Basic Blade Services Software on ATCA-7368

Programmer's Reference

P/N: 6806800L95C

June 2014

ARTESYN™
EMBEDDED TECHNOLOGIES

Trademarks

Artesyn Embedded Technologies, Artesyn and the Artesyn Embedded Technologies logo are trademarks and service marks of Artesyn Embedded Technologies, Inc.© 2014 Artesyn Embedded Technologies, Inc. All other product or service names are the property of their respective owners.

Intel® is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ and all other Java-based marks are trademarks or registered trademarks of Oracle America, Inc. in the U.S. and other countries.

Microsoft®, Windows® and Windows Me® are registered trademarks of Microsoft Corporation; and Windows XP™ is a trademark of Microsoft Corporation.

PICMG®, CompactPCI®, AdvancedTCA™ and the PICMG, CompactPCI and AdvancedTCA logos are registered trademarks of the PCI Industrial Computer Manufacturers Group.

UNIX® is a registered trademark of The Open Group in the United States and other countries.

Notice

While reasonable efforts have been made to assure the accuracy of this document, Artesyn assumes no liability resulting from any omissions in this document, or from the use of the information obtained therein. Artesyn reserves the right to revise this document and to make changes from time to time in the content hereof without obligation of Artesyn to notify any person of such revision or changes.

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 The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of Artesyn.

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

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).

Contact Address

Artesyn Embedded Technologies
Marketing Communications
2900 S. Diablo Way, Suite 190
Tempe, Arizona 85282

Artesyn Embedded Technologies
Lilienthalstr. 17-19
85579 Neubiberg/Munich
Germany

Contents

List of Tables

List of Figures

About this Manual

Overview of Contents

This manual is divided into the following chapters and appendices.

Abbreviations

This document uses the following abbreviations:

Abbreviation	Definition
API	Application Programming Interface
AdvancedTCA	Advanced Telecommunications Computing Architecture
ATCA	Advanced Telecommunications Computing Architecture
BBS	Basic Blade Services
BIOS	Basic Input Output System
CGL	Carrier Grade Linux
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
ECC	Embedded Communications Computing
eSW	Embedded Software
FCU	FUF Command Line Utility
FM	Fault Management
FPGA	Field Programmable Gate Array
FRI	Firmware Recovery Image
FRU	Field Replaceable Unit
FUF	Firmware Upgrade Facility
FWH	Firmware Hub
GPIO	General Purpose Input/Output
HPI	Hardware Platform Interface
HPM	Hardware Platform Management
I/O	Input Output
IP	Internet Protocol
IPM	Intelligent Platform Management
IPMB	Intelligent Platform Management Bus

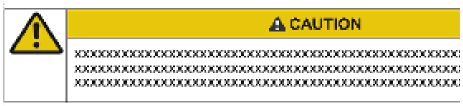
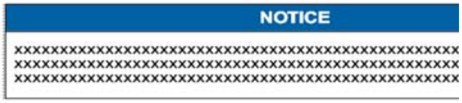
Abbreviation	Definition
IPMC	Intelligent Platform Management Controller
IPMI	Intelligent Platform Management Interface
LED	Light Emitting Diode
LHC	Link Health Check
LSP	Linux Support Package
LUN	Logic Unit Number
MAC	Media Access Control
MIB	Management Information Base
NTP	Network Time Protocol
OEM	Original Equipment Manufacturer
OSDL	Open Source Development Labs
PC	Personal Computer
PCI	Peripheral Component Interconnect
PCIx	PCI Express
PICMG	PCI Industrial Computers Manufacturers Group
PXE	Preboot Execution Environment
RAM	Random Access Memory
ROM	Read Only Memory
RPM	RedHat Package Manager
RTM	Rear Transition Module
SAF	Service Availability Forum
SAS	Serial Attached SCSI
SATA	Serial ATA
SCSI	Small Computer System Interface
SDR	Sensor Data Record
SMI	Serial Management Interface
SNMP	Simple Network Management Protocol

Abbreviation	Definition
SSD	Solid State Disk
SSH	Secure Shell
SSU	Synchronization Supply Unit
TAR	Tape Archive
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
UDP	User Datagram Protocol
USB	Universal Serial Bus

Conventions

The following table describes the conventions used throughout this manual.

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

Notation	Description
. . .	Omission of information from example/command that is not necessary at the time being
..	Ranges, for example: 0..4 means one of the integers 0,1,2,3, and 4 (used in registers)
|	Logical OR
	Indicates a hazardous situation which, if not avoided, could result in death or serious injury
	Indicates a hazardous situation which, if not avoided, may result in minor or moderate injury
	Indicates a property damage message
	No danger encountered. Pay attention to important information

Summary of Changes

See the table below for manual revisions and changes.

Part Number	Date	Description
6806800L95A	January 2011	First Version
6806800L95B	December 2011	GA Version
6806800L95C	June 2014	Re-branded to Artesyn

Chapter 1

Introduction

1.1 Overview

This manual is applicable to the following part numbers:

- ATCA-7368-0GB
- ATCA-7368-0GB-LS
- ATCA-7368-0GB-CE
- ATCA-7368-0GB-LS-CE
- ATCA-7368-L-CE
- ATCA-7368-LSL-CE

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

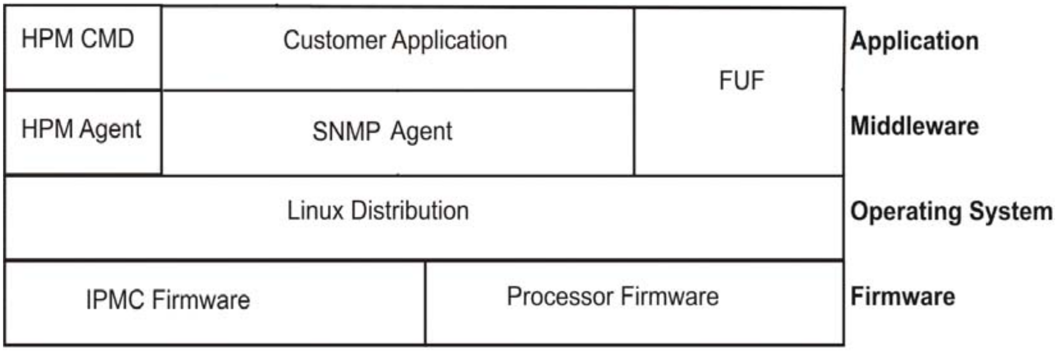
- BSP package to build WindRiver PNE Linux 4.0 for the ATCA-7368 blade.
- Several custom hardware management functions for the unique hardware of the blade and firmware upgrading facilities.
- A set of management routines for Linux and all hardware interfaces. Management access includes support for SNMP and a local console interface based on a standard Linux command shell.

1.2 Software Building Blocks

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

Figure 1-1 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

BBS for the ATCA-7368 consists of the following main software and services:

- Firmware Upgrade Facility
  The Firmware Upgrade Facility (FUF) provides a uniform way to upgrade firmware on Artesyn blades, regardless on which flash locations the firmware is stored. FUF upgrades the BIOS firmware as well as the IPMC firmware (via HPM agent). The FUF currently consists of a Firmware Upgrade Command Line Utility (FCU), flash device drivers, and specially prepared firmware recovery image files. The FUF can be used on switch and node blades.

- Linux Operating System
  Wind River Enterprise Linux 4.0 (Carrier Grade Linux) is the operating system for BBS blades and modules. The operating system comes with kernel 2.6.34.6. Various Linux services (above the kernel) will be activated by the BBS installation scripts.

- Hardware Platform Management
  Hardware Platform Management (HPM) in AdvancedTCA systems is based on Intelligent Platform Management Interface specification (IPMI). IPMI commands can be complex and cumbersome. Using a certain set of commands, HPM facilitates the blade or module-level hardware management.

- SNMP Agent
  As each BBS blade is individually managed, the default installation script installs and initializes the "Net-SNMP" agent.

- HPI-B
  This release contains an RPM which contains all necessary files for developing HPI-B applications. For further information refer to the System Management Interface Based on HPI-B (Centellis 2000/4440) User's Guide.

Installing the Basic Blade Services Software

2.1 Overview

Artesyn provides software images, including software updates, to its licensed customers. In order to obtain the latest BBS software versions, contact your local sales representative.

Generally, there are three typical ways of installing/booting the BBS software on ATCA-7368:

- Diskless client boot via network
- Installation and booting from SATA/SAS hard disk
- Installation and booting from on-board USB disk

On ATCA-7368 the SAS-HDD and SATA-HDD can reside in the AMC Bay or RTM when an RTM-ATCA-7368, RTM-ATCA-7360, RTM-ATCA-736X-DD blade which has the HDD installed. AMC Bay also supports SATA-HDD AMC without the installation of any RTMs. As an option a Solid State Disk (SSD) with SATA interface (SATA cube) can be placed on the front board.

For all these options you need to set up an external TFTP server to retrieve the required BBS files. Furthermore you need to do some initial configurations. Table 2-1 provides an overview of the main steps you need to take for the three installation/boot options. The detailed procedures can be found in the following sections.

Table 2-1 BBS Installation/Boot Options - Main Set-Up and Configuration Steps

Installation/Boot Option	Main Set-Up and Configuration Steps
Diskless client boot. Refer *Configuring ATCA-7368 for Diskless Client Boot of the BBS Software* on page 32.	1. Set up and configure external TFTP boot server 2. Configure DHCP server 3. Configure PXE boot options 4. Configure ATCA-7368 BIOS to boot from network
Installation and booting from SATA/SAS hard disk. The hard disk can be located on the RTM or locally (as SATA cube) on the front board. Refer *Installing BBS Software on Hard Disk Drive* on page 33.	1. Set up and configure external TFTP boot server 2. Configure DHCP server 3. Configure PXE boot options 4. Configure ATCA-7368 BIOS for network boot 5. Boot BBS initrd image 6. Install BBS image on hard disk (via install script) 7. Configure ATCA-7368 BIOS to boot from hard disk

Table 2-1 BBS Installation/Boot Options - Main Set-Up and Configuration Steps (continued)

Installation/Boot Option	Main Set-Up and Configuration Steps
Installation and booting from on-board USB flash. Refer *Installing BBS Software on On-Board USB Disk* on page 37.	1. Set up and configure external TFTP boot server 2. Configure DHCP server 3. Configure PXE boot options 4. Configure ATCA-7368 BIOS for network boot 5. Boot BBS initrd image 6. Install BBS image on USB flash (via install script) 7. Configure ATCA-7368 BIOS to boot from USB flash

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

2.1.1 Installation Scripts

The following table describes the installation scripts required for installing the operating system and blade utilities for ATCA-7368. These installation scripts require a TFTP and a DHCP server to download the installation files.

Table 2-2 Installation Scripts

	linuxrc	flashrfsrc	flashrc
Installed packages	Kernel, RFS and additional packages	Kernel, RFS and additional packages	Kernel, Ramdisk (including BBS packages)
Features	Enhanced partition layout, SMART-/Timezone and NTP configuration, Autoconfig	Simple partition layout and autoconfig	Timezone-/NTP configuration
Suitable for devices	HDD, SSD > 30 GB	HDD, SSD or on-board USB flash	On-board USB flash or USB stick

2.1.2 Package Information

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

The BBS distribution contains the following packages.

Table 2-3 BBS Distribution Packages

Description	File Name
Network Boot	
Operating System Kernel (Kernel version 2.6.34.6)	`kernel`
Ramdisk image for netboot	`ramdisk.image.gz`
Hard Disk Installation	
Root file system for hard disk installation	`rootfs.tar.gz`
Kernel command line	`default.bbs-atca7368`
Check sum of all files and RPMs	`files.sha1sum`
BIOS package	`atca-7368_em_bios_<version>.rpm`
IPMI firmware package (front blade)	atca-7368_em_ipmc_<CPU type>_<version>.rpm atca-7368_em_ipmc_<CPU type>-noamc_<version>.rpm
IPMI booter package (front blade)	atca-7368_em_ibb_<CPU type>_<version>.rpm atca-7368_em_ibbl_<CPU type>-noamc_<version>.rpm
Blade FRU data	atca-7368_em_frud_<CPU type>_<version>.rpm atca-7368_em_frud_<CPU type>-noamc_<version>.rpm
IPMI firmware package (RTM-ATCA-7368)	`artm-7368_em_mmcf_<version>.rpm`
IPMI firmware booter package (RTM-ATCA-7368)	`artm-7368_em_mmcb_<version>.rpm`
Blade FRU data (RTM-ATCA-7368)	`artm-7368_em_frud_<version>.rpm`
MMC(all) Firmware (RTM-ATCA-736X-DD)	artm-736xdd_em_mmcall_<version>.rpm
MMC Firmware (RTM-ATCA-7360)	artm-7360_em_mmcf_<version>.rpm

Table 2-3 BBS Distribution Packages (continued)

Description	File Name
MMC Boot loader Firmware (RTM-ATCA-7360)	artm-7360_em_mmcb_<version>.rpm
FPGA firmware package	`atca-7368_em_fpga_<version>.rpm`
Flash Drive support package	bbs-flashdriver-atca7301-1.0.1-1-pne20.rpm
Firmware Upgrade functionality - Upgrade tool for BIOS, FPGA and IPMC on front blade and RTM	`bbs-fuf-atca7368-<version>.rpm`
Hardware Platform Management consisting of daemon and client	`bbs-hpmagentcmd-atca7368-<version>.rpm`
Board control utility to get FPGA data	`bbs-boardctrl-atca7368-<version>.rpm`
Flashrom package (update tool for BIOS flash via SPI interface)	`bbs-flashrom-atca7368-<version>.rpm`
Flash driver for SFMEM module	`bbs-sfmem-atca7368-<version>.rpm`
HPI-B package	`bbs-hpib-<version>`
HPI-B client package	`bbs-hpib-clientsrc-<version>`
HPI-B developer package	`bbs-hpib-devel-<version>`
Uncompressed kernel image that has accessible, associated, and debuginfo for crash dump analysis usage.	`vmlinux`

The following rpm commands are useful to review package information.

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

2.1.3 Accessing the ATCA-7368 via Serial Console

In most procedures described in the following sections you need to invoke Linux commands or configure BIOS settings. In order to do this, you need to access the ATCA-7368 via the face plate serial port. If using a serial console or terminal emulator, the default serial port settings are:

- 9600 baud
- No parity
- Eight data bits
- One stop bit
- Flow control: xon/xoff
- Emulated terminal type: VT100

If you wish to access Linux via a Linux shell, the default account login is **`root`** with the password **`root`**. Refer *Login* on page 43, for more information.

2.2 Configuring TFTP, DHCP and PXE

For all installation and boot options, you need to set up and configure a TFTP server. Furthermore, for the diskless client and hard disk installation/boot option, you need to configure the system's DHCP server and configure PXE boot options. All related steps are described in the following section.

2.2.1 Create /tftpboot Directory and Copy Target Files

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

Creating the /tftpboot Directory and Copying the Target Files

To create the expected directory structure and copy the needed files, follow these steps.

1. On the host create a `/tftpboot` directory, if it does not already exist

   ```
   mkdir /tftpboot
   ```

2. Create a subdirectory for ATCA-7368, for example:
   `mkdir /tftpboot/ATCA7368`

3. Depending on the boot/installation option, copy or move the required installation files to the subdirectory. Refer Table 2-3 on page 25 for BBS distribution packaging.

The exact file names in your BBS release may be different. Refer the release notes applicable to your particular release. Table 2-3 on page 25 is only an example of a possible packaging with example file names.

Please ensure that the file attributes are set to 755, so that PXE can access and load the files.

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

2.2.2 Configuring a TFTP Server

The instructions in this section can be used to configure standard TFTP servers (BSD compatible) that are under the control of `xinetd`. The exact configuration settings depend on the particular system configuration, the following instructions are only meant as a general guideline.

Configuring a TFTP Server

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

1. Create (or edit) the file `/etc/inetd.d/tftp`. Depending on the particular system environment, it may contain the following entries.

```
#/etc/inetd.d/tftp
service tftp
{
   socket_type = dgram
```

```
        wait = yes
        user = root
        log_on_success += USERID
        log_on_failure += USERID
        server = /bin/in.tftpd
        server_args = -r blksize /tftpboot
        disable = no
        protocol = udp
    }
```

2. Create the directory `/tftpboot` and add the needed files as described in *Create /tftpboot Directory and Copy Target Files* on page 27.

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

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

Your TFTP server is now configured.

2.2.3 Configuring DHCP

The DHCP configuration file on an TFTP server (for example ATCA-F120 or an external TFTP server) resides in `/etc/dhcpd.conf`. Make sure this file contains the following entries (IP addresses may be different in your configuration):

```
#
# Sample dhcpd configuration file
#
#

allow bootp;
allow booting;
authoritative;
filename "pxelinux.0";
ddns-update-style ad-hoc;

option domain-name "booting.com";
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;

#Base 1 interfaces
subnet 192.168.21.0 netmask 255.255.255.0 {
   range 192.168.21.100 192.168.21.125;
   option broadcast-address 192.168.21.255;
}


#Base 2 interfaces
subnet 192.168.22.0 netmask 255.255.255.0 {
   range 192.168.22.100 192.168.22.125;
   option broadcast-address 192.168.22.255;
}
```

Restart DHCP service on your Linux DHCP server by issuing the following commands and make sure your DHCP service starts successfully against your configuration files:

```
#/etc/init.d/dhcp stop
```

```
#/etc/init.d/dhcp start
```

2.2.4 Configuring PXE

PXE determines which kernel and root file system image a blade gets from the server. The PXE environment as well as the bootable images usually reside in the `/tftpboot` directory on the server. The initial boot file is called `pxelinux.0` and the PXE configuration directory is in the `/tftpboot/pxelinux.cfg`. The default configuration file is called `/tftpboot/pxelinux.cfg/default`.

Example default file:

```
DEFAULT ATCA7368/kernel ramdisk_size=716800 console=ttyS0,9600n8
initrd=ATCA7368/ramdisk.image.gz root=/dev/ram0 ip=none ro pci=lastbus=255
quiet nopat
```

In this configuration, the same images are served to all blades in the chassis. In order to distinguish between blades and to serve different images, you can use different default files and link them to different' MAC addresses of different blades.

Depending on the particular BBS release, an example default file for the ATCA-7368 may be contained in the BBS package (check the release notes applicable to your blade release). This file contains all required kernel parameters. In order to use the default file, you need to link it to the MAC address of the ATCA-7368 as described below.

Example:
The following example shows how to set up the PXE environment for an ATCA-7368 blade. This is done by creating a new default file and linking it to the MAC address of the ATCA-7368 boot Ethernet interface, which is `00:80:42:1d:da:07` in the example.

PXE expects that the file name should be prefixed with "01" and all the characters in the file name are lower case letters.

Setting up the PXE Environment

Proceed as follows:

1. Make a new subdirectory in `/tftpboot`
   #**mkdir -p /tftpboot/ATCA7368**

2. Copy the corresponding boot image and RPMs to this directory.

3. Set up a new default files in `/tftpboot/pxelinux.cfg`, for example default.atca7368.
   The contents of default.atca7368 are:

```
DEFAULT ATCA7368/kernel ramdisk_size=716800 console=ttyS0,9600n8
initrd=ATCA7368/ramdisk.image.gz root=/dev/ram0 ip=none ro
pci=lastbus=255 quiet nopat
```

4. Link the MAC address of the blade to its boot default file, for example:
   ```
   #cd /tftpboot/pxelinux.cfg
   #ln -s default.atca7368 01-00-80-42-1d-da-07
   ```

2.3 Installation Procedures

The following subsections list the different BBS installation procedures.

2.3.1 Configuring ATCA-7368 for Diskless Client Boot of the BBS Software

This section describes the steps you need to take for performing diskless client boot of the BBS software.

Configuring BIOS for Diskless Client Boot

To configure BIOS for diskless client boot, proceed as follows:

1. Connect to the blade via the serial interface.

2. Power up or reboot the blade.

3. Quickly hold down the **<F2>** key on your keyboard until the BIOS menu appears.

4. Select **ADVANCED** on the top menu.

5. Scroll down to **BOOT FEATURES** by using the arrow keys.

6. Press **<ENTER>**.

7. Make sure that the following settings are enabled:

   `Base-Interface Network boot` or `Front Panel Network Boot` (depending on the interface you want to boot from).
   If any of these settings is disabled, enable the setting(s) and press **<F10>** or select **Exit Saving Changes**. This will save the new settings and restart the BIOS. After the restart, press **<F2>** to enter BIOS again and continue with the BIOS configuration.

8. Depending on which interface you want to boot from, put either **`Base Network 1, Base Network 2`**, **`FrontPanel Network 1, or FrontPanel Network 2`** to the first position of the **`Boot priority`** order list.

9. Save and exit.

Rebooting the Blade

To reboot a blade, proceed as follows:

1. Reboot the blade via:
   - Shelf manager
   - Opening and closing the lower handle switch on the face plate
   - Pressing the reset button on the face plate

2. Observe that the blade is getting a DHCP address and is loading the kernel and ramdisk image:

```
Try to load: pxelinux.cfg/<address>
boot:
Loading <blade/module>/kernel..............
Loading <blade/module>/ramdisk-image.gz....
```

3. Once the blade has fully come up, log on to the serial console as **`root`** with the default password **`root`**.

2.3.2 Installing BBS Software on Hard Disk Drive

This section describes how to install and boot the BBS software from hard disk. The BBS software can be installed on the following hard disk types:

- SAS/SATA hard disk drive installed on RTM or
- SATA cube on front board (optional)

The installation process starts with the booting of an initial ramdisk via network. The initial ramdisk is then used to copy (via TFTP) and interactively install the kernel, the root file system, and other BBS software on the disk.

The following procedures describe these steps in detail.

Configuring BIOS for Diskless Network Boot

To configure BIOS for network boot, proceed as follows:

1. Connect to the blade via the serial interface.
2. Power up or reboot the blade.
3. Quickly hold down the **`<F2>`** key on your keyboard until the BIOS menu appears.
4. Select **`ADVANCED`** on the top menu.
5. Scroll down to **`BOOT FEATURES`** by using the arrow keys.
6. Press **`<ENTER>`**.
7. Make sure that the following settings are enabled:

   `Base-Interface Network boot` or `Front Panel Network Boot` (depending on the interface you want to boot from).
   If any of these settings is disabled, enable the setting(s) and press **`<F10>`** or select **`Exit Saving Changes`**. This will save the new settings and restart the BIOS. After the restart, press **`<F2>`** to enter BIOS again and continue with the BIOS configuration.
8. Depending on which interface you want to boot from, put either **`Base Network 1, Base Network 2`**, **`FrontPanel Network 1,`** or **`FrontPanel Network 2`** to the first position of the **`Boot priority`** order list.
9. Save and exit.

Installing Files and Configuring TFTP on the ATCA-7368

After the system has come up, install Linux with the following procedure:

1. Login as **`root`**.
2. Identify the Linux device name of the hard disk on which you want to install BBS. To do so, enter **`fdisk -l`**. This displays available hard disks, their Linux device names and also the storage capacity.

3. Run the `linuxrc` script from the `/opt/bladeservices/tools` directory:
   `./linuxrc`
   The hard disk installation begins by checking for necessary commands on the system.

4. Enter the information requested by the script, such as the TFTP server address from where the software is loaded, NTP server address, and time zone.

5. Above steps installs all the BBS packages that are available after tftpboot and Linux Boot loader, on the hard disk.

Refer *Installing BBS Using Hard Disk* on page 131, for step-wise output of the installation and configuration procedure.

Performing the Final Configuration on the ATCA-7368

The final configuration includes configuring the host name and password and setting the time zone.

1. Configure the host name:
   ```
   Choose a hostname for this machine [ ]
   ```
   There is no default hostname. Enter a value here.

2. Configure the root password:
   ```
   Enter new UNIX password:
   Retype new UNIX password:
   ```
   There is no default root password.

Now the boot loader `grub` is installed. After that you need to configure BIOS to boot from the hard disk as described in the following procedure.

Configuring the ATCA-7368 BIOS to Boot from Hard Disk

To configure BIOS on an ATCA-7368 blade, proceed as follows:

1. Connect to the blade via the serial interface.

2. Power up or reboot the blade.

3. Quickly hold down the **<F2>** key on your keyboard until the BIOS menu appears.

4. Select **BOOT** on the top menu.

5. Scroll down to **BOOT OPTIONS** by using the arrow keys.

6. Press **<ENTER>**.

7. Depending on the hard disk type and the location where the hard disk is installed, make sure that the corresponding BIOS setting shown in the following table is enabled.

Hard Disk	BIOS Menu and Setting Which Must Be Enabled
SAS hard disk installed on RTM	“Boot Features” -> "ARTM SAS boot"
SAS hard disk installed on AMC module in AMC bay 1 (upper AMC bay)	“Boot Features” -> "ARTM SAS boot"
SATA hard disk installed on AMC module in AMC Bay 4 (lower AMC bay)	"Advanced Chipset Control" -> "Serial ATA"

If the desired setting was previously NOT enabled, enable the desired setting and press **<F10>** or select **Exit Saving Changes**. This will save the new settings and restart the BIOS. After the restart, press **<F2>** to enter BIOS again and continue with the BIOS configuration.

8. Put the hard disk which you want to boot from to the first position of the Boot priority order list.

9. Save and exit.
   After a successful reboot, you can logon as root using the password you have defined during the final configuration.

2.3.3 Installing BBS Software on On-Board USB Disk

The ATCA-7368 BBS supports network boot via tftp. You can use `flashrfsrc` script to install the root file system on the ATCA-7368 on-board flash and to boot from it. The `flashrfsrc` script performs the following tasks:

- formats the ATCA-7368 on-board flash device
- transfers the kernel, root file system, and BBS packages from the tftp server to the on-board flash device
- installs and configures the GRUB boot loader

Executing the flashrfsrc script will erase all the data existing on the on-board flash.

Configuring BIOS for Diskless Network Boot

To configure BIOS for network boot, proceed as follows:

1. Connect to the blade via the serial interface.
2. Power up or reboot the blade.
3. Quickly hold down the <F2> key on your keyboard, until the BIOS menu appears.
4. Select **`BOOT`** on the top menu.
5. Scroll down to **`BOOT OPTIONS`** by using the arrow keys.
6. Press <ENTER>.
7. Ensure that the following settings are enabled:
   `Base Network 1, Base Network 2, FrontPanel Network 1,` or **`FrontPanel Network 2`**; depending on the interface you want to boot from. If any of these settings is disabled, enable the setting(s) and press <F10> or select **`Exit Saving Changes`**, to save the new settings and restart the BIOS. After the restart, press <F2> to enter BIOS again and continue with the BIOS configuration.

8. Depending on which interface you want to boot from, put either **`Base Network 1, Base Network 2`**, **`FrontPanel Network 1,`** or **`FrontPanel Network 2`** to the first position of the Boot priority order list.

9. Save and exit.

Installing Root File System on the on-board Flash (Optional)

To install the OS and BBS software on the on-board flash:

1. Start the blade using `tftp-boot`.

2. Execute `flashrfsrc` script (available in the directory of `/opt/bladeservices/tools/`) after an initial netboot. It allows an automatic installation of the OS and BBS packages on the on-board USB flash disk. Provide the device name on which the software is to be installed when you are prompted for it. You can modify the configuration file `flashConfig.default` as per your requirement and use it for the installation.

3. Enter the information requested by the script, such as the TFTP server address from where the software is loaded, NTP server address, and time zone.

4. Above steps installs all the BBS packages that are available after tftpboot and Linux Boot loader, on the hard disk.

5. Reboot the blade.

6. Press <F2> to configure the BIOS.

7. In the Boot menu, move `onboard:USBHdd SMART eUSB` to the first option in the Boot Priority Order list.

8. Save and exit BIOS settings and continue booting.

After successful installation, the OS is loaded from the on-board USB flash disk drive.

Refer Appendix A, *Installing and Configuring BBS,* on page 131, for step-wise output of the installation and configuration procedure.

2.4 Upgrading the Software

Software updates are usually delivered as rpm-files. To install the files on the disk of the blade, copy the new RPM file to the blade, stop the application using this rpm, remove the original files (using the `rpm -e <package>` command) and install the newly copied rpm (using the command `rpm -Uvh <package-name>` command).

To upgrade the BBS software for diskless clients, you have to delete the installation files in the `/tftpboot` directory on the tftpserver, copy the new installation files into this directory, and follow the instructions in *Configuring ATCA-7368 for Diskless Client Boot of the BBS Software* on page 32.

2.5 Adapting the BBS Software to Customer’s Needs

The BBS software structure allows a maximum flexibility with regards to customer’s adaptations. Software packages can easily be installed into or removed from existing installations.

The following adaptations are possible:

- Modifying the NetBoot root file system
- Modifying the hard disk installation
- Modifying the hard disk installation procedure
- Modifying the Configuration of the Artesyn-Supplied CGL Kernel

2.5.1 Modifying the NetBoot Root File System

The netboot root file system is stored in the file `ramdisk.image.gz` on the TFTP server. In order to change the system’s behavior regarding network booting blades, you have to modify the root file system.

As root:

```
# cd /tftpboot/<blade or module to be modified>
# mkdir mnt
# gunzip ramdisk.image.gz
# mount -o loop ramdisk.image mnt
# pushd mnt
# ..... /* make all modifications and enhancements: delete, add or
change files*/
# popd
# umount mnt
# gzip -9 ramdisk.image
```

The blade will now boot the modified root file system.

2.5.2 Modifying Hard Disk Installation

The hard disk installation can be changed after the blade has been installed or by modifying the file `rootfs.tar.gz` prior to the installation. After modifying this file, you have to compute and add the sha1 checksum of the modified root file system to the files.sha1sum in the installation directory on the TFTP server.

The example below shows how to change the default login behavior.

```
# cd /tftpboot/...   (cd to the directory containing the
rootfs.tar.gz you want to modify)
# mkdir rootfs
# cd rootfs
# tar xzf ../rootfs.tar.gz
* Make your modifications and enhancements to the root filesystem
in the current directory
# tar czf ../rootfs.tar.gz .
# cd ..
# sha1sum rootfs.tar.gz
*Correct the sha1sum for rootfs.tar.gz in files.sha1.sum
```

2.5.3 Modifying the Hard Disk Installation Procedure

The hard disk installation procedure is based on the `files.sha1sum` in the installation directory on the TFTP server. All packages which are copied to the blade during installation are listed in the `files.sha1sum` together with their sha1sum. The standard installation process accepts rpm, tar, and tgz files and all files that have "kernel" in the file name.

The packages from `files.sha1sum` are installed in the same sequence as listed in the file `files.sha1sum`. The installation process re-calculates the sha1sum of the packages on the blade and compares it to the sha1sum determined by `files.sha1sum`. This ensures a protection against errors and faults during file transmission. The user will be notified in case of mismatch. In that case, you have to repeat the installation procedure.

The root file system must precede the rpm files in the **`files.sha1sum`** file.

2.5.4 Modifying the Configuration of the Artesyn-Supplied PNE Linux Kernel

The current kernel configuration of a running ATCA-7368 installation can be retrieved using the Linux command `zcat /proc/config.gz` or from `atca7368_em_bsp/template/board/atca7368/linux/atca7368.cfg` in the LSP directory of the Release package.

To modify the configuration of the CGL kernel supplied by Artesyn, consult your local Artesyn sales representative for assistance and further information.

Linux Distribution Description

3.1 Overview

This chapter describes the Linux distribution of ATCA-7368 BBS.

3.2 Distribution Description

The BBS for the ATCA-7368 is based on Wind River Enterprise Linux 4.0, which is a Linux distribution built on Linux 2.6.34.6 kernel technology.

3.3 Reliability

The hard disk installation is configured to use the journaling file system `ext3`. In this distribution majority of errors that are caused due to improper shutdown are fixed automatically during the boot process. Catastrophic errors that cannot be fixed automatically will yield to a command prompt, allowing the super user to execute the `fsck` command on the affected partition.

3.4 Login

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

If you use a serial console or terminal emulator, the serial/RTM port settings are 9600 baud, no parity, 8 data bits, and 1 stop bit.

If you use secure shell 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 user "admin".If you use SSH, refer *Network Services Configuration* on page 47 for default IP address assignments.

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

The following table lists available default login accounts.

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

3.5 Long POST/Diagnostics

The long POST (Power-On Self test) is an extension to the standard POST which the ATCA-7368 executes after power-up. It is executed during the booting of the Linux operating system and includes higher-level tests. This section describes which tests are by default executed during the long POST, how to obtain the results of these tests and how to add your own test routines.

3.5.1 Default Test Routines

The long POST test routines are implemented as Linux scripts which are invoked during the Linux boot phase. The test scripts which are to be executed need to be defined in the IPMI boot parameter variable `runLP` or as additional parameter in the kernel command line (`runLP=`...`,` ...`,`). Further details are given in *Configuring the Long POST Behavior* on page 45.

Each test routine displays the test status on the console and writes it to the Linux log module (via `logger`). Furthermore, each test routine writes status information to the IPMI sensor "System Firmware Progress" (type `0xF0`). The used event data values are Artesyn-specific. The following table provides details.

Table 3-1 Long POST Standard Test Routines - Generated IPMI Data

Action	Data Written to System Firmware Progress Sensor
Test routine is started	Offset 0: 0x02 Offset 1: 0xFD Offset 2: 0x1E
Tests routine detects an error.	Offset 0: 0x00 Offset 1: 0xFD Offset 2: 0x1E

The following table lists the names of the default long POST test routines and describes which tests each routine performs.

Table 3-2 Long POST Default Test Routines

Test Routine Name	Description
cpuspeed	This tool gives an overview on the active performance governors and the frequency per core.
memSize	Checks the amount of memory physically installed and the memory seen by the Operating system.
rtctest	Tests the functionality of the real time clock.
eccTest.sh	This scripts tests checks the ECC Error counter in the memory controller.

3.5.2 Configuring the Long POST Behavior

The names of the test scripts which are to be executed have to be defined in the IPMI system boot parameter variable `runLP` as a comma-separated list, for example as follows: runLP=cpuspeed,memSize.sh,rtctest,eccTest.sh

The scripts are expected to be located in the following directory: `/opt/bladeservices/tools/LP`. So in order to add your own scripts, simply add an entry to the IPMI boot variable `runLP` or add an appropriately defined kernel boot parameter `runLP` and place the script(s) in `/opt/bladeservices/tools/LP`. Depending on your system configuration, you may want to design your test scripts to generate console output, write to the log module and store any results in the IPMI System firmware progress sensor as done by the default test scripts.

The IPMI boot parameter can be set by using the hpm command **`bootparamset`** (**`hpm -c bootparamset`**).

When Linux is booted, the Linux start-up script `/etc/init.d/LPmain.sh` is executed. It reads and analyses the IPMI boot parameter variable `runLP` and invokes the listed test scripts (if any) in the given order. For more advanced customizations, you may want to modify the `/etc/init.d/rd.d/LPmain.sh` script.

The `LPmain.sh` provides the following options:

Table 3-3 Long POST Script LPmain.sh - Options

Option	Description
start	Starts the Long POST for the specified test cases.
status	Gives information about the test cases to be executed during long POST and shows whether longPOST is switched on.
enableLP	Enables the long POST. The Long POST will be executed during the next OS startup.
disableSP	Disables the long POST. The long POST will not be executed during the next OS startup.

3.6 Linux Services Initialization

Table 3-4 describes the generic Linux run levels. Table 3-1 describe the services configured to start in the various Linux run levels. Per default, the blade first runs run level S and then boots into run level 3 as configured by the factory.

Table 3-4 Generic Linux Run Levels

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

3.6.1 RC Scripts

In addition to the rc-scripts of the Wind River PNE 4.0 Linux configuration the following start/stop scripts are added to ATCA-7368.

Run Level	Script Name	Description
rc2.d	`S20kdump`	Starts the kdump service.
rc3.d	`S01bbsrpms`	Installs bbs-rpms during initial blade startup, for example, after blade installation or boot via network boot.
	`S02bbsinit`	Starts boardctrl driver and the optional persistent memory drivers (pram and sfmem when the optional memory module is installed).
	`S03longPost`	Performs some basic blade tests, before most of the OS services are started.
	`S09hpm`	Starts the hpmagent.
	`S10ethDevOrdering`	Performs the eth-device reordering and renaming.
	`S57bbsvlan`	Configures ip-addresses for the fabric interfaces and brings up the base and fabric interfaces.
	`S99osBootSensor`	Specifies the boot device and checks if watchdog is started or not. It stops the watchdog if it is started already.
rc6.d	`K05hpm`	Stops the hpmagent.

3.7 Network Services Configuration

The following sections describe the default configuration for network services.

3.7.1 ATCA-7368 Ethernet Interfaces

The ethernet devices, such as eth0, eth1, and eth2 in Linux distribution are renamed to more meaningful name in ATCA-7368, such as base1, base2, and fabric1. This renaming is done in the `/etc/init.d/ethDevOrdering.sh` script, before the network startup.

The following table specifies the Ethernet devices supported by ATCA-7368.

Device Name	Description	Speed	Location	IP address
base1	Base Interface 1	1GbE	Base blade -> Backplane	Obtained by the DHCP client request.
base2	Base Interface 2	1GbE	Base blade -> Backplane	Obtained by the DHCP client request.
fabric1	Fabric Interface 1	10GbE	Base blade -> Fabric Interface on Backplane	Static IP address. It is computed as: `192.168.<fabricIf>.<slotnumber*10>` `fabricIf` can have value of ;'11" for Fabric Interface 1 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.
fabric2	Fabric Interface 2	10GbE	Base blade -> Fabric Interface on Backplane	Static IP address. It is computed as: `192.168.<fabricIf>.<slotnumber*10>` `fabricIf` can have value of ;'11" for Fabric Interface 1 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.
pan0	Front Panel Interface ETH0	1GbE	Base blade	No IP address assigned.
pan1	Front Panel Interface ETH1	1GbE	Base blade	No IP address assigned.

3.8 Tools

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

3.8.1 Performance Tool

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	Dump CPU Frequency/Governor Info
-h	Help
-p	Print supported governors
-s	Set 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@ACPI4-C-9:xxxxxx#:/opt/bladeservices/tools#
/opt/bladeservices/tools/cpuspeed

####### CPU Frequency Info #######

Number Of Cores: 12

MinFrequency:    1596000

MaxFrequency:    1730000

Available Governors: ondemand userspace powersave performance

####### Frequency Info Per Core #######

Core: Governor:     CurrentFrequency:

    0  performance      1730000

    1  performance      1730000

    2  performance      1730000

    3  performance      1730000

    4  performance      1730000

    5  performance      1730000

    6  performance      1730000

    7  performance      1730000

    8  performance      1730000

    9  performance      1730000

   10  performance      1730000

   11  performance      1730000
```

3.8.2 IPMIBPAR

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

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

The following example describes the steps required to change the BootOrder from SAS-HDD to Base Network1.

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

```
root@ACPI4-C-9:~# 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 = 608 (0x260)

0000  5c 02 77 68 65 61 3d 6f 6e 00 68 79 70 65 72 5f   <\.whea=on.hyper_>

0010  74 68 72 65 61 64 69 6e 67 3d 6f 6e 00 6c 69 6d   <threading=on.lim>

0020  69 74 5f 63 70 75 69 64 3d 6f 66 66 00 65 64 5f   <it_cpuid=off.ed_>

0030  62 69 74 3d 6f 6e 00 68 77 5f 70 72 65 66 65 74   <bit=on.hw_prefet>

0040  63 68 65 72 3d 6f 6e 00 61 64 6a 5f 63 61 63 68   <cher=on.adj_cach>

0050  65 5f 70 72 65 66 65 74 63 68 3d 6f 6e 00 6c 31   <e_prefetch=on.l1>
```

```
0060  5f 70 72 65 66 65 74 63 68 3d 6f 6e 00 64 61 74   <_prefetch=on.dat>
0070  61 72 65 75 73 65 5f 6f 70 74 6d 69 7a 65 3d 6f   <areuse_optmize=o>
0080  6e 00 76 69 72 74 75 61 6c 69 7a 61 74 69 6f 6e   <n.virtualization>
0090  3d 6f 6e 00 74 75 72 62 6f 5f 6d 6f 64 65 3d 6f   <=on.turbo_mode=o>
00a0  6e 00 72 74 5f 65 72 72 5f 6c 6f 67 3d 6f 66 66   <n.rt_err_log=off>
00b0  00 70 63 69 5f 6c 6f 67 3d 6f 66 66 00 76 74 5f   <.pci_log=off.vt_>
00c0  64 3d 6f 66 66 00 69 6e 74 5f 6d 61 70 3d 6f 6e   <d=off.int_map=on>
00d0  00 65 63 63 5f 73 75 70 70 6f 72 74 3d 6f 6e 00   <.ecc_support=on.>
00e0  68 77 5f 6d 65 6d 5f 74 65 73 74 3d 6f 6e 00 70   <hw_mem_test=on.p>
00f0  61 74 72 6f 6c 5f 73 63 72 75 62 3d 6f 6e 00 64   <atrol_scrub=on.d>
0100  65 6d 61 6e 64 5f 73 63 72 75 62 3d 6f 66 66 00   <emand_scrub=off.>
0110  61 6c 6c 5f 75 73 62 5f 64 65 76 69 63 65 3d 6f   <all_usb_device=o>
0120  6e 00 75 73 62 5f 32 2e 30 5f 63 6f 6e 74 5f 6d   <n.usb_2.0_cont_m>
0130  6f 64 65 3d 6f 6e 00 75 73 62 3d 66 70 5f 6f 6e   <ode=on.usb=fp_on>
0140  2c 72 74 6d 5f 6f 6e 2c 6f 6e 62 6f 61 72 64 5f   <,rtm_on,onboard_>
0150  6f 6e 00 75 73 62 5f 62 6f 6f 74 3d 6f 6e 00 62   <on.usb_boot=on.b>
0160  61 75 64 72 61 74 65 3d 33 38 34 30 30 00 6f 73   <audrate=38400.os>
0170  5f 62 6f 6f 74 5f 77 61 74 63 68 64 6f 67 3d 6f   <_boot_watchdog=o>
0180  66 66 2c 35 2c 72 65 73 65 74 00 66 72 6f 6e 74   <ff,5,reset.front>
0190  6e 65 74 5f 62 6f 6f 74 3d 6f 6e 00 62 61 73 65   <net_boot=on.base>
01a0  6e 65 74 5f 62 6f 6f 74 3d 6f 6e 00 66 61 62 72   <net_boot=on.fabr>
01b0  69 63 6e 65 74 5f 62 6f 6f 74 3d 6f 6e 00 61 72   <icnet_boot=on.ar>
01c0  74 6d 5f 6e 65 74 5f 62 6f 6f 74 3d 6f 66 66 00   <tm_net_boot=off.>
01d0  61 72 74 6d 5f 73 61 73 5f 62 6f 6f 74 3d 6f 6e   <artm_sas_boot=on>
```

```
01e0  00 61 72 74 6d 5f 66 63 5f 62 6f 6f 74 3d 6f 66   <.artm_fc_boot=of>
01f0  66 00 73 73 63 5f 73 75 70 70 6f 72 74 3d 6f 6e   <f.ssc_support=on>
0200  00 6e 75 6d 62 65 72 5f 6c 6f 63 6b 3d 6f 6e 00   <.number_lock=on.>
0210  62 6f 6f 74 5f 6f 72 64 65 72 3d 66 72 6f 6e 74   <boot_order=front>
0220  6e 65 74 2c 66 72 6f 6e 74 6e 65 74 2c 62 61 73   <net,frontnet,bas>
0230  65 6e 65 74 30 2c 62 61 73 65 6e 65 74 31 2c 65   <enet0,basenet1,e>
0240  66 69 73 68 65 6c 6c 00 73 6d 62 69 6f 73 5f 65   <fishell.smbios_e>
0250  76 65 6e 74 5f 6c 6f 67 3d 6f 6e 00 00 00 58 e2   <vent_log=on...X.>


IPMI Boot Parameter:
whea=on
hyper_threading=on
limit_cpuid=off
ed_bit=on
hw_prefetcher=on
adj_cache_prefetch=on
l1_prefetch=on
datareuse_optmize=on
virtualization=on
turbo_mode=on
rt_err_log=off
pci_log=off
vt_d=off
int_map=on
```

```
ecc_support=on
hw_mem_test=on
patrol_scrub=on
demand_scrub=off
all_usb_device=on
usb_2.0_cont_mode=on
usb=fp_on,rtm_on,onboard_on
usb_boot=on
baudrate=38400
os_boot_watchdog=off,5,reset
frontnet_boot=on
basenet_boot=on
fabricnet_boot=on
artm_net_boot=off
artm_sas_boot=on
artm_fc_boot=off
ssc_support=on
number_lock=on
boot_order=frontnet0,frontnet1,basenet0,basenet1,efishell
smbios_event_log=on
```

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

```
whea=on
hyper_threading=on
```

```
limit_cpuid=off

ed_bit=on

hw_prefetcher=on

adj_cache_prefetch=on

l1_prefetch=on

datareuse_optmize=on

virtualization=on

turbo_mode=on

rt_err_log=off

pci_log=off

vt_d=off

int_map=on

ecc_support=on

hw_mem_test=on

patrol_scrub=on

demand_scrub=off

all_usb_device=on

usb_2.0_cont_mode=on

usb=fp_on,rtm_on,onboard_on

usb_boot=on

baudrate=38400

os_boot_watchdog=off,5,reset

frontnet_boot=on

basenet_boot=on
```

```
fabricnet_boot=on

artm_net_boot=off

artm_sas_boot=on

artm_fc_boot=off

ssc_support=on

number_lock=on

boot_order= basenet0,basenet1,frontnet0,frontnet1,efishell

smbios_event_log=on
```

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

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

Chapter 4

Firmware Upgrade Facility

4.1 Overview

The Firmware Upgrade Facility (FUF) provides a uniform way to upgrade firmware on Artesyn hub blades and node blades. It consists of a Firmware Upgrade Command-line Utility (FCU), flash device drivers, and specially prepared firmware recovery image files. On the ATCA-7368 FUF allows you to upgrade the following firmware types:

- BIOS firmware
- IPMC firmware
- MMC firmware on RTM
- FPGA image
- FRU data

4.2 Firmware Recovery Image Files

FCU supports specially prepared firmware recovery image (FRI) files as well as firmware images in the HPM.1 format. HPM.1 is a PICMG standard to upgrade IPMCs.

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

The following image files are currently supported.

Filename	Description
`atca-7368_em_bios_<version>.fri`	FRI format BIOS image for ATCA-7368. Only upgradable through Artesyn FUF.
`atca-7368_em_bios_<version>.hpm`	HPM.1 format BIOS image for ATCA-7368
`atca-7368_em_ipmc_<version>.hpm`	IPMC firmware for ATCA-7368. The "version" include the presentations of CPU type, with/without AMC slot and the firmware version. These image files were classified according to the CPU type and stored in different directories respectively.
`atca-7368_em_ibbl_<version>.hpm`	IPMC boot loader for ATCA-7368. The "version" include the presentations of CPU type, with/without AMC slot and the firmware version. These image files were classified according to the CPU type and stored in different directories respectively.
`atca-7368_em_frud_<version>.hpm`	FRU data for ATCA-7368.The "version" include the presentations of CPU type, with/without AMC slot and the firmware version. These image files were classified according to the CPU type and stored in different directories respectively.
`atca-7368_em_fpga_<version>.hpm`	FPGA image for ATCA-7368
`artm-7368_em_mmcf_<version>.hpm`	IPMC Firmware for RTM-ATCA-7368. Stored in directory artm-7368.
`artm-7368_em_mmcb_<version>.hpm`	IPMC boot loader for RTM-ATCA-7368. Stored in directory artm-7368.
`artm-7368_em_frud_<version>.hpm`	FRU data for RTM-ATCA-7368. Stored in directory artm-7368.
artm-7360_em_mmcb_<version>.hpm	MMC boot loader for RTM-ATCA-7360. Stored in directory artm-7360.
artm-7360_em_mmcf_<version>.hpm	MMC firmware for RTM-ATCA-7360. Stored in directory artm-7360.
artm-736xdd_em_mmcall_<version>.hpm	MMC firmware and boot loader for RTM-ATCA-736X-DD. Stored in directory artm-736xdd.

4.3 Backup Concept

The BIOS firmware for the ATCA-7368 is stored in redundant, persistent memory devices. This allows the firmware image in one bank to serve as a backup for the other bank. This is particularly useful for firmware upgrades.

During normal operation, the CPU on the ATCA-7368 determines which bank to boot from based on a chip select signal controlled by the IPMC. This bank is considered the active boot device. FCU will only allow you to upgrade an inactive device. FUF allows you to upgrade only the BIOS boot bank from which the blade has not booted. This means that you need to reboot the blade in case you want to upgrade both the banks of the BIOS flash. The BIOS firmware image can be programmed via the payload by using FUF or via IPMC.

The IPMC firmware consists of a boot loader as well as an active and a stand-by IPMI firmware. The boot loader maintains both the active and stand-by firmware in the flash memory of the ATCA-7368. Please note that the BootLoader Firmware is not installed in the BBS by default, as the BootLoader update must be performed on Artesyn request only.

Each time the IPMC firmware is upgraded, the most recent firmware version is kept in flash memory and the older firmware version is overwritten by the new one. Once the new IPMI firmware is programmed, the IPMC resets itself to boot from the new image. The boot loader validates the new IPMC firmware. Provided the IPMC can power up successfully the current image is made active and the previously active image is made backup. In case of power-up failures, the boot loader automatically recovers from crisis and boots from the previous image.

BIOS BootBlock shall verify BIOS integrity by checksum calculation. If the active BIOS is found corrupt a switch to the backup BIOS shall occur (initiated by BIOS BootBlock). The switch shall also occur in case the BootBlock code is missing or corrupt. This BIOS independent switch is triggered by IPMC watchdog logic.

The FPGA bank can be updated via FCU or IPMC. This means that a corrupt FPGA image can be restored using IPMC.

The following sample output displays the information regarding BIOS, IPMI, and FPGA. Depending on your setup, you may get a different output.

```
fcu -q
```

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

OPERATION : Query
```

```
RESULT    : SUCCESS
MESSAGE   : Device              : atca-7368-cpu
            Part number         : 0106866J03A
            Part revision       : REV.A
                BANK                : A
                Firmware Name       : AMI-BIOS
                Firmware Version    : 1.0.17 Build 0004
                Marked for next use : no
```

--> Installed BIOS Version on Bank0

```
               BANK                : B
                Firmware Name       : AMI-BIOS
                Firmware Version    : 1.0.16
                Marked for next use : yes
```

--> Installed BIOS Version on Bank1

```
OPERATION : Query
RESULT    : SUCCESS
MESSAGE   : Device              : atca-7368-hpm.1-ipmc
            Part number         : 0106866J03A
            Part revision       : REV.A
        BANK                : A - Operational
                Firmware Name       : H8S-AMCc F/W
                Firmware Version    : 2.0.15040000
```

--> Installed IPMI FW on Bank0

```
                BANK                : B - Rollback
                Firmware Name        : H8S-AMCc F/W

                Firmware Version     : 2.0.15040000
```

--> Installed IPMI FW on Bank1

```
                BANK                 : D - Operational

                Firmware Name        : H8S-AMCc B/L

                Firmware Version     : 2.0.15020000
```

--> Installed IPMI FW Boot Loader

```
                BANK                 : G - Operational

                  Firmware Name      : H8S-AMCc F/I

                Firmware Version     : 2.0.15040000
```

--> Internal IPMI Bank (FRU Info)

```
                BANK                 : J - Operational

                Firmware Name : H8S-AMCc F/C

                 Firmware Version : 0.0.00000000
```

--> Internal IPMI Bank (FRU Info Carrier) No subject to be updated

```
                BANK                 : M - Operational

                Firmware Name        : FPGA

                Firmware Version     : 16.0.00000000
```

--> Installed FPGA Version

```
                BANK                 : P - Operational

                 Firmware Name        : BIOS

                Firmware Version     : 1.0.17000000
```

-->Installed BIOS Version (seen by IPMC)

```
                BANK                : S - Operational
                  Firmware Name       : BIOS
                Firmware Version    : 0.0.00000000
```

--> Second BIOS Bank (currently not seen by IPMC)

```
OPERATION : Query
RESULT    : SUCCESS
MESSAGE   : Device              : artm-7368-hpm.1-ipmc
            Part number         : 0106867J03A
            Part revision       : REV.A
            IPMI address        : MMC=0x72


                BANK                : A - Operational
                Firmware Name       : AVR-AMCm F/W
                Firmware Version    : 2.0.01000000
```

--> Installed IPMI FW for ARTM on Bank1

```
               BANK                : B - Rollback
                Firmware Name       : AVR-AMCm F/W
                Firmware Version    : 2.0.01000000
```

--> Installed IPMI FW for ARTM on Bank2

```
                BANK                : D - Operational
                Firmware Name       : AVR-AMCm B/L
                Firmware Version    : 2.0.01000000
```

--> Installed IPMI Booter FW for ARTM

```
                BANK                : G - Operational
                Firmware Name       : AVR-AMCm F/I

                Firmware Version    : 2.0.01000000

--> Internal IPMI Bank (FRU Info)

                BANK                : H - Rollback

                Firmware Name       : AVR-AMCm F/I

                Firmware Version    : 2.0.01000000

--> Internal IPMI Bank (FRU Info)

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

4.4 fcu—Firmware Upgrade Command-Line Utility

Description

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

- Query the current versions of firmware installed on the ATCA-7368 and determine which firmware devices are active.
- Verify that a specified upgrade image is sound and compatible with the current hardware.
- Upgrade a firmware image and update the firmware version number string in FRU data accordingly, referring to the IIF file.
- Mark a device to be used as the boot source on the next reset.
- Show the version of a specified firmware image file and compare the version of a specified firmware image file with the version of an installed firmware image.

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

FCU works in conjunction with device drivers created specifically for the flash devices on Artesyn blades.

The FCU verify and upgrade operations require specially prepared FRI or HPM files (see *Firmware Recovery Image Files* on page 59).

FCU also relies on the Hardware Platform Management Agent daemon to interact with the local IPMC. Most commands will fail if the HPM Agent is not running. For information on configuring and running HPM Agent, see Chapter 5, *Hardware Platform Management,* on page 77.

Synopsis

```
fcu --help

fcu --version

fcu -q [-d <device-id>]

fcu -v -f <filename>

fcu -u -f <filename>
fcu -a -f <filename>

fcu -m -b <bank-letter> -d <device-id>
```

Parameters

`-a`
`--full-upgrade`

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

`-b <bank-letter>`
`--bank=<bank-letter>`

Specifies the bank to mark for next boot, where `<bank-letter>` is the letter designating a specific bank. For BIOS banks, possible values are A and B. This option is used with the mark operation. Use the query option `-q` to display available banks.

`-c`
`-compare`

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

`-d <device-id>`
`--device=<device-id>`

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

Device ID	Description
`atca-7368-cpu`	BIOS firmware image on ATCA-7368
`atca-7368-hpm.1-ipmc`	IPMC firmware and FPGA image on ATCA-7368
`artm-7368-hpm.1-ipmc`	RTM MMC firmware on RTM-ATCA-7368

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

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

```
--force
```

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

```
--help
```

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

This option needs a target destination `-t` argument added when working with the IPMC or ARTM.

```
-m
--mark
```

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

Currently, the mark operation only supports CPU firmware devices.

```
-q
--query
```

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

```
-s
show
```

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

`-u`
`--upgrade`

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

You must specify the to-be-upgraded firmware filename and a right corresponding IIF filename in the same time to run the upgrade. FCU will not check whether the IIF information really matches the firmware files or not.

`-v`
`--verify`

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

`--version`

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

Usage

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

- --compare --file=`<filename>`
- --full-upgrade --file=`<firmware-filename>`

A full-upgrade combination will perform verify, upgrade and mark actions together. The operation will automatically upgrade and specify the currently 'Rollback' bank as the bank to be used for next power-up.

- --help
- --mark --bank=`<bank-letter>` --device=`<device-id>`
- --query
- --query --device=`<device-id>`
- --show --file=`<filename>`
- --upgrade --file=`<filename>`
- --upgrade --mark --file=`<filename>`
- --upgrade --file=`<filename>`
- --verify --file=`<filename>`
- --verify --upgrade --file=`<filename>`
- --verify --upgrade --mark --file=`<filename>`
- --version

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

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

- **`fcu –vf /opt/bladeservices/rom/<filename>`**
- **`fcu –q –d <device-id>`**
- **`fcu –q –d <device-id>`**
- **`fcu –mb a –d <device-id>`**

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

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

4.5 Upgrading a Firmware Image

This section describes recommended procedures for upgrading firmware devices. The procedures for upgrading BIOS and IPMC differ slightly.

NOTICE

The upgrade fails if the following is not taken into consideration:
Upgrade only one bank at a time, then reboot and verify the upgrade using the query option. If the upgrade fails and both banks become corrupted for any reason, the ATCA-7368 will be rendered inoperable.
To prepare the ATCA-7368 for a BIOS upgrade, the on-board DIP-switches of the ATCA-7368 must be set such that all BIOS flashes are writable. This is the default configuration. Refer to the ATCA-7368 Installation and Use guide for further details about DIP switch settings.

4.5.1 BIOS Upgrade

The BIOS can only be upgraded from the ATCA-7368 on which the BIOS is running. You have to upgrade the BIOS by using `fcu`.

Upgrading the BIOS Firmware

Follow these steps to upgrade the BIOS. The shown file names and paths are only meant as an example and should be replaced with file names and paths applicable to your configuration.

1. Query the current BIOS firmware images on the blade.
   ```
   fcu -qd atca-7368-cpu
   ```

2. Show the version of the new BIOS file (to verify that it has actually a newer version than the already installed BIOS)
   ```
   fcu --show -f /opt/bladeservices/rom/atca-7368_em-
   bios_<verision>.fri
   ```
   or
   ```
   fcu --show -f /opt/bladeservices/rom/atca-7368_em-
   bios_<verision>.hpm
   ```

3. Upgrade the firmware image:
   ```
   fcu --upgrade -f /opt/bladeservices/rom/atca-7368_em-
   bios_<verision>.fri
   ```
   or
   ```
   fcu --upgrade -f /opt/bladeservices/rom/atca-7368_em-
   bios_<verision>.hpm
   ```

   FCU writes the new image and then reads back the image and performs a binary compare to ensure that the write was successful. If the upgrade was not successful, you will see an error message. Try the upgrade again. If it is still not successful, contact your Artesyn representative.

4. Query the new image to ensure that the version information is correct,
   ```
   fcu -qd atca-7368-cpu
   ```

5. Mark the new image as active so that it will be used for the next boot, for example:

```
fcu --mark -b <bank-letter> -d atca-7368-cpu
```

where `<bank-letter>` is the letter of the upgraded bank, for example: a

ATCA-7368 payload should be power-cycled after a BIOS upgrading to make the updated BIOS active. Note that the installed AMC and RTM will also be power-cycled automatically when the ATCA-7368 payload is power-cycled.

4.5.2 IPMC/MMC Firmware, Bootloader and FRU Data Upgrade

Upgrading the IPMC/MMC Firmware

Follow these steps to upgrade an IPMC/MMC. The shown file names and paths are only meant as an example and should be replaced with file names and paths applicable to your configuration.

The general procedure to upgrade the MMC image of an RTM is the same, except that you need to use `artm-7368-hpm.1-ipmc` as device ID.

1. Query the current IPMC firmware images on the blade.

   ```
   fcu -q -d atca-7368-hpm.1-ipmc
   ```

2. Show the version of the new IPMC file (to verify that it has actually a newer version than the already installed image):

   ```
   fcu --show -f /opt/bladeservices/rom/<board type>/atca-7368_em_ipmc_<version>.hpm
   ```

3. Upgrade the firmware image:

   ```
   fcu --upgrade -f /opt/bladeservices/rom/<board type>/atca-7368_em_ipmc_<version>.hpm
   ```

   Once the new IPMI firmware is programmed, the IPMC resets itself to boot from the new image. The boot loader validates the new IPMC firmware. Provided the IPMC can power up successfully the current image is made active and the previously active image is made backup. In case of power-up failures, the boot loader automatically recovers from crisis and boots from the previous image.

4. Query the new image to ensure that the version information is correct:
   ```
   fcu -qd atca-7368-hpm.1-ipmc
   ```

If the version you just installed is now the active image, the upgrade was successful.

4.5.3 FPGA Upgrade

Upgrading the FPGA Firmware

The ATCA-7368 uses an EEPROM which contains the FPGA firmware.

The following procedure describes how to upgrade the FPGA image stored in the user-programmable EEPROM. The shown file names and paths are only meant as an example and should be replaced with file names and paths applicable to your configuration.

1. Query the current FPGA firmware images on the blade.
   ```
   fcu -q -d atca-7368-hpm.1-ipmc
   ```

The FCU tool reads the FPGA firmware version directly from the FPGA and not from the EEPROM. Therefore, you need to perform a blade power cycle before you can see the version of a newly installed FPGA firmware.

2. Show the version of the new FPGA file (to verify that it has actually a newer version than the already installed image):
   ```
   fcu --show -f /opt/bladeservices/rom/atca-7368_em_fpga_<version>.hpm
   ```

3. Upgrade the firmware image.
   There are two options to do this: via the `--full-upgrade` option and via the `--upgrade` option. Both the options are equivalent.
   ```
   fcu --full-upgrade -f /opt/bladeservices/rom/atca-7368_em_fpga_<version>.hpm
   ```
   or
   ```
   fcu --upgrade -f /opt/bladeservices/rom/atca-7368_em_fpga_<version>.hpm
   ```

This upgrades the user-programmable FPGA EEPROM with the specified FPGA image file.

4. Power-cycle the blade.

If the blade fails to start up after an FPGA upgrade, you need to reload the FPGA EEPROM via IPMI. Refer to the *ATCA-7368 Installation and Use Guide* for further details about the system boot option functionality as supported by the ATCA-7368.

Chapter 5

Hardware Platform Management

5.1 Overview

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

The HPM package consists of:

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

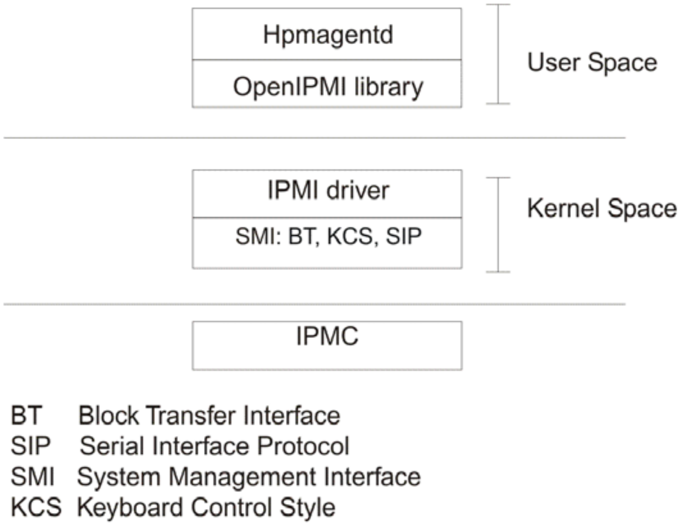
The `hpmcmd` sends a given HPM command to the `hpmagentd` and displays the received response on the console. The `hpmagentd` executes the incoming HPM commands and returns the result to a `hpmcmd` client.

HPM commands include:

- Retrieving and modifying FRU data
- Reading and controlling status of IPMI-controlled LEDs
- Executing shutdown and reboot scripts in response to cold reset or graceful reboot requests
- Communicating local slot location information

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

Figure 5-1 Software Levels of the HPM Architecture


Hpmagentd
OpenIPMI library
User Space
IPMI driver
SMI: BT, KCS, SIP
Kernel Space
IPMC
BT Block Transfer Interface
SIP Serial Interface Protocol
SMI System Management Interface
KCS Keyboard Control Style

The SMI (System Management Interface) driver provides the low level interface for communicating to IPMC with a KCS driver.

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

5.2 hpmagentd—HPM Agent Daemon

Description

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

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

This daemon has an init script called hpm that will start the daemon in run level 2 with the default settings.

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

Synopsis

```
hpmagentd [-l log-level] [-r reboot-script] [-s shutdown-script]

hpmagentd {-i | -u | -h | -v}
```

Parameters

`-l log-level`

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

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

`-r reboot-script`

Specifies a graceful reboot script that will be called when a blade graceful reboot request is received by the IPMC, where reboot-script is the complete path and filename of the target script. The default is `/opt/bladeservices/bin/hpmreboot` (see *hpm—Shutdown and Reboot Scripts* on page 81).

`-s shutdown-script`

Specifies a shutdown script that will be called when a blade shutdown request is received by the IPMC, where shutdown-script is the complete path and filename of the target script. The default is `/opt/bladeservices/bin/hpmshutdown` (see *hpm—Shutdown and Reboot Scripts* on page 81).

`-i`

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

`-u | -h`

Displays a brief message about command usage.

`-v`

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

5.3 hpm—Start-Up Script

Description

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

Synopsis

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

Parameters

`start`

Starts the hpm agent daemon.

`stop`

Terminates the hpm agent daemon.

`restart`

Terminates and then starts the hpm agent daemon.

`force-reload`

Terminates and then starts the hpm agent daemon.

5.4 hpm—Shutdown and Reboot Scripts

Description

At any time during normal operation, a shelf manager may issue a shutdown (FRU Activation Deactivate) or graceful reboot (FRU Control Reboot) request to the IPMC on a given blade. The IPMC then forwards this information to the HPM agent. The HPM agent listens for such requests from the IPMC. When it receives a request, it calls the respective script to run the reboot or shutdown sequence. In case of a shutdown indication, all running processes should be notified about the shutdown. In case of a reboot notification, the payload is responsible for invoking the reboot procedure. The IPMC is not involved in this process. This allows processes currently running on the blade to prepare for shutdown. After the notification, it takes roughly 30 seconds before the payload is powered off.

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

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

Synopsis

```
hpmshutdown
```

```
hpmreboot
```

5.5 hpmcmd—HPM Command Utility

Description

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

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

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

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

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

From there you can start executing commands.

Synopsis

```
hpmcmd [-p new-prompt] [-o output] [-i input | -c command]

hpmcmd [--prompt new_prompt] [--output_file output] [--input_file

 input | -cmd_line command]
```

Parameters

`-p new-prompt`

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

`-i input-file`

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

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

`-o output-file`

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

`-c command`

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

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

5.5.1 Command Overview

The following table lists all commands from the hpmcmd program available on the ATCA-7368. You can display this list and a short command description using the help command (see section *help* on page 98). A detailed description of the commands is given in section *Supported Commands* on page 86.

Table 5-1 Command Overview

Command	Description
bootbankget	Gets the bootbank to boot from
bootbankset	Sets the bootbank to boot from
bootparamerase	Erase boot parameter value
bootparamget	Get boot parameter value
bootparamset	Set a boot parameter value
bye	Exit the hpmCmd program
chinfo	Retrieve channel info
cmd	Execute any IPMI command
deviceid	Gets the Device Id.
exit	Exit the hpmcmd program
frudata	Allows to get FRU info in hex numbers
fruinfoget	Gets string fields from the FRU
fruinv	Allows to get the FRU size and addressable units
fruread	Allows to read x number of bytes from the FRU
fruwrite	Allows to write x number of bytes from the FRU
help	List of hpmcmd commands.
ipmbaddress	Shows the local board IPMB address
ipmcdevice	Shows the payload interface to the IPMC
ipmcstatus	Gets the IPMC Status
ledget	Gets the state of a specific FRU LED
ledprop	Get the LED properties for this FRU.
ledset	Controls the state of a specific FRU LED

Table 5-1 Command Overview (continued)

Command	Description
loglevelget	Gets the hpmagentd log level
loglevelset	Sets the hpmagentd log level(0-7)
macaddress	Lists the MAC addresses
motshelftype	Gets the Artesyn Shelf Type from the Shelf FRU (Board Product Name)
partnumber	Gets the board part number
physlotnumber	Gets the board physical slot number
portget	Shows the current state E-Key governed intfs
portset	Enables/Disables ports in a channel
quit	Exit the hpmcmd program
rebootpath	Gets hpmagentd reboot script path
sdr	Shows the SDR records
sdr_dump	Shows the SDR records in raw format
sdrinfo	Shows SDR information
sendcmd	Sends an IPMI command
shelfaddress	Gets the Shelf Address String
shelfslots	Gets number of slots in the shelf
shutdownpath	Gets hpmagentd shutdown script path
slotmap	Prints the slotmap of the shelf
slotnumber	Shows the board logical slot number
solcfgget	Get SOL configuration parameter
solcfgset	Set SOL configuration parameter
upgrade	Allows to upgrade the IPMC firmware
version	Shows the hpmCmd version and the hpmagentd version
watchdog	Control Payload WDT functionality

5.5.2 Supported Commands

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

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

5.5.2.1 bye

Description

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

Synopsis

```
bye
```

5.5.2.2 bootbankget

Description

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

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

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

Synopsis

```
bootbankget <payload_cpu_selector>
```

Parameters

`payload_cpu_selector`

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

Example

```
hpmcmd -c bootbankget 0
```

5.5.2.3 bootbankset

Description

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

Synopsis

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

Parameters

`payload_cpu_selector`

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

`newBootBank`

Can be set to BANK0 or BANK1

Example

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

5.5.2.4 bootparamerase

Description

This command allows you to erase data which is stored in the IPMC boot parameters storage area. The data which is stored in this area can be accessed from the IPMI subsystem and also from the OS and boot firmware. The storage area can for example be used in order to pass boot parameters to the boot firmware. For further details, refer to the *ATCA-7368 Installation and Use Guide*.

Synopsis

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

Parameters

`section`

Section within the IPMC storage area in which data is to be erased. Possible values are:

- USER
- DEFAULT
- TEST
- OS_PARAM

`name`

Name of the parameter which is to be erased

`-t`

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

Example

```
hpmcmd -c bootparamerase USER boot_order
```

```
Successful bootparamerase Operation
```

5.5.2.5 bootparamget

Description

This command allows you to read data which is stored in the IPMC boot parameters storage area. The data which is stored in this area can be accessed from the IPMI subsystem and also from the OS and boot firmware. The storage area can for example be used in order to pass boot parameters to the boot firmware. For further details, refer to the *ATCA-7368 Installation and Use Guide*.

Synopsis

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

Parameters

`section`

Section within the IPMC storage area from which data is to be read. Possible values are:

- USER
- DEFAULT
- TEST
- OS_PARAM

`name`

Name of the parameter whose value is to be read

`-t`

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

Example

```
hpmcmd -c bootparamget USER boot_order

boot_order = sashdd,sata3,sata1,basenet0,basenet1
```

5.5.2.6 bootparamset

Description

This command allows you to write data to the IPMC boot parameters storage area. The data which is stored in this area can be accessed from the IPMI subsystem and also from the OS and boot firmware. The storage area can for example be used in order to pass boot parameters to the boot firmware. For further details, refer to the *ATCA-7368 Installation and Use Guide*.

Synopsis

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

Parameters

`section`

Section within the IPMC storage area where to write the data to. Possible values are:

- USER
- DEFAULT
- TEST
- OS_PARAM

`name`

Name of the parameter which is to be set

`value`

Value of the parameter

`-t`

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

Example

```
hpmcmd -c bootparamset USER
bootboot_order=sashdd,sata3,sata1,basenet0,basenet1,usbonboard
```

```
Successful bootparamset Operation
```

5.5.2.7 chinfo

Description

Retrieve channel information

Synopsis

```
chinfo channel [-t ipmbAddr[:mmcAddr]]
```

Parameters

`channel`

Channel number

`-t`

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

Example

`hpmcmd -c chinfo 0`

```
root@ACPI4-C-9:~#  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

root@ACPI4-C-9:~# hpmcmd -c chinfo 4
Channel Medium Type   : System Interface (KCS, SMIC, or BT)
Channel Protocol Type : KCS
Session Support       : session-less
Active Session Count  : 0
Protocol Vendor ID    : 001BF2

root@ACPI4-C-9:~# hpmcmd -c chinfo 1
Channel Medium Type   : Asynch. Serial/Modem (RS-232)
Channel Protocol Type : TMode

Session Support       : session-less

Active Session Count  : 0

Protocol Vendor ID    : 00400AProtocol

Vendor ID             : 00400A
```

5.5.2.8 cmd

Description

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

Synopsis

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

Parameters

`ipmi address`

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

`netfn cmd`

Identifies the command type.

`cmd data`

Specifies the message data associated with the command.

Example

GetDeviceId command to the local IPMC:

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

GetDeviceId command to the remote IPMC on address 9a:

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

GetDeviceId command to the remote IPMC on address 7a:

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

5.5.2.9 deviceid

Description

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

Synopsis

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

Parameters

`-t`

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

Example

```
hpmcmd -c deviceid
```

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

5.5.2.10 exit

Description

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

Synopsis

```
exit
```

5.5.2.11 frudata

Description

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

Synopsis

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

Parameters

`fruid`

Is 0 for the main blade and 1 for the rear transition module.

`-t`

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

Example

```
hpmcmd -c frudata 0
```

5.5.2.12 fruinfoget

Description

This command retrieves information from the specified FRU.

Synopsis

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

Parameters

`fruid`

Is `0` for the main blade and `1` for the rear transition module.

`field`

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

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

`-v`

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

`-t`

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

Example

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

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

```
hpmcmd -c fruinfoget 0
```

5.5.2.13 fruinv

Description

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

Synopsis

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

Parameters

`fruid`

Is 0 for the main blade and 1 for the rear transition module (if supported).

`-t`

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

Example

```
hpmcmd -c fruinv 0
```

5.5.2.14 fruread

Description

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

Synopsis

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

Parameters

`fruid`

Is 0 for the main blade and 1 for the rear transition module (if supported).

`startAddress`

Is the starting address in decimal.

`nbytes`

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

Example

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

5.5.2.15 fruwrite

Description

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

Synopsis

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

Parameters

`fruid`

Is 0 for the main blade.

`startAddress`

Starting address in decimal numbers.

`nBytes`

is the number of bytes to write in decimal. nBytes cannot exceed16 because of IPMI message size limitations.

5.5.2.16 help

Description

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

Synopsis

```
help
```

5.5.2.17 ipmbaddress

Description

This command retrieves the blade IPMB address.

Synopsis

```
ipmbaddress
```

5.5.2.18 ipmcdevice

Description

This command retrieves the payload tty device.

Synopsis

```
ipmcdevice
```

5.5.2.19 ipmcstatus

Description

This command retrieves the IPMC operating mode, payload control and outstanding events.

Synopsis

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

Parameters

`-v`

Verbose mode to get additional information operation

Example

```
hpmcmd -c ipmcstatus -v
```

5.5.2.20 ledget

Description

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

Synopsis

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

Parameters

`fruid`

Is 0 for the main blade and 1 for the rear transition module (if supported).

`led`

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

`-t`

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

Example

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

5.5.2.21 ledprop

Description

This command displays the FRU LED properties under IPMC control.

Synopsis

```
ledprop <fruid>
```

Parameters

`fruid`

0 for the main board and 1 for the RTM.

Example

```
hpmcmd -c ledprop 0

FRU LEDs under IPMC control:

LED0 = BLUE

LED1 = RED or AMBER

LED2 = GREEN
```

5.5.2.22 ledset

Description

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

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

Synopsis

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

Parameters

`fruid`

Is 0 for the main blade and 1 for the rear transition module (if supported).

`led`

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

ON = enable override state and turn LED on.
OFF = enable override state and turn LED off.
BLINK = enable override state and blink LED; off_duration and on_duration specify the blink duration; the default on and off duration is 300 ms.
LOCAL = cancel override state and restore LED control to the IPMC, that is, local state.
TEST = run lamp test for specified on_duration, then restore prior state.

`offms`

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

`onms`

Only valid if operation is BLINK or TEST:
If operation is BLINK, 10–2500 in 10-millisecond increments
If operation is TEST, 100-12800 in 100-millisecond increments

`color`

LED0 = BLUE
LED1 = RED
LED2 = GREEN
LED3 = AMBER

`-t ipmbAddr`

Sends the command to ipmbAddr.

Example

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

5.5.2.23 loglevelget

Description

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

Synopsis

```
loglevelget
```

Example

```
hpmcmd -c loglevelget
```

```
Loglevel 5 (NOTICE)
```

5.5.2.24 loglevelset

Description

This command sets the level of message logging for hpmagentd.

Synopsis

```
loglevelset <newLogLevel>
```

Parameters

`newLogLevel`

Is one of the standard syslog levels:

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

Example

```
hpmcmd -c loglevelset 7
```

5.5.2.25 macaddress

Description

This command retrieves a list of available MAC addresses.

Synopsis

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

Parameters

`-t ipmbAddr`

Sends the command to ipmbAddr.

Example

```
hpmcmd -c macaddress
BASE Interface Channel 0 : 00:0E:0C:85:E9:91

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

5.5.2.26 motshelftype

Description

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

Synopsis

```
motshelftype
```

Example

```
hpmcmd -c motshelftype
CHS1406
```

5.5.2.27 partnumber

Description

This command retrieves the part number of the main blade.

Synopsis

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

Parameters

```
-t ipmbAddr
```

Sends the command to ipmbAddr.

Example

```
hpmcmd -c partnumber
```

5.5.2.28 physlotnumber

Description

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

Synopsis

```
physlotnumber
```

5.5.2.29 portget

Description

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

Synopsis

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

Parameters

`interface`

Valid values are:

BASE | FABRIC | UPDATE

`channel`

an integer in the following range:
1–16 for Base
1–15 for Fabric
1 for Update
The value of channel must be valid for the blade. For example, node blades have only 2 channels for the base interface; using a value of 4 will return an error.

`-t ipmbAddr`

Sends the command to ipmbAddr.

Example

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

5.5.2.30 portset

Description

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

Synopsis

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

Parameters

`intf`

Valid values are:
BASE | FABRIC | UPDATE

`chan`

an integer in the following range:
1–16 for Base
1–15 for Fabric
1 for Update
The value of channel must be valid for the blade. For example, node blades have only 2 channels for the base interface; using a value of 4 will return an error.

`grpid`

Always 0 according to current shelf FRU information

`type`

Valid values are:

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

`typeX`

Always 0 in current implementation. Valid values are:
0 (for 1000Base-BX)
1 (for 10GBase-BX4)
2 (for FC-PI)

`ports`

A sequence of ports to act on.
For base and update channels, port is always 0.
For fabric channels, port can specify up to 4 ports as specified in PICMG 3.1:
Option 1: 0
Option 2: 01
Option 9: 0123

`oper`

Valid values are DISABLE or ENABLE.

Example

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

5.5.2.31 quit

Description

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

Synopsis

```
quit
```

5.5.2.32 rebootpath

Description

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

Synopsis

```
rebootpath
```

Example

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

5.5.2.33 sdr

Description

This command shows the SDR records.

Synopsis

```
sdr
```

Example

```
hpmcmd -c sdr

recID 1: management controller device locator record

    I2C slave addr:   49

    Channel number:   00

    Power state:      06

    Global init:      0C

    Capabilities:     2D
```

```
        Entity Id:       PICMG front board
        Entity instance: 60
        OEM:             00
        Id string:       AC4 MC Locator

    recID 2: full sensor record
        owner is IPMB 92 sensor num 00 on lun 00 channel 00
        logical entity: PICMG front board - instance 60
        AC4 HS Carrier : FRU hot swap : sensor-specific discrete
```

5.5.2.34 sdr_dump

Description

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

Synopsis

```
sdr_dump
```

Example

`hpmcmd -c sdr_dump`

```
SDR Records:
    01 00 51 12 19 92 00 cc 2d 00 00 00 a0 60 00 ce  "..Q....ì-...?`.?"
    41 43 34 20 4d 43 20 4c 6f 63 61 74 6f 72        "AC4 MC Locator"
```

5.5.2.35 sendcmd

Description

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

Synopsis

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

Parameters

`IPMBaddress`

Destination IPMB address in hex digits.

`netfn`

IPMI request net function in hex digits.

`cmd`

IPMI request command in hex digits

`data0 ... dataN`

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

Example

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

5.5.2.36 sdrinfo

Description

This command shows the SDR information.

Synopsis

```
sdrinfo
```

Example

```
hpmcmd -c sdrinfo
SDR Information:
```

```
LUN 0 has 054 sensors; static sensor population

LUN 1 has 000 sensors

LUN 2 has 000 sensors

LUN 3 has 000 sensors
```

The LUN sensor number mentioned above are in decimal format.

5.5.2.37 shelfaddress

Description

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

Synopsis

```
shelfaddress
```

Example

```
hpmcmd -c shelfaddress
01
```

5.5.2.38 shelfslots

Description

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

Synopsis

```
shelfslots
```

Example

```
hpmcmd -c shelfslots
```

```
14 slots      //e.g. in a Centellis 4440 System
```

5.5.2.39 shutdownpath

Description

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

Synopsis

```
shutdownpath
```

Example

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

5.5.2.40 slotmap

Description

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

Synopsis

```
slotmap
```

Example

```
hpmcmd -c slotmap

Physical Slot : 01 02 03 04  . 05 06 07 08  . 09 10 11 12  . 13 14

Logical  Slot : 13 11 09 07  . 05 01 03 04  . 02 06 08 10  . 12 14

IPMB  Address : 9A 96 92 8E  . 8A 82 86 88  . 84 8C 90 94  . 98 9C
```

5.5.2.41 slotnumber

Description

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

Synopsis

```
slotnumber
```

Example

```
hpmcmd -c slotnumber
9
```

5.5.2.42 solcfgget

Description

Retrieves the current serial over LAN (SOL) configuration. SOL is a feature which allows you to redirect the serial console of the blade via an IPMI session over the network. Refer to the blade's hardware user manual for further details.

Synopsis

```
solcfgget channel [param] [-t ipmbAddr[:mmcAddr]]
```

Parameters

`channel`

Channel number

`param`

The configuration parameter whose value you want to retrieve. Possible values are:

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

`-t`

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

5.5.2.43 solcfgset

Description

Sets a serial over LAN (SOL) configuration parameter. SOL is a feature which allows you to redirect the serial console of the blade via an IPMI session over the network. Refer to the blade's hardware user manual for further details.

Synopsis

```
solcfgset channel param value [-t ipmbAddr[:mmcAddr]]
```

Parameters

`channel`

Channel number

`param`

The configuration parameter whose value you want to retrieve. Possible values are:

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

`value`

The value which you want to set

`-t`

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

5.5.2.44 upgrade

Description

This command is used to upgrade the IPMC firmware.

It is only possible to upgrade the firmware remotely from one blade to another, not from the blade itself. In case of an RTM upgrade the front blade will be powered down.

Synopsis

```
upgrade <image> -f <filepath>
```

Parameters

`image`

Full path of the upgrade image file

`-f filepath`

Full path of the upgrade image file. This operation will make the current image the backup one.

5.5.2.45 version

Description

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

Synopsis

```
version
```

Example

```
hpmcmd -c version
hpmagentcmd version bbs 1.3.12 build 2.pne30
```

5.5.2.46 watchdog

Description

This command is used handle the payload BMC watchdog.

Synopsis

```
watchdog set <tmr_use> <tmr_action> <pre_timeout> <flags> <lsb_val>
<msb_val>
watchdog set default

watchdog get
watchdog start
watchdog stop
watchdog reset
```

Parameters

`set`

Possible values are

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

HPI-B Software

6.1 Overview

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

This BBS release contains an HPI-B library package. For more information on Artesyn Embedded Technologies’s HPI-B implementation, refer to the *System Management Interface Based on HPI-B User’s Guide*.

Board Control Module

7.1 Overview

Board control is a kernel module which provides access to the board FPGA. The board control module creates a boardinfo directory and a file of `BSP_VERSION_INFO` in the `/proc` file system that contains general information on the ATCA-7368. The following table describes the information in boardinfo directory.

File	Description	Sample output
`/proc/BSP_VERSION_INFO`	Shows the BSP version	`02_02_0000`
`board_name`	Shows the board name, as provided by the BIOS.	`PCA,ATCA-7368/0GB/6E`
`board_version`	Shows the board version, as provided by the BIOS.	`0106866J03A`

File	Description	Sample output
`bios_version`	Shows the BIOS version.	`1.0.10`
`board_serial number`	Shows the serial number of the board, as provided by the BIOS.	`To be filled by O.E.M.`

`fpga`	Shows additional FPGA information.	`FPGA version: 0x10`
		`. . .`

`summary`	Shows a summary of the board state (FPGA registers) and BIOS provided information.	`Board Vendor:          Emerson`
		`Board Name:            PCA,ATCA-7368/0GB/6E`
		`Board Version:         0106866J03A`
		`Board Serial Number: To be filled by O.E.M.`
		`BIOS Vendor:           Emerson`
		`BIOS Version:          1.0.10`
		`BIOS Release Date:     01/07/2011`
		`Last Reset Source: PowerOn Reserved`
		
		`Memory Module:`
		`  Device/Bank: DIMM_Socket_P05/DIMM_CPU#0`
		`  Size:                 4096 Mbyte`
		`  Data Width:           64 Bit`
		`  Manufacturer:         Samsung`
		`IPMI`
		`  Interface Type:      1 KCS (Keyboard Control Style)`
		`  IPMI Spec Rev:        2.0`
		`  I2C Slave Addr:       0x9A`
		`  NV Stor.Dev.Addr:     Not Present`
		`  Base Addr:            0x00000CA3`
		`  IRQ:                  0x0`

7.2 Board Control Tool

The board control module provides an IOCTL interface which can be used by the userland applications. The following sections describes userland applications, such as LEDCTRL and FPGA_TEST.

7.2.1 LEDCTRL

Description

Allows to control the 3 front panel LEDs, according to their capabilities.

LEDCTRL can be found at `/opt/bladeservices/bin/ledctrl`.

Synopsis

```
ledctl [options] [led1] [led2] ...
```

Here, `led<n>` are zero-based LED numbers. If no LED numbers are given, the option is applied to all the available LEDs.

The options can have following values.

Option	Description
`-n`	Print number of available LEDs.
`-i`	Display information about LED capabilities.
`-s`	Print current LED settings.
`-c <color>`	Set LED(s) <color> to: g[reen], y[ellow], r[ed], b[lue], a[mber], hdd, or eth.
`-b <freq>`	Set blink frequency to: off or p[ermanent].

7.2.2 FPGA_TEST

Description

Dumps the FPGA register set.

FPGA_TEST can be found at `/opt/bladeservices/bin/fpga_test`.

Synopsis

```
fpga_test -d
```

Here, -d option is used to dump the complete FPGA register set.

Kernel and Root File System Config using PNE 4.0

8.1 Building Kernel and Root File System

This section provides an introduction for building the Linux kernel and root file system for the ATCA-7368 with Wind River PNE 4.0.

8.1.1 Prerequisites

The local RedHat Linux system should have Wind River PNE 4.0 installed.

The latest release of LSP is delivered and verified with the following Wind River 4.0 GA release packages.

Item	DVD Names
PNE4.0 – Base Package	DVD-R159194.1-1-00 DVD-R159195.1-1-00

8.1.2 Additional Kernel Patches

Table 8-1, describes the additional kernel packages required for ATCA-7368. These patches are stored in `atca7368_em_bsp/templates/board/atca7368/linux/` and are applied automatically during project configuration.

Table 8-1 ATCA-7368 specific kernel patches

Patch name	Description
`i8042.patch`	Suppress invalid error message.
`i7core_edac.patch`	Add EDAC driver for Westmere.
`dma_ioat.patch`	Add DMA IOAT support.
`coretemp.patch`	Uses correct tjmax temperature.

Implicit source NAT change warning message is not needed.

8.1.3 Project Setup

To setup the project, extract the ATCA-7368 BSP package on your build machine. You need access to Wind River PNE4.0.

```
tar -xzvf /atca7368_em_bsp_02_02_0000.tgz
```

You can configure the ATCA-7368 platform project using either of the following:

- Project configure script
- Wind River work bench for PNE 4.0

The ATCA-7368 BSP package is stored in the `atca7368_em_bsp` folder. The BSP consists of the following sub-folders and files:

- `atca7368_em_bsp/bbs/` - It contains the ATCA-7368 setup and configuration scripts, along with the pre-build BBS packages.
- `atca7368_em_bsp/dist` and `atca7368_em_bsp/packages` - It contains additional/modified tools/services or specific patches.
- `atca7368_em_bsp/packages` - It contains the firmware packages in eSW format used by FUF to make the firmware upgrading.
- `atca7368_em_bsp/templates/` - It contains the board specific configuration files. It consists of the following files, along with other sub-folders and files:
  - `atca7368_em_bsp/templates/board/atca7368/linux/atca7368.scc`- The kernel configuration file.
  - `atca7368_em_bsp/templates/board/atca7368/pkglist.add` and `atca7368_em_bsp/templates/board/atca7368/pkglist.remove` - RFS package list files.

8.1.3.1 Project Configure Script

You can use the following script to configure the project,

```
atca7368_em_bsp/bbs/scripts/setEnv.sh.
```

Parameters MUST be specified for this script:

```
setEnv <ATCA7368_OBJ_DIR> <PROJECT_PATH> <PROJECT_LAYER_PATH>
<WINDRIVER_INSTALL_PATH>
```

ACPI4_C_OBJ_DIR: Directory where your BBS application making modules object files and eSW will be put.

PROJECT_PATH: Directory of your PNE platform project. In this case:
`/home/ec7538/atca7368/ga01`

PROJECT_LAYER_PATH: Directory where you have installed `atca7368_em_bsp`. In this case:
`/home/ec7538/atca7368/ga01/cml/atca7368_em_bsp`

WINDRIVER_INSTALL_PATH: Directory where the WindRiver PNE 4.0 workbench installed and the WindRiver PNE 4.0 `wrenv.sh` file can be found. In this case:
`/opt/windriver/PNE4.0/`

To use this script you should create a new project directory, in this case:
`/home/ec7538/atca7368/ga01/prj`.

After successfully set your building environment using `setEnv.sh`, now you can run another script of `atca7368_setup.sh` to configure your ATCA-7368 PNE 4.0 workspace against the configuration in the `atca7368_em_bsp` and build the project (including kernel and root file system).

8.1.4 Kernel Configuration

The ATCA-7368 uses the 64-bit kernel configuration. The kernel configuration file, is stored as `atca7368_em_bsp/templates/board/atca7368/linux/atca7368.scc` and a file of `atca7368.cfg` is referred by the file of `atca7368.scc`. You can directly modify `atca7368.cfg` to tune your own custom-built kernel. The kernel configuration will be applied automatically, if you are using the layer structure.

8.1.5 Root File System Configuration

The file system provided with ATCA-7368 supports the multilibs feature of Wind River PNE 4.0, that allows running of both 32-bit and 64-bit applications on the board. As a result, the package-list contains 32-bit packages also. You can adapt the packages used in your projects as per the requirement. The ATCA-7368 `pkglist.add` and `pkglist.remove` files are stored in `atca7368_em_bsp/templates/board/atca7368/` directory.

The original PNE 4.0 does not include a tftp server. Therefore an open source tftp server is incorporated (see **`atca7368_bsp_em/dist/tftp/`**).

The onboard Intel 82576/82599 ethernet controller driver igb/ixgbe is built as an loadable Linux module by the script of `atca7368_em_bsp/bbs/scripts/atca7368_make_bbs.sh` (instead of building from PNE kernel).

Check the source code and adapted Makefiles at `atca7368_em_bsp/bbs/src/igb.tgz` and `atca7368_em_bsp/bbs/src/ixgbe.tgz`.

8.1.6 Making BBS modules

BBS modules delivered to you in source code format and igb/ixgbe driver source code are available here `atca7368_em_bsp/bbs/src`. You may need to modify the script of `atca7368_em_bsp/bbs/scripts/atca7368_make_bbs.sh` (delete those modules you do/could not need to be built from source or adapt the source code repository directory) according to your building environment and project requirement.

Please run the script of `atca7368_em_bsp/bbs/scripts/atca7368_make_bbs.sh` to build your BBS modules.

8.1.7 Getting Root File System and RAMDISK Image

The root file system and RAMDSIK (including BBS packages) is created after successfully building kernel and root file system followed by post configuration of the root file system, such as create admin-user, change ownership for ntp-scripts, create additional links to rc-scripts, add additional device nodes, and remove locals to save space in RAMDISK. The post configuration tasks are performed by `atca7368_make_image.sh` script stored in `atca7368_em_bsp/bbs/scripts` directory. Before executing the `atca7368_make_image.sh` script, update `PROJECT_PATH` and `PROJECT_LAYER_PATH` as per the location of `atca7368_em_bsp`. For example:

```
PROJECT_PATH=/home/ec7538/atca7368/ga01/prj/
PROJECT_LAYER_PATH=/home/ec7538/atca7368/ga01/cml/atca7368_em_bsp
```

A sample output of the `atca7368_make_image.sh` script is:

```
bash-3.2$ sudo ./atca7368_make_image.sh

--- Project Path:        /home/ec7538/atca7368/ga01/prj

--- Project layer path:  /home/ec7538/atca7368/ga01/cml/atca7368_em_bsp

--- Path to Result files:/home/ec7538/atca7368/ga01/prj

--- Path to Patch files:
/home/ec7538/atca7368/ga01/cml/atca7368_em_bsp/bbs

--- Path to Initramfs:
/home/ec7538/atca7368/ga01/cml/atca7368_em_bsp/bbs/initramfs



--- results from build process:

Symbol Files:   /home/ec7538/atca7368/ga01/prj/export/atca7368-vmlinux-
symbols-WR4.0.0.0_cgl

System Map:     /home/ec7538/atca7368/ga01/prj/export/atca7368-System.map-
WR4.0.0.0_cgl

Kernel:         /home/ec7538/atca7368/ga01/prj/export/atca7368-
default_kernel_image-WR4.0.0.0_cgl

RootFileSystem: /home/ec7538/atca7368/ga01/prj/export/atca7368-cgl-
glibc_cgl-dist.tar.bz2
```

```
LinuxModules:   /home/ec7538/atca7368/ga01/prj/export/atca7368-linux-
modules-WR4.0.0.0_cgl.tar.bz2


============================================================

=                                                          =

=     Building rootfs and ramdisk image for ACPI4-C        =

=                                                          =

============================================================
```

Appendix A

Installing and Configuring BBS

A.1 Installing BBS Using Hard Disk

After the system comes up, install Linux with the following procedure:

1. Login as **`root`**.
2. Identify the Linux device name of the hard disk on which you want to install BBS. To do so, enter **`fdisk -l`**. This displays available hard disks, their Linux device names and also the storage capacity. An easy way to identify a particular hard disk is by its storage capacity. Refer to the respective hardware user manuals for information about the storage capacities of the hard disks used in your configuration. Another way to identify a particular hard disk, is via the device name. Linux uses different device names for different hard disk types. The exact format, however, differs between Linux versions and distributions. Refer to your Linux documentation for further details.
3. Run the `linuxrc` script from the `/opt/bladeservices/tools` directory:
   `./linuxrc`
   The hard disk installation begins by checking for necessary commands on the system. The screen output will look similar to this:

```
Checking for necessary commands...
awk               [exists]
chroot            [exists]
mount             [exists]
umount            [exists]
tar               [exists]
gzip              [exists]
mkdir             [exists]
rmdir             [exists]
rm                [exists]
cp                [exists]
mv                [exists]
```

```
    date              [exists]
    chmod             [exists]
    chown             [exists]
    grep              [exists]
    dd                [exists]
    stty              [exists]
    sed               [exists]
    …                 …
Necessary commands found, safe to continue...
------
The following disks are available:
/dev/sda: 40.0 GB
/dev/sdb: 4110 MB

 Following default settings were found:
export AUTO_DEV_NAME=sda
export AUTO_TFTPSRV=192.168.22.55
export AUTO_TFTPLOC=ATCA7368
export AUTO_TZ=n
export AUTO_NTPUSE=n
export AUTO_MDATEUSE=n
 Do you want to use predefined values? [y/n] n
------
 Select the disk/flash device where you want to have the
filesystem installed (e.g. sdc) []: sda
 ------

 Verifying disk device: /dev/sda...done.
```

4. Start the installation by entering **y** for yes.

   ```
   Do you wish to begin the installation? [y/n]
   ```

   There is no default answer to this question. Choosing y will begin the installation. Choosing n will abort the installation.

   Starting the installation will cause the hard disk drive to be partitioned and formatted while displaying the message:

   ```
   Disk /dev/sda: 4864 cylinders, 255 heads, 63 sectors/track
   Old situation:
   Units = mebibytes of 1048576 bytes, blocks of 1024 bytes, counting from 0

   Device    Boot Start    End    MiB    #blocks   Id  System
   /dev/sda1         0+   1004-  1005-   1028159+  83  Linux
   /dev/sda2   *  1004+   6008-  5005-   5124735   83  Linux
   /dev/sda3      6008+  11013-  5005-   5124735   83  Linux
   /dev/sda4     11013+  38154- 27142-  27792450    5  Extended
   /dev/sda5     11013+  16017-  5005-   5124734+  83  Linux
   /dev/sda6     16017+  21022-  5005-   5124734+  83  Linux
   /dev/sda7     21022+  22560-  1538-   1574369+  82  Linux swap/Solaris
   /dev/sda8     22560+  26560-  4001-   4096574+  83  Linux
   /dev/sda9     26560+  27078-   518-    530144+  83  Linux
   /dev/sda10    27078+ 27596-   518-    530144+  83  Linux
   New situation:
   Units = mebibytes of 1048576 bytes, blocks of 1024 bytes, counting from 0

   Device    Boot Start    End    MiB    #blocks   Id  System
   ```

```
/dev/sda1          0+    1004-   1005-    1028159+  83  Linux
/dev/sda2   *   1004+    6008-   5005-    5124735   83  Linux
/dev/sda3       6008+   11013-   5005-    5124735   83  Linux
/dev/sda4      11013+   38154-  27142-   27792450    5  Extended
/dev/sda5      11013+   16017-   5005-    5124734+  83  Linux
/dev/sda6      16017+   21022-   5005-    5124734+  83  Linux
/dev/sda7      21022+   22560-   1538-    1574369+  82  Linux swap/Solaris
/dev/sda8      22560+   26560-   4001-    4096574+  83  Linux
/dev/sda9      26560+   27078-    518-     530144+  83  Linux
/dev/sda10     27078+   27596-    518-     530144+  83  Linux

Successfully wrote the new partition table

Re-reading the partition table ...

If you created or changed a DOS partition, /dev/foo7, say, then use dd(1)
to zero the first 512 bytes:  dd if=/dev/zero of=/dev/foo7 bs=512 count=1
(See fdisk(8).)
```

5. Choose Dynamic IP Configuration.
   You will be prompted to either accept dynamic IP configuration for the installation interface, or to choose static configuration:
   Artesyn recommends to use dynamic IP configuration for all blades
   `Do you wish use static IP management [y/N]?`
   Enter **N** or press <Enter>.

6. Choose the TFTP Server.
   The TFTP server houses all of the files necessary for the installation. There is no default choice.
   `Which TFTP Server do you wish to use? [xxx.xxx.xxx.xxx]`
   A series of 'pings' are sent to the server at the given address to ensure connectivity. If the connection to the tftp server cannot be established, the query will be repeated.

7. Select the TFTP server installation directory.
   A known set of files is expected to be available on the TFTP server to proceed the installation. Only the location of these files on the TFTP server can be configured.
   Note: Please enter the directory name without leading or trailing slashes.
   `What is the installation files directory? [ ]`
   For example: **`ATCA7368`**

8. Downloading of Files
   During the download of the files the following is displayed, for example:

```
 Downloading files.sha1sum from ATCA7368....Done.

 Downloading kernel from ATCA7368....Done.

 Downloading rootfs.tar.gz from ATCA7368....Done.

 Downloading modules.tar.bz2 from ATCA7368....Done.

 Downloading atca-7368_em_bbs-bios_1_0_0.rpm from ATCA7368....Done.

 Downloading atca-7368_em_bbs-boardctrl_1_0_1.rpm from
ATCA7368....Done.

 Downloading atca-7368_em_bbs-flashrom_0_1_1.rpm from ATCA7368....Done.

 Downloading atca-7368_em_bbs-fpga_1_0_0.rpm from ATCA7368....Done.

 Downloading atca-7368_em_bbs-fuf_1_3_8.rpm from ATCA7368....Done.

 Downloading atca-7368_em_bbs-ibbl_2_0_1.rpm from ATCA7368....Done.

 Downloading atca-7368_em_bbs-ipmc_2_0_1.rpm from ATCA7368....Done.

 Downloading atca-7368_em_bbs-hpmagentcmd_1_3_12.rpm from
ATCA7368....Done.

 Downloading artm-7368_em_bbs-mmcb_2_0_1.rpm from ATCA7368....Done.

 Downloading artm-7368_em_bbs-mmcf_2_0_1.rpm from ATCA7368....Done.

 Downloading initrd0.img from ATCA7368....Done.

 Downloading vmlinux from ATCA7368....Done.
```

9. Set the time zone, if necessary. The time zone (24-hour clock) is by default set to US/Eastern on all the blades. In order to change the time zone, enter **y** for Yes when being prompted. See the following output example.

```
Your current time zone is set to US/Eastern
Do you want to change that? [n]:
y
```

```
 1) Africa
 2) Americas
 3) Antarctica
 4) Arctic Ocean
 5) Asia
 6) Atlantic Ocean
 7) Australia
 8) Europe
 9) Indian Ocean
10) Pacific Ocean
11) none - I want to specify the time zone using the Posix TZ format.
#?8
Please select a country.
 1) Aaland Islands18) Greece          35) Norway
 2) Albania       19) Guernsey        36) Poland
 3) Andorra       20) Hungary         37) Portugal
 4) Austria       21) Ireland         38) Romania
 5) Belarus       22) Isle of Man     39) Russia
 6) Belgium       23) Italy           40) San Marino
 7) Bosnia & Herzegovina  24) Jersey  41) Serbia
 8) Britain (UK)  25) Latvia          42) Slovakia
 9) Bulgari       26) Liechtenstein   43) Slovenia
10) Croatia       27) Lithuania       44) Spain
11) Czech Republic28) Luxembourg      45) Sweden
12) Denmark       29) Macedonia       46) Switzerland
13) Estonia       30) Malta           47) Turkey
14) Finland       31) Moldova         48) Ukraine
15) France        32) Monaco          49) Vatican City
16) Germany       33) Montenegro
17) Gibraltar     34) Netherlands
#?16
```

Choose a time zone out of the list that is displayed and enter the corresponding number. After choosing Germany, for example, the following output would be displayed.

```
The following information has been given:
Germany
Therefore TZ='Europe/Berlin' will be used.
Local time is now:      Sun Jan 2 12:58:05 CET 2005.
Universal Time is now:  Sun Jan 2 11:58:05 UTC 2005.
Is the above information OK?
1) Yes
2) No
#?1
```

10. Set the time used on the blade. It is possible to set the time automatically using an NTP server if an NTP server is available, or to set it manually. It is strongly advised that one of two methods is used to ensure that a valid date and time is set on the system before the installation of files begins.
    If a valid NTP server is available, answer the following question with **y** and enter the IP address of the NTP server.

    ```
     Do you wish to use NTP to set the current time? [Y/n]
    Please enter the NTP server address [xxx.xxx.xxx.xxx]
    ```

    If a NTP sever is not available, then the time can be set manually. To do so, answer the following question with y and enter the date and time manually.

    ```
    Do you wish to set the date manually? [Y/n]y
    Enter date in 'MM/DD/YYYY' format. [] Enter time in 'HH:MM'
    and 24-hour format. []
    ```

    The values entered are validated and ensure accuracy.
    The value that is either gathered from the NTP server or entered manually is written to the hardware clock of the blade.

11. Check SHA1 Checksums and Install
    Once the files have been downloaded, the SHA1 checksums of the downloaded files are compared to their expected values and if they are correct, the root file system is un-compressed and finally the BBS software’s RPMs are installed.

A.2 Setting up the kdump Utility on a Hard Disk Driver Installed System

Kexec and kdump are new features in the 2.6 mainstream kernel. The purpose of these features is to ensure faster boot up and creation of reliable kernel vmcores for postmortem diagnostic purposes in case the system crashes.

Kexec

Kexec is a fastboot mechanism which allows booting a Linux kernel from the context of already running kernel without going through BIOS.

Kdump

Kdump is a new kernel crash dumping mechanism and is very reliable because the crash dump is captured from the context of a freshly booted kernel and not from the context of the crashed kernel. Kdump uses kexec to boot into a second kernel whenever system crashes. This second kernel often called as capture kernel, boots with very little memory and captures the dump image.

The first kernel reserves a section of memory that the second kernel uses to boot (in our case the memory region is 64M~256M). Kexec enables booting the capture kernel without going through BIOS hence contents of first kernel's memory are preserved, which is essentially the kernel crash dump.

This feature has been integrated for Hard Disk Driver installed BBS system. With pre-configured default settings, user can easily configure their system to get and analysis the dumped vmcore dump file.

Following are the list of files with procedure to perform the postmortem analysis:

1. Configuration files:

File	Configurable	Comments
/etc/kdump.conf	YES	Configures where to put the kdump vmcore files. Default location is /var/crash.
/etc/sysconfig/kdump	NO	Configures the capture kernel related options. We use the same kernel for both the first kernel and the capture kernel with re-locatable feature enabled.
/etc/init.d/kdump	NO	Start and stop kdump crash recovery service. The kdump init script provides the support necessary for loading a capture kernel into memory at system boot up time, and for copying away a vmcore at system panic time.
/boot/grub.conf	NO	Boot parameters ’crashkernel' to reserve a chunk of memory for the capture kernel, in our case 'crashkernel=256M@64M'.

2. kdump.conf option description.
   The kdump.conf file configures manually or automatically to save the dumped core file /proc/vmcore. It also configures the location to store the core file.

   - Typical kdump.conf file looks like as follows:

```
auto_dump yes
#raw /dev/sda1
#ext3 /dev/sda1
#ext3 LABEL=/boot
#ext3 UUID=03138356-5e61-4ab3-b58e-27507ac41937
#net my.server.com:/export/tmp
#net netdumpuser@192.168.16.100
path /var/crash
#core_collector makedumpfile -c
#link_delay 60
#default shell
```

- Configuration option and value description.

Option	Value Description
`auto_dump <yes/no>`	Set to "yes" if you want the capture kernel to dump the core file `/proc/vmcore` to your specified location. After dumping the kernel will reboot the system to a fresh state. Set to "no" will prevent the capture kernel from saving core files automatically. And you have to save the core file manually to your preferred location after the capture kernel boots up. It is always necessary to reboot after you finish saving.
`raw <partition>`	This will store `/proc/vmcore` into given `<partition>`.
`net <nfs mount>`	This will mount fs and copy `/proc/vmcore` to `<mnt>/var/crash/%HOST-%DATE/`, DNS supported.
`net <user@server>`	This will scp `/proc/vmcore` to `<user@server>:/var/crash/%HOST-%DATE/`, DNS supported. NOTE: make sure user has necessary write permissions on server.
`<fs type> <partition>`	This will mount `-t <fs type> <partition> /mnt` and copy `/proc/vmcore` to `/mnt/var/crash/%DATE/`. NOTE: `<partition>` can be a device node, label or uuid.

Option	Value Description
`path <path>`	Append path to the filesystem device which you are dumping to. It is ignored for raw device dumps. If unset, it will use the default path `/var/crash`.
`core_collector makedumpfile <options>`	This directive allows you to use the dump filtering program `makedumpfile` to retrieve your core with reduced file size. See `/bin/makedumpfile --help` for a list of options. Note that the `-i` and `-g` options are not needed here, as the `initrd` will automatically be populated with a config file for the running kernel.
`link_delay <seconds>`	Some network cards may takes long time to initialize, and some spanning tree enabled networks do not transmit user traffic for long period after the link state changes. This optional parameter defines a wait period after a link is activated in which the `initramfs` will wait before attempting to transmit user data.
`default <reboot | shell>`	This command is used to declare which action is to be performed either `reboot` or `shell` instead of mounting root fs and running init process. `reboot`: It simply reboots the system and cancels the core that you are trying to retrieve. `shell`: If the default action is shell, then drop to an msh session inside the initramfs from where you can try to record the core manually. Exiting this shell reboots the system. NOTE: If no default action is specified, the initramfs will mount the root file system and runs init.

You can use "#" as prefix to comment out those options which you do not want to use.

3. Debug information files.
   To support postmortem analysis, the kernel-debuginfo file have been provided as `/boot/vmlinux` in an HDD installed BBS system. The uncompressed kernel image file has kernel symbols built-in for debug purpose usage.

4. Make your own configuration.

All the users need to modify the configuration file `/etc/kdump.conf` to specify the location where vmcore dumped files to be put-up.
Restart the kdump service by running of the following command to make your modified configuration activated.

```
root@ATCA-7368-13:#service kdump restart
```

5. Make the analysis.
   Use the analysis tool 'crash' with the option of '--no_data_debug' to make the postmortem analysis. For example:

   ```
   root@ATCA-7368-13:~#crash --no_data_debug /boot/vmlinux-
   2.6.34.6-grsec-WR4.0.0.0_cgl /var/crash/2010-11-08-
   11\:06/vmcore
   ```

6. An example.
   You can force-crash your system by echo'ing a c into `/proc/sysrq-trigger`:

   ```
   root@ATCA-7368-13:~# echo "c" > /proc/sysrq-trigger
   ```

   You should see some panic output, followed by the system restarting into the kdump kernel. When the boot process gets to the point where it starts the kdump service, your vmcore should be copied out to disk (by default in `/var/crash/<YYYY-MM-DD-HH:MM>/vmcore`), then the system rebooted back into your normal kernel.
   Once back to your normal kernel, you can use the crash kernel in conjunction with the kernel-debuginfo file to perform the postmortem analysis, for example:

   ```
   root@ATCA-7368-13:~# crash --no_data_debug /boot/vmlinux-
   2.6.34.6-grsec-WR4.0.0.0_cgl /var/crash/2010-11-08-
   12\:05/vmcore

   crash 5.0.8

   … …

   GNU gdb (GDB) 7.0

   … …

   please wait... (gathering kmem slab cache data)

   please wait... (gathering module symbol data)

   WARNING: invalid kernel module size: 0

   please wait... (gathering task table data)
   ```

```
please wait... (determining panic task)
KERNEL: /boot/vmlinux-2.6.34.6-grsec-WR4.0.0.0_cgl
    DUMPFILE: /var/crash/2010-11-08-11:06/vmcore
        CPUS: 6
        DATE: Mon Nov  8 19:05:52 2010
      UPTIME: 00:05:11
LOAD AVERAGE: 0.00, 0.02, 0.00
       TASKS: 212
    NODENAME: ATCA-7368-13
     RELEASE: 2.6.34.6-grsec-WR4.0.0.0_cgl
     VERSION: #1 SMP PREEMPT Wed Nov 3 02:08:00 HKT 2010
     MACHINE: x86_64  (1994 Mhz)
      MEMORY: 8 GB
       PANIC: "Oops: 0002 [#1] PREEMPT SMP " (check log for details)
         PID: 4982
     COMMAND: "bash"
        TASK: ffff880214b5af40  [THREAD_INFO: ffff880216f82000]
         CPU: 0
       STATE: TASK_RUNNING (PANIC)

crash> bt
PID: 4982   TASK: ffff880214b5af40  CPU: 0   COMMAND: "bash"
 #0 [ffff880216f83e40] __handle_sysrq at ffffffff8136ec6f
 #1 [ffff880216f83e80] write_sysrq_trigger at ffffffff8136ed11
 #2 [ffff880216f83e90] proc_reg_write at ffffffff8117623f
 #3 [ffff880216f83ee0] vfs_write at ffffffff81118ce3
 #4 [ffff880216f83f30] sys_write at ffffffff81118f7a
    RIP: 00000039986c2870  RSP: 00007fffa9ef51e0  RFLAGS: 00010206
    RAX: 0000000000000001  RBX: ffffffff8100302b  RCX: 0000000000000400
    RDX: 0000000000000002  RSI: 00007fa2914f3000  RDI: 0000000000000001
    RBP: 00007fffa9ef5a04   R8: 000000000000000a   R9: 00007fa291b15700
    R10: 0000000000000022  R11: 0000000000000246  R12: 0000000000000002
    R13: 000000399894f780  R14: 00007fa2914f3000  R15: 0000000000000002
    ORIG_RAX: 0000000000000001  CS: 0033  SS: 002b

crash> q
```

Related Documentation

B.1 Artesyn Embedded Technologies - Embedded Computing Documentation

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

1. Go to www.artesyn.com/computing.
2. Under SUPPORT, click TECHNICAL DOCUMENTATION.
3. Under FILTER OPTIONS, click the Document types drop-down list box to select the type of document you are looking for.
4. In the Search text box, type the product name and click GO.

Table B-1 Artesyn Embedded Technologies - Embedded Computing Publications

Document Title	Publication Number
ATCA-7368 Installation and Use Guide	6806800M12
RTM-ATCA-7360 Installation and Use	66806800J08
Centellis 2000 Preliminary Installation and Use	6806800G45
Centellis 4440 Installation and Use	6806800H23

B.2 Related Specifications

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

Table B-2 Related Specifications

Document Title	Source
IPMI Specifications http://www.intel.com/design/servers/ipmi	
IPMI Spec V.2.0	Intel Corporation, Hewlett-Packard, DEC, NEC
IPMI Platform Management FRU Information Storage Definition V1.0, September 27, 1999	Intel Corporation
PCI Industrial Computer Manufacturers Group (PICMG) Specifications http://www.picmg.org	
PICMG 3.0 Revision 2.0 Advanced Telecommunications Computing Architecture (AdvancedTCA) Base Specification	PICMG

B.3 References

The following table lists references documentations for which the BBS software is implemented.

Table B-3 Additional Resources

Document Title	Source
Embedded SW Delivery Format Description for ATCA HW Platform	Jarmo Kant, NSN

B.4 Additional Resources

The following table lists additional resources which may be useful in working with Artesyn's AdvancedTCA systems.

Table B-4 Additional Resources

Resource	Source
OpenHPI open source software project http://openhpi.org	
OpenHPI 1.0 Manual	OpenHPI
OpenHPI NetSNMP Subagent Development Manual	OpenHPI
Net-SNMP http://net-snmp.sourceforge.net/	
Pigeon Point Systems http://www.pigeonpoint.com	
IPM Sentry Shelf-External Interface Reference	Pigeon Point Systems
IPM Sentry Shelf Manager User Guide	Pigeon Point Systems
OpenIPMI http://openipmi.sourceforge.net/	

ARTESYN™
EMBEDDED TECHNOLOGIES