Basic Blade Services Software on ATCA-7480

Programmer's Reference

P/N: 6806800T30B

September 2019

Trademarks

The stylized "S" and "SMART" is a registered trademark of SMART Modular Technologies, Inc. and “SMART Embedded Computing” and the SMART Embedded Computing logo are trademarks of SMART Modular Technologies, Inc. All other names and logos referred to are trade names, trademarks, or registered trademarks of their respective owners. These materials are provided by SMART Embedded Computing as a service to its customers and may be used for informational purposes only.

Disclaimer*

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

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 The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of SMART EC.

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

Limited and Restricted Rights Legend

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

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

SMART Embedded Computing, Inc.

2900 S. Diablo Way, Suite 190

Tempe, Arizona 85282

USA

*For full legal terms and conditions, visit *www.smartembedded.com/ec/legal*

Table of Contents

Table of Contents

List of Figures

List of Figures

List of Tables

List of Tables

About this Manual

Overview of Contents

This manual is divided into following chapters and appendices.

Abbreviations

This document uses the following abbreviations:

Abbreviation	Definition
ATCA	Advanced Telecommunications Computing Architecture
BBS	Basic Blade Services
BIOS	Basic Input Output System
BSD	Berkeley Software Distribution
CGL	Carrier Grade Linux
DHCP	Dynamic Host Configuration Protocol
FCU	FUF Command Line Utility
FPGA	Field Programmable Gate Array
FRI	Firmware Recovery Image

Abbreviation	Definition
FRU	Field Replaceable Unit
FUF	Firmware Upgrade Facility
GPIO	General Purpose Input/Output
HPI	Hardware Platform Interface
HPM	Hardware Platform Management
IPMB	Intelligent Platform Management Bus
IPMC	Intelligent Platform Management Controller
IPMI	Intelligent Platform Management Interface
IQR	Interrupt Request
LSP	Linux Support Package
MAC	Media Access Control
NTP	Network Time Protocol
OEM	Original Equipment Manufacturer
PCI	Peripheral Component Interconnect
PICMG	PCI Industrial Computers Manufacturers Group
PXE	Preboot Execution Environment
QSFP	Quad Small form-factor Pluggable
RPM	Red Hat Package Manager
RTM	Rear Transition Module
SAS	Serial Attached SCSI
SATA	Serial ATA
SDR	Sensor Data Record
SFP	Small form-factor Pluggable
SMI	Serial Management Interface
SNMP	Simple Network Management Protocol
SSD	Solid State Disk
SSH	Secure Shell

Abbreviation	Definition
TFTP	Trivial File Transfer Protocol

Conventions

The following table describes the conventions used throughout this manual. .

Notation	Description
`0x00000000`	Typical notation for hexadecimal numbers (digits are 0 through F), for example used for addresses and offsets
`0b0000`	Same for binary numbers (digits are 0 and 1)
bold	Used to emphasize a word
`Screen`	Used for on-screen output and code related elements or commands. `Sample of Programming used in a table (9pt)`
`Courier + Bold`	Used to characterize user input and to separate it from system output
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
..	Ranges, for example: 0..4 means one of the integers 0,1,2,3, and 4 (used in registers)
|	Logical OR
	Indicates a hazardous situation which, if not avoided, could result in death or serious injury

Notation	Description
	Indicates a hazardous situation which, if not avoided, may result in minor or moderate injury
	Indicates a property damage message
	Indicates a hot surface that could result in moderate or serious injury
	Indicates an electrical situation that could result in moderate injury or death
	Indicates that when working in an ESD environment care should be taken to use proper ESD practices
	No danger encountered, pay attention to important information

Summary of Changes

See the table below for manual revisions and changes.

Part Number	Date	Description
6806800T30B	September 2019	Rebranded to SMART Embedded Computing template. Light edits.
6806800T30A	December 2014	Initial version

Introduction

1.1 Overview

This manual describes the Basic Blades Services (BBS) software functionality based on Red Hat Enterprise Linux (RHEL) 6.5 and 7.0. The BBS software provides a set of services that support the blade on which the software is installed. BBS includes:

- Several custom hardware management functions for the unique hardware embedded on the blade.
- A set of management routines for Linux and all hardware interfaces. Management access includes support for Simple Network Management Protocol (SNMP) and a local console interface based on standard Linux command shell.l.

1.2 Support for other Linux Operating Systems

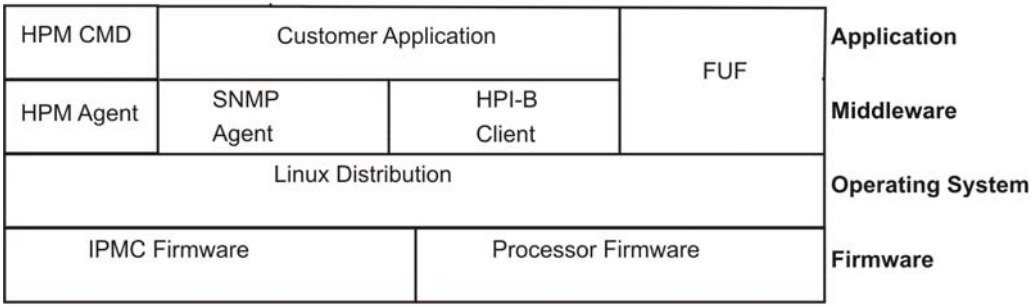
The BBS functionality can be made available for other Linux distributions like Ubuntu 14.04 on request. Deviations in the functionality will be described in the respective release notes.

1.3 Software Building Blocks

BBS includes a common set of functionality, which is available for all ATCA blades. It also includes a unique set of functionality, which is tailored particularly for this blade.

The following figure depicts the architecture of the BBS software.

Figure 1-1 BBS Architecture



HPM: Hardware Platform Management
FUF: Firmware Upgrade Facility
SNMP: Simple Network Management Protocol
IPMC: Intelligent Peripheral Management Controller
HPI-B: Hardware Platform Interface (Version B)

BBS for the ATCA-7480 consists of the following main software and services:

- Firmware Upgrade Facility (FUF): Provides a uniform way to upgrade firmware on SMART EC blades, regardless on which flash locations the firmware is stored. FUF upgrades the BIOS firmware and the IPMC firmware via HPM agent. Currently, FUF consists of 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.
- Linux Operating System
  Red Hat Linux 6.5 is the base operating system (OS) for BBS. Various Linux services (on the kernel) will be activated by the BBS installation scripts..

NOTICE

The operating system is not part of the BBS release. The customer is responsible for installing an appropriate Linux OS on the blade.

- Hardware Platform Management
  Hardware Platform Management (HPM) in ATCA systems is based on the Intelligent Platform Management Interface (IPMI) specification. IPMI commands can be complex and cumbersome. Using certain sets of commands, HPM facilitates the blade- and the module-level hardware management.
- HPI-B
  The HPI-B Software packages can be received from your local SMART EC sales representative. For further information, refer to the *System Management Interface Based on HPI-B (Centellis 1440/2000/4440) User's Guide*.

Installing the Basic Blade Services

Software

2.1 Overview

The BBS software packages require a pre-installed host operating system with a kernel 2.6.34 or later, or RHEL 6.5 like OS with kernel backports.

The best way to perform such a host OS installation is using an unattended network installation with a kick start file. For more details, see *Kickstart File for Unattended RHEL6.5 Installation on page 97*.

2.1.1 Package Information

The BBS software is delivered as RPM packages, which can be installed on your target operating system.

The BBS distribution contains software packages listed in the following table.

Table 2-1 Software Packages

Software	Package Name
Kernel command line	`default.bbs-atca7480`
Checksum of all files and RPMs	`files.sha1sum`
HPM Agent command tool	`bbs-hpmagentcmd-atca7480-<version>-<OS distro>.rpm`
Firmware upgrade utility	`bbs-fcu-atca7480-<version>-<OS distro>.rpm`
Board control utility to get FPGA data	`bbs-boardctrl-atca7480-<version>-<OS distro>.rpm`
BIOS image	`bbs-bios-atca7480-<version>.rpm`
Field-Programmable Gate Array (FPGA) image	`bbs-fpga-atca7480-<version>.rpm`
IPMC Firmware incl Booter (Front board) (should not be used)	`bbs-ipmc-all-atca7480-<version>.rpm`
IPMC Firmware wo Booter (Front board)	`bbs-ipmc-atca7480-<version>.rpm`

Table 2-1 Software Packages (continued)

Software	Package Name
IPMC Booter (Front board)	`bbs-ipmc-boot-atca7480-<version>.rpm`
IPMC Firmware (ARTM)	`bbs-artm-atca7360-<version>.rpm`
IPMC Booter Firmware (ARTM)	`bbs-artm-boot-atca7360-<version>.rpm`
IPMC FW for Dual Disk ARTM	`bbs-artm-rtm450n-atca7480-<version>.rpm`
NMU Package	`bbs-nmu-atca7480-<version>-.rpm`

The following RPM commands are useful to review package information.

Command	Description
`rpm -qa`	Lists all the installed packages. Use `rpm -qa | grep hpi` to list only HPI packages.
`rpm -ql <package-name>`	Lists the content of a package, where `package-name` is the name of a specific package. For example, `rpm -ql bbs-fcu.`
`rpm -qi <package-name>`	Lists the information about a package, where `package-name` is the name of a specific package. For example, `rpm -qi bbs-fcu.`
`rpm -qf <path to file>`	Finds out to which RPM file belongs to.

2.1.2 Accessing the ATCA-7480 via Serial Console

To invoke the Linux commands or configure BIOS setting, first you need to access the ATCA-7480 via the face plate serial port. For using a serial console or a terminal emulator, the default serial port settings are:

- 115200 baud
- No parity
- Eight data bits
- One stop bit
- Flow control: xon/xoff
- Emulated terminal type: VT100

If you want to access Linux via a Linux shell, the default account user name and password are **root**. For more information, refer *Login on page 23*.

2.2 BBS Installation

This section describes installing BBS on host operating systems, such as RHEL and Ubuntu. For installing the host OS, refer to the respective release notes provided with the distribution or by vendor.

Verify that following packages are installed along with the OS.

- pciutils
- OpenIPMI
- ipmitools
- trousers

Installing BBS

Copy the BBS package to your target system and run the installation script. This script performs the following:

- Sets up the udev rules for network device renaming
- Installs the BBS packages
- Installs the i40e driver for the Intel XL710 40G NIC devices
- Updates the path variable

NOTICE

The installation script requires root permissions.

During installation of the i40e driver, the initramfs and the Linux modules will be updated if needed.

2.3 Upgrading the BBS Software

BBS software updates are usually delivered as RPM files. Use the following procedure to upgrade the BBS on a hard disk:

To upgrade the BBS on hard disk

1. Copy the new RPM files to the blade.
2. Stop all BBS related applications.
3. Remove the **boardctrl** and the **nmu_sfp** kernel driver.
4. Remove the previous files using the `rpm -e` <package> command.
5. Install the newly copied files using the `rpm -Uvh` <package-name> command.
6. Restart the applications, the **boardctrl**, and the **nmu_sfp** driver again.

Operating System

3.1 Distribution Description

The BBS for the ATCA-7480 blade is based on RHEL 6.5 and 7.0. BBS package can be made available for other operating systems on request.

3.2 Login

A Linux shell can be accessed via the face plate serial port. If you use a serial console or terminal emulator, the serial/RTM port settings are 115200 baud, no parity, 8 data bits, and 1 stop bit.

If you use Secure Shell (SSH) server, it starts in run levels 2–5 and listens on all the Ethernet interfaces. Root login for SSH is not permitted, you need to log in as `admin` user. If you use SSH, refer to *Network Services Configuration on page 25* 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` to set in the `/etc/ssh/sshd_config` file. To effect the changes, you must either reboot the blade or run the `/etc/init.d/ssh restart` command.

The following table lists available default login accounts.

Login Name	Password	Description
admin	bbsadmin	Non-privileged user account
root	root	Privileged user account

3.3 Linux Services Initialization

The following table lists the generic Linux run-levels and the services configured to start in the various Linux run-levels. By default, the blade first runs run-level 'S' and then boots into run-level '3' as configured by the factory.

Table 3-1 Generic Linux Run Levels

Run Level	Description
S	Startup
0	Halt system

Table 3-1 Generic Linux Run Levels (continued)

Run Level	Description
1	Single-user mode
2	Multiuser mode
3	Multiuser mode with network (default)
4	Not used
5	Not used
6	Reboot system

3.3.1 RC Scripts

The initialization mechanism of Host OS initializes the Linux services. For RHEL 6.5 and 7.0 classic Linux initialization, RC scripts are used.

NOTICE

This section is applicable only for RHEL host operating system.

Other operating systems use different startup/init mechanisms. For example, upstart

Following table provides a list of RC scripts used for starting the BBS services.

Table 3-2 RC Scripts

RunLevel	ScriptName	Description
rc0.d	K56hpm	Kills hpmagent daemon.
rc1.d	K56hpm	Kills hpmagent daemon.
rc3.d	S09hpm	Starts hpmagent daemon.
rc3.d	S01bbsrpms	Installs BBS related RPMS.
rc3.d	S02bbsinit	Loads boardctrl driver.
rc3.d	S10ethDevOrdering	Renames and sets the network devices.
rc3.d	S11bbsSetHostName	Sets host name dependent on slot number and product name.

Table 3-2 RC Scripts (continued)

RunLevel	ScriptName	Description
rc3.d	S57bbsVlan	Sets up the Fabric network interface based on shelf slot number.
rc3.d	S99osBootSensor	Informs IPMC about successful boot and disables OS boot supervision in case it is enabled.
rc3.d	S99postProcessing	Removes initial Boot flag.
rc3.d	S99setNetwIrqAffinity	Assigns Interrupt Requests (IRQs) of the network devices to the appropriate CPU.
rc6.d	K56hpm	Kills hpmagent daemon.

3.4 Network Services Configuration

The following sections describe the default configuration for network services.

3.4.1 ATCA-7480 Ethernet Interfaces

The Ethernet devices, such as eth0, eth1, and eth2 in Linux distribution are renamed to more meaningful names in ATCA-7480, such as base1, base2, and fabric. For “eth” device ordering of the front blade, udev rules (`/etc/udev/rules.d/26-network.rules`) are used. For ARTM, the “eth” network device is renamed using the `netdev-renamertool`.

The following table specifies the Ethernet devices supported by ATCA-7480.

Device Name	Description	Speed	Location	IP address	Driver Name
base1, base2	Base Interface	1 GbE	Front Board -> Backplane	Obtained by the DHCP client request.	Intel - igb

fabric1_1, fabric1_2, fabric2_1 and fabric2_2	Fabric Interface	KR4 - 40 GbE	Front Board -> Fabric Interface on Backplane Fabric1_2 and fabric2_2 are only available in combination with a dual-dual star backplane	Static IP address.
				It is computed as:
				`192.168.<fabricIf>.<slotnumber*10>`
				`fabricIf can have value of; "11" for Fabric Interface1 and "12" for Fabric Interface2.`
				`slotnumber specifies the logical slot number converted to decimal.`
				`The setup of the IP Addresses for Fabric IF is done in the /etc/init.d/bbsvlan.sh file.`
					Intel i40e

front1, front2	Front Panel Interface	1GbE	Front Board	No IP address assigned.	Intel - igb
rtm1..n (optional)	RTM Panel Interface	1GbE, 10GbE, 40GbE	RTM (optional)	No IP address assigned.	Dependent on ARTM (Intel igb/ixgbe, i40e)

Front Board Network Devices Setup

Table 3-3 Front Board Network Devices

PCIDev	IF Name	Driver
0000:01:00.0	Base1	igb
0000:01:00.1	Base2	igb
0000:01:00.2	Front1	igb

Table 3-3 Front Board Network Devices (continued)

PCIDev	IF Name	Driver
0000:01:00.3	Front2	igb
0000:04:00.0	Fabric1_1	i40e
0000:04:00.1	Fabric1_2	i40e
0000:81:00.0	Fabric2_1	i40e
0000:81:00.1	Fabric2_2	i40e

The BBS for ATCA-7480 sets up the network interfaces of ARTM-736X-10G/ARTM-736X-10G-SP.

ARTM-736X-10G-SP Configuration Setup

PCI Dev	IF Name	Driver
0000:0a:00.0	rtm8	igb
0000:0a:00.1	rtm7	igb
0000:0a:00.2	rtm6	igb
0000:0a:00.3	rtm5	igb
0000:05:00.0	rtm10	ixgbe
0000:05:00.1	rtm9	ixgbe
0000:06:00.0	rtm2	ixgbe
0000:06:00.1	rtm1	ixgbe
0000:84:00.0	rtm4	ixgbe
0000:84:00.1	rtm3	ixgbe

ARTM-736X-10G Configuration Setup

PCI Dev	IF Name	Driver
0000:0a:00.0	rtm8	igb
0000:0a:00.1	rtm7	igb
0000:0a:00.2	rtm6	igb
0000:0a:00.3	rtm5	igb

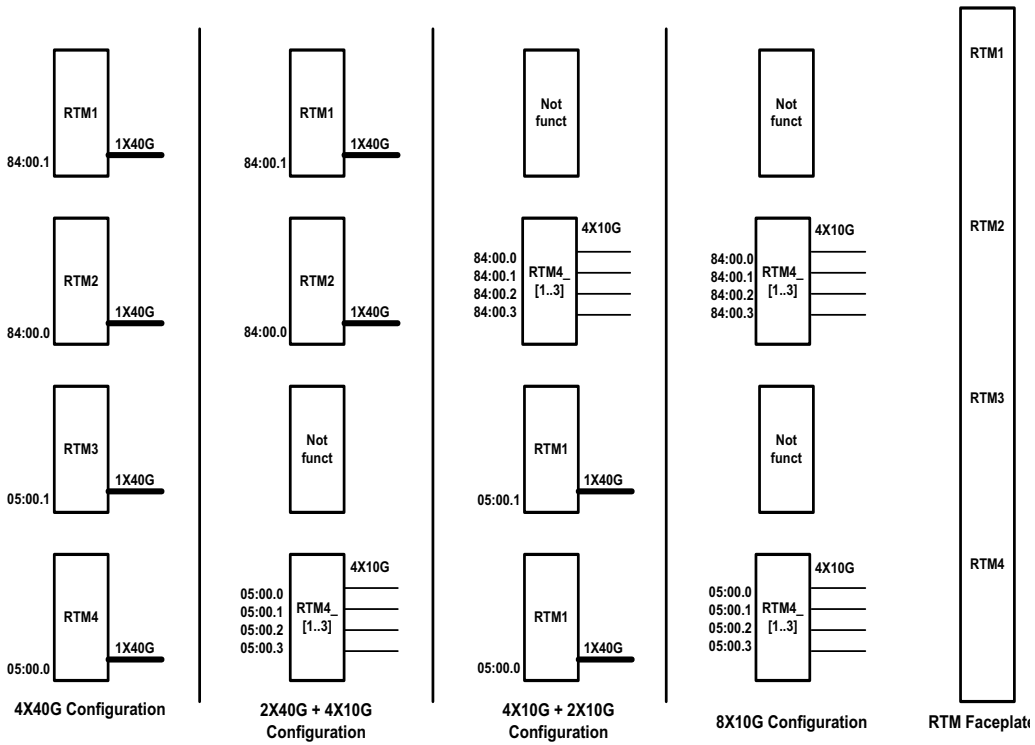
PCI Dev	IF Name	Driver
0000:06:00.0	rtm2	ixgbe
0000:06:00.1	rtm1	ixgbe
0000:84:00.0	rtm4	ixgbe
0000:84:00.1	rtm3	ixgbe

Setup RTM-ATCA-7480 Configuration

The ARTM-7480 is equipped with two dual port XL710 NIC devices. Dependent on the configuration file (NVM), the device can be used as 2x 40Gb/s device or 4x10Gb/s device. To support both configurations, two EEPROMs are provided per XL710 NIC on the ARTM.

Generally, the 40G configuration is stored for the first NVM and the 4x10G configuration is stored on the second NVM.

Figure 3-1 RTM-7480 Network Interface Naming



The NVM configuration can be read using `hpmcmd -c nvmbankget <device 0..1>` and it can be changed using `hpmcmd -c nvmbankset <device 0..1> <bank 0..1>`. For more details, see *nvmbankget on page 65* and *nvmbankset on page 65*. When changing the NVM bank image, board power cycle is required.

The NVM image can be updated using lanconf64 tool (for efi, DOS and Linux). Due to license issues, this tool can be retrieved only from Intel.

The first EEPROM contains the NCM image for the 2x40Gb/s configuration and the second EEPROM contains the 4x10Gb/s. You can switch between these configurations using the IPMI command, get/set boot options on the MMC. As an alternative, the hpmcmd nvmbankget/nvmbankset can be used to get/set the configuration. After switching the NVM banks, a complete power-cycle of the board needs to be performed to get the configuration up and running.

NOTE: The NVM images can be reprogrammed with the Intel tool, lanconf.

3.4.2 ATCA-7480 Fabric Interface

The ATCA-7480 blade is equipped with two Intel XL710 Dual 40-Gb/s dual-port Ethernet controller devices, which are responsible for handling the dual-star Fabric Interface (fabric1_1 and fabric2_1) or the dual-dual star fabric (fabric1_1, fabric1_2, fabric2_1, and fabric2_2).

3.4.2.1 Network Driver - IRQ Assignment

In order to gain maximum network performance, it is inevitable to assign the interrupts for the network devices to the CPU nodes, the devices are physically connected. The IRQs must also not be routed to the first CPU core.

This IRQ assignment for the network devices is performed by the script:
`/opt/bladeservices/tools/set_irq_affintiy_.sh`

3.5 Tools

This section describes `CPUSpeed` and `IPMIBPAR` tools, which can be used to change the processor performance governors and IPMI Boot Parameter list.

3.5.1 Performance Tool - cpuspeed

The performance tool - cpuspeed allows to change the processor performance governors and the core frequency (for userspace governor) on a per core base. It utilizes data stored in the `/sys/device/system/cpu` directory.

The following table describes various governors:

Governor	Description
Performance	Core is running with maximum frequency.
Ondemand	Cores in idle state are running at lowest frequency. When the core is changed to the utilized state, the frequency of the core is changed to maximum.
Powersave	Core is running with minimum frequency.
Userspace	Core frequency can be adjusted by the user (in steps).

If the P-States are limited by BIOS, the required driver is not loaded and therefore the CPUSpeed tool cannot work.

CPUSpeed supports the following options.

Option	Description
-d	Dumps CPU Frequency/Governor Info
-h	Provides help
-p	Print supported governors
-s	Sets governor/frequency. It supports the following options: **-c:** Specifies the core. Valid values are 0 .. 15. Omitting this option means, all cores. **-f:** Specifies the frequency. Valid values are 1596000 .. 2129000. This parameter is ignored except for 'userspace governor'. **-g:** Specifies governors, such as performance, powersave, ondemand, and userspace.

Example:

```
root@atca7480:/proc/boardinfo# cpuspeed -s -g performance
```

This command sets the performance governor for all CPU core.

NOTICE

The CPUSpeed tool is not usefull for RHEL7.0 based Operating Systems, as PState handling has been completely changed and the cpu frequency is controlled without CPUSpeed governor.

3.5.2 IPMIBPAR

The `IPMIBPAR tool` is used to change the IPMI Boot Parameter list when Linux is up and running. It supports the following options:

Option	Description
-d	Enables debug output.
-a xx	IPMB Address, if not present, local IPMC is used.
-i	Gets device ID.
-g	Gets IPMI Boot Parameter USER area.
-s file	Stores IPMI Boot Parameter (USER area), read from file.
-h	Provides Help.

The following example describes the steps required to change the BootOrder from On-board SATA disk to FrontNetwork2.

1. Read the IPMI boot parameter USER area from IPMC.

```
root@ATCA-7480:~# ipmibpar -g
ipmibpar - Version 1.02 - IPMI Boot
Parameter Demo
Copyright 2008 Emerson Network Power Embedded Computing Inc.
Read System Boot Options from USER area (local IPMC)
Hexdump IPMI Boot Parameter:
Size = 222 (0xde)
0000  da 00 61 72 74 6d 5f 6e 65 74 5f 62 6f 6f 74 3d   <..artm_net_boot=>
0010  6f 66 66 00 61 72 74 6d 5f 73 61 73 5f 62 6f 6f   <off.artm_sas_boo>
0020  74 3d 6f 6e 00 62 61 73 65 6e 65 74 5f 62 6f 6f   <t=on.basenet_boo>
0030  74 3d 6f 6e 00 62 61 75 64 72 61 74 65 3d 31 31   <t=on.baudrate=11>
0040  35 32 30 30 00 62 6f 6f 74 5f 6f 72 64 65 72 3d   <5200.boot_order=>
0050  73 61 74 61 6f 6e 62 6f 61 72 64 2c 66 72 6f 6e   <sataonboard,fron>
0060  74 6e 65 74 32 2c 62 61 73 65 6e 65 74 31 2c 65   <tnet2,basenet1,e>
0070  66 69 73 68 65 6c 6c 2c 75 73 62 68 64 64 00 63   <fishell,usbhdd.c>
0080  6f 6d 5f 74 65 72 6d 5f 74 79 70 65 3d 76 74 31   <om_term_type=vt1>
0090  30 30 2b 00 66 72 6f 6e 74 6e 65 74 5f 62 6f 6f   <00+.frontnet_boo>
00a0  74 3d 6f 6e 00 69 70 6d 69 5f 69 72 71 3d 6f 66   <t=on.ipmi_irq=of>
00b0  66 00 6f 73 5f 62 6f 6f 74 5f 77 61 74 63 68 64   <f.os_boot_watchd>
00c0  6f 67 3d 6f 66 66 2c 35 2c 72 65 73 65 74 00 75   <og=off,5,reset.u>
00d0  73 62 5f 62 6f 6f 74 3d 6f 6e 00 00 5b 50          <sb_boot=on..[P>
IPMI Boot Parameter:
artm_net_boot=off
artm_sas_boot=on
basenet_boot=on
baudrate=115200
boot_order=sataonboard,frontnet2,basenet1,efishell,usbhdd
```

```
com_term_type=vt100+
frontnet_boot=on
ipmi_irq=off
os_boot_watchdog=off,5,reset
usb_boot=on
```

2. Save the received IPMI Boot Parameter list into a file (for example, bootparam.log) and change the boot order as follows.

```
artm_net_boot=off
artm_sas_boot=on
basenet_boot=on
baudrate=115200
boot_order=frontnet2,sataonboard,basenet1,efishell,usbhdd
com_term_type=vt100+
frontnet_boot=on
ipmi_irq=off
os_boot_watchdog=off,5,reset
usb_boot=on
```

3. Write the IPMI parameter list file (for example, bootparam.log).

```
ipmibpar -s <filename>
```

3.5.3 CoreTemp

At `/opt/bladeservices/tools` you can find a script named, `coreTempDump`, which reads and displays the CPU temperature on a per core base. It also shows whether the temperature exceeds the maximum allowed core temperature (indication for CPU throttling).

Syntax: `coreTempDump.sh`

Firmware Upgrade Facility

4.1 Overview

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

4.2 Firmware Recovery Image Files

FCU supports specially prepared Firmware Recovery Image 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 BBS software in `/opt/bladeservices/rom` directory, when the blade-specific firmware support packages are installed.

The following image files are currently supported.

Filename	Description
`atca-7480-cpu-<version>.hpm`	BIOS image
`atca-7480-ipmc-boot-<version>.hpm`	Firmware for the IPMC booter
`atca-7480-ipmc-<version>.hpm`	Firmware for the IPMC
`atca-7480-ipmc-all-<version>.hpm`	Combined image for IPMC and IPMC booter (only delivered when the update of the booter is required)
`atca-74807480-spi--<version>.bin.hpm`	Glue FPGA Bitstream

NOTICE

The `ipmc-boot` and the `ipmc-all` firmware will not be installed on the board any more. If there is a need to upgrade the boot-loader, this will be explicitly stated in the release notes. BIOS will only be released in hpm.1 file format.

4.3 fcu—Firmware Upgrade Command-Line Utility

fcu displays upgrade capabilities of firmware devices on a board. If the board is managed by an IPMC, then fcu requests IPMC to determine the board type. A board may have multiple devices, which are low-level hardware components, like BIOS, FPGA, and IPMC and so on. fcu abstracts firmware upgrade operations to a common set of operations for all devices. The fcu user does not need to know internals of a device. fcu identifies a device by its name. The name is a combination of the board name and the device properties.

For example, the BIOS for the ATCA-7480 has the name "atca-7480-cpu". fcu provides the following upgrade operations:

- Query the device to return the firmware version and other information
- Query the firmware image to return the firmware version and other information
- Validate the firmware image
- Verify whether the image is applicable on the target device
- Upgrade the device with the given firmware image
- Compare the firmware image with an installed firmware image on the device
- Mark a bank of the device to become active after the next reset or power cycle
- Activate a bank of the device immediately

4.3.1 Query Operation

Using the `Query` operation, fcu returns 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. `Query` operation shows all the banks of a device. One of these banks is the active version, which means that the device was booted with the firmware installed in that bank. The device might have a second bank that contains the rollback version. You can switch to the rollback version with the help of activate operation or with mark operation in combination with a reboot or power-cycle.

If the device supports the mark operation, the `query` operation shows which bank is marked for next use. Furthermore, fcu shows the capabilities of a device. Device capabilities are set of fcu operations, such as manual, automatic rollback, or self test implemented. The following example shows the BIOS of the ATCA-7480 board.

```
root@ATCA7480:~# fcu --query -d atca-7480-cpu
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Query
 #10 Device   : atca-7480-cpu
  Bank #1 -      Active Version: 0.5.00000000
  Bank #0 -    Rollback Version: 0.5.00000000
```

```
  Bank marked for next use: #1
********************[[[[[ REPORT END ]]]]]********************
```

In the first line, the number of the device and its name are displayed. In Bank1, the active version is stored and this bank is also marked for next use. Bank0 contains the rollback version. The device supports the `upgrade`, `mark`, and `compare` operation. The `activate` operation is not supported. The version is always a combination of decimal numbers.

4.3.2 Show Operation

`Show` operation does not access any device. It only operates with the firmware image and it shows the metadata, which is part of the image. Furthermore, it validates the firmware image to compare the checksum part of the metadata against the checksum of the raw image. The output of the `show` operation is similar to the output of the `query` operation.

```
root@ATCA7480:~# fcu --show --file /opt/bladeservices/rom/atca-7480-cpu-0.5.0.hm
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Show
 Manufacturer : EMERSON
 Board        : atca-7480
 #00 Device   : ATCA7480 BIOS Image
  Bank #0 -             Version: 0.05.00000000

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

Additionally, this operation shows the name of the manufacturer and the name of the board to which the firmware of this image is compatible to. A firmware image does not have multiple banks. So you can see only one bank.

4.3.3 Mark Operation

`Mark` operation selects the bank that is used after the next reset or reboot. With this operation, you have to specify the name of the device and the bank.

NOTICE

Not all devices support this operation.

4.3.4 Activate Operation

`Activate` operation is similar to that of `mark` operation. Because, this operation also sets a bank to active state; but the bank is activated immediately, which is not in the case of `mark` operation. Not every device supports this operation. IPMCs that have the HPM.1 must support the `Activate` operation on the rollback bank and the deferred bank. If the rollback bank is activated by this operation, a manual rollback is performed.

4.3.5 Compare Operation

With `Compare` operation, you can compare firmware images on the target with images in the upgrade file. You have to specify the firmware image file and the bank which you want to compare.

4.3.6 Upgrade Operation

`Upgrade` operation uploads the firmware image to the device. To have a valid firmware image in one bank, fcu tries to protect the active bank being overwritten by a new firmware image. For devices like IPMC, this protection is done by the IPMC firmware itself, which communicates with fcu during the image upload. The IPMC firmware selects the bank to write the new image too. Fcu determines this bank by reading an IPMI sensor, which knows the active bank. For that reason, you have to specify the firmware image with the `Upgrade` operation.

4.3.7 Verify Operation

`Verify` operation checks if the firmware image is applicable on a device of the blade. It prevents you from writing a firmware image into a non-compatible device. The `Verify` operation is always done before an Upgrade operation. So, you do not have to specify it explicitly when you perform an upgrade.

4.3.8 Command-Line Options

```
fcu --help
```
Usage

```
fcu [operations] [operands]
```
Operations

```
--query
-q      Set to perform a query operation
```

--upgrade
-u Upgrade the unused version of firmware
This operation requires the --file flag

--help
-h Display this help message

--show
-s Display information about the target which is included in the given upgrade file
This operation requires the --file flag

--verify
-v Set to perform a verification of an upgrade file
This operation does not install the upgrade image
This operation requires the --file flag to be set

--mark
-m Mark the specified bank as next to boot
This operation depends on the --bank and the --device flag

--activate
-r Activate the specified bank
This operation depends on the --bank and the --device flag

--compare
-c Compare the operation firmware with the image specified by the --file flag

--version
Display the version of this utility

Operands

--device=<device name>
-d Device to perform operation on

--file=<filename>
-f Filename of the firmware file

--bank=<bankletter>
-b Bank-letter for mark/compare command

--level=[0-7]
Severity level of logging, 7 logs everything, default is 5

--log=ARG
File name for logging

4.4 Upgrading Firmware Image

This section describes recommended procedures for upgrading firmware devices.

NOTICE

The shown file names and paths are only for example and should be replaced with file names and paths applicable to your configuration.

4.4.1 BIOS Firmware Upgrade of the ATCA-7480 Blade

Follow these steps for BIOS firmware upgrade on the ATCA-7480 blade:

1. Query the current firmware versions.

```
root@ATCA-7480:/root# fcu --query -d atca-7480-cpu
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Query
 #08 Device   : atca-7480-cpu
  Bank #1 -      Active Version: 0.5.00000000
  Bank #0 -    Rollback Version: 0.5.00000002
 Bank marked for next use: #1
  Capabilities        :
       Upgrade operation   : (x)
          Mark operation   : (x)
      Activate operation   : ( )
      Compare operation    : (x)
       Service affected    : ( )
       Manual Rollback     : (x)
       Automatic Rollback  : ( )
       Self Test           : ( )

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

2. Upgrade bank 0 to version 0.9.4

```
root@ATCA-7480:/mnt# fcu -uf /opt/bladeservices/rom/atca-7480-cpu-
0.9.4.hpm
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Upgrade
Hardware Write Protection off!
```

```
  erasing ...100 %
  writing ...100 %
Result   : Success
********************[[[[[ REPORT END ]]]]]********************
```

3. Mark Bank0 for next use.

```
root@ATCA-7480:/mnt# fcu -m -b 0 -d atca-7480-cpu
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Mark
Result   : Success
********************[[[[[ REPORT END ]]]]]********************
```

4. Check if the new firmware version (here 0.9.4) is in Bank0 and whether the Bank0 is marked.

```
root@ATCA-7480:/mnt# fcu -q -d atca-7480-cpu
********************[[[[[REPORT BEGIN]]]]]********************
Operation: Query
 #08 Device   : atca-7480-cpu
  Bank #1 -      Active Version: 0.9.00000003
  Bank #0 -    Rollback Version: 0.9.00000004
 Bank marked for next use: #0
  Capabilities        :
       Upgrade operation   : (x)
          Mark operation   : (x)
      Activate operation   : ( )
      Compare operation    : (x)
       Service affected    : ( )
       Manual Rollback     : (x)
       Automatic Rollback  : ( )
       Self Test           : ( )

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

NOTE: BIOS will be activated after a powercycle.

4.4.2 FPGA Upgrade

The FPGA on the ATCA-7480 board can also be updated with the FCU tool.

To upgrade FPGA on the ATCA-7480 blade:

1. Run the following command:
   ```
   fcu -uf /opt/bladeservices/rom/atca-7480-spi-0.0.11.bin.hpm
   ```
   After a successful upgrade, following is displayed:
   ```
   ********************[[[[[REPORT BEGIN]]]]]********************
   Operation: Upgrade
   preparation stage started
   capabilities of target and image header successfully compared
   properties of component FPGA retrieved
   properties of component FPGA retrieved
   preparation stage successfully finished
   preparing upload
   upload stage of ATCA7480 FPGA Image started
   initializing upload
   uploading ...100 %
   finishing upload
   upload stage finished
   activation stage of new firmware started
   Result   : Success
   ********************[[[[[ REPORT END ]]]]]********************
   ```
2. Switch the FPGA bank.
   To read the current FPGA bank, run the following command:
   ```
   fcu -m -b BANK<0/1> -d "PYLD FPGA"
   ```
3. Power cycle the board to run the new FPGA image.

Hardware Platform Management

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, SMART EC provides the Hardware Platform Management (HPM) package, which provides a set of commands that are based on IPMI commands. These commands are easy to use than the IPMI command itself. An HPM command can encapsulate a sequence of IPMI commands, for example, reading the FRU Inventory 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 utility, hpmcmd
- Script framework for managing shutdown, reboot, and local ekeying events

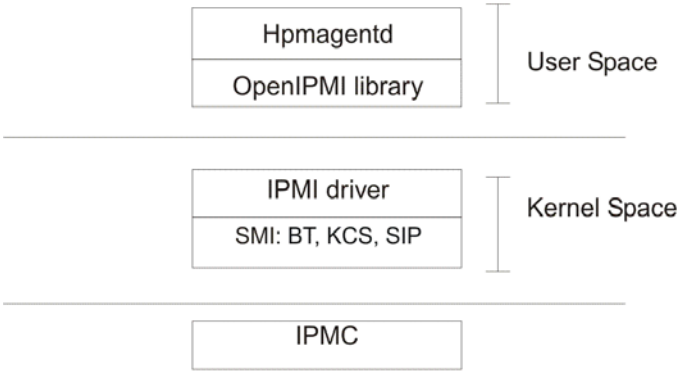
The HPM daemon is responsible to wait for events from the IPMC to perform a graceful shutdown/reboot of the Operating System and to react when the link state of a channel's port is changed.

The utility hpmcmd displays the response of commands on the console in a human-readable format. HPM commands include:

- Retrieving and modifying FRU data
- Reading and controlling status of IPMI-controlled LEDs
- Communicating local slot location information
- Retrieving the event messages from the SEL of the IPMC

The hpmagentd and hpmcmd make use of OpenIPMI driver to talk to the local IPMC. The following figure shows the software levels that are involved in the HPM architecture:

Figure 5-1 Software Levels of the HPM Architecture



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

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 *Supported IPMI Commands and Feature Set chapters in ATCA-7480 Installation and Use Manual*.

5.2 hpmagentd—HPM Agent Daemon

5.2.1 Description

The `hpmagentd` is the service to process events from the local IPMC. For any incoming event, it calls the respective script which is part of the hpmagentcmd package. Event data is passed to a script by command line arguments. You can modify a script to fulfill your requirements. The following events are handled by the daemon:

Graceful Shutdown - When the IPMC receives an FRU activation request to deactivate an FRU, then it redirects the command to the hpmagentd through the IPMI driver. The hpmagentd invokes the shutdown script which is located in `/opt/bladeservices/bin/hpmshutdown`. By default, within the script `shutdown -h now` is called, which initiates a shutdown of the Linux immediately.

NOTICE

IPMC powers down the processor in any case after a certain time. You may adjust this time with the Graceful Shutdown Timeout parameter of the IPMC, which can be set with a SetSystemBootOption IPMI command.

Graceful Reboot - On receiving an FRU control request to gracefully reboot the payload, the IPMC sends the command to the hpmagentd. The daemon invokes the reboot script which is available at `/opt/bladeservices/bin/hpmreboot`.

Ekeying Events - When the IPMC modifies the link state of a port, then it notifies the hpmagentd about that change. If the link goes down, then the script hpmekeydown is invoked, otherwise, the daemon calls hpmekeyup. Both scripts are placed in `/opt/bladeservices/bin`. By default, these scripts are empty. To identify which port was changed, the hpmagentd passes an argument to an hpmekey script in the format: First the interface is specified: BC for Base Channel, FC for Fabric Channel, UC for Update Channel, and AMC for AMC Channel. The channel number and the port numbers of the channel are specified in the below example.

For example, the ports 1,2,3,4 of the channel 1 of a base interface is changed, then the argument looks like: "BC1.1,2,3,4".

5.2.2 Deployment

By default, the HPM daemon is installed in `/opt/bladeservices/bin` directory. With the `hpmagentd` binary, the scripts `hpmreboot, hpmshutdown, hpmekeyup`, and `hpmekeydown` are stored in that directory. Additionally, there are `init` script hpm to start and stop the daemon and the script `hpmvar` which exports some important variables to `/etc/default/hpmvars` to describe the board.

Synopsis

```
hpmagentd [options]
```

Parameters

```
--log =<file name>
```

You may specify a log file for the daemon with this option. If you do not use it, then the `hpmagentd` logs to the syslog.

`-l --level`
Specifies the level of message logging, where 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

`-v --version`
Displays the version of the daemon

`-L --disable-led`
Disable the LED management

`-r --reboot-script=<script>`
Use the specified Blade Reboot script. The default script is `/opt/bladeservices/bin/hpmreboot`

`-s --shutdown-script=<script>`
Use the specified Blade Shutdown script. The default script is `/opt/bladeservices/bin/hpmshutdown`

`-u --ekey-up=<script>`
Called when a port is enabled. Default is `/opt/bladeservices/bin/hpmekeyup`

`-d --ekey-up=<script>`
Called when a port is disabled. Default is `/opt/bladeservices/bin/hpmekeydown`

`-h --help`
Displays the help message

`-i --dont-daemonize`
Run interactively

5.2.3 hpm - Init.d Script

The hpm init.d script allows to start, stop, and restart the hpmagentd. It can be linked to a run-level to automatically start the daemon at Linux boot time and to stop it when Linux shuts down.

Synopsis

```
hpm { start | stop | restart | status }
```

Parameters

`Start` - Starts the hpmagentd

`Stop` - Stops the hpmagentd

`Restarts` - Stops and Starts the hpmagentd again

`Status` - Reports if the hpmagentd is currently running or not

5.3 hpmcmd—HPM Command Utility

5.3.1 Overview

The HPM command utility communicates directly to the IPMC through the IPMI driver, which is part of the Linux kernel. It takes care of translating the user-friendly commands into elaborated IPMI commands that the IPMC is able to understand. Those IPMI commands are transferred to the local IPMC. The HPM command utility can be started in interactive mode, where a prompt is displayed and the user enters commands; or it can process a single command.

By default, the hpmcmd binary is installed in `/opt/bladeservices/bin` directory.

Synopsis

```
hpmcmd [options]
```

Parameters

`-c` Processes a single command

`--help -h` Displays this help message

`-v` Verbose mode for some commands
Some commands like `fruinfoget`, print more details if these options is given.
Commands which do not support the verbose option ignores it.

`-t` Sends the command to a remote target.
For more information, refer *Target Addressing with hpmcmd on page 46*.
If this option is not given, then the command goes to the local IPMC.

`-p`	Changes the prompt when `hpmcmd` runs in interactive mode.
`-o`	Prints results to a file

5.3.2 Target Addressing with hpmcmd

Using the `-t` option, you can send commands to other IPMCs or MMCs, which participate on an IPMB.

syntax: `-t <IPMB address>[: MMC address]`. The addresses must be set in hexadecimal format.

To send the command to an AMC attached on this blade use:

```
-t0:72 or -tlc:c0
```

To send the command to another IPMC type:

```
-t 92
```

To send the command to an MMC, which is attached on another blade in the shelf specify:

```
-t 82:72
```

5.3.3 Command Overview

The following table lists all commands from the hpmcmd program available on ATCA-7480. You can display this list and a short description about command using the help command (see section *help on page 58*). A detailed description of the commands is given in section *Supported Commands on page 48*.

Table 5-1 Command Overview

Command	Description
bootbankget	Gets the bootbank to boot from
bootbankset	Sets the bootbank to boot from
bootparamerase	Erases boot parameter value
bootparamget	Gets the boot parameter value
bootparamset	Sets a boot parameter value
chinfo	Retrieves the channel information
cmd	Executes IPMI commands
deviceid	Gets the Device ID

Table 5-1 Command Overview (continued)

Command	Description
frudata	Allows to get FRU info in hexadecimal numbers
fruinfoget	Gets the string fields from 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 to the FRU
fwprogevent	Sends a Firmware Progress Sensor Event
help	Gets the list of commands
ipmbaddress	Gets the IPMB address
ipmcstatus	Gets the IPMC status
lancfgget	Gets the LAN configuration parameter
lancfgset	Sets the LAN configuration parameter
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
macaddress	Lists the MAC addresses
partnumber	Gets the board part number
physlotnumber	Gets the board physical slot number
portget	Shows the current state of network interfaces governed by E_Keying events
portset	Enables/Disables the ports in a channel
posttypeget	Gets the posttype to run at boot
posttypeset	Sets the posttype to run at boot
sdr	Shows SDR records

Table 5-1 Command Overview (continued)

Command	Description
sdr_dump	Shows the SDR records in raw format
sdrinfo	Shows SDR information
sel	Shows SEL records
selclear	Erases all contents from the SEL
selinfo	Shows SEL information
sendamc	Sends an IPMI request to an MMC behind a remote IPMC
sendcmd	Sends an IPMI request to an IPMB address IPMC
serialoutputget	Determines which serial output source goes to a particular serial port connector
serialoutputset	Selects the serial output source of the serial port connector
shelfaddress	Gets the Shelf Address String
shelfslots	Prints the number of slots in the shelf
shelftype	Gets the Shelf Type from the Shelf FRU (Board Product Name)
slotmap	Prints the slotmap of the shelf
slotnumber	Shows the board slot number
solcfgget	Gets SOL configuration parameter
solcfgset	Sets SOL configuration parameter
version	Shows the hpmcmd version
watchdog	Controls Payload WDT functionality

5.3.4 Supported Commands

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

5.3.4.1 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 SMART EC ATCA blades is stored in redundant persistent memory devices. This allows the firmware image in one bank to serve as a backup for 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 as active boot device. Because, you can change the “active” device with the hpmcmd `bootbankset` command. Active status does not necessarily indicate which device was used on the last boot. It simply represents which device is set to be used on the next boot.

Synopsis

```
bootbankget <payload_cpu_selector>
```

Parameters

`payload_cpu_selector`
Is an integer between 0 and the number of CPU devices supported on the blade. On the ATCA-7480 the `payload_cpu_selector` is 0.

Example

```
hpmcmd -c bootbankget 0
BANK1
```

```
hpmcmd -c bootbankget 1
BANK0
```

5.3.4.2 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. On the ATCA-7480 the `payload_cpu_selector` is 0.

`newBootBank`
Can be set to `BANK1, BANK2...`

Example

```
hpmcmd -c bootbankset 1 BANK 1
```

5.3.4.3 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.

5.3.4.4 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`.

5.3.4.5 bootparamset

Description

This command sets the boot parameter value.

Synopsis

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

Parameters

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

`name`
Specifies name of the parameter.

`t`
Sends the command to `ipmbAddr:mmcAddr`.

5.3.4.6 cmd

Description

This command allows you to enter the 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. The `ipmi` command is 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 specification.
`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`

5.3.4.7 deviceid

Description

This command retrieves the raw IPMI `GetDeviceID` command response and decodes the IPMI message.

Synopsis

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

Parameters

`-t`
Sends the command to `ipmbAddre:mmcAddr.`

Example

```
root@ATCA-7480:~# hpmcmd -c deviceid

DEVICEID INFORMATION
--------------------
Device Id           =  0x00
Device Revision     =  0x00
Device Mode         =  Normal Operation ; Supports Device SDR
Firmware Revision   =  1.20
IPMI Version        =  2.0
Device Support      =  IPMB Evnt Gen; FRU; SEL; Sensor;
Manufacturer ID     =  0x000065CD
Product ID          =  0x201A
Auxiliary Revision  =  0x00000024 4
```

5.3.4.8 chinfo

Description

Retrieves information about an IPMI channel.

Synopsis

`Chinfo <channel>`

Parameters

`Channel`
Channel number

Example

```
root@ATCA7480:~# hpmcmd -c chinfo 0
Channel Medium Type   : IPMB (I2C)
Channel Protocol Type : IPMB-1.0
Session Support       : session-less
Active Session Count  : 0
Protocol Vendor ID    : 001BF2
```

5.3.4.9 frudata

Description

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

Synopsis

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

Parameters

`fruid`
Is 0 for the ATCA-7480 front board and 1 for the ARTM.

`-t`
Sends the command to `ipmbAddr:mmcAddr`.

Example

```
root@ATCA74780:~# hpmcmd -c frudata 0
01 00 00 01 0b 14 00 df 01 0a 19 20 40 90 c7 41
52 54 45 53 59 4e e3 50 43 41 2c 41 54 43 41 2d
37 34 38 30 2f 31 30 35 57 2f 30 47 42 2f 52 31
2e 30 2f 43 46 47 30 30 30 30 c7 45 31 35 32 44
38 43 ca 30 31 30 36 38 34 38 4e 30 31 c2 32 20
```

```
c1 00 00 00 00 00 00 14 01 09 19 c7 41 52 54 45
53 59 4e c9 41 54 43 41 2d 37 34 38 30 da 41 54
43 41 2d 37 34 38 30 2f 30 47 42 2f 52 31 2e 30
2f 43 46 47 30 30 30 30 c5 52 2e 31 2e 30 c7 45
```

5.3.4.10 fruinfoget

Description

This command retrieves information from the specified FRU.

Synopsis

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

Parameters

`fruid`
Is 0 for the ATCA-7480 front board and 1 for the ARTM.

`field`
Is one of the following data fields. If no field is specified, it retrieves the entire FRU information.

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

```
root@ATCA7480:~# hpmcmd -c fruinfoget 0
Common Header:
    Format Version = 1

Board Info Area:
    Version   = 1
    Language Code           = 25
    Mfg Date/Time          = Dec 22 00:00:00 2013 (9453600 minutes since
1996)
    Board Manufacturer      = ARTESYN
    Board Product Name      = PCA,ATCA-7480/105W/0GB/R1.0/CFG0000
    Board Serial Number     = E152D8C
    Board Part Number       = 0106848N01
    FRU Programmer File ID  = 2

Product Info Area:
    Version   = 1
    Language Code           = 25
    Manufacturer Name       = ARTESYN
    Product Name            = ATCA-7480
    Product Part / Model#   = ATCA-7480/0GB/R1.0/CFG0000
    Product Version         = R.1.0
    Product Serial Number   = E152D8C
    Asset Tag               =
    FRU Programmer File ID  =

Multi Record Area:

    PICMG LED Descriptor Record (ID=0x2f)
        Version = 0

    OEM MAC Addresses Record (ID=0x01)
        Version = 1

    Emerson Unknown Record (ID=0x10)

    PICMG Board Point-to-Point Connectivity Record (ID=0x14)
        Version = 1

    AMC Carrier Information Table Record (ID=0x1a)
```

```
    Version = 0

AMC Carrier Activation and Current Management Record (ID=0x17)
    Version = 0

AMC Carrier Point-to-Point Connectivity Record (ID=0x18)
  Version = 0

AMC Point-to-Point Connectivity Record (ID=0x19)
    Version = 0
```

5.3.4.11 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 ATCA-7480 Front board and 1 for the ARTM.

`-t`
Sends the command to `ipmbAddr:mmcAddr`.

Example

```
FrontBoard:
root@ATCA-7480:~# hpmcmd -c fruinv 0
FruSize = 2048
Accessed Units = Bytes

ARTM:
root@ATCA-7480:~# hpmcmd -c fruinv 1
FruSize = 512
Accessed Units = Bytes
```

5.3.4.12 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 ATCA-7480 Front board and 1 for the ARTM.

`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

```
root@ATCA7480:~# hpmcmd -c fruread 0 0 100
01 00 00 01 0b 14 00 df 01 0a 19 20 40 90 c7 41
52 54 45 53 59 4e e3 50 43 41 2c 41 54 43 41 2d
37 34 38 30 2f 31 30 35 57 2f 30 47 42 2f 52 31
2e 30 2f 43 46 47 30 30 30 30 c7 45 31 35 32 44
38 43 ca 30 31 30 36 38 34 38 4e 30 31 c2 32 20
c1 00 00 00 00 00 00 14 01 09 19 c7 41 52 54 45
53 59 4e c9
```

5.3.4.13 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 ATCA-7480 Front board and 1 for the ARTM.

`startAddress`
Is the starting address for writing.

`hexval1 .. hexvalN`
Is the hexadecimal value to write.

`-t`
Sends the command to `ipmbAddr:mmcAddr.`

5.3.4.14 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, "01" for 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.

5.3.4.15 help

Description

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

Synopsis

```
help
```

5.3.4.16 ipmbaddress

Description

This command retrieves the blade IPMB address.

Synopsis

```
ipmbaddress
```

Example

```
Hpmcmd -c ipmbaddress
Ipmbaddress is 0x88
```

5.3.4.17 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
IPMC Mode                =  NORMAL
Payload Control          =  Enabled
IPMC Outstanding Events  =   None
```

5.3.4.18 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 ATCA-7480 Front board and 1 for the ARTM.

`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
Current State = OVERRIDE
State Function/(ms) Duration(ms)      Color
-------------------------------------------
Override         Off       Always        Red
```

5.3.4.19 ledprop

Description

This command displays the FRU LED properties under IPMC control.

Synopsis

```
ledprop <fruid>
```

Parameters

`fruid`
Is 0 for the ATCA-7480 Front board and 1 for the ARTM.

Example

```
hpmcmd -c ledprop 0
FRU LEDs under IPMC control:    LED0 = Blue, Default: Blue
LED1 = Amber,Red, Default: Red
LED2 = Amber,Green,Red Default: Green
LED3 = Amber, Default: Amber
```

5.3.4.20 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 ATCA-7480 Front board and 1 for the ARTM.

`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
```

5.3.4.21 loglevelget

Description

This command retrieves the current `hpmagentd` log level. The log level of hpmcmd can be set with the environment variable, `HPMCMD_LOG_LEVEL`. For example, export HPMCMD_LOG_LEVEL=7 to set debug level. All log messages are sent to the syslog.

Log levels:
0 = Emergency
1 = Alert
2 = Critical
3 = Error

4 = Warning
5 = Notice
6 = Information
7 = Debug

Synopsis

`loglevelget`

Example

`hpmcmd -c loglevelget`

`5`

5.3.4.22 shelftype

Description

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

Synopsis

`shelftype`

Example

`hpmcmd -c shelftype`

`AXP-1440`

5.3.4.23 macaddress

Description

This command retrieves the list of available MAC addresses.

Synopsis

`macaddress [fruid]`

Parameters

`Fruid`
Is 0 for the ATCA-7480 Front board and 1 for the ARTM

Example

```
FrontBoard:
root@ATCA7480:~# hpmcmd -c macaddress 0
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:7a:f8
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:7a:f9
Base Interface             MAC Addr :  ec:9e:cd:11:7a:f6
```

```
Base Interface             MAC Addr :  ec:9e:cd:11:7a:f7
Fabric Interface           MAC Addr :  ec:9e:cd:11:7a:fa
Fabric Interface           MAC Addr :  ec:9e:cd:11:7a:fb
Fabric Interface           MAC Addr :  ec:9e:cd:11:7a:fc
Fabric Interface           MAC Addr :  ec:9e:cd:11:7a:fd
Serial over Lan Interface  MAC Addr :  ec:9e:cd:11:7a:fe
Serial over Lan Interface  MAC Addr :  ec:9e:cd:11:7a:ff
```

```
ARTM:
root@ATCA7480:~# hpmcmd -c macaddress 1
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f2
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f3
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f4
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f5
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f6
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f7
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f8
Front or Rear Panel Interface MAC Addr :  ec:9e:cd:11:77:f9…
```

5.3.4.24 lancfgget

Description

Gets LAN configuration parameter

Synopsis

`lancfgget <channel> [param]`

Parameters

`channel`
channel number

`param`

- auth-type-support
- auth-type-enables
- ip-addr
- ip-addr-src
- mac-addr
- subnet-mask
- ipv4-header-params
- primary-rmcp-port

secondary-rmcp-port
bmc-generated-arp-control
gratuidous-arp-interval
default-gateway-addr
default-gateway-mac-addr
backup-gateway-addr
backup-gateway-mac-addr
community-string
num-destinations
destination-type
destination-addr
vlan-id
vlan-prio
rmcp-cipher-support
rmcp-ciphers
rmcp-priv-levels
dst-addr-vlan-tags

Example

```
root@ATCA-7480:~# hpmcmd -c lancfgget 1
IP Address         : 192.168.25.10
Subnet Mask        : 255.255.0.0
Default Gateway    : 172.16.0.1
```

5.3.4.25 lancfgset

Description

Sets LAN configuration parameter

Synopsis

```
lancfgset <channel> <param> <value>
```

Parameters

channel
channel number

Example

```
root@ATCA-7480:~# hpmcmd -c lancfgset 1 ip-addr 192.168.25.10
Successful lancfgset Operation
```

5.3.4.26 nvmbankget

Description

This command retrieves the nvm bank for the specified XL710 NIC device. (only for ARTM-7480)

Synopsis

```
nvmbankget <nic_device_selector>
```

Parameters

`nic_device_selector`
XL710 NIC Device number (ARTM-7480). Valid entries are ’0’ and ’1’.

Example

```
root@ATCA7480:~# hpmcmd -c nvmbankget 0
XL710 NIC Device: 0 NVM BANK: 1
```

5.3.4.27 nvmbankset

Description

This command sets the nvm bank for the specified XL710 NIC device. (only for ARTM-7480)

Synopsis

```
nvmbankset <nic_device_selector> <bank>
```

Parameters

`nic_device_selector`
XL710 NIC Device number (ARTM-7480). Valid entries are ’0’ and ’1’.

bank
NVM Bank for specified NIC device (ARTM-7480). Valid entries are ’0’ and ’1’.

Example

```
root@ATCA7480:~# hpmcmd -c nvmbankset 0 1
```

5.3.4.28 partnumber

Description

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

Synopsis

```
partnumber
```

Example

```
hpmcmd -c partnumber
```

5.3.4.29 physlotnumber

Description

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

Synopsis

```
hpmcmd -c physlotnumber
```

5.3.4.30 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]
```

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.

Example

```
root@ATCA-7480:~# hpmcmd -c portget

  STATE   INTERFACE  CHANNEL  LINKTYPE  LINKEXT  GROUP  PORTS
-------------------------------------------------------------
 ENABLED     BASE       1        BASE      0       0    0
 ENABLED     BASE       2        BASE      0       0    0
 ENABLED    FABRIC      1       ETHER      4       0    0,1,2,3
```

```
DISABLED   FABRIC     1   RESERVED     1      0   0,1,2,3
DISABLED   FABRIC     1      ETHER     3      0   0
 ENABLED   FABRIC     2      ETHER     4      0   0,1,2,3
DISABLED   FABRIC     2   RESERVED     1      0   0,1,2,3
DISABLED   FABRIC     2      ETHER     3      0   0
DISABLED   FABRIC     3   RESERVED     1      0   0,1,2,3
DISABLED   FABRIC     3      ETHER     4      0   0,1,2,3
DISABLED   FABRIC     3      ETHER     3      0   0
DISABLED   FABRIC     4   RESERVED     1      0   0,1,2,3
DISABLED   FABRIC     4      ETHER     4      0   0,1,2,3
DISABLED   FABRIC     4      ETHER     3      0   0
DISABLED   UPDATE     3        OEM     0      0   0
DISABLED   UPDATE     4        OEM     0      0   0
DISABLED   UPDATE     5        OEM     0      0   0
```

5.3.4.31 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
```

5.3.4.32 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. Is 0 for the GPP, 1 for the SPP.

Example

```
hpmcmd -c posttypeget 1
LONG
```

5.3.4.33 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.

5.3.4.34 sdr

Description

This command shows the SDR records.

Synopsis

```
sdr
```

Example

```
root@ATCA7480:~# hpmcmd -c sdr

recID 1: management controller device locator record
     I2C slave addr:  44
    Channel number:  00
    Power state:     06
    Global init:     0C
    Capabilities:    2D
    Entity Id:       PICMG front board
    Entity instance: 60
    OEM:             00
    Id string:       ATCA-7480


recID 2: full sensor record
    owner is IPMB 88 sensor num 00 on lun 00 channel 00
    logical entity: PICMG front board - instance 60
```

```
    Hot Swap Carrier : FRU hot swap : sensor-specific discrete

recID 3: full sensor record
    owner is IPMB 88 sensor num 01 on lun 00 channel 00
    logical entity: RTM - instance 60
    Hot Swap RTM : FRU hot swap : sensor-specific discrete

recID 4: full sensor record
    owner is IPMB 88 sensor num 02 on lun 00 channel 00
    logical entity: PICMG front board - instance 60
    IPMB Physical : IPMC physical link : sensor-specific discrete

recID 5: full sensor record
    owner is IPMB 9C sensor num 06 on lun 00 channel 00
    logical entity: PICMG front board - instance 60
    Version change : 2B : sensor-specific discrete
```

5.3.4.35 sdr_dump

Description

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

Synopsis

```
sdr_dump
```

Example

```
root@ATCA7480:~# hpmcmd -c sdr_dump
SDR Records:
01 00 51 00 12 88 00 cc 2d 00 00 00 30 00 00 00  "..Q....Ì-...0..."
00 00 00 00 7c 00 00 00 00 00 00 00 00 00 00 00  "....|..........."
e0 06 02 00 00 00 00 00 02 00 00 00 00 00 00 00  "à..............."
```

5.3.4.36 sdrinfo

Description

This command shows the SDR information.

Synopsis

```
sdrinfo
```

Example

```
root@ATCA-7480:~# hpmcmd -c sdrinfo
SDR Information:
LUN 0 has 068 sensors; dynamic sensor population
LUN 1 has 000 sensors
LUN 2 has 000 sensors
LUN 3 has 000 sensors
```

5.3.4.37 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.

5.3.4.38 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
```

5.3.4.39 shelfaddress

Description

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

Synopsis

```
shelfaddress
```

Example

```
hpmcmd -c shelfaddress
01
```

5.3.4.40 shelfslots

Description

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

Synopsis

```
shelfslots
```

Example

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

5.3.4.41 slotmap

Description

This command displays a slotmap table for the shelf where the blade is installed.

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
------------------------------------------------------------
```

5.3.4.42 slotnumber

Description

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

Synopsis

```
slotnumber Parameters
```

Example

```
hpmcmd -c slotnumber
4
```

5.3.4.43 sel

Description

Shows the SEL records.

Synopsis

```
Sel
```

Example

```
root@ATCA7480:~# hpmcmd -c sel
0x01A2: Event: at: Dec 31 19:00:00 1969; from:(0xee,0,0);
sensor:(0xd7,30); event:(0x6f,asserted): a0 65 00
0x01A3: Event: at: Dec 31 19:00:00 1969; from:(0xee,0,0);
sensor:(0xda,34); event:(0x6f,asserted): a0 01 00
```

```
0x01A4: Event: at: Dec 31 19:00:00 1969; from:(0xee,0,0);
sensor:(0x08,31); event:(0x6f,asserted): 00 ff ff
0x01A5: Event: at: Dec 31 19:00:00 1969; from:(0xee,0,0);
sensor:(0x08,32); event:(0x6f,asserted): 00 ff ff
```

5.3.4.44 selinfo

Description

Shows the SEL information.

Synopsis

```
Sel
```

Example

hpmcmd -c selinfo

```
root@ATCA7480:~# hpmcmd -c selinfo
SEL version: 1.5
Number of log entries: 1023
Free space: 0 bytes
Events have been dropped due to lack of space in the SEL
Last addition timestamp: Sep 22 12:06:55 2014
Last erase timestamp: Dec 31 19:00:00 1969
Supported operations:
 - Reserve command supported
```

5.3.4.45 selclear

Description

Erases all contents of the SEL.

Synopsis

```
Selclear
```

Example

hpmcmd -c selclear

5.3.4.46 serialoutputget

Description

Determines which serial output source goes to a particular serial port connector.

Synopsis

```
serialoutputget <Serial Connector Type> <Instance Number>
```

Parameters

```
Serial Connector Type -
```
0 Faceplate Connector
1 Backplane Connector
2 On-board Connector
3 On-board Device (route to another chipset)

```
Instance Number -
```
zero-based instance number

Example

```
hpmcmd -c serialoutputget 0 1
Serial Output Selector: 1
```

5.3.4.47 serialoutputset

Description

Select the serial port output source for a serial port connector.

Synopsis

```
serialoutputset <Serial Connector Type> <Instance Number> <Serial Output
Selector>
```

Parameters

```
Serial Connector Type -
```
0 Faceplate Connector
1 Backplane Connector
2 On-board Connector
3 On-board Device (route to another chipset)

```
Instance Number -
```
zero-based instance number

```
Serial Output Selector -
```
an integer value >= 0

5.3.4.48 solcfgget

Description

Gets SOL configuration parameter

Synopsis

```
solcfgget <channel> [param]
```

Parameters

channel
channel number

param

- enable
- authentication
- char-settings
- retry
- nonvolatile-bit-rate
- volatile-bit-rate
- payload-channel
- payload-port

Example

```
root@ATCA7480:~# hpmcmd -c solcfgget 1
Enabled                           : true
Force Encryption                  : false
Force Authentication              : false
Privilege Level                   : Admin
Character Accumulate Interval (ms): 5
Character Send Threshold          : 1
Retry Count                       : 1
Retry Interval                (ms): 50
Non-Volatile Bit Rate        (kps): 115.2
Volatile Bit Rate            (kps): 115.2
Payload Port                      : 623
```

5.3.4.49 solcfgset

Description

Sets SOL configuration parameter

Synopsis

```
solcfgset <channel> <param> <value>
```

Parameters

`channel`
channel number

`param`

force-encryption true|false
force-authentication true|false
privilege-level user|operator|administrator|oem
char-accumulate-interval 1-1275 (ms)
char-send-threshold 0-255
etry-count 0-255
retry-interval 0-2550 (ms)
non-volatile-bitrate 9.6|19.2|38.4|57.6|115.2
volatile-bitrate 9.6|19.2|38.4|57.6|115.2
port 0-255

5.3.4.50 version

Description

This command displays the version of the hpmcmd software.

Synopsis

```
version
```

Example

```
hpmcmd -c version
3.11.3
```

5.3.4.51 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

Quick Assist Technology

6.1 ColetoCreek Acceleration

The RTM-ATCA-748X-40G is equipped with two optional ColetoCreek Acceleration devices (DH895x). These devices can be used for data compression/decompression and cryptographic tasks.

6.2 Software Overview

Intel provides software for ColetoCreek Acceleration devices, which includes drivers and acceleration code that runs on the Intel® Architecture (IA) CPUs and on the accelerator devices.

The software development package can be downloaded from the Intel website. For other Linux distributions, the QAT package must be compiled and installed manually.

6.3 The Intel Acceleration Software Subsystem

A subsystem (provided by Intel), which includes the software components that provide acceleration to applications running on the PCH. It contains:

- Services (Cryptographic, Data Compression)
  Includes the firmware that drives the various workload slices in the accelerator(s), and the associated Intel® architecture Service libraries that expose these workloads via Application Program Interfaces (API). These service libraries use the Acceleration Driver Framework (ADF) to plug into the OS and gain access to the hardware to communicate with the firmware.
- Intel® QuickAssist Technology APIs
  The Intel® QuickAssist Technology APIs provide service level interfaces for customer applications or Ecosystem Middleware to access the accelerator(s) in the PCH. More detail on the APIs and associated architecture are found in the Crystal Forest Software - Programmer's Guide from Intel.
- Acceleration Driver Framework (ADF)
  The ADF includes infrastructure libraries that provide various services to the different software components of the acceleration drivers. The software framework is used to provide the acceleration services API to the application. A configuration file enables customization of system operation.

The detailed information on the ColetoCreek APIs and software can be found at *www.intel.com*.

6.4 QuickAssist Software on the ATCA-7480 board

The BBS for ATCA-7480 comes with precompiled sample code from Intel for demonstration purpose.

Before installation of the RPM packages in the Root File system (for example, during network boot or first boot from disk) the version of the DH895x devices is read. Depending on the version information, the suitable RPM is installed.

When the ATCA-7480 blade is up and running, you can get version information as follows:

```
root@ATCA7480:~# /etc/init.d/accDevInfo.sh
###########################################
#    ColetoCreek Accelerator Device Info  #
###########################################
   DeviceSteppingSKU
-----------------------------------------
   09:00.0 A0 SKU3 or 4
   85:00.0 A0 SKU3 or 4
```

6.5 Running the QuickAssist Demo Software

6.5.1 Grub/netboot configuration

The `grub-` or `netboot` configuration file must be enhanced with the following parameters:

```
acpi_enforce_resources=lax intel_iommu=off
```

6.5.2 Installation of the QuickAssist Demo Software

To run the QuickAssist Demo software, install the `bbs-qat-atca7480-_<Version>-<OS distro> rpm-package`. This is usually done automatically during board startup. The package installs the required kernel drivers, demo applications, firmware, and test suites.

6.5.3 Start of the QuickAssist Demo Software

After installation of the RPM package, the QuickAssist service can be started with the **`qat_service start`**

```
root@ATCA7480:~# /etc/init.d/qat_service start
alg: No test for __gcm-aes-aesni (__driver-gcm-aes-aesni)
qat_1_6_adf 0000:09:00.0: PCI INT A -> GSI 42 (level, low) -> IRQ 42
qat_1_6_adf 0000:09:00.0: setting latency timer to 64
qat_1_6_adf 0000:85:00.0: PCI INT A -> GSI 64 (level, low) -> IRQ 64
```

```
qat_1_6_adf 0000:85:00.0: setting latency timer to 64
Loading SAL Module ...
qat_mem_module_init: get dynamic major 238
Loading QAE MEM Module ...
Processing file: /etc/dh895xcc_qReading config file.
a_dev0.conf
Starting acceleration device icp_dev0.
Resetting device icp_dev0
qat_1_6_adf 0000:09:00.0: irq 623 for MSI/MSI-X
qat_1_6_adf 0000:09:00.0: firmware: requesting dh895xcc/mof_firmware.bin
qat_1_6_adf 0000:09:00.0: firmware: requesting dh895xcc/mmp_firmware.bin
Started AE 0
....
Started AE 11
Processing file:Reading config file.
 /etc/dh895xcc_qStarting acceleration device icp_dev1.
Resetting device icp_dev1
a_dev1.conf
qat_1_6_adf 0000:85:00.0: irq 656 for MSI/MSI-X
qat_1_6_adf 0000:85:00.0: firmware: requesting dh895xcc/mof_firmware.bin
qat_1_6_adf 0000:85:00.0: firmware: requesting dh895xcc/mmp_firmware.bin
Started AE 0
...
Started AE 11
There is 2 acceleration device(s) in the system:
 icp_dev0 - type=dh895xcc, inst_id=0, node_id=0,  bdf=09:00:0, #accel=6,
#enginp
 icp_dev1 - type=dh895xcc, inst_id=1, node_id=1,  bdf=85:00:0, #accel=6,
#enginp
root@ATCA7480:~#

root@ATCA7480:~# cpa_sample_code signOfLife=1
icp_sal_userStartMultiProcess("SSL") started
qaeMemInit started
*** QA version information ***
device ID= 0
firmware = 2.1.0
mmp      = 1.0.0
software = 2.1.0
hardware = A0
*** END QA version information ***
```

```
*** QA version information ***
device ID= 1
firmware = 2.1.0
mmp      = 1.0.0
software = 2.1.0
hardware = A0
*** END QA version information ***
....
Inst 0, Affin: 0, Dev: 0, Accel 0, EE 0, BDF 09:00:00
Inst 1, Affin: 2, Dev: 0, Accel 0, EE 0, BDF 09:00:00
Inst 2, Affin: 1, Dev: 1, Accel 0, EE 0, BDF 85:00:00
Inst 3, Affin: 3, Dev: 1, Accel 0, EE 0, BDF 85:00:00
--------------------------------------
API                     Data_Plane
Session State           STATELESS
Algorithm               DEFLATE
Huffman Type            STATIC
Mode                    ASYNCHRONOUS
Direction               DECOMPRESS
Packet Size             8192
Compression Level       3
Corpus                  CALGARY_CORPUS
Number of threads       4
Total Responses         1584
Total Retries           3976
Clock Cycles Start      3572122098612
Clock Cycles End        3572126921442
Total Cycles            4822830
CPU Frequency(kHz)      1796022
Throughput(Mbps)        51904
Compression Ratio       0.51
--------------------------------------
```

Now, several tests are executed on both the DH895x devices on the Accelerator module.

The detailed information on Intel QuickAssist Technology can be found in:

- *Intel Communications Chipset 89xx Series Software Release Notes*
- *Intel Communications Chipset 89xx Series Software Programmer's Guide*
- *Intel QuickAssist Technology API Programmer's Guide*
- *Intel QuickAssist Technology Cryptographic API Reference Manual*
- *Intel QuickAssist Technology Data Compression API Reference Manual*

Network Management Utility

7.1 Overview

The Network Management Utilities tool can be used for the supervision of QSFP+/SFP devices and enable/disable the laser of QSFP+/SFP. The `nmucmd -c show trx` command displays all SFPs that are connected to the board. In order to use this, the `nmu_sfp` driver must be loaded.

The `show` command displays transceiver information, which is retrieved from the EEPROM of the SFP. Therefore, the QSFP+/SFP device must be compliant to the SFF-8472 specification. With the `trx` command, you can modify transceiver settings that are accessible in the EEPROM.

7.2 Installing nmucmd

7.2.1 NMU Application

The NMU application is provided as RPM package. It can be installed using `-Uvh --nodeps --force bbs-nmucmd-atca7480-<version>-<os>.rpm`.

After installation, the nmucmd tool can be found at `/opt/bladeservices/bin/`.

7.2.2 nmu_sfp driver

The nmucmd tool requires access to the EEPROM data of the QSFP+/SFP devices. This access is provided by `nmu_sfp` kernel driver module.

The `nmu_sfp` driver depends on the Intel i40e device driver for the Intel XL710 NIC devices (for example, on RTM-ATCA-748X-40G) and ixgbe for the Intel 82599 NIC devices (for example, on RTM-ATCA-7xxx-10G).

NOTICE

Make sure that the `nmu_sfp` driver is loaded after the Intel network device drivers, such as i40e and ixgbe. When removing, `nmu_sfp` is removed first and then Intel network device drivers.

You might get kernel crashes if you do not follow this sequence.

7.3 Syntax of nmucmd Tool

```
root@ATCA-7365:~# /opt/bladeservices/bin/nmucmd -c help
-----------------------------------------------------------------
         Command  -  Description
-----------------------------------------------------------------
            help  -  list of commands.
            show  -  Prints information about different network
                     entities.
             trx  -  Modify transceiver settings
         version  -  Shows the version of this tool.

root@ATCA-7365:~# /opt/bladeservices/bin/nmucmd -c show

HELP: show switch
    Dumps switch information

SYNOPSIS:
    show switch

HELP: show trx
    Dumps transceiver information

SYNOPSIS:
    show trx [switch] [port] [property]
    where:
              switch  -  name of the swith
                port  -  number of the port
            property  -  vendor-name
                         vendor-part-number
                         vendor-revision
                         vendor-serial-number
                         vendor-date
                         connector-type
                         ethernet-code
                         fibre-ch-link-length
                         fibre-technology
```

```
fibre-media
encoding
bitrate
tx-state
rx-state
tx-fault
internal-temperature
internal-supply-voltage
laser-bias-current
capability-tx-fault
capability-soft-tx-fault
capability-tx-disable
capability-soft-tx-disable
capability-rx-los
capability-soft-rx-los
```

Show Command

The `nmucmd` tool is based on the NMU API, which is part of the BLSV API. In the NMU architecture, an SFP is part of the hierarchical organization among switches and ports. A board may have multiple switches and a switch may have multiple ports. An SFP is always connected to a port. In case of ATCA-7480, there is no switch chip on the board, but the NMU defines a virtual one. You can see it using `nmucmd -c show switch`. For more information, see *Show Switch/NIC Device Type on page 86*.

Using command, `nmucmd -c show <switch name> [portnumber] [property]`, you can get the information about a QSFP+/SFP type. When the `portnumber` is specified in the command, the port specific QSFP+/SFP data is shown. If `portnumber` is not specified, all QSFP+/SFPs that are associated with the switch are displayed.

The property field is optional and using `nmucmd -c help show` command, you can get a list of the possible properties such as, vendor name, temperature, serial number, and so on. Without the property option, all properties of a QSFP+/SFP are displayed. For user, it is difficult to find out which switch and port number combination is associated to which network interface. `nmucmd` supports to specify either the switch and port number or only the name of the network interface.

trx Command

With the `trx` command, you can modify QSFP+/SFP settings. Using `nmucmd -c help trx` command, you can get a list of settings. For more information, see *Enable/disable tx-state of Specified Port on page 88*.

7.4 Examples

```
show bitrate of device i40e at port 2:
nmucmd -c show trx i40e 2 bitrate
    bitrate              : 10000 MBd
```

7.4.1 Show Switch/NIC Device Type

```
show switch/NIC type:
nmucmd -c show switch
i40e: Intel 40G NIC
```

7.4.2 Show Data of all Ports

```
show data of all ports assigned to the i40e device:
nmucmd -c show trx Transceivers of switch i40e
  Transceiver info at port 1:
    eth-device-name      : rtm4
    vendor-name          : AVAGO.
    vendor-part-number   : AFBR-79E4Z-D
    vendor-revision      : 01
    vendor-serial-number : QB310286
    vendor-date          : 08/02/11
    connector-type       : MPO
    ethernet-code        : 40G BASE-SR4
    encoding             : 64b/66b Line Code
    bitrate              : 10300 MBd
    tx-state             : Enabled
    rx-state             : Available
    tx-fault             : Ok
    internal-temperature : 22.88 degrees Celsius
    internal-supply-voltage : 3306000 uV
    laser-bias-current      : 6828 uA
     rx-input-power          : 790.30 uW (-1.02 dbm) type=average
    capability-tx-fault        : supported
    capability-soft-tx-fault   : supported
    capability-tx-disable      : supported
    capability-soft-tx-disable : supported
    capability-rx-los          : supported
    capability-soft-rx-los     : supported
  Transceiver info at port 2:
```

```
    eth-device-name      : rtm3
    vendor-name          : Tyco Electronics.
    vendor-part-number   : 2053638-3
    vendor-revision      : S
```

7.4.3 Show Data of a Specified Port

```
show all data assigned to i40e - port1:
nmucmd -c show trx i40e 1
  Transceiver info at port 1:
    eth-device-name      : rtm4
    vendor-name          : AVAGO.
    vendor-part-number   : AFBR-79E4Z-D
    vendor-revision      : 01
    vendor-serial-number : QB310286
    vendor-date          : 08/02/11
    connector-type       : MPO
    ethernet-code        : 40G BASE-SR4
    encoding             : 64b/66b Line Code
    bitrate              : 10300 MBd
    tx-state             : Enabled
    rx-state             : Available
    tx-fault             : Ok
    internal-temperature : 22.88 degree Celsius
    internal-supply-voltage : 3306000 uV
    laser-bias-current      : 6786 uA
    rx-input-power          : 789.30 uW (-1.03 dbm) type=average
    capability-tx-fault        : supported
    capability-soft-tx-fault   : supported
    capability-tx-disable      : supported
    capability-soft-tx-disable : supported
    capability-rx-los          : supported
    capability-soft-rx-los     : supported
```

7.4.4 Show tx-state of Specified Port

```
show tx-state of device i40e at port 1:
nmucmd -c show trx i40e 1 tx-state
    tx-state             : Enabled
```

7.4.5 Enable/disable tx-state of Specified Port

```
set tx-state of device i40e at port 1:
nmucmd -c trx i40e 1 tx-state on
nmucmd -c show trx i40e 1 tx-state
    tx-state             : Enabled
nmucmd -c trx i40e 1 tx-state off
nmucmd -c show trx i40e 1 tx-state
    tx-state             : Disabled
```

7.4.6 Show Device Name of Specified Port

```
show eth device at i40e port 2:
nmucmd -c show trx i40e 2 eth-device-name
….eth-device-name       : rtm3
```

7.4.7 Display QSFP+/SFP Temperature

```
show SFP temperature of device i40e at port 1:
nmucmd -c show trx i40e 1 internal- temperature
internal-temperature : 22.88 degrees Celsius
```

7.4.8 Display Used Bitrate

```
show bitrate of device nmu at port 1:
nmucmd -c show trx i40e 1 bitrate
bitrate: 10300 MBd
```

HPI-B Software

8.1 Overview

To ease the implementation of highly available systems with off-the-shelf building blocks, the Service Availability Forum (SA Forum) and HPI-B defines a set of platform-independent programming interfaces to monitor and control systems, such as AdvancedTCA systems, designed to provide high availability. HPI provides applications and middleware a consistent, standardized interface for managing hardware components.

You can get the HPI-B Software packages from local SMART EC sales representative. For more information on SMART Embedded Computing's HPI-B implementation, refer to the *System Management Interface Based on HPI-B User's Guide.*

Board Control Module

9.1 Overview

Board control is a Linux kernel module, which provides access to the Glue logic FPGA of the ATCA-7480 board. The `boardctrl` driver is installed during boot, which provides access to the BIOS version registers. After successful installation, an `init` script for loading/unloading the driver can be found at `/etc/init.d/boardctrl`.

The boardctrl driver is a prerequisite for the firmware upgrade tool fcu.

After loading the driver, you will find data provided at `/proc/boardinfo`, shown in the table below.

File	Description	Sample output
bios	Shows the version and release date of the currently installed BIOS	BIOS Vendor : ARTESYN BIOS Version : 0.4.1 BIOS Release date: 08/05/2014Jul/26/2012
board	Shows the board name/version as provided by the BIOS.	Board Vendor : ARTESYN Board Name : PCA,ATCA-7480/105W/0GB/R1.0/CFG0000 Board Version : R.1.0 Board Serial Number:
eccInfo	Reads the Error counters of the 2-memory controller of the board	Corrected Error Count per Rank: 1 2 3 4 5 6 7 8 CPU0 - IMC0: 0000 0000 0000 0000 0000 0000 0000 0000 CPU0 - IMC1: 0000 0000 0000 0000 0000 0000 0000 0000 CPU1 - IMC0: 0000 0000 0000 0000 0000 0000 0000 0000 CPU1 - IMC1: 0000 0000 0000 0000 0000 0000 0000 0000

File	Description	Sample output
tempInfo	Reads some temperature sensors (for example, CPU package temperature, maximum temperature of the DIMM channels, PCH package temperature.	0xff:0x1e.0x00-0x00c8: Package_Temperature: 67 C 0xff:0x1e.0x00-0x00cc: PPO_Temperature: 62 C 0xff:0x1e.0x00-0x0060: MEM_TRML_Temperature: Channel0_Max_Temperature: 47 C Channel1_Max_Temperature: 45 C Channel2_Max_Temperature: 0 C Channel3_Max_Temperature: 0 C 0x7f:0x1e.0x00-0x00c8: Package_Temperature: 74 C 0x7f:0x1e.0x00-0x00cc: PPO_Temperature: 69 C 0x7f:0x1e.0x00-0x0060: MEM_TRML_Temperature: Channel0_Max_Temperature: 36 C Channel1_Max_Temperature: 38 C Channel2_Max_Temperature: 0 C Channel3_Max_Temperature: 0 C 0x00:0x1f.0x06-0x0010: Wellsburg Temperature: 42 C
fpga	Shows some additional FPGA Information (CPU presence, CPU Type, and so on)	Board Module Version: 0x7480 FPGA version: 0x0A BIOS Reset Source: 0x00 OS Reset Source: 0xC1 IPMC Reset Source: 0x00 LED Control Register: 0x00
summary	Shows the summary of the board state (FPGA registers) and BIOS provided information	root@ATCA7480:~# cat /proc/boardinfo/summary Board Vendor: ARTESYN Board Name: PCA,ATCA-7480/105W/0GB/R1.0/CFG0000 Board Version: R.1.0 Board Serial Number: BIOS Vendor: ARTESYN BIOS Version: 0.4.1 BIOS Release Date: 08/05/2014 Memory Module: Device/Bank: J11/CPU 0 Size: 8192 Mbyte Data Width: 64 Bit Manufacturer: 0x2C00 Memory Module: Device/Bank: J13/CPU 0 ….

9.2 Board Control Tool

The board control module provides an IOCTL (input/output control) interface, which can be used by the userland applications. As an example, `fpgatool` can be used.

9.2.1 FPGATOOL

Description

Tool to read/write to the FPGA register set.

NOTE: Some of the registers are write-protected and cannot be overwritten by userland applications.

FPGATOOL can be found at `/opt/bladeservices/bin/fpgatool`.

Synopsis

```
root@ATCA7480:~# fpgatool

usage: fpgatool <command> <parameter> Available commands: read write dump info
```

Example

```
read from register 0x50 (LED Control register) root@ATCA7480:~# fpgatool read 0x50

Offset: 0x50  Value: 0x00 - 00000000

write to register 0x50 (LED Control register)

root@ATCA7480:~# fpgatool write 0x50 0x01

Old Value: 0x00 - New Value: 0x01

dump complete register set:

root@ATCA7480:~# fpgatool dump
```

Kernel and Root File System Configuration

10.1 Kernel configuration

The ATCA-7480 uses the 64-bit kernel configuration that comes along with OS distribution. The kernel configuration file is usually stored in the boot directory of the host system.

10.1.1 Additional Kernel Patches

The following table provides the kernel patch, which applies during the kernel build.

Table 10-1 Kernel Patches

Name	Host OS	Description
NA	RHEL 6.5	No Patches available.

10.1.2 Additional Kernel Drivers

Intel XL710 40Gb/s NIC requires a new network driver (i40e), which is not included in the current OS release. As long as this driver is not available, you need to download, compile and install the driver by yourself.

You can download the latest driver and function driver (required for virtualization) from: http://sourceforge.net/projects/e1000/files/i40e%20stable/

Installing i40e device driver

1. After downloading the driver, copy it to the driver package in the target system.
2. Extract the TAR file using `tar xzvf i40e-<version>.tgz` command.
   In the above command, replace `<version>` with the version of the device driver.
3. Go to `i40e-<version>/src/` directory and run `make install` command to install the driver.

Network Installation with RHEL

A.1 Kickstart File for Unattended RHEL6.5 Installation

This section provides an example of an anaconda-kickstart for automated network installation of the ATCA-7480 blade with RHEL6.5.

```
#version=RHEL6
# System authorization information
install

# Use NFS installation media
nfs --server=<ip address of your NFS server>
     --dir=<directory where the unpacked RHEL65 image can be found>
    --opts="rw,nfsvers=3"

auth --enableshadow --passalgo=sha512
firewall --service=ssh
selinux --enforcing
text

# Run the Setup Agent on first boot
#Setup Agent
firstboot --disable
ignoredisk --only-use=sda
# Keyboard layouts
keyboard us
# System language
lang en_US.UTF-8

# Network information
network  --bootproto=dhcp --device=eth1
network  --bootproto=dhcp --device=eth2
network  --bootproto=dhcp --device=eth3
network  --hostname=ATCA7480_RHEL65.localdomain

#root password
rootpw --plaintext root

# System timezone
timezone Europe/Berlin --isUtc
```

```
# User: Admin Password: Admin
user --homedir=/home/admin --name=admin -
password=$6$vMk8jmr7bC4iCwSA$ZEWVviSlkSTMXLwAbo1MRB9jKN09HHlOF66kv2EtV1.
ajpO49mY7JMh9DtRsyabYgfxjqEedoGa23n8QT7ryC. --iscrypted --gecos="Admin"

# System bootloader configuration
bootloader --location=mbr --driveorder=sda --append="crashkernel=auto
console=ttyS0,115200 selinux=0 pci=lastbus=255
acpi_enforce_resources=lax
intel_iommu=off ipmi_si.type=kcs ipmi_si.trydefaults=0"

# delete Master Boot record
zerombr

# use automatic partition layout
autopart

# Partition clearing information
clearpart --all --initlabel --drives=sda
reboot

%packages
@Base
@Core
@debugging
@development
@development
@network-server
@network-server
@network-tools
@nfs-file-server
@performance
@php
@security-tools
@server-platform
@server-platform-devel
@storage-server
@system-admin-tools
OpenIPMI
OpenIPMI-tools
```

```
bind-utils
binutils-devel
dhclient
dhcp
kernel-devel
redhat-rpm-config
rpm-build
tftp
tftp-server
tigervnc
tigervnc-server
watchdog
cpupowerutils
qemu-kvm
libvirt
python-virtinst
tpm-tools
powertop
net-snmp-utils
mcelog
expect
telnet
vim-enhanced
%end
```

Related Documentation

B.1 SMART Embedded Computing Documentation

The documentation listed is referenced in this manual. Technical documentation can be found by using the Documentation Search at *https://www.smartembedded.com/ec/support/* or you can obtain electronic copies of SMART EC documentation by contacting your local sales representative.

Table B-1 SMART EC Publications

Document Title	Publication Number
ATCA-7480 Installation and Use Guide	6806800T17
RTM-ATCA-748X-40G Installation and Use Guide	6806800T16