 16	14	n/a	15
				SFP 17	15	n/a	16
				SFP 18	16	n/a	17
				SFP 19	17	n/a	18
				SFP 20	18	n/a	19

6.6 6X10G RTM Ports

Figure 6-2 6X10G RTM Ports

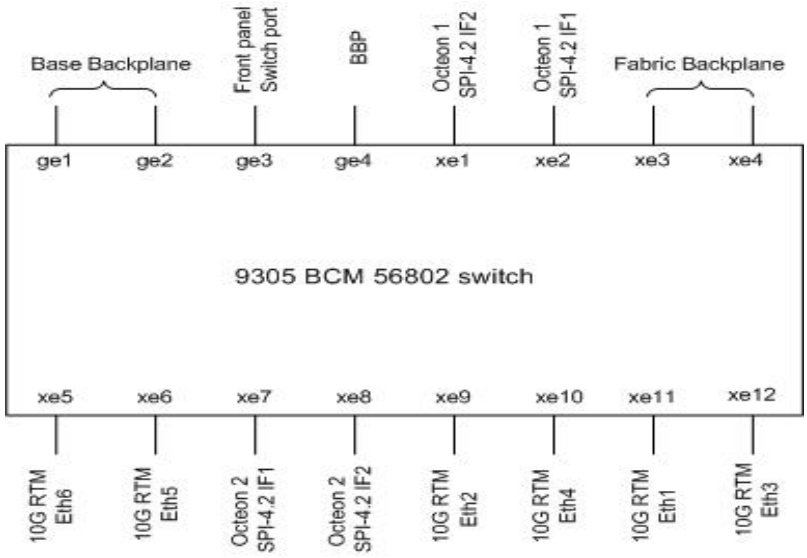


Table 6-4 ARTM-9305 6x10G Ports

ATCA-9305				ARTM-9305 6x10 GbE
Description	BCM56802 Port	SRS Port	U-boot Port	Description
Base	1	ge1	0	
Base	2	ge2	1	
Front panel RJ45	3	ge3	2	
MPC8548	4	ge4	3	

Table 6-4 ARTM-9305 6x10G Ports (continued)

ATCA-9305				ARTM-9305 6x10 GbE
Description	BCM56802 Port	SRS Port	U-boot Port	Description
Octeon 1 SPI-1	5	xe1	4	
Octeon 1 SPI-0	6	xe2	5	
Fabric1	7	xe3	6	
Fabric0	8	xe4	7	
Not connected on BCM56803	9			
Not connected on BCM56803	10			
Zone 3 (J30)	11	xe5	Not supported in u-boot	SFP+ (ETH6)
Zone 3 (J30)	12	xe6	Not supported in u-boot	SFP+ (ETH5)
Octeon 2 SPI-0	13	xe7	10	
Octeon 2 SPI-1	14	xe8	11	
Zone 3 (J31)	15	xe9	Not supported in u-boot	SFP+ (ETH2)
Zone 3 (J31)	16	xe10	Not supported in u-boot	SFP+ (ETH4)
Zone 3 (J31)	17	xe11	Not supported in u-boot	SFP+ (ETH1)
Zone 3 (J31)	18	xe12	Not supported in u-boot	SFP+ (ETH3)

A.1 Artesyn Embedded Technologies - Embedded Computing Documentation

The publications listed below are referenced in this manual. You can obtain electronic copies of Artesyn Embedded Technologies - Embedded Computing publications by contacting your local Artesyn sales office. For released products, you can also visit our Web site for the latest copies of our product documentation.

1. Go to www.artesyn.com/computing/support/product/technical-documentation.php.
2. Under FILTER OPTIONS, click the **Document types** drop-down list box to select the type of document you are looking for.
3. In the Search text box, type the product or document name and click **Filter**.

Table A-1 Artesyn Embedded Technologies - Embedded Computing Publications

Document Title	Publication Number
ATCA-9305 SRstackware 2.1 Protocol Demo Guide	6806800J46
ATCA-9305 Datasheet	*Datasheet*
ATCA-9305: ATCA Blade with Dual Cavium Processors User's Manual (April 2009)	10009109-01
RTM-ATCA-9305 IO Installation and Use	10010926-02
SRstackware on ATCA-9305 Protocol Demo Guide	6806800K25

A.2 Related Specifications

For additional information, refer to the following table for related specifications. As an additional help, a source for the listed document is provided. Please note that, while these sources have been verified, the information is subject to change without notice.

Table A-2 Related Specifications

Organization	Document Title
Intel developer.intel.com/design/servers/ipmi	Platform Management FRU Information Storage Definition v1.0
Intel developer.intel.com/design/servers/ipmi	IPMI Specification V2.0
PICMG picmg.org/specifications.stm	PICMG 3.0 Revision 2.0 Advanced TCA Base Specification PICMG 3.1 Revision 1.0 Specification Ethernet/Fiber Channel
Broadcom	BCM8725 Preliminary Data Sheet (Doc. No. 8725-DS01-R, September 19, 2007)
Marvell	Prestera-DX Packet Processors Hardware Design Guide (MV-S300644-00, Rev. D, December 27, 2006) 92DX5126 Hardware Specifications (MV-S104553-00, Rev. C, March 5, 2008)
Service Availability Forum Specifications http://www.saforum.org	SAI-HPI-B.01.01 Hardware Platform Interface Specification
Service Availability Forum Specifications http://www.saforum.org	SAI-AIS-A.01.01 Application Interface Specification
Service Availability Forum Specifications http://www.saforum.org	SAI-HPI-SNMP-B.01.01
Service Availability Forum Specifications http://www.saforum.org	SAIM-HPI-B.01.01-ATCA SAF HPI-to-AdvancedTCA Mapping Specification

ARTESYN™
EMBEDDED TECHNOLOGIES