Dialogic®

Dialogic® Multimedia Blade for AdvancedTCA®

Technical Product Specification

February 2007

05-2535-002

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation. Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic Corporation does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic Corporation or its subsidiaries may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic Corporation does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic Corporation or its subsidiaries and no such licenses are provided except pursuant to a signed agreement with Dialogic Corporation. More detailed information about such intellectual property is available from Dialogic Corporation's legal department at 9800 Cavendish Blvd., Montreal, Quebec, Canada H4M 2V9. **Dialogic Corporation encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Dialogic is a registered trademark of Dialogic Corporation. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time.

The names of actual companies and products mentioned herein are the trademarks of their respective owners.

Publication Date: February 2007

Document Number: 05-2535-002

Contents

Figures

Tables

Revision History

Document No.	Publication Date	Description of Revisions
05-2535-001	December 2006	Initial version of document.
05-2535-002	February 2007	Made global changes to reflect Dialogic brand. In "Getting Started" chapter, updated operating system information in "Installing the Operating System" section. In "Administration via Command Line Interface (CLI)" chapter, added procedure for "Configuring PSTN Trunks and Clock Fallback". Added new chapter on "Troubleshooting". In "Using Shelf Managers with the Multimedia Platform" appendix, renamed appendix and added "Using Intel NetStructure® MPCMM001 Chassis Management Module" section.

1.0 Introduction

1.1 Document Organization

This document gives operational information and technical specifications related to the Dialogic® Multimedia Blade for AdvancedTCA® (hereinafter referred to as the Multimedia Blade). The following summarizes the focus of each section in this document.

Chapter 1.0, "Introduction" gives an overview of the information contained in this document as well as a glossary of acronyms and important terms.

Chapter 2.0, "Product Overview" provides a functional block diagram of the Multimedia Blade and discusses some of its key features.

Chapter 3.0, "Getting Started" provides procedures for installing and configuring the hardware and software.

Chapter 4.0, "Administration via Command Line Interface (CLI)" provides information about using the Command Line Interface (CLI) agent software for local or remote configuration and monitoring of the Multimedia Blade.

Chapter 5.0, "Administration via SNMP" provides information about using the Simple Network Management Protocol (SNMP) agent software for local or remote monitoring and limited configuration of the Multimedia Blade.

Chapter 6.0, "Controls, Indicators, and Connectors" includes an illustration of LEDs, connector locations, connector descriptions, and pinout tables.

Chapter 7.0, "Hardware Management" provides a high-level overview of the IPMI implementation based on the PICMG* 3.0 and IPMI 2.0 specifications.

Chapter 8.0, "BIOS Features" provides an introduction to the BIOS features of the Multimedia Blade.

Chapter 9.0, "BIOS Settings" describes the interactive menu system of the BIOS Setup program, which allows users to configure the BIOS for a given system.

Chapter 10.0, "Specifications" provides specifications for the Multimedia Blade in such areas as operation, connectivity, power, environmental, and more.

Chapter 11.0, "Troubleshooting" provides information about troubleshooting your system.

Appendix A, "Supported IPMI Commands" lists the IPMI commands supported by the Multimedia Blade.

Appendix B, "BIOS Error Messages" lists the BIOS error messages supported by the Multimedia Blade.

Appendix C, "POST Checkpoints" describes the Power On Self Test (POST) codes generated by the BIOS.

Appendix D, "Using Shelf Managers with the Multimedia Platform" provides information about shelf managers, including: firmware version required for the Intel NetStructure® MPCMM001 Chassis Management Module; procedures for upgrading Schroff Shelf Manager firmware; and procedures for reprogramming the Multimedia Platform RTM FRU file, in case it was overwritten.

1.2 Glossary

The following is a list of acronyms and terms used in this document.

ACPI	Advanced Configuration and Power Interface
AdvancedMC	Advanced Mezzanine Card
AdvancedTCA	Advanced Telecommunications Compute Architecture
BIOS	Basic Input/Output Subsystem. ROM code that initializes the computer and performs some basic functions.
Blade	An assembled PCB card that plugs into a chassis
CLI	Command Line Interface
DIMM	Dual Inline Memory Module
DSP	Digital Signal Processor
ECC	Error Correcting Code
EEPROM	Electrically Erasable Programmable Read-Only Memory
Fabric Board	A board capable of moving packet data between Node Boards via the ports of the backplane. This is sometimes referred to as a switch.
Fabric Slot	A slot supporting a link port connection to/from each Node Slot and/or out of the chassis
FPGA	Field Programmable Gate Array
FRB	Fault Resilient Booting
FRU	Field Replaceable Unit
FSB	Front-Side Bus
FWH	Firmware Hub
GbE	Gigabit Ethernet
GPIO	General Purpose I/O
Hyper-Threading Technology (HT Technology)	Allows a single (or dual) physical processor, to appear as two (or quad) logical processors to a HT Technology-aware operating system.
I^2C	Inter-Integrated Circuit. A two-wire interface commonly used to carry management data.
IBA	Intel® Boot Agent. Software that allows your networked client computer to boot using a program code image supplied by a remote server.
ICH	I/O Controller Hub
IDE	Integrated Device Electronics. A common, low-cost disk interface.

IPMB	Intelligent Platform Management Bus. A physical two-wire medium to carry IPMI information.
IPMC	Intelligent Platform Management Controller. An ASIC on the baseboard responsible for low-level system management.
IPMI	Intelligent Platform Management Interface. A programming model for system management.
KCS	Keyboard Controller Style interface
LPC Bus	Low Pin Count Bus. A legacy I/O bus that replaces ISA and X-bus.
MAC	Media Access Control
MCA	Machine Check Architecture
MCH	Memory Controller Hub
MIB	Management Information Base
MMC	Module Management Controller
NEBS	Network Equipment Building System. A set of telco standards for equipment emissions, thermal, shock, contaminants, and fire suppression requirements.
NMI	Non-Maskable Interrupt. A low-level PC interrupt.
Node Board	A board capable of providing and/or receiving packet data to/ from a Fabric Board via the ports of the networks. This term is used interchangeably with SBC.
Node Slot	A slot supporting port connections to/from one or more Fabric slots. A Node Slot is intended to accept a Node Board.
OID	Object Identifier
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
PEF	Platform Event Filtering
Physical Port	A port that physically exists. It is supported by one of many physical (PHY) type components.
PICMG	PCI Industrial Computer Manufacturers Group
POST	Power On Self Test
PSTN	Public Switched Telephone Network
ROM	Read-Only Memory
RTM	Rear Transition Module
SAS	Serial Attached SCSI
SBC	Single Board Computer. This term is used interchangeably with Node Board.
SCSI	Small Computer System Interface
SDR	Sensor Data Record
SEL	System Event Log. Actions logged by the management controller.
SFF	Small Form Factor

SMBus	System Management Bus. Similar to I^2C.
SMI	System Management Interrupt. A low-level PC interrupt that can be initiated by the chipset or management controller. Used to service IPMC or handle things like memory errors.
SMM	Systems Management Mode
SMS	System Management Software or Standard Microsystems Corporation*
SNMP	Simple Network Management Protocol
SOL	Serial Over LAN
SPOR	Standby Power On Reset
USB	Universal Serial Bus. A general-purpose peripheral interconnect. USB 1.1 operates up to 12 Mbps. USB 2.0 operates up to 480 Mbps.
WDT	Watchdog Timer

2.0 Product Overview

This chapter provides an overview of the Dialogic® Multimedia Blade for AdvancedTCA® (hereinafter referred to as the Multimedia Blade). The following topics are discussed:

- Product Applications
- Block Diagram
  - Baseboard
  - Advanced Mezzanine Card
  - Rear Transition Module
- Overview of Main Subsystems
- External Connections
- Real-Time Clock

2.1 Product Applications

The Multimedia Blade is a powerful and cost-effective product that can be used to deliver applications such as voice and video mail, color ringback tones, unified messaging, and audio conferencing over IP and PSTN interfaces in wireline and wireless environments using standard protocols for session and media control.

The Multimedia Blade is designed to meet the needs of Telecom Equipment Manufacturers (TEMs) and other vendors who are building next-generation multimedia processing solutions, especially in the IP Multimedia Subsystem (IMS) framework, while enabling migration from solutions deployed in existing TDM networks. The platform provides a comprehensive set of building blocks that developers need to create multimedia server and gateway solutions. Its flexibility allows a choice between running customer-provided software from a local hard drive or standard remote interfaces for media control, configuration, and management.

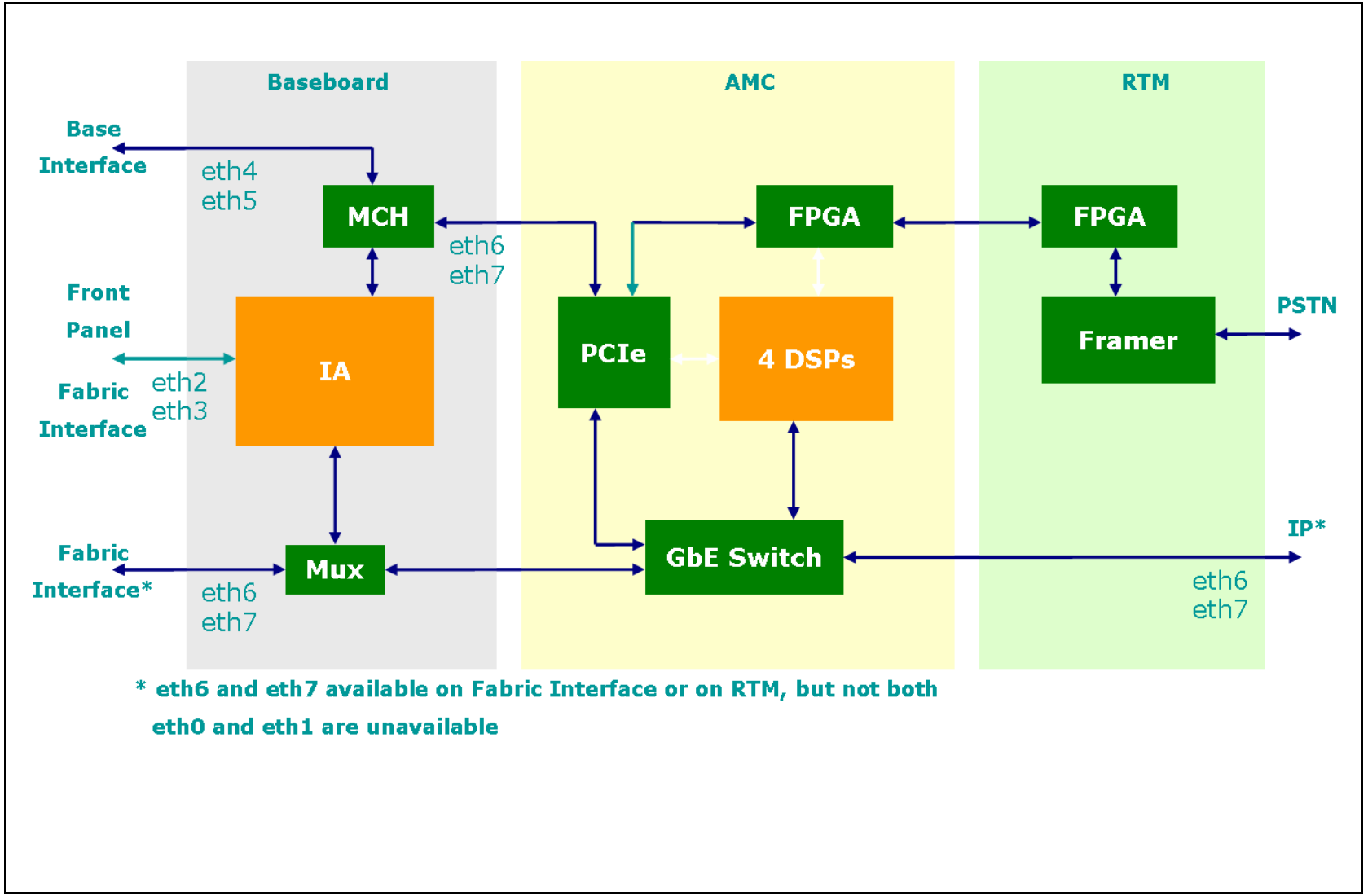
2.2 Block Diagram

Figure 1 shows the functional blocks of the Multimedia Blade, including the baseboard, Advanced Mezzanine Card (AdvancedMC) and Rear Transition Module (RTM).

2.2.1 Baseboard

The baseboard is a hot-swappable SBC with backplane connections to Gigabit Ethernet (GbE) ports on the Base and Fabric interface. The Fabric interface of the Multimedia Blade supports option 2 of the PICMG 3.1 specification.

Figure 1. Multimedia Blade Block Diagram



The front panel provides one USB port, one serial console port, two SAS ports (one physical SAS connector) and two Gigabit Ethernet ports.

Note: The front panel connectors should not be used during normal (run time) operation. During normal operation, connections to the PSTN and Ethernet are made via connectors on the Rear Transition Module (RTM).

For storage, the board includes an on-board Serial Attached SCSI (SAS) small form factor (SFF) hard disk. In addition, there is a front panel SAS connector for external SAS drive connections and SAS connections to the RTM.

The Multimedia Blade incorporates an Intelligent Platform Management Controller (IPMC) that monitors critical hardware functionality of the board such as temperature and voltage, responds to commands from the shelf manager, and reports events.

Power is supplied to the Multimedia Blade through two redundant -48 V power supply connections.

2.2.2 Advanced Mezzanine Card

The Advanced Mezzanine Card (AdvancedMC), which is pre-installed on the baseboard, has four digital signal processors (DSPs) used for echo cancellation and transcoding offload.

2.2.3 Rear Transition Module

The Multimedia Blade can be used with either a PSTN RTM or an IP-only RTM:

- The PSTN RTM (product code MMB480RTMTE01A) provides 16 spans of E1/T1 (clear channel only) and two Gigabit Ethernet interfaces.
- The IP-only RTM (product code MMBRTMIP01A) provides two Gigabit Ethernet interfaces.

2.3 Overview of Main Subsystems

The following sections discuss:

- Dual-Core Intel® Xeon® Processor LV 2.0 GHz Processors
- Chipset
- Intelligent Platform Management Controller

2.3.1 Dual-Core Intel® Xeon® Processor LV 2.0 GHz Processors

The Multimedia Blade supports two Dual-Core Intel® Xeon® processor LV 2.0 GHz processors with 667 MHz front side bus, with the following features:

- Dual processor support, including an enhanced bus arbitration protocol, power-optimized 667 MHz front-side bus (FSB), and a 2 MB shared L2 cache per processor that enables up to four high-performance cores per platform
- 36-bit memory addressing that supports more than 4 GBytes of memory
- FSB address, data parity, and an enhanced error reporting mechanism through the MCA (Machine Check Architecture) that ensures reliability and data integrity

2.3.2 Chipset

The Multimedia Blade uses the Intel® E7520 Chipset, which comprises the following major components:

- Intel® E7520 Memory Controller Hub (MCH)
- Intel® 6300ESB I/O Controller (ICH)
- Intel® 82571EB Gigabit Ethernet Controller

2.3.3 Intelligent Platform Management Controller

The Intelligent Platform Management Controller (IPMC) is a management subsystem providing monitoring, event logging, and recovery control. The IPMC serves as the gateway for management applications to access the platform hardware. It complies with PICMG 3.0 and IPMI v2.0.

2.4 External Connections

The following sections discuss:

- Gigabit Ethernet
- Serial Attached SCSI (SAS) Controller
- USB 2.0
- Serial Port

2.4.1 Gigabit Ethernet

The Multimedia Blade implements six Gigabit Ethernet (GbE) interfaces from three separate Gigabit Ethernet controllers. Two of these interfaces are routed to the Base interface, and the other four are routed to the Fabric interface on the AdvancedTCA backplane to support the PICMG* 3.0 (Base) and 3.1 option 2 (Fabric) specifications. Refer to the block diagram in Figure 1.

The AdvancedMC GbE ports are connected to the backplane Fabric interface (Port 0, Channel 1 and 2) and RTM Ethernet ports. The RTM Ethernet ports and Fabric interface ports share the Ethernet controllers.

2.4.2 Serial Attached SCSI (SAS) Controller

The Multimedia Blade has an 8-port Serial Attached SCSI (SAS) controller and one on-board SAS hard drive. The SAS controller is connected to the PCI-X bus of the Intel® 6300ESB I/O Controller Hub. All SAS ports have a serial point-to-point interface using a differential transmit/receive pair. The SAS controller has a flash device that is used to store its firmware.

2.4.3 USB 2.0

The Multimedia Blade has one front panel USB connector that supports USB 2.0 and 1.1. USB supports Plug & Play* and hot swapping operations (OS level) that allow USB devices to be automatically attached, configured, and detached, without rebooting. There is a second USB port that is routed to the RTM.

2.4.4 Serial Port

The Multimedia Blade supports one serial port connected to the Intel® 6300ESB I/O Controller Hub. The serial port is routed to the front panel RJ-45 connector for normal operation.

2.5 Real-Time Clock

The Multimedia Blade real-time clock is integrated into the ICH. It is derived from a 32.768 kHz crystal.

The real-time clock is powered by a total capacitance of 0.47 F when main power is not applied to the board. This capacitor powers the real-time clock for a minimum of 2 hours while external power is removed from the Multimedia Blade.

Note: If the board is powered down for greater than 2 hours, the system date and time and BIOS settings **revert to the factory defaults** and will have to be reset. Ensure that the date and time are set before starting the Dialogic Services.

3.0 Getting Started

This chapter provides procedures for installing and configuring the Dialogic® Multimedia Blade for AdvancedTCA® and Dialogic® Multimedia Software Release for AdvancedTCA® (hereinafter referred to as the Multimedia Blade and Multimedia Software). The major installation steps are:

- Installing the Rear Transition Module (RTM)
- Installing the Multimedia Blade
- Connecting Serial Port to Terminal Emulator
- Setting up the BIOS
- Installing the Operating System
- Updating the Ethernet Driver
- Obtaining a License File
- Installing the Multimedia Software
- Configuring the System
- Subnet Connectivity
- Verifying the System

Note: Before performing these procedures, check the *Release Guide* for system requirements. Make sure that your system meets the specified hardware and software requirements.

The following information is also included in this chapter for reference purposes:

- Digital Ground to Chassis Ground Connectivity
- Identifying Multimedia Blade Ethernet MAC Addresses

3.1 Digital Ground to Chassis Ground Connectivity

Digital ground is tied to chassis ground through a screw that threads into a standoff in the metal backing plate. The PCB hole for this screw is non-plated, and is surrounded by a large gold plated copper pad on the PCB top side and a copper-free area on the PCB bottom side. The large copper pad is tied to logic ground by several vias around its perimeter. The screw head bears on the copper pad and threads into the backside plate to make the connection. A plastic shoulder washer is used under the screw head to keep the grounds isolated. Boards ship standard with this washer in place. If it is necessary to short the grounds for regulatory reasons, remove the washer and reinstall the screw.

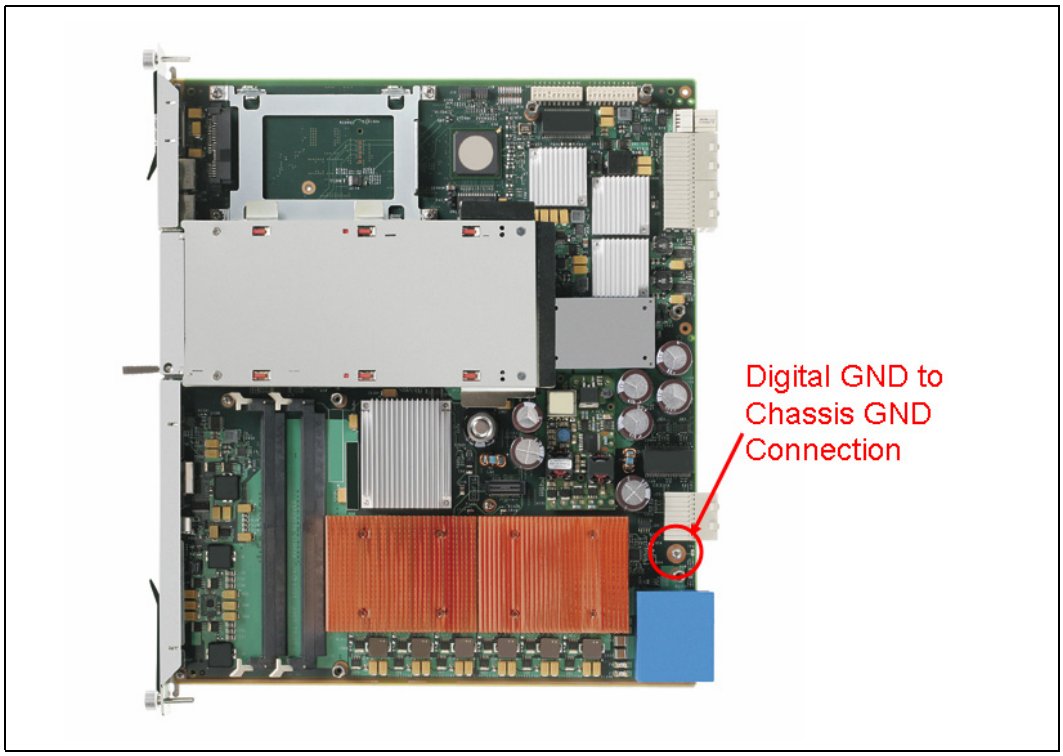
Note: For the Multimedia Blade, the default is that digital ground is isolated from chassis ground.

To connect the digital ground to the chassis ground, follow this procedure:

1. Remove the top metal cover from the board. The hard drive and DIMM covers do not need to be removed separately. It is OK to remove the entire top cover with the hard drive and DIMM covers still in place.
2. Remove the screw with the plastic shoulder washer. See Figure 2 for the screw location above the P10 blue connector.
3. Remove the plastic insulating washer to connect digital ground to chassis ground.
4. Store the plastic insulation washer in a safe place. It may be needed to isolate the digital ground from chassis ground in the future.
5. Reinstall the screw and tighten to 4 in-lb (0.45N-m) using a torque screwdriver.
6. Reinstall the top metal cover to the board.

Note: Digital ground is also called logic ground. Chassis ground is also known as shelf ground.

Figure 2. Digital Ground to Chassis Ground Screw Location



3.2 Identifying Multimedia Blade Ethernet MAC Addresses

The backing plate of the Multimedia Blade has a MAC address label to assist users in determining the MAC address of each Ethernet port on the SBC.

Figure 3 shows the location of the MAC address label on the Multimedia Blade backing plate. Table 1 breaks down the mapping of the Ethernet interfaces to the labeling on the Multimedia Blade faceplate and in the operating system.

The first MAC address (port A) can be retrieved remotely by using an IPMI command. From that first MAC address (port A), each port increments by 0x01h.

In the IPMI specification (23.2), parameter #5 (MAC address) of the Get LAN Configuration Parameters Command will be populated automatically with the base interface MAC address.

Figure 3. MAC Address Label on Multimedia Blade Backing Plate

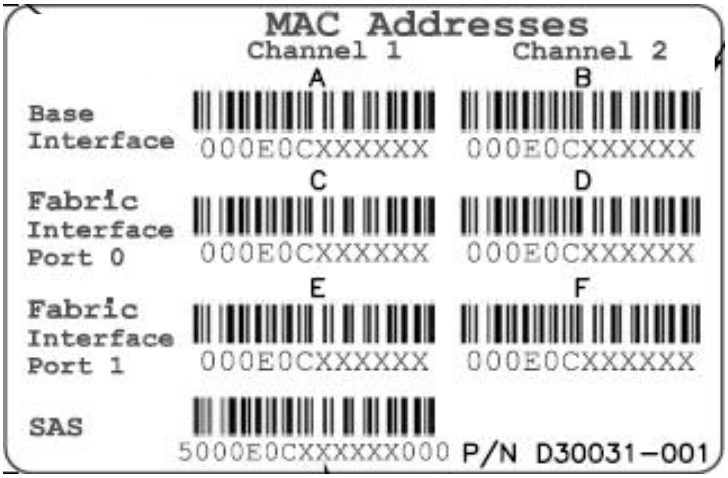


Table 1. Ethernet Port Mapping

BIOS PXE Boot Port	Identification on Multimedia Blade Faceplate	Operating System Port	MAC Address	PCI Express Port Connection to Memory Controller Hub (MCH)	PICMG 3.0 Port Definition
0200*	Port A	eth4	xxxxxxxxxxN	B1	Base Interface; Port 0, Channel 1
0201*	Port B	eth5	xxxxxxxxxxx (N+1)	B1	Base Interface; Port 0, Channel 2
0300*	Port C	eth0	xxxxxxxxxxx (N+2)	A0	Fabric Interface; Port 0, Channel 1 (Ports C and D can be jointly selected to connect to the backplane from the on-board Ethernet controller or GbE ports from the AdvancedMC slot.)
0301*	Port E	eth2	xxxxxxxxxxx (N+4)	A1	Fabric Interface; Port 1, Channel 1 (Ports E and F can be jointly selected to connect to backplane (default setting) or front panel RJ45.)
0400*	Port D	eth1	xxxxxxxxxxx (N+3)	A0	Fabric Interface; Port 0, Channel 2 (Ports C and D can be jointly selected to connect to the backplane from the on-board Ethernet controller or GbE ports from the AdvancedMC slot.)
0401*	Port F	eth3	xxxxxxxxxxx (N+5)	A1	Fabric Interface; Port 1, Channel 2 (Ports E and F can be jointly selected to connect to backplane (default setting) or front panel RJ45.)

* Slot xxyy refers to:

- xx = PCI bus number, for example 02 means bus 2.
- yy = Bits 7-3 is the PCI device number; bits 2-0 is the PCI function number. For example, 00 means device 0, function 0; 01 means device 0, function 1 etc.

Note: For the Ethernet operating system port, this is the default unless the default parameters are changed. For example, pci=xxx boot parameter.

3.3 Installing the Rear Transition Module (RTM)

Note: The RTM and Multimedia Blade are installed in corresponding slots in the rear and front of the chassis, respectively. When installing boards in a chassis with the power on, the RTM must be installed first, before the main board is installed.

The RTM has two ejector handles (top and bottom) to help insert the RTM into the chassis. Install the RTM as follows:

1. Install the RTM from the rear of the chassis. Open the ejector handles and push the RTM into the chassis card slot. Once the ejector handles slide into the top and bottom notches in the chassis, push both ejector handles toward the faceplate of the board until the handles click into place.
2. Tighten the RTM faceplate screws so the board is firmly seated in the chassis.

Continue with Section 3.4, Installing the Multimedia Blade.

3.4 Installing the Multimedia Blade

Install the Multimedia Blade as follows:

1. Make a note of the Multimedia Blade serial number, which is shown on a label on the box as well as on the board itself. The serial number begins with PJ followed by six digits (PJnnnnnn). The serial number will be needed for licensing.
2. Check DIP switches SW4 and SW5 to ensure that all switches are in the ON position. For location of SW4 and SW5, see Figure 9, “Backplane and On-board Connector and DIP Switch Locations” on page 66.
3. Install the Multimedia Blade into the front slot of the chassis corresponding to the rear slot where the RTM was installed. The Multimedia Blade has two ejector handles (top and bottom) to help insert the Multimedia Blade into the chassis. Open the ejector handles and push the board into the chassis card slot. Once the ejector handles slide into the top and bottom notches in the chassis, push both ejector handles toward the faceplate of the board until the handles click into place.

   The ejector handles provide a positive cam action, which ensures that the board is properly seated. Once the bottom ejector handle is closed, the Hot Swap switch is engaged and this starts the normal power on sequence.

For information about front panel LEDs, see Section 6.2.3, “Front Panel LEDs” on page 74.

Continue with Section 3.5, Connecting Serial Port to Terminal Emulator.

3.5 Connecting Serial Port to Terminal Emulator

For initial setup and configuration, connect the serial port to a terminal emulator. Console redirection to the front panel serial port is enabled by default in the BIOS. This setting redirects the text output of the BIOS and operating system to the RJ-45 serial port on the Multimedia Blade faceplate. The default settings are 9600, 8, n, 1 with no flow control.

For information about serial port connector pin assignments, see Section 6.2.1, “Front Panel Connectors” on page 70.

Continue with Section 3.6, Setting up the BIOS.

3.6 Setting up the BIOS

To access the BIOS Setup program, press the F4 key as soon as the BIOS starts booting.

Note: To use F4, the keyboard must be mapped accordingly.

A complete description of the BIOS Setup program is given in Chapter 9.0, "BIOS Settings". In most cases, the BIOS defaults provide the correct configuration for board use. However, it is necessary to set the following:

- On the **Main menu**, set the system time and system date.

Note: If the board is powered down or out of the chassis for greater than 2 hours, the system time and date revert to the factory defaults and will have to be reset. Ensure that the time and date are set before starting the Dialogic Services.

- On the **Advanced menu, IDE Configuration submenu**, disable IDE (default is P-ATA only).
- On the **Advanced menu, Ethernet Ports Direction Configuration submenu**, select Yes to change Ethernet Ports C&D Direction (default is No). The BIOS screen should look like this:

```
************************************************
* Ethernet Ports Direction Configuration Menu
* ********************************************
* Current Eth Ports E&F Direction :Front Panel
* Change Eth Ports E&F Direction [No]
*
* Current Eth Ports C&D Direction :Onboard GbEth
* Change Eth Ports C&D Direction [Yes]
*
```

- On the **Advanced menu, Remote Access Configuration Submenu**, change the Serial Port Mode settings from 9600,8,n,1 to 115200,8,n,1. (After setting the BIOS, change the baud rate in the terminal program to 115,200.)
- On the **Boot menu, Boot Device Priority submenu**, set the first boot device to the device where you will be installing the Linux operating system from (for example, USB disk or SCSI device).

Note: After setting the boot device, confirm that the setting took effect; reset it if necessary.

Continue with Section 3.7, Installing the Operating System.

3.7 Installing the Operating System

The Multimedia Blade and Multimedia Software can be used with the following operating systems:

- Red Hat* Enterprise Linux* Advanced Server Release 4.0 with Update 2 or Update 3
- Red Hat* Enterprise Linux* Enterprise Server Release 4.0 with Update 2
- SUSE Linux SLES* Release 9 SP3

Install the operating system per instructions provided by the operating system vendor, using all of the default package selections.

Note: There is no video card, so ensure that your installation method does not rely on one.

After installing the operating system, reboot.

Note: When using **Red Hat Linux**, the following is a list of services that should be enabled on the Multimedia Blade when performing the initial install. It is also recommended that the user disable SELinux.

- Acpid
- Arptable_jt
- Autofs
- Ct_intel
- Firstboot
- Haldaemon
- ipmi
- Irqpbalance
- Kudzu
- Lmsensors
- Mdmonitor
- Messagebus
- Microcode
- Netfs
- Network
- Portmap
- Raw devices
- Rhnsd
- Rpcgssd
- Rpcidmapd
- Smartd
- Sshd
- Syslog
- Xfs
- Xinetd
- Ypbind

Note: When using **SUSE Linux**, the SUSE Linux kernel **.config** file should be configured for an Intel Pentium 4 processor (at a minimum) as follows:

- CONFIG_M586 should be disabled and CONFIG_MPENTIUM4 should be enabled, depending on your processor.
- CONFIG_PREEMPT should be enabled.
- CONFIG_HPET_RTC_IRQ should be disabled.

Although the **.config** file can be manually configured, it is recommended that you use the provided configuration script from the website:
http://resource.intel.com/telecom/support/hmplinux/hmp12/suse/index.htm

Continue with Section 3.8, Updating the Ethernet Driver.

3.8 Updating the Ethernet Driver

The Ethernet driver that was delivered with the operating system does not support the network controller on the Multimedia Blade baseboard. To use the network, you need to install a replacement Ethernet driver version on the Multimedia Blade as follows:

1. Download the E1000 driver from the following website:
   http://developer.intel.com/design/telecom/support/
2. Untar the file.
3. Go to the e1000-7.2.9 directory and `cd` to the src directory.
4. Enter the following commands:
   ```
   make clean
   make install
   ```

Continue with Section 3.9, Obtaining a License File.

3.9 Obtaining a License File

Note: A verification license is provided with the Multimedia Software, activated and ready to use, but this license is only intended to be used for verifying that the software installation was successful.

You can obtain a license file either before or after you install the Multimedia Software, but you need to obtain a file before you can proceed with using the Multimedia Software. The license file contains authorization for a specified number of channels with or without multimedia processing:

- 250 channels voice, 250 channels multimedia
- 500 channels voice
- 500 channels voice, 250 channels multimedia

Contact your VAD to obtain a license file. You will need the board serial number as discussed in Section 3.4, “Installing the Multimedia Blade” on page 24.

Continue with Section 3.10, Installing the Multimedia Software.

3.10 Installing the Multimedia Software

To obtain the Multimedia Software, go to the following website:
http://developer.intel.com/design/telecom/support/

Follow the instructions/prompts for downloading the software. The Multimedia Software is provided in a tar file. After unpacking the tar file, proceed with the software installation as follows:

1. Ensure that you are logged in to the Linux system as **root**.
2. Enter the following command to start the install script:

```
./install.sh
```

Messages similar to the following are displayed:

```
Installing Dialogic(R) Multimedia Software Release for ATCA
Redistributable Runtime
Please wait, installation is in progress...

Do you wish to configure the SNMP Agent Listening Port (default is 161) [y/n] ?
```

3. To use the default SNMP Agent Listening port, type n. Otherwise, type y and respond to the prompt for a port number.

   Note: No range checking is done, so you are responsible for providing a usable address for the SNMP Agent Listening port if you are not using the default address.

   You are then prompted for the CLI Agent Telnet port:

```
Do you wish to configure the CLI Agent Telnet Port (default is 23) [y/n] ?
```

4. To use the default CLI Agent Telnet port, type n. Otherwise, type y and respond to the prompt for a port number.

   Note: No range checking is done, so you are responsible for providing a usable address for the CLI Agent Telnet port if you are not using the default address.

   Software installation is now complete, and the following messages are displayed:

```
Installation of Multimedia Software Release for ATCA successfully completed.

------------------------------------------------------------------------------
--                                                                          --
-- NOTE:                                                                    --
--                                                                          --
-- (1) The default Verification License has been automatically installed.   --
-- (2) The Dialogic(R) system services will automatically --
--     start every time the system is rebooted. To start and stop system    --
--     services manually, use the dlstop and dlstart scripts found in       --
--     /usr/dialogic/bin.                                                   --
-- (3) For administration via SNMP, use any SNMP browser, library or tools--
-- (4) For administration via CLI, use the telnet command                   --
-- (5) For details, refer to documentation                                  --
--                                                                          --
-- NEXT STEPS:                                                              --
-- You need to do the following before using the product:                   --
-- (1) Configure your network                                               --
-- (2) Configure an appropriate license (higher density) via SNMP or CLI    --
-- (3) Configure the Media Ethernet Interfaces via SNMP or CLI              --
-- (4) If you are using PSTN, refer documentation on how to configure       --
--     PSTN via SNMP or CLI                                                 --
-- (5) Before using the software, you must also ensure that the             --
--     environment variables are set by performing the following action:    --
--     - Logout and login                                                   --
--                                                                          --
------------------------------------------------------------------------------
```

5. Following the successful installation of the Multimedia Software, log out and log back in.

   The Dialogic services start automatically every time the system is rebooted. (To stop and start system services manually, use the dlstop and dlstart commands found in /usr/dialogic/bin.)

   A series of messages are displayed, as follows:

```
STARTING DIALOGIC SERVICES...

Starting ipmi_msghandler                                   [  OK  ]
Starting ipmi_si                                           [  OK  ]
Starting ipmi_devintf                                      [  OK  ]
Starting ipmi_watchdog                                     [  OK  ]
Starting tvl2_sysctl                                       [  OK  ]
Starting csme                                              [  OK  ]
Starting tvl2drv_load                                      [  OK  ]
Starting CORBA Event Server:                               [  OK  ]
Starting CORBA Name Server:                                [  OK  ]
Starting Error Logger :                                    [  OK  ]
Starting Device Mapper :                                   [  OK  ]
Starting Timeslot Doler :                                  [  OK  ]
```

```
Starting Board Manager Service :                          [  OK  ]
Starting SNMP Agent Service :                             [  OK  ]
Starting CLI Agent Service :                              [  OK  ]

STARTING DIALOGIC MEDIA SERVER

. . . . . .

Dialogic Media Server is ACTIVE

#
```

Continue with Section 3.11, Configuring the System.

3.11 Configuring the System

After installing the Multimedia Software, you should check configuration parameters and perform additional configuration as required. Configuration steps include:

- Configuring the network, e.g., Ethernet 4 if using Base interface (optional), Ethernet 6 if using Fabric/RTM (mandatory). This is done from the operating system, as follows:
  - For Red Hat Linux, use the `/usr/sbin/netconfig` program to configure a static or dynamic IP address.
  - For SUSE Linux, use the `/sbin/yast` program to configure a static or dynamic IP address.
- Activating the license (mandatory). For example, to activate the license by using the Command Line Interface, see Section 4.6, “Activating a License” on page 41.
- Configuring the Media Ethernet Interfaces, i.e., the four IP addresses for the AdvancedMC DSPs (mandatory). For example, to configure the Media Ethernet Interfaces using the Command Line Interface, see Section 4.7, “Configuring the Media DSP IP Addresses” on page 42.
- Configuring the PSTN if applicable, e.g., trunk configuration, clock fallback, etc. For example, to configure the PSTN trunks and clock fallback using the Command Line Interface, see Section 4.8, “Configuring PSTN Trunks and Clock Fallback” on page 43.

There is a choice of two interfaces for configuring and monitoring the Multimedia Blade: Command Line Interface (CLI) and Simple Network Management Protocol (SNMP). They both have access to the same configuration and monitoring data. SNMP is MIB-based, and CLI is text command line-based.

For detailed information about CLI, see Chapter 4.0, “Administration via Command Line Interface (CLI).” For detailed information about SNMP, see Chapter 5.0, “Administration via SNMP.”

3.12 Subnet Connectivity

Media data flow in the system is designed to travel over the eth6/eth7 interfaces, and network routing is used to ensure correct operation. Routes must be configured such that media processing elements and gateways to media endpoints are correctly routed over the eth6/eth7 interfaces.

The system can be configured for:

- Single subnet; all traffic (control and media) travels over single RTM or Fabric interface (simplest configuration). See Figure 4.

- Single subnet with two interfaces; the default route is eth4/eth5, and explicit routes are added to force access to gateway over eth6/7. See Figure 5. The commands to set up this configuration (from the operating system shell) are:

```
route add -net 192.168.0.0 netmask 255.255.255.0 eth4
route add -net 192.168.0.0 netmask 255.255.255.0 eth6
route add default gw 192.168.0.1 eth4
# routes to media processors
route add -host 192.168.0.20 eth6
route add -host 192.168.0.21 eth6
route add -host 192.168.0.22 eth6
route add -host 192.168.0.23 eth6
# add routes to media endpoint gateway via media interfaces
route add -host 192.168.0.100 eth6
route add -network 192.168.3.0 netmask 255.255.255.0 gw
192.168.0.100
```

- Single subnet with two interfaces; the default route is eth6/eth7, and explicit routes are added to force access to host over eth4/5. See Figure 6. The commands to set up this configuration (from the operating system shell) are:

```
route add -net 192.168.0.0 netmask 255.255.255.0 eth4
route add -net 192.168.0.0 netmask 255.255.255.0 eth6
# default gateway is 192.168.0.100 through eth6
route add default gw 192.168.0.100 eth6
# route to NFS server via eth4
route add -host 192.168.0.4 eth4
```

- Two subnets; media traffic travels over RTM or Fabric interface, and other traffic travels over Base interface. See Figure 7.

Figure 4. Single Subnet, Single Network Interface Configuration

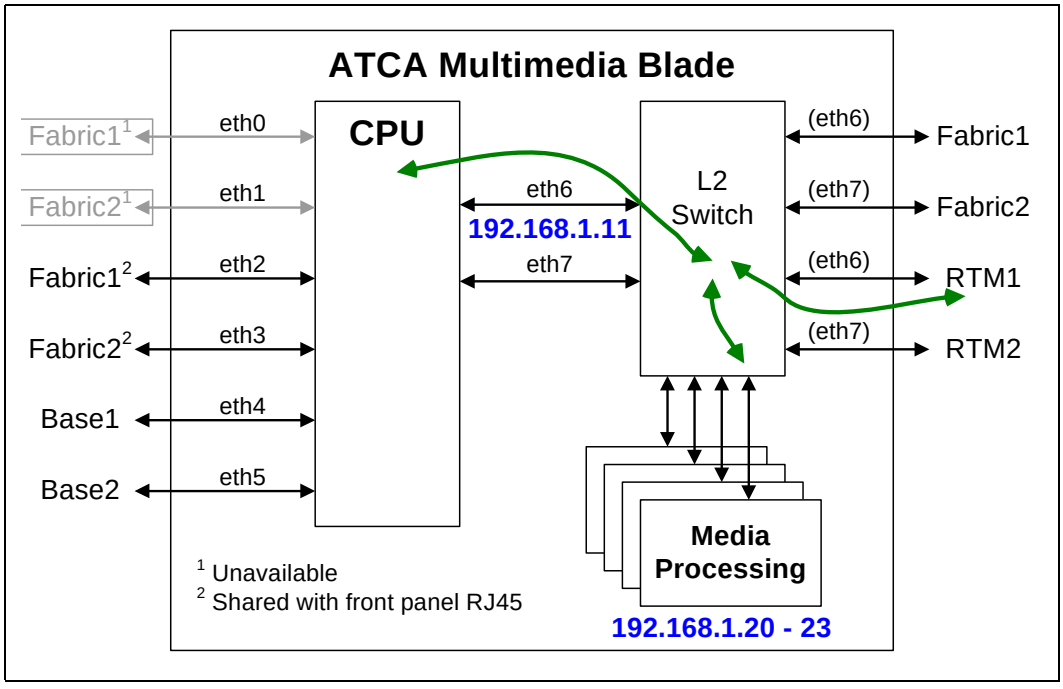


Figure 5. Single Subnet, Two Interfaces, Default Route eth4 / eth5

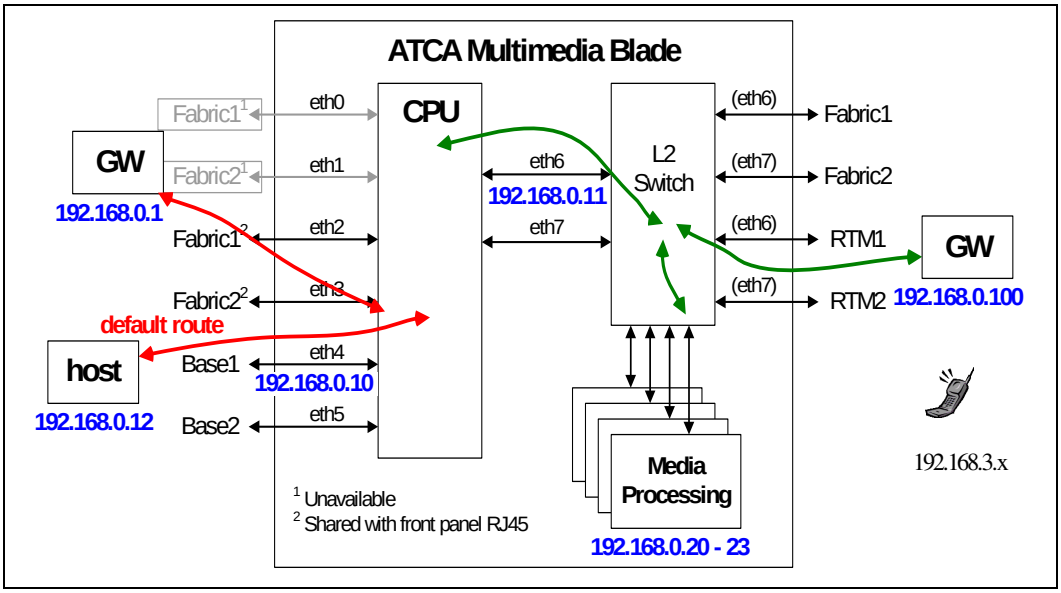


Figure 6. Single Subnet, Two Interfaces, Default Route eth6 / eth7

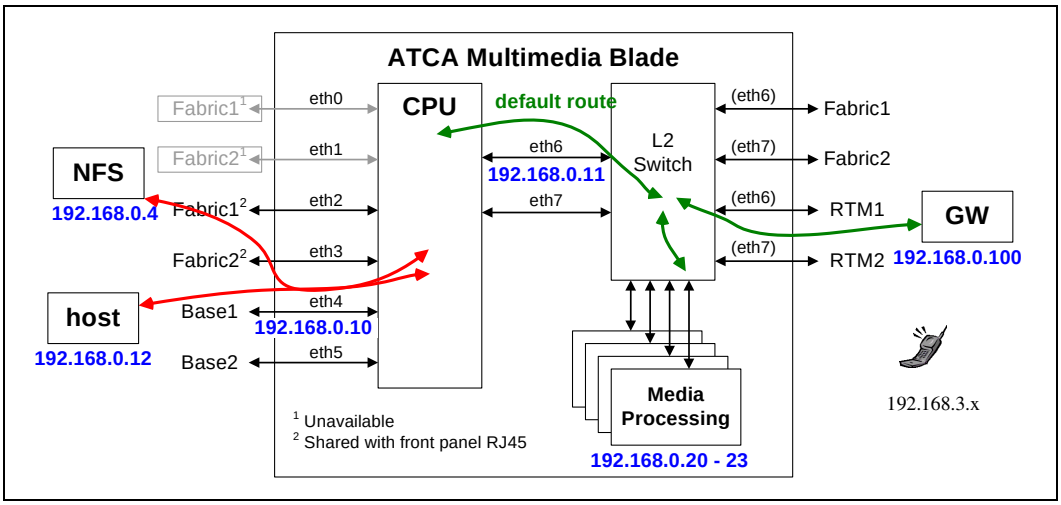
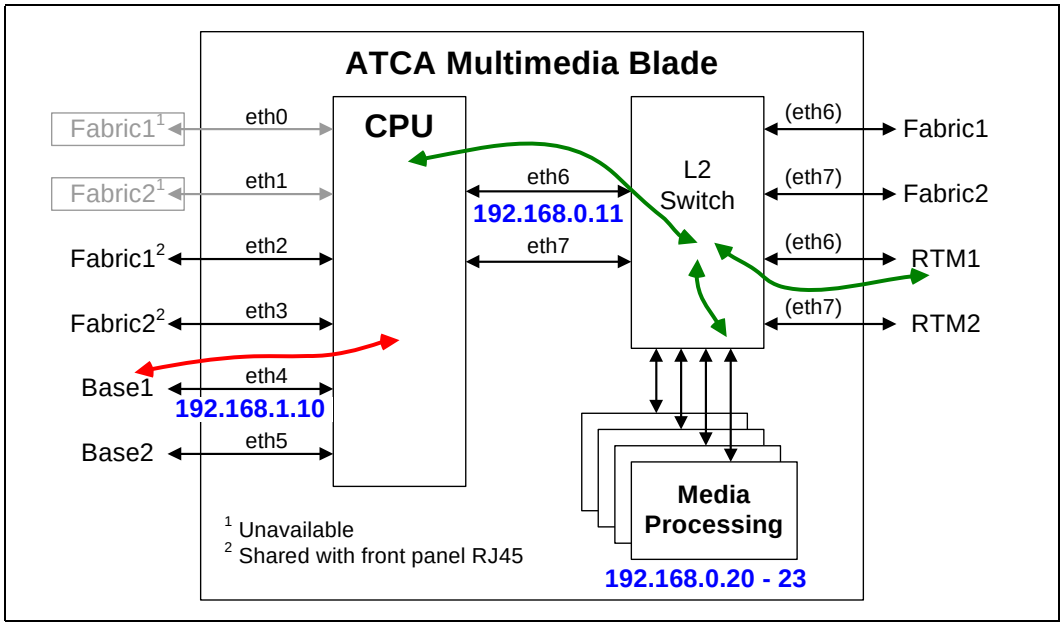


Figure 7. Two Subnets



3.13 Verifying the System

If you would like to verify that you have set up the system properly, you can run any of the demos that are provided with the Multimedia Software. This step is optional, but recommended. See the directory /usr/dialogic/demos for demos.

Note: Only the Multimedia demo (/usr/dialogic/demos/MultiMedia/MultiMediaDemo/Release) has been customized to work on the ATCA Multimedia Platform. The other demos in /usr/dialogic/demos will require modification.

4.0 Administration via Command Line Interface (CLI)

This chapter provides information about using the Command Line Interface (CLI) software for local or remote configuration and monitoring of the Dialogic® Multimedia Blade for AdvancedTCA® (hereinafter referred to as the Multimedia Blade). It gives general information about the CLI as well as some commonly used procedures. The following topics are discussed:

- CLI Overview
- CLI Account Management Commands
- CLI Show Command
- CLI Conf Command
- Stopping and Restarting the Media Service
- Activating a License
- Configuring the Media DSP IP Addresses
- Configuring PSTN Trunks and Clock Fallback
- Stopping and Starting the CLI Agent
- Error Handling and Logging
- CLI Command Quick Reference

4.1 CLI Overview

CLI is one of two interfaces that can be used to configure and monitor the Multimedia Blade. The other is Simple Network Management Protocol (SNMP). They both have access to the same configuration and monitoring data. SNMP is MIB-based, and CLI is text command line-based. For information about using SNMP for configuration management, see Chapter 5.0, "Administration via SNMP."

When the CLI is running, events are displayed on the CLI terminal.

4.1.1 Access to the CLI

The CLI agent provides multi-user access for remote and/or local configuration management and monitoring of the Multimedia Blade. The CLI agent is included with the Dialogic® Multimedia Software Release for AdvancedTCA® (hereinafter called Multimedia Software) and is automatically started when the Multimedia Software is started using a system reboot or the **dlstart** command.

Up to three users can access the CLI at the same time. Each login session is controlled by a session timer, so when users forget to log off the Multimedia Blade, CLI will do it for them.

The CLI agent is a server process that monitors Telnet port 23 by default. Users can access the CLI agent using the **telnet** command from any system supporting the Telnet protocol. For example:

```
> telnet 192.168.12.34
```

If you are already logged on to the Multimedia Blade, you can use the loopback IP address to access the CLI:

```
> telnet 127.0.0.1
```

Once you successfully Telnet to the Multimedia Blade, you will be presented with a banner, followed by a login prompt.

```
+-------------------------------------------------------+
|                                                       |
|  .:::::::::::::::::::::::::::::::::::::::::::::::.   |
|   ::: Dialogic® ATCA Multimedia Platform        :::   |
|  `:::::::::::::::::::::::::::::::::::::::::::::::'   |
|                                                       |
|                                                       |
| For HELP:                                             |
|   Use '?' at command prompt                           |
|   Use '-h' within commands                            |
|                                                       |
+-------------------------------------------------------+

Login :
```

After you successfully log in, you can use CLI commands to view and modify Multimedia Blade settings and parameters, depending on your access level. The CLI command prompt is:

```
CLI>
```

4.1.2 Default Usernames and Passwords for CLI Users

CLI supports three levels of users. Each level has different capabilities and command access within the CLI; see Table 2.

Table 2. CLI Usernames and Passwords

Role	Username	Password	Description
Account manager	acctmgr	acctmgr	Allowed to add, delete, and update all user information.
Craftsperson	craft	craft	Allowed to read blade data except account manager specific data.
Administrator	admin	admin	Allowed to read and write blade data except account manager specific data.

4.1.3 CLI Commands by User Role

The CLI consists of basic commands used to configure and monitor the Multimedia Blade, for example:

- **show** - To view blade information
- **conf** - To configure blade information

The basic commands are then followed by one or more subcommands that describe the type of data or specific data item you wish to view or modify. Table 3 lists the available commands as they apply to a user's role.

Table 3. CLI Users and Command Privileges

Command	Account Manager	Craftsperson	Administrator	Description
adduser	Yes	No	No	Add users to the system
clear	Yes	Yes	Yes	Clear display
conf	No	No	Yes	Change blade parameters
deleteuser	Yes	No	No	Delete users from system
editpword	Yes	Yes	Yes	Change your password
edituser	Yes	No	No	Modify user account data
keepalive	Yes	Yes	Yes	Specify number of seconds to wait between CLI commands before auto logout for this session
list	Yes	No	No	List CLI user accounts
logout	Yes	Yes	Yes	Log off
man	Yes	Yes	Yes	Help
prompt	Yes	Yes	Yes	Change the display prompt for the current session
reload	No	No	Yes	Reboot the blade
restart	No	No	Yes	Reboot the blade (alias for reload)
show	Yes	Yes	Yes	Show blade parameters and statistics
who	Yes	Yes	Yes	Display users currently logged on to CLI agent

4.1.4 CLI Help

The CLI agent provides context-level help at any point during command input by typing **-h** followed by a carriage return. For example:

```
CLI> show -h
Usage:show
    arp         ARP table
    controller  Controller status and configuration
                 <ds1>
    hardware    Hardware specific information
                 <amc> <rtm>
    icmp        ICMP information
    interface   Interface status and configuration
                 <desc> <ds1> <ethernet> <media>
    ip          IP information
                 <addr> <arp> <traffic> <interface>
    license     Media license information
    msml        Remote media server settings
                 <access-list>
    system      System information
    tcp         Status of TCP connections
                <brief> <statistics>
    udp         Status of UDP connections
                <brief> <statistics>
    version     System version information
```

For the list of available commands based on your login permission, enter **?** at the command prompt. For example:

```
Login :admin
Password :*****
CLI> ?
clear           conf            editpword       logout          man
prompt          reload          restart          show           who
```

The keyboard up/down arrows allow you to scroll through the input history buffer, and thus repeat the last line(s) you typed. Using the left and right arrow keys, you can edit the input line for updated values without having to retype the entire line.

4.2 CLI Account Management Commands

This section covers the CLI account management commands used to provision and maintain CLI users and permissions.

As stated earlier, there are three levels of users in the CLI. The account manager user is responsible for provisioning and managing all other users with the CLI. The account manager user can add new users, delete users, update user passwords or context (user role), and set user aging parameters. Aging parameters are used to automatically remove passwords and user accounts after a configurable time period and are provided to increase system security.

The commands to create, modify, and delete users have the form:

[**add**/**edit**/**delete**]**user** [*username*] <*parameters*>

where the command prefix is one of **add**, **edit**, or **delete**; *username* identifies the user; and the *parameters* define the managed user and optional settings. The *username* and *parameters* can be entered in any order.

Table 4 lists the account management commands.

Table 4. Account Management Commands

Command	Parameters	Description	Examples
adduser	[-g <password aging>] [-a <account aging>] [-i <context>] username password	Add user to the system. Default password aging: 60 Default account aging: 180 Contexts are ACCTMGRCTX for account manager, CRAFTCTX for craftsperson, ADMINCTX for administrator.	CLI> adduser -i ADMINCTX foo bar • Creates user “foo” with password “bar” and permissions “ADMINCTX”
deleteuser	username	Deletes user “username” account	CLI> deleteuser foo • Deletes user “foo”
editpword	password	Change current user password.	CLI> editpword foo foobar • Updates current user password to “foobar”
edituser	[-g <password aging>] [-a <account aging>] [-i <context>] [-p <new password>] username	Edits account information for the specified user.	CLI> edituser foo -p bar • Changes user “foo” password to “bar”

4.3 CLI Show Command

This section covers the CLI **show** commands used to view Multimedia Blade data.

The **show** command is used to view any/all configuration parameters and statistics. It has the form:

show <*subcommand*>

where *subcommand* can be a terminating command that results in data output, such as **show system**, or a submenu category such as **show tcp**, which requires additional arguments before output is displayed (for example, **show tcp brief**).

The initial command and subcommand are case sensitive and require complete command entry, for example:

```
CLI> show interface
```

Parameters following the command and subcommand will match to the shortest unique command. For instance, the following are legal inputs to the **interface** subcommand:

```
CLI> show interface e
CLI> show interface Ethernet
CLI> show interface de
CLI> shoe interface ds
```

If the user enters a non-unique parameter, a help message is displayed with the list of legal parameters. For example:

```
CLI> show interface d
ambiguous option - expected
       desc      Display link/administration status for interfaces
       ds1       DS1 interface information
       ethernet  Ethernet interface information
       media-ethernet  Media ethernet interface information
       <cr>      Dump all interface information
```

Table 5 lists frequently used **show** commands. (See also Section 4.11, “CLI Command Quick Reference” on page 45.)

Table 5. Show Commands

Subcommand	Description	Additional Subcommands/Parameters
arp	Shows the ARP table	None
controller	Shows information on ds1 interfaces (alias for “interface” but only for ds1)	**ds1** || <*type id*> || <cr> For example: CLI> show controller CLI> show controller ds1 0
hardware	Shows hardware information	<*type*> || <*type id*> || <cr> For example: CLI> show hardware CLI> show hardware amc CLI> show hardware amc 1 CLI> show hardware rtm
interface	Shows information on interfaces	<**desc**> || <*type*> || <*type id*> || <cr> For example: CLI> show interface ethernet CLI> show interface eth 0 CLI> show interface desc
icmp	Shows ICMP statistics	None

Table 5. Show Commands (Continued)

Subcommand	Description	Additional Subcommands/Parameters
ip	Shows configuration/statistics on IP interfaces	<**interface**> || <**interface** *type id*> || <**traffic**> || <**addr**> || <**arp**> || <**route**> For example: CLI> show ip interface ethernet 0 CLI> show ip address
license	Shows license information	None
msml	Shows remote media control settings	<**access-list**> || <cr> For example: CLI> show msml CLI> show msml access-list
system	Shows system information	None
tcp	Shows TCP connection information	<**brief**> || <**statistics**> For example: CLI> show tcp brief CLI> show tcp statistics
udp	Shows UDP connection information	<**brief**> || <**statistics**> For example: CLI> show udp brief CLI> show udp statistics
version	Shows system version information	None

4.4 CLI Conf Command

This section covers the CLI **conf** commands used to configure Multimedia Blade data.

The **conf** command is used to update configuration parameters. It has the general form:

conf <*subcommand*> <*parameter*> <*value*>

where *subcommand* identifies the group being configured, such as **interface** or **system**. Following the command and subcommand are a simple parameter/value pair or additional subcategories followed by a parameter/value setting.

For example, the following is a simple parameter/value setting that updates the system group, field name, to “My NetMedia Box”:

```
CLI> conf system name 'My NetMedia Box'
```

The following is a more complex example updating a media interface 0, sub category ip, key field address 192.168.1.2, parameter netmask to value 255.255.255.0:

```
CLI> conf interface media 0 ip address 192.168.1.2 netmask 255.255.255.0
```

The same rules for the **show** command entry also apply to the **conf** command. The command and subcommand are case sensitive and must be entered as shown. Subcategories and parameters are case insensitive and match on the shortest, most unique input.

Table 6 lists frequently used **conf** commands. (See also Section 4.11, "CLI Command Quick Reference" on page 45.)

Table 6. Conf Commands

Subcommand	Description	Additional Subcommands / Parameters
controller	Configures ds1 interface related fields	<**ds1** *n*> <*parameter field*> <*value*> || <**adminstatus** *value*> <**trap** *value*> For example: CLI> conf controller ds1 0 adminstatus up CLI> conf controller ds1 0 trap CLI> conf controller ds1 0 no trap CLI> conf controller ds1 0 linecode ami
interface	Configures interface related fields	<**ethernet** *n*> <**ip** *parameter value*> || <**adminstatus** *value*> <**trap** *value*> For example: CLI> conf interface ethernet 0 adminstatus up CLI> conf interface ethernet 0 no trap CLI> conf interface media 0 ip address 210.123.66.43 CLI> conf interface med 0 ip address 210.123.44.42 addr 214.237.5.32
ip	Configures IP related fields	<**ttl**> For example: CLI> conf ip ttl 300
msml	Configures remote media control fields	<**access-list** *ipaddr status*> <**sip-addr** *value*> || <**sip-port** *value*> <**protocol** *value*> For example: CLI> conf msml enable CLI> conf msml sip-a 1.2.3.4 CLI> conf msml sip-p 23342 CLI> conf msml access 2.3.4.5 CLI> conf msml access no 2.3.4.5
system	Configures MIB II system parameters	<**contact**> || <**location**> || <**name**> || <**ipmedia**> For example: CLI> conf system name 'IPMS Media Blade' CLI> conf system ipmedia start

4.5 Stopping and Restarting the Media Service

This section covers how to stop/start the Media Service.

4.5.1 Stopping the Media Service

In order to stop the Media Service, the Media Service must be running. This can be found by retrieving the system run status found in the system group.

To stop the Media Service, use the **conf** command to update the system ipmedia setting to stop. Starting or stopping the media results in a number of events being generated by the system and displayed to any user logged on the CLI. These events are also translated into SNMP traps and can be monitored using an SNMP browser.

The following demonstrates stopping the Media Service. The **show system** command is entered first, which shows the current run status as Active. After **conf system ip stop** is entered, shutdown event notifications are displayed.

```
CLI> show system
IPMS ATCA Blade
```

```
Name:          System Name Not Defined
Location:     System Location Not Defined
Contact:       System Contact Not Defined

System is ready
Run Status:             Active
Media Support:      Video Option not Support with this Product.
Media processing last request:   Start
PSTN Trunk Limit: 4
Uptime:     55 second(s)
Last Boot Reason: Power on
Services:    7
Form Factor: ATCA
Model:       IPMS-MDL-707
Vendor:      Dialogic Corp,  Serial Number: SN-123456789
CLI> conf system ip stop
updated
CLI>
EVENT-- SYSTEM Run State [Shutdown]
CLI>
EVENT-- SYSTEM Run State [Inactive]
CLI>
EVENT-- SYSTEM [not ready]
```

4.5.2 Restarting the Media Service

In order to restart the Media Service, the Media Service must be stopped. This can be found by retrieving the system run status found in the system group.

To start the Media Service, use the **conf** command to update the system ipmedia setting to start. Starting or stopping the media results in a number of events being generated by the system and displayed to any user logged on the CLI. These events are also translated into SNMP traps and can be monitored via an SNMP browser.

The following demonstrates starting the Media Service. The **show system** command is entered first, which shows the current run status as Inactive. After **conf system ip start** is entered, startup event notifications are displayed.

```
CLI> show system
IPMS ATCA Blade
Name:          System Name Not Defined
Location:     System Location Not Defined
Contact:       System Contact Not Defined

System is ready
Run Status:             Inactive
Media Support:      Video Option not Support with this Product.
Media processing last request:   Start
PSTN Trunk Limit: 4
Uptime:     55 second(s)
Last Boot Reason: Power on
Services:    7
Form Factor: ATCA
Model:       IPMS-MDL-707
Vendor:      Dialogic Corp,  Serial Number: SN-123456789
CLI> conf system ip start
updated
CLI>
EVENT-- SYSTEM Run State [Start]
CLI>
EVENT-- SYSTEM Run State [Initializing]
CLI>
EVENT-- AMC Bay(1) Run State [Initializing]
CLI>
EVENT-- RTM Amc Bay(9) Run State [Initializing]
```

```
CLI>
EVENT-- AMC Bay(1) Run State [Active]
CLI>
EVENT-- RTM Amc Bay(9) Run State [Active]
CLI>
EVENT-- SYSTEM Run State [Active]
CLI>
EVENT-- SYSTEM [ready]
```

4.6 Activating a License

This section shows how to use the CLI to activate a license. You must be logged in to the CLI as administrator to activate a license.

You can activate the verification license supplied with the software or another license you have obtained. See Section 3.9, “Obtaining a License File” on page 27.

1. At the CLI prompt, enter the **conf license directory** command and specify the path to the license file (the directory where you downloaded the license). For example:

```
CLI> conf license directory /testing/license
updated
CLI>
```

2. At the CLI prompt, enter the **conf license activate** command and specify the license file name. The license will be activated on the next ipmedia restart. For example:

```
CLI> conf license activate tvl2_atca_mm_license.xml
updated   - Activates on next ipmedia restart.
CLI>
```

3. To complete the license activation, log out of the CLI, then enter the **dlstop** and **dlstart** commands. The **dlstop** command stops the Dialogic services, and the **dlstart** command starts the Dialogic services. (In the example below, complete output from the **dlstop** and **dlstart** commands is not shown.)

```
CLI> logout
Connection closed by foreign host.
# dlstop

STOPPING DIALOGIC SERVICES...
.
.
.

# dlstart

STARTING DIALOGIC SERVICES...
.
.
.

STARTING DIALOGIC MEDIA SERVER

. . . . . .

Dialogic Media Server is ACTIVE
```

4.7 Configuring the Media DSP IP Addresses

This section explains how to modify the default Media DSP IP addresses. The AdvancedMC, which is pre-installed on the Multimedia Blade baseboard, has four digital signal processors (DSPs) used for echo cancellation and transcoding offload. See Figure 1, “Multimedia Blade Block Diagram” on page 16.

The default IP addresses assigned to the DSPs are:

```
"192.168.0.20",
"192.168.0.21",
"192.168.0.22",
"192.168.0.23"
```

These addresses need to be modified to fit the end user’s network layout. The new addresses should be in the same subnet as the AdvancedMC Ethernet interface; this will be OS interface eth6 or CLI interface eth6.

To view the current settings for the media interfaces use the **show** command. There are four media DSP interfaces that need IP address configuration within your specific network. They are interface media 0 through media 4:

```
CLI> show interface media

Media Interface(s):
medeth0 is administratively up, current state is up
External interface:
Type: Amc-4, Ifindex: 20
Description:media-1
Interface trap unknown Last change: 0 seconds
Hardware address: 00:0E:0C:AE:98:11
Ip Address: 192.168.0.23
    Netmask: 0.0.0.0, Broadcast Bit: 0, Reasemble Max: 0
MTU 0 bytes, Speed 0
Rcvd: 0 octets
    0 subnetwork unicast packets
    0 subnetwork broadcast/multicast packets
    0 discarded packets
    0 error packets
    0 unknown/unsupported protocol packets
Sent: 0 octets
    0 subnetwork unicast packets
    0 subnetwork broadcast/multicast packets
    0 discarded packets
    0 error packets
    0 packet queue length

medeth1 is administratively up, current state is up
External interface:
Type: Amc-3, Ifindex: 21
Description:media-2
Interface trap unknown Last change: 0 seconds
Hardware address: 00:0E:0C:AE:98:10
Ip Address: 192.168.0.22
    Netmask: 0.0.0.0, Broadcast Bit: 0, Reasemble Max: 0
MTU 0 bytes, Speed 0
Rcvd: 0 octets
    0 subnetwork unicast packets
    0 subnetwork broadcast/multicast packets
…
…
…
```

To update the four addresses, use the CLI **conf interface** command, specifying the current IP address and the new IP address. The command format is:

conf interface media <*id*> **ip address** <*current ip address*> **addr** <*new ip address*>

For example:

```
CLI> conf interface media 0 ip address 192.168.0.23 addr 212.799.43.08
Updated
```

By using the up arrow, you can repeat the previous command. Using the left/right arrows allows you to edit the command line and easily complete the CLI entry for the next media DSP interface.

4.8 Configuring PSTN Trunks and Clock Fallback

The Multimedia Blade Rear Transition Module provides 16 E1/T1 trunks. This section provides the following procedures:

- Configuring Trunk Parameters
- Setting the Clock Fallback List

4.8.1 Configuring Trunk Parameters

1. Stop the media subsystem:

```
CLI> conf system ipmedia stop
```

2. Configure the system for the total number of trunks to use (must be a multiple of 4; 16 is the maximum):

```
CLI> conf system pstn-trunks 16
```

Note: From this point on, each trunk must be configured on a per trunk basis; ranges are not allowed. Trunk 1 is shown in this example. These steps must be repeated for all trunks.

3. Set the trunk out of service:

```
CLI> conf interface ds1 1 adminstatus down
```

4. Set the line type:

```
CLI> conf interface ds1 1 linetype <supported line type>
```

Note: For certain trunk parameters, the PSTN trunks are segmented into groups with each group consisting of 4 trunks. The same line type must be selected for trunks within the same group even though the line type is set individually for each trunk. For example, PSTN trunks 1-4 must be set to the same line type. Likewise, trunks 5-8 must have the same line type but it may differ from the line type selected for trunks 1-4.

5. Set the line code (e.g., b8zs, hdb3, ami):

```
CLI> conf interface ds1 1 linecode <supported line coding ** must be compatible
with line type>
```

Note: The line code can be set before the line type. The line code and line type parameters must be compatible; otherwise the system will return an error when the configuration is applied.

6. Set other trunk parameters as needed.
   To display trunk parameters, use the **show** command:

```
CLI> show interface ds1 1
```

Note: All parameters must be set before the line is put back into service.

7. Set the trunk back to in-service state to have new configuration validated and applied if there are no errors.

```
CLI> conf interface ds1 1 adminstatus up
```

8. Start the media subsystem:

```
CLI> conf system ipmedia start
```

4.8.2 Setting the Clock Fallback List

The clock fallback list specifies the priority of the clock sources. The board's E1/T1 trunks and its internal oscillator can be specified as clock sources.

The clock source and its priority in the fallback list are specified in the same command:

conf interface ds1 clock fallback-list <clock source> <priority>

For example, the following commands specify trunks 1, 5, 9, 13, and the internal oscillator as priorities 1-5 on the clock fallback list:

```
CLI> conf interface ds1 clock fallback-list 1 1
CLI> conf interface ds1 clock fallback-list 5 2
CLI> conf interface ds1 clock fallback-list 9 3
CLI> conf interface ds1 clock fallback-list 13 4
CLI> conf interface ds1 clock fallback-list internal 5
```

4.9 Stopping and Starting the CLI Agent

This section covers how to stop and start the CLI agent independent from the media application software. It also shows how to change the CLI listening port.

4.9.1 Stopping the CLI Agent

The CLI and SNMP agent software is automatically started at system reboot and when the media application software suite is started using **dlstart**. It is automatically stopped when the media application suite is terminated using **dlstop**. In order for the CLI to function, the Multimedia Software must be running, because the CLI has dependencies on back-end processes for data storage and retrieval. However, you can bring the CLI and SNMP agents up and down as often as needed while the Multimedia Software is up. (The SNMP agent software is discussed in Chapter 5.0, "Administration via SNMP.")

The CLI can be stopped using the **dlservices** command:

```
# dlservices cli stop
Shutting CLI Agent Service : [  OK  ]
```

At this point, users will no longer be able to Telnet to the Multimedia Blade and monitor or provision the blade using CLI.

4.9.2 Starting the CLI Agent

To restart the CLI agent, use the **dlservices** command:

```
# dlservices cli start
Starting CLI Agent Service : [  OK  ]
```

At this point, CLI is monitoring Telnet port 23 and waiting for user login requests again.

4.9.3 Changing the CLI Listening Port

The default CLI listening port is 23. It can be changed during the software installation process as explained in Section 3.10, “Installing the Multimedia Software” on page 27. To change it at any other time, use **dlservices**. For example:

```
# dlservices cli port 2323
```

This changes the CLI listening port to the specified port, 2323 in this example. Port changes take effect on the next **dlstart** or **dlservices cli start**.

Note: No range checking is done, so you are responsible for providing a usable address for the CLI listening port if you are not using the default address.

4.10 Error Handling and Logging

This section covers CLI error handling and logging.

4.10.1 CLI Errors

The CLI displays error messages for invalid inputs on the terminal screen. The error messages typically include a list of valid inputs for the command. For example:

```
CLI> show interface med 12

Media Interface(s):
unknown interface id - expected value
    <0 - 3>          Media Ethernet Interfaces
    <cr>              show all interface data

CLI> conf interface med 0 ip addr 192.168.99.90 addr a.b.c.d
bad value - expected
    A.B.C.D           Dotted ip address
CLI> Connection closed by foreign host.
```

4.10.2 CLI Logging

The CLI agent logs all update attempts per user and Error, Warning, and Info messages to a log file. The log file can be found:

```
/var/log/ngmit.log
```

The file contains all logging information related to provisioning, monitoring, and starting the Multimedia Services. All messages are prefixed with the date, time, machine name, and application. For CLI, the application is 'CLIAGENT'. The following is an example of a log message for the invalid IP address update:

```
Jun  6 18:30:43 mymachine CLIAGENT: From Component: IsIPV4DottedIp : Invalid Ip
Address format[Ip Address contains non digits]
```

4.11 CLI Command Quick Reference

This section shows the CLI commands and expected input.

4.11.1 Show

- arp
- icmp

- system
- tcp
  - — brief
  - — statistics
- udp
  - — brief
  - — statistics
- interface
  - — ethernet <*id*>
  - — media <*id*>
  - — desc
  - — ds1 <*id*>
  - — ds1 clock
- ip
  - — addr
  - — arp
  - — route
  - — traffic
  - — interface
    ethernet <*id*>
    media <*id*>
    desc
- hardware
  - — amc <*id*>
  - — rtm
- license
- msml
  - — access-list
- version

4.11.2 Conf

- system
  - — name *value* (string)
  - — location *value* (string)
  - — contact *value* (string)
  - — ipmedia *value* (start, stop)
  - — pstn-trunks *value* (0,4,8,12,16)
- ip
  - — ttl *value* (integer)

- interface
  - — ethernet id
    adminstatus *value* (up/down/testing)
  - — trap (<cr> enables)
  - — no
    trap (<cr> disables)
  - — ip
    address *value* <secondary> (ipaddress)
    address *value value* <secondary> (ipaddress netmask)
    address no *value* (ipaddress)
    address *value1* address *value2* (change ipaddr1 to ipaddr 2)
    address *value* netmask *value* (ipaddress netmask)
    address *value* broadcast *value* (ipaddress 0/1)
  - — ds1 id
    adminstatus *value* (up/down/testing)
    alarmtrap (<cr> enables)
    circuitid *value* (string)
    linecode *value* (b8zs, hdb3, ami)
    linelength *value* (integer - in meters)
    linetype *value* (esf, d4, e1, e1crc, e1mf, e1ctcmf)
    loopbackconfig *value* (inward, line, noloop)
    trap (<cr> enables)
    no
      - trap (<cr> disables)
      - alarmtrap (<cr> disables)
  - — ds1 clock
    fallback-list
      - *value value* (internal/ds1*id*, priority - inserts)
      - no *value* (internal/ds1*id* - deletes)
  - — media id
    trap (<cr> enables)
    no
      - trap (<cr> disables)

    ip
      - address *value* (ipaddress)
      - address no *value* (ipaddress)
      - address *value1* address *value2* (change ipaddr1 to ipaddr2)
- license
  - — *directory value* (text - path to license file)
  - — activate *value* (text - license file name)
  - — oem-function *value* (text - function name)
  - — oem-library *value* (text - library name)
- msml
  - — protocol (<cr> enables)
  - — ip *value* (ipaddress)
  - — port *value* (port)

— transport *value* (udp, tcp-udp)

— no
access-list (<cr> disables)
protocol (<cr> disables)

— access-list (<cr> enables)
value (ipaddress - inserts)
no

- *value* (ipaddress - deletes)

— http-caching

— schema-validation

5.0 Administration via SNMP

This chapter provides information about using the Simple Network Management Protocol (SNMP) agent software for local or remote monitoring and limited configuration of the Dialogic® Multimedia Blade for AdvancedTCA® (hereinafter referred to as the Multimedia Blade). It is assumed that the reader of this chapter has SNMP knowledge.

The first part of this chapter gives an overview and discusses some of the tasks that can be performed using SNMP. The following topics are discussed:

- SNMP Administration Overview
- System Requirements
- Stopping and Starting the SNMP Agent
- Username and Password
- Trap Configuration
- Stopping and Restarting the Media Service
- Media Ethernet Interface Configuration
- Resolving an Application Failure

Following this is reference information about the MIBs:

- Public MIB 2 Module Reference
- Public DS1 Module Reference
- Private System Environment Module Reference
- Private Hardware Module Reference
- Private External Interfaces Module Reference
- Private TDM Clock Module Reference
- Private Media Control Protocol Module Reference

5.1 SNMP Administration Overview

This section describes the administration tasks that can be performed using the SNMP agent software with the Dialogic® Multimedia Software Release for AdvancedTCA® (hereinafter referred to as Multimedia Software). It also describes the various Management Information Bases (MIBs) that can be used with this software, details the Multimedia Software agent framework, and discusses security authentications, such as username and passwords.

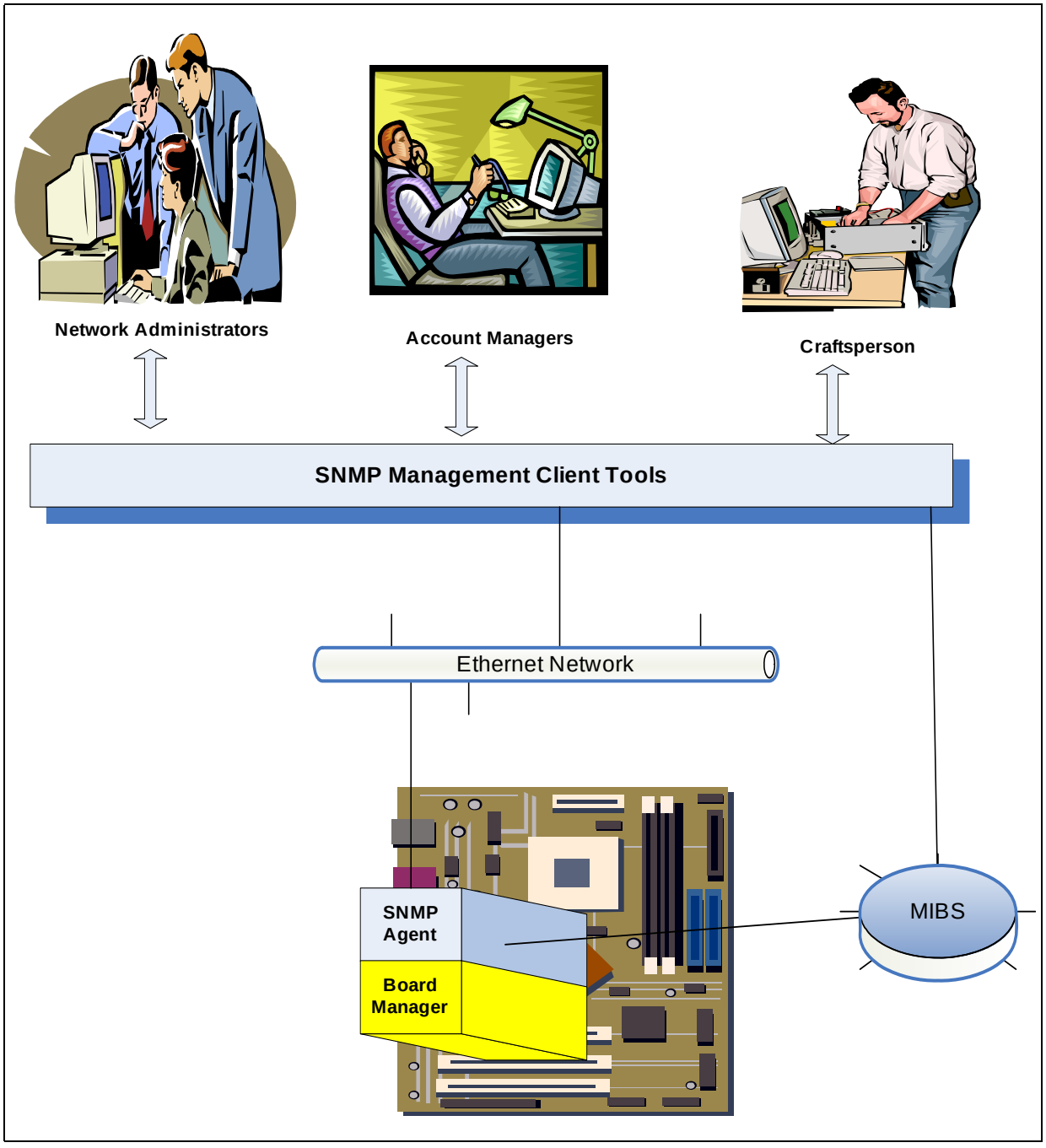
5.1.1 Common Administration Tasks

SNMP is a standard IP network management mechanism for exchanging management information between SNMP agents that typically reside on a managed device. The main job of SNMP is to manage network devices. The Multimedia Blade is a device that resides on a network; thus, it is a perfect candidate for SNMP management.

The SNMP agent provides remote (or local) monitoring and configuration. There are two types of security that the SNMP agent is governed by, namely, V2 and V3 securities. Security is discussed in Section 5.4, “Username and Password” on page 56.

Figure 8 shows the three types of SNMP users that exist in the Multimedia Software SNMP framework: craftsperson, administrator, and account manager. These various user roles are covered in Section 5.4. It also shows that the SNMP framework relies on the IP network and therefore, the network has to be properly set up before using SNMP.

Figure 8. General SNMP Architecture



The SNMP agent runs on the Multimedia Blade and listens for SNMP requests on the TCP/UDP 161 port. Once the SNMP agent processes the SNMP request, it delegates the management request to the Board Manager Service. The Board Manager Service from the end user point of view is transparent.

Most important, Figure 8 shows that the SNMP agent implementation and the client request types are described by the MIBs, which are implemented and exposed by this framework. The MIBs are covered in Section 5.9 through Section 5.15.

After you complete the preparations for using the SNMP framework software, as described in Section 5.2, "System Requirements" on page 54, you can perform the following administration tasks:

- If necessary, stop and start the SNMP agent software independent from the media application software, and change the SNMP listening port. (See Section 5.3, "Stopping and Starting the SNMP Agent" on page 56.)
- Perform user management such as adding, deleting, and modifying passwords. (See Section 5.4, "Username and Password" on page 56.)
- Configure various traps (events) such as system ready, run status, AdvancedMC run status, agent cold start, and Ethernet interface change notifications. Configure the trap forwarding table. (See Section 5.5, "Trap Configuration" on page 57.)
- Start and stop the media service. (See Section 5.6, "Stopping and Restarting the Media Service" on page 59.)
- Configure the DSP IP addresses. (See Section 5.7, "Media Ethernet Interface Configuration" on page 59.)
- Check system logs to help resolve problems. (See Section 5.8, "Resolving an Application Failure" on page 61.)
- Manage MIB-2 information such as system information, interfaces, TCP, UDP, and IP information. (See Section 5.9, "Public MIB 2 Module Reference" on page 62.)
- Manage DS1 information. (See Section 5.10, "Public DS1 Module Reference" on page 63.)
- Retrieve system information such as run status, product license, serial number, etc. (See Section 5.11, "Private System Environment Module Reference" on page 63.)
- Retrieve AdvancedMC hardware information and run status. (See Section 5.12, "Private Hardware Module Reference" on page 64.)
- Retrieve interface types. (See Section 5.13, "Private External Interfaces Module Reference" on page 64.)
- Configure and check status of the TDM clock. (See Section 5.14, "Private TDM Clock Module Reference" on page 64.)
- Configure and check status of the Media Control Protocol. (See Section 5.15, "Private Media Control Protocol Module Reference" on page 64.)
- Retrieve SNMP agent information such as agent version, IP address, port number, up time, and operational status. (See the private AdventNet AGENT-CONFIG-MIB MIB Module. User must have knowledge of how to read SNMP MIB files.)

5.1.2 MIBs

This section lists, defines, and gives the location of the supported MIBs. For specific information about each MIB, refer to the MIBs reference in Section 5.9 through Section 5.15.

A MIB is a specification containing definitions of management information so that networked systems can be remotely monitored, configured, and controlled.

A MIB resides on a managed node. The MIB definition sets the limits on what can be managed. The MIB defines which variables or parameters will be accessed, as well as how each value will be identified, encoded, and interpreted. A MIB contains objects (units of management information) divided into scalars and tables, which are identified by object identifiers (OIDs). These objects are exchanged between the managed node and the SNMP manager. Each MIB defines three types of OIDs: read-only OID, read-write OID, and traps.

MIBs can be either private (enterprise) MIBs or standard MIBs. SNMP agent software currently supports both public and private MIBs. The user can find all supported MIBs in the following directory in a given Multimedia Blade: /usr/dialogic/mibs/.

Note: Any public MIB can be obtained from the Internet or other sources. Public MIBs have been included in the Multimedia Blade MIB directory as a convenience to the user. Also, note that all groups in a given MIB module may not be fully supported. For more information, see the MIBs reference in Section 5.9 through Section 5.15.

Table 7 lists the MIB modules that the SNMP agent software supports. The table does not include all the MIB modules that are included by the described MIB files. For all MIBs, view /usr/dialogic/mibs.

Table 7. Supported MIBs

MIB Module	File Name (/usr/dialogic/mibs)	Private/ Public	Description
AGENT-CONFIG-MIB	AGENT-CONFIG-MIB.mib	Public	Trap sink configuration
AGENT-SNMP-CONFIG-MIB	AGENT-SNMP-CONFIG-MIB.mib	Public	Trap sink configuration
HOST-RESOURCES-MIB	rfc2790-HostResource.mib	Public	Host resource management such as disk, memory, software versions, software installed, etc. ***Note:*** Support planned for a future release.
RFC1213	RFC1213-MIB.mib	Public	MIB-2 - IP based information
RFC2495	RFC2495DS1-MIB.mib	Public	DS1 trunks
SNMP-USER-BASED-SM-MIB	SNMP-USER-BASED-SM-MIB.mib	Public	SNMP V3 Security - user authentication
SNMP-VIEW-BASED-ACM-MIB	SNMP-VIEW-BASED-ACM-MIB.mib	Public	SNMP V3 Security- user authorization.
Intel-Common-MIB	Intel-Common-Mib.mib	Private	Identifiers' branch typically contains objects, i.e., Intel branch project. Use for setting OID branch.
Intel-Prod-Tel-MCPD-IPMS-Root	Intel-Prod-Tel-MCPD-IPMS-Root.mib	Private	Identifiers' branch typically contains objects, i.e. Dialogic Multimedia Platform specific OID
IPMS-ATCA-HARDWARE	ipms_atca_hardware.mib	Private	ATCA related hardware information and configuration
IPMS-EXT-IF	ipms_ext_if.mib	Private	Configuration and status information for all other types of connections not covered by standard publicly defined MIB, such as interface IP address assignment
IPMS-MEDIA-CTL-PROTOCOL	ipms_media_ctl_protocol.mib	Private	Configuration of the IP media server protocol control

Table 7. Supported MIBs (Continued)

MIB Module	File Name (/usr/dialogic/mibs)	Private/ Public	Description
IPMS-SOFTWARE-UPGRADE	ipms_software_upgrade.mib	Private	Software upgrade ***Note:*** Support planned for a future release.
IPMS-SYS-ENV	ipms_sys_env.mib	Private	Configuration of the IP media server environment variable and media process control
IPMS-TDM-CLOCK	ipms_tdm_clock.mib	Private	Configuration of the IP Media Server TDM clock and TDM clock status

5.2 System Requirements

This section lists the software requirements for using the SNMP agent with the Multimedia Software release. It mainly concentrates on the SNMP client requirements for the management side.

It is important to understand that there are two ways that a user can manage the Multimedia Blade:

- First, the user can manage the Multimedia Blade using SNMP client tools. Since SNMP supports remote management, the client manager station exists as separate computer equipment, and the SNMP client tools must be installed in this equipment.
- Second, the user can use Secure Shell (ssh) program to log in to the Multimedia Blade and execute SNMP command line interface (CLI). This approach may not be suitable for managing multiple blades since the ssh operation is required for each managed blade. Nevertheless, for managing a blade at a time, this approach can be used.

In the following subsections, each client software requirement is addressed for each management approach.

5.2.1 Requirement for Remote Management Stations

The Remote Management Station can be based on either Linux/Unix or Windows Operating System. However, this document focuses on the Linux OS.

If users already have a management SNMP platform, such as their own browsers and SNMP development environment, then they need only load the appropriate Multimedia Software MIBs. This can be done by either copying the MIBs from the distribution or from the /usr/dialogic/mibs directory after the Multimedia Software has been installed on the blade.

If the Remote Management Station does not contain any SNMP client support, there are many possibilities for purchasing a commercial toolset or obtaining a free version. For example, the Net-SNMP client library is freely available from Open Source and can be downloaded from http://net-snmp.sourceforge.net/. Some SNMP management commercial tools include AdventNet (http://www.adventnet.com/), HP Openview (http://h20229.www2.hp.com), and MG-Soft (http://www.mg-soft.com).

Note: These are only suggestions and there are no restrictions imposed in the selection of any SNMP client toolset. Note that many Linux operating systems already come with the Net-SNMP toolset.

A basic SNMP client toolset (or libraries) is composed of the following:

- MIB compiler

- SNMP library such as C/Java binding. These libraries can be used for writing client management applications using the given APIs.
- A set of command line tools such as get, sets, table gets, trap manager, etc. These command line tools are simple client executables that can be easily called from either the command line or can be embedded in a script language to build very powerful remote or local configuration scripts.
- Other tools may offer GUIs such as SNMP browsers, trap displays, etc.

The free Net-SNMP is an example of an available client toolset and contains only the following:

- MIB compiler
- SNMP library such as C binding. These libraries can be used for writing client management applications using the given APIs.
- Command line tools

For the Red Hat Advance Server libraries, only the following SNMP client libraries are needed:

Note: Any version greater than 5.1 is acceptable.

- net-snmp-libs-5.1.2-11.EL4.6
- net-snmp-5.1.2-11.EL4.6

In addition, you will need to build the following Net-SNMP CLI clients. This SNMP CLI allows the user to script configuration.

- snmpbulkget
- snmpget
- snmpset
- snmptable
- snmptranslate
- snmptrapd
- snmpvacm
- snmpbulkwalk
- snmpdf
- snmpgetnext
- snmpstatus
- snmptest
- snmptrap
- snmpusm
- snmpwalk

Consult the Net-SNMP site for obtaining the libraries and obtaining the Net-SNMP CLI.

5.2.2 Requirements for Local Managed Nodes

Net-SNMP client libraries are required to use the SNMP CLI. The user is required to install the OS; therefore, the user also must install the Net-SNMP client environment.

5.3 Stopping and Starting the SNMP Agent

This section describes how to stop and start the SNMP agent software independent from the media application software. It also shows how to change the SNMP listening port.

5.3.1 Stopping the SNMP Agent

The CLI and SNMP agent software is automatically started at system reboot and when the media application software suite is started using **dlstart**. It is automatically stopped when the media application suite is terminated using **dlstop**. In order for the CLI to function, the Multimedia Software must be running, because the CLI has dependencies on back-end processes for data storage and retrieval. However, you can bring the CLI and SNMP agents up and down as often as needed while the Multimedia Software is up. (The CLI agent software is discussed in Chapter 4.0, "Administration via Command Line Interface (CLI).")

The SNMP agent can be stopped using the **dlservices** command:

```
# dlservices snmp stop
Shutting SNMP Agent Service : [  OK  ]
```

5.3.2 Starting the SNMP Agent

To restart the SNMP agent, use the **dlservices** command:

```
# dlservices snmp start
Starting SNMP Agent Service : [  OK  ]
```

5.3.3 Changing the SNMP Listening Port

The default SNMP listening port is 161. It can be changed during the software installation process as explained in Section 3.10, "Installing the Multimedia Software" on page 27. To change it at any other time, use **dlservices**. For example:

```
# dlservices snmp port 166
```

This changes the SNMP listening port to the specified port, 166 in this example. Port changes take effect on the next **dlstart** or **dlservices snmp start**.

Note: No range checking is done, so you are responsible for providing a usable address for the SNMP listening port if you are not using the default address.

5.4 Username and Password

The Multimedia Software SNMP agent handles both SNMP protocol 1/2c and Version 3. The Multimedia Blade has been preconfigured with various default users and passwords.

The following subsections explain each default user role and the respective username and passwords. Username and passwords can be changed, added, and deleted.

5.4.1 SNMP Version 1 and 2c

When using SNMP version 1 or 2c, the following SNMP User Community exists (Table 8).

Table 8. Version 1 and 2c Community Names

Community Name	Access
Craftsperson	Read only
Administrator	Read and write

5.4.2 SNMP Version 3

When using SNMP version 3, the following SNMP User Roles exist (Table 9).

Table 9. Version 3 Usernames and Passwords

Role	Username	Password	Auth Protocol	Context	Privacy Protocol	Role Description
Account Manager	AccountManager	AccountManager	MD5	auth	none	Allowed to add, delete, and change usernames and passwords. Allowed to add trap sink.
Administrator	Administrator	Administrator	MD5	auth	none	Allowed to read and write all supported MIBs except for Account Manager operations.
Craftsperson	Craftsperson	Craftsperson	MD5	auth	none	Allowed to read all supported MIBs except for Account Manager operations.
Account Manager (SHA)	AccountManagerSHA1	AccountManagerSHA1	SHA	auth	none	Allowed to add, delete, and change usernames and passwords. Allowed to add trap sink.
Administrator (SHA)	AdministratorSHA1	AdministratorSHA1	SHA	auth	none	Allowed to read and write all supported MIBs except for Account Manager operations.
Craftsperson (SHA)	CraftspersonSHA1	CraftspersonSHA1	SHA	auth	none	Allowed to read all supported MIBs except for Account Manager operations.
Note: All users use security level = authNoPriv						

5.5 Trap Configuration

This section discusses how to configure the forwarding trap table and all the traps generated by the system.

5.5.1 Supported Traps

Supported traps are listed and described in Table 10.

Table 10. Supported Traps

Trap Name	Private / Public	Description
Dsx1LineStatusChange	Public	Sent when the value of an instance dsx1LineStatus changes. It can be used by a network management system to trigger polls. When the line status change results from a higher level line status change (i.e., ds3), then no traps for the ds1 are sent.
coldStart	Public	Signifies that the SNMP entity, supporting a notification originator application, is reinitializing itself and that its configuration may have been altered.
authenticationFailure	Public	Signifies that the SNMP entity has received a protocol message that is not properly authenticated. While all implementations of SNMP entities *may* be capable of generating this trap, the snmpEnableAuthenTraps object indicates whether this trap will be generated.
ipmsSysEnvRunStatus	Private	Reports the run state of the overall IP Media Server system. The software that comprises the entire IP Media Server is treated as a system. Set to one of the following: • ipmsSysEnvStatusUnknown • ipmsSysEnvBootingUp • ipmsSysEnvStatusInitialize • ipmsSysEnvStatusActive • ipmsSysEnvStatusInActive • ipmsSysEnvStatusFault • ipmsSysEnvShutdown • ipmsSysEnvHotSwap Any time there is a state change, an SNMP notification (trap) is sent out.
ipmsSysEnvSysReady	Private	Reports when the IPMS application operationally ready is achieved. Sent also as a notification whenever the state changes whether from Active or to Active state. The companion ipmsSysEnvRunStatus should be used by the SNMP client to retrieve a more detailed run status.
ipmsHwAmcRunStatusNotification	Private	Notification is sent when the run status is changed on an AdvancedMC.
ipmsExtIfStatusChangedNotification	Private	Sends an SNMP trap when external interface link status change (up or down) occurs. The SNMP notification message includes an identification of the specific external interface the state change applies to using the ifIndex value to identify a particular of the ifTable entry.
ipmsTdmClkRefSwitchoverNotification	Private	Sent when clock source switchover situation occurs.

5.5.2 Configuration of Traps

The purpose of the trap configuration is to register with the SNMP agent of all Manager Stations that are interested in receiving SNMP traps.

There are two ways that an end user can configure the V3 trap forwarding table. (The configuration of this table also applies to V1/V2c.)

- The user is allowed to add and remove entries in the /usr/dialogic/conf/snmp/v3managertable.txt. This file contains the persistence information that is read by the SNMP agent upon boot up in order to find out the manager stations that have registered for traps.

  For Example:

```
192.168.99.90 SNMP_VERSION_2c public noAuthUser USM
NOAUTHNOPRIV 6 noAuth 5 0 ACTIVE
192.168.99.90 SNMP_VERSION_3 Administrator Administrator USM
NOAUTHNOPRIV 6 noAuth 5 0 ACTIVE
```

Note: Every time the user changes the v3managertable.txt file, the SNMP agent must be restarted in order for the changes to take place.

Note: Make sure that the security values for the trap browser or the Manager Application receiving the trap application, which has been configured to work with SNMP V3, match the V3 user security parameters set in /usr/dialogic/conf/snmp/v3managertable.txt file.

- The end user manipulates the trap forwarding table using SNMP. In this case, the changes take effect immediately and no restart of the SNMP agent is required.

5.6 Stopping and Restarting the Media Service

This section describes how a user is able to use SNMP to start and stop the media service.

5.6.1 Stopping the Media Service

In order to stop the Media Service, the Media Service must be running. The status can be checked by retrieving the run status using the ipmsSysEnv group.

To stop the Media Service, simply set the ipmsSysEnv group ipmsSysEnvMediaProcCtl attribute to ipmsMediaProcStop. If the Management Station has registered itself to receive traps from the SNMP agent, then the Management Station and all registered Stations should receive the Run Status Trap, which notifies the user that the Media Service has been stopped.

5.6.2 Restarting the Media Service

In order to start the Media Service, the Media Service must be in the stop state. The status can be checked by retrieving the run status using the ipmsSysEnv group.

To start the Media Service, simply set the ipmsSysEnv group ipmsSysEnvMediaProcCtl attribute to ipmsMediaProcStart. If the Management Station has registered itself to receive traps from the SNMP agent, then the Management Station and all registered Stations should receive two traps, namely, the Run Status Trap, which notifies the user that the media services Run Status changed, and the System Ready Notification.

5.7 Media Ethernet Interface Configuration

This section explains how to modify the default Media DSP IP addresses. The AdvancedMC, which is pre-installed on the Multimedia Blade baseboard, has four digital signal processors (DSPs) used for echo cancellation and transcoding offload. See Figure 1, “Multimedia Blade Block Diagram” on page 16.

The default IP addresses assigned to the DSPs are:

```
"192.168.0.20",
"192.168.0.21",
"192.168.0.22",
"192.168.0.23"
```

If the end user starts the board and it fails, it is because the user must change the IP addresses for each Media (DSP) device according to its network configuration.

Media DSP IP address configuration can be done using a MIB browser. Configuration can also be performed by using Net-SNMP command line tools as described below.

Note: For further information about Net-SNMP, consult the Net-SNMP command line help and the help document at http://net-snmp.sourceforge.net/.

5.7.1 Changing Default DSP IP Addresses

The general steps required to change the DSP IP addresses are as follows:

1. Determine which interfaces represent the DSP devices. Remember that the Multimedia Blade contains several interfaces. Refer to Section 5.7.2, “Displaying DSP IP Addresses” on page 60.
2. Delete the appropriate table entry from ipmsExtIfIpAddrTable. Refer to Section 5.7.3, “Deleting DSP IP Addresses” on page 60.
3. Create the appropriate table entry in the ipmsExtIfIpAddrTable. Refer to Section 5.7.4, “Creating DSP IP Addresses” on page 61.
4. Once all changes have been made for all the DSP IP addresses, restart the media service as described in Section 5.6.2, “Restarting the Media Service” on page 59.

5.7.2 Displaying DSP IP Addresses

This section shows an example using Net-SNMP command line tools that can be used to display the IPMS External Interface IP Address Table. This table contains the IP addresses of all IP interfaces in the Multimedia Blade.

1. Dump out the External Interface Type table. This table relates a particular interface index to its interface type. In the case of the DSP, the interface type is AMC1 through AMC4.

   ```
   snmptable -v3 -a MD5 -A Administrator -l authNoPriv -n auth -u
   Administrator -m ALL -M /usr/dialogic/mibs 192.168.99.90
   ipmsExtIfTable
   ```

2. Determine which interface index represents the DSPs, then type:

   ```
   ./snmptable -v3 -a MD5 -A Administrator -l authNoPriv -n auth -
   u Administrator -m ALL -M  /usr/dialogic/mibs 192.168.99.90
   ipmsExtIfIpAddrTable
   ```

The above command dumps the External Interface IP Address Table. Correlate the interface index from the above command with the output of this command. The matching set represents the DSP IP addresses. Notice that typically the DSP interfaces start at 20.

5.7.3 Deleting DSP IP Addresses

Identify which interface to delete (i.e., DSPs), then use Net-SNMP CLI to submit the following command to delete the entry:

```
./snmpset -v3 -a MD5 -A Administrator -l authNoPriv -n auth -u
Administrator -m ALL -M ../mibs 192.168.99.90
ipmsExtIfIpAddrRowStatus. 192.168.99.90 i 6
```

In this command:

- `i 6` = integer 6 (means destroy row status)

- `ipmsExtIfIpAddrRowStatus. 192.168.99.90` = IP address key and 20 is the interface index value. This table required a compound key. (These values are obtained from previous displays of the IP addresses).

Note: Be aware the following issue needs to be resolved: After successfully deleting/creating an IP address entry, the user may get invalid information when retrieving the table back. There are some cache problems with the framework that must be worked out. If the user waits a short amount of time after either a delete or an update of an IP address entry, then the IP address table shows the correct entries.

5.7.4 Creating DSP IP Addresses

Creating a new DSP IP entry is as simple as deleting a row. The following is an example of a Net-SNMP CLI command:

```
./snmpset -v3 -a MD5 -A Administrator -l authNoPriv -n auth -u
Administrator -m ALL -M ../mibs 192.168.99.90
ipmsExtIfIpAddrRowStatus. 192.168.99.90 i 4
```

In this command:

- `i 4` = integer 4 (means create row status)

5.8 Resolving an Application Failure

SNMP return error codes are not too robust. In many cases when the end users encounter errors, they may need to rely on the system logs in order to identify certain system problems.

The Multimedia Software uses log files to store information, warnings, and errors.

The log file for the entire OA&M and SNMP infrastructure can be found in the /var/log/ directory. The file ngmit.log represents the log file for the OA&M SNMP log information generated by the OA&M framework.

Example of a log message in ngmit.log file:

```
Jun  2 15:57:35 mymachine Board Manager: From Component: MIB2-OS-
Adaptor : Ds1 empty
```

Notice the format of the log message. Basically all messages have the following generic format:

<Date> <machine name> <OA&M component> <Error Message>

Note: The syslogd service must be running in order for logging to work. Also, the file /etc/syslog.conf must contain the following configuration lines for NGMIT:

```
# NGMIT Logging and Tracing
local1.* /var/log/ngmit.log
```

The configuration entries are automatically configured for the system when the Multimedia Software has been properly installed.

5.9 Public MIB 2 Module Reference

Table 11 describes which groups in the given MIB modules are implemented, and describes any deviations from the MIB if taken. It serves as the Module Conformance. Each MIB entry is not described because the MIB itself should be self-descriptive.

Table 11. MIB 2 Conformance Table

RFC1213					
MIB Group	**MIB Attribute**	**Read**	**Write**	**Supported**	**Comments**
System(1.3.6.1.2.1.1)	sysDesc (1)	Yes	Yes	Yes	
	sysObjectID (2)	Yes	No	Yes	
	sysUpTime (3)	Yes	No	Yes	
	sysContact(4)	Yes	Yes	Yes	
	sysName(5)	Yes	Yes	Yes	
	sysLocation(6)	Yes	Yes	Yes	
	sysServices(7)	Yes	Yes	Yes	
	sysORLastChange(8)	No	No	No	Never supported
	SysORTable(9)	No	No	No	Never supported
interfaces(1.3.6.1.2.1.2)	ifNumber(1)	Yes	No	Yes	
	ifTable(2)	Yes	Yes	Yes	Table has only one write value
	The only column from this table that is writable is ifAdminStatus (.1.7)	Yes	Yes	Yes	
at (1.3.6.1.2.1.3)	Not supported	No	No	No	Deprecated and never supported
ip (1.3.6.1.2.1.4)	ipForwarding(1)	Yes	No	Yes	Write not supported in backend
	ipDefaultTTL(2)	Yes	No	Yes	Write not supported in backend
	ipInReceives(3)	Yes	No	Yes	
	ipInHdrErrors(4)	Yes	No	Yes	
	ipInAddrErrors(5)	Yes	No	Yes	
	ipForwDatagrams(6)	Yes	No	Yes	
	ipInUnknownProtos(7)	Yes	No	Yes	
	ipInDiscards(8)	Yes	No	Yes	
	ipInDelivers(9)	Yes	No	Yes	
	ipOutRequests(10)	Yes	No	Yes	
	ipOutDiscards(11)	Yes	No	Yes	
	ipOutNoRoutes(12)	Yes	No	Yes	
	ipReasmTimeout(13)	Yes	No	Yes	
	ipReasmReqds(14)	Yes	No	Yes	
	ipReasmOKs(15)	Yes	No	Yes	
	ipReasmFails(16)	Yes	No	Yes	
	ipFragOKs(17)	Yes	No	Yes	
	ipFragFails(18)	Yes	No	Yes	
	ipFragCreates(19)	Yes	No	Yes	
	ipAddrTable (20)	Yes	No	Yes	All cols supported
	ipRouteTable(21)	Yes	No	Yes	Support for table fields is read-only

Table 11. MIB 2 Conformance Table (Continued)

RFC1213					
MIB Group	**MIB Attribute**	**Read**	**Write**	**Supported**	**Comments**
	ipNetToMediaTable	Yes	No	Yes	Read-only
icmp (1.3.6.1.2.1.5)	All	Yes	No	Yes	All attributes are read-only and all are supported
tcp (1.3.6.1.2.1.6)	All scalar are read only	Yes	No	Yes	
	tcpConnTable(13) - all read is supported	Yes	No	Yes	tcpConnState (13.1.1) write is not supported
udp (1.3.6.1.2.1.7)	All scalar are supported	Yes	No	Yes	
	udpTable (5.1)	Yes	No	Yes	
egp (1.3.6.1.2.1.8)	Not supported	No	No	No	Never, not supported by backend
snmp (1.3.6.1.2.1.11)	All scalar are supported	Yes	No	Yes	

5.10 Public DS1 Module Reference

Table 12 describes which groups in the given MIB modules are implemented, and describes any deviations from the MIB if taken. It serves as the Module Conformance. Each MIB entry is not described because the MIB itself should be self-descriptive.

Table 12. DS1 MIB Conformance Table

Module	MIB Attribute	Read	Write	Supported	Comments
dsx1configTable (1.3.6.1.2.1.10.18.6)	dsx1LineType (.1.5)	Yes	Yes	Yes	Valid writable if AdminStatus =down
	dsx1LineCoding(.1.6)	Yes	Yes	Yes	Valid writable if AdminStatus =down
	dsx1CircuitIdentifier(.1.8)	Yes	Yes	Yes	
	dsx1SendCode(1.7)	Yes	No	Yes	Writes not supported by backend
	dsx1LineLength(1.15)	Yes	Yes	Yes	Valid writable if AdminStatus =down
	dsx1LoopbackConfig(1.9)	Yes	Yes	Yes	Valid writable if AdminStatus =down
	dsx1TransmitClockSource	Yes	No	Yes	Writes not supported by backend
	dsx1CurrentTable(.18.7)	Yes	No	Yes	
	dsx1IntervalTable(18.8)	Yes	No	Yes	
	dsx1IntervalTable(18.9)	Yes	No	Yes	
	dsx1LineStatusChangeTrapEnable	Yes	Yes	Yes	
	Not the far ends, and fact tables are not supported	No	No	No	
	Also dsx1ChanMappingTable not supported	No	No	No	
Note: dsx1configTable(18.6) all fields can be read; however, a few are only writable. Also, none of the far end tables are supported.					

5.11 Private System Environment Module Reference

The ipmsSysEnv Module (1.3.6.1.4.1.343.2.15.2.1.1) contains basic information about the vendor, form factor type, blade model, run status, license information, and the attribute to start and stop the media service. Consult the MIB for specific definitions.

5.12 Private Hardware Module Reference

The ipmsAtcaHardware Module (1.3.6.1.4.1.343.2.15.2.1.2.1) describes basic information about the base Multimedia Blade. Consult the MIB for specific definitions. Only the HwResetEntireCard field in this table is writable.

Notes:

- The ipmsHwBaseboardStatusLEDTable Module (1.3.6.1.4.1.343.2.15.2.1.2.1.8) reflects the baseboard LED states.
- The ipmsHwAmcTable Module (1.3.6.1.4.1.343.2.15.2.1.2.1.9) contains basic information about the AdvancedMC devices.
- The ipmsHwAmcStatusLEDTable Module (1.3.6.1.4.1.343.2.15.2.1.2.1.11) reflects the AdvancedMC LED states.

5.13 Private External Interfaces Module Reference

The ipmsExtIf (1.3.6.1.4.1.343.2.15.2.1.3) Module allows the end user to correlate all Ethernet interfaces available in the hardware. Each interface is uniquely identified by its interface index field. The interface index is associated with the MIB 2 interface table.

This MIB module contains two main tables. The ipmsExtIfEntry table simply associates the various types of Ethernet interfaces to their corresponding types. The possible Ethernet types are:

1. baseInterface(1)
2. fabricInterface(2)
3. rearTransitionModuleInterface(3)
4. fabricRTMInterface(4)
5. aMC1Interface(5) (represents the DSP)
6. aMC2Interface(6) (represents the DSP)
7. aMC3Interface(7) (represents the DSP)
8. aMC4Interface(8) (represents the DSP)
9. other(9)

The other table mainly allows the user to set the Ethernet IP configuration such IP address, mask, etc.

5.14 Private TDM Clock Module Reference

The ipmsTDMClock Module (1.3.6.1.4.1.343.2.15.2.1.4) describes the configuration of the IP media server TDM clock and TDM clock status. Consult the MIB for specific definitions.

5.15 Private Media Control Protocol Module Reference

The ipmsMediaCtlProtocol Module (1.3.6.1.4.1.343.2.15.2.1.6) describes the configuration of the IP media server protocol control. Consult the MIB for specific definitions.

6.0 Controls, Indicators, and Connectors

This chapter provides information about the controls, indicators, and connectors on the Dialogic® Multimedia Blade for AdvancedTCA® (hereinafter referred to as the Multimedia Blade). The following topics are discussed:

- Backplane and On-board Connectors
- Front Panel
- AdvancedMC
- Rear Transition Module
- Ejector Handles

6.1 Backplane and On-board Connectors

Connectors along the rear edge of the Multimedia Blade are divided into three distinct zones, as described in Section 2.3 of the PICMG 3.0 Specification:

- Zone 1 for system management and power distribution (P10)
- Zone 2 for data fabric (J23)
- Zone 3 for the rear transition module (RTM) (J30, J31, J32)

Figure 9 shows the locations of the backplane and on-board connectors and Table 13 explains the function of each connector. This is followed by information about:

- Power Distribution Connector (P10)
- AdvancedTCA Data Transport Connector (J23)
- Zone 3 Rear Transition Module Connectors (J30, J31, J32)
- Alignment Blocks

Figure 9. Backplane and On-board Connector and DIP Switch Locations

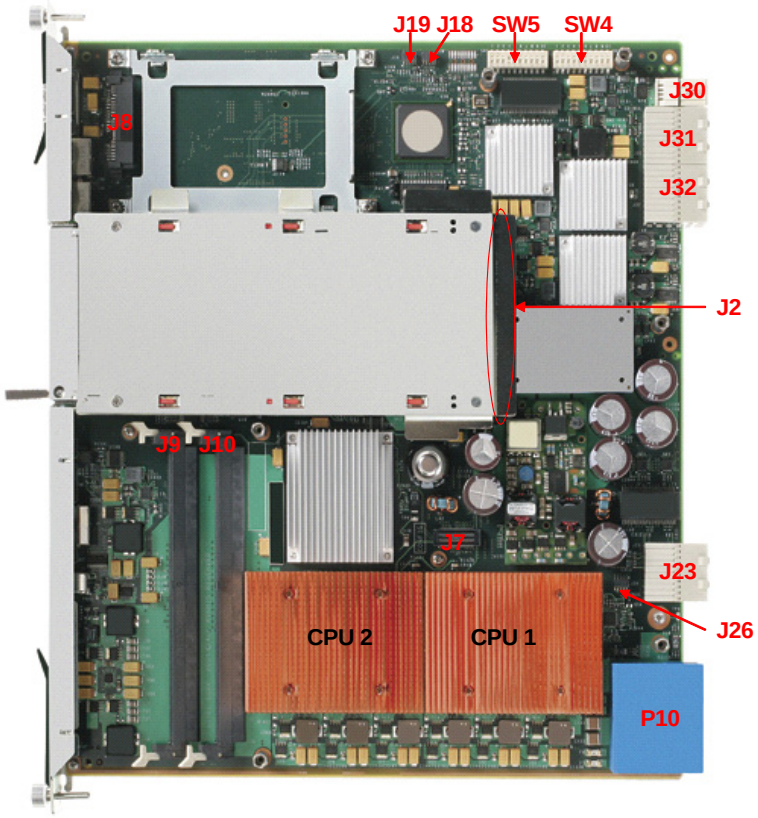


Table 13. Backplane and On-board Connector Assignments (Sheet 1 of 2)

Connector	Description
CPU 1	Processor #1 (CPU 1 is not a connector)
CPU 2	Processor #2 (CPU 2 is not a connector)
J2	AdvancedMC* connector
J7	ITP700 connector
J8	Serial attached SCSI (SAS) hard disk connector
J9	DIMM 1
J10	DIMM 2
J18	JTAG header 2 (for manufacturing test purposes only)
J19	JTAG header 1 (for manufacturing test purposes only)
J23	AdvancedTCA data transport for Base and Fabric interfaces (Zone 2)
J26	ITP JTAG header
J30	RTM power (Zone 3)
J31	RTM data and control connector (Zone 3)

Table 13. Backplane and On-board Connector Assignments (Sheet 2 of 2)

J32	RTM data and control connector (Zone 3)
P10	AdvancedTCA power and IPMB
SW4	DIP switches ***Note:*** All switches should remain in the default ON position.
SW5	DIP switches ***Note:*** All switches should remain in the default ON position.

6.1.1 Power Distribution Connector (P10)

Zone 1 consists of P10, a 34-pin Positronic* header connector that provides the following signals:

- Two -48 VDC power feeds (four signals each; eight signals total)
- Two IPMB ports (two signals each, four signals total)
- Geographic address (eight signals)
- Two ground pins
- 12 unused (but physically present) pins

Figure 10 shows the mechanical drawing of the connector. The pin assignments are given in Table 14.

Figure 10. Power Distribution Connector (Zone 1) P10

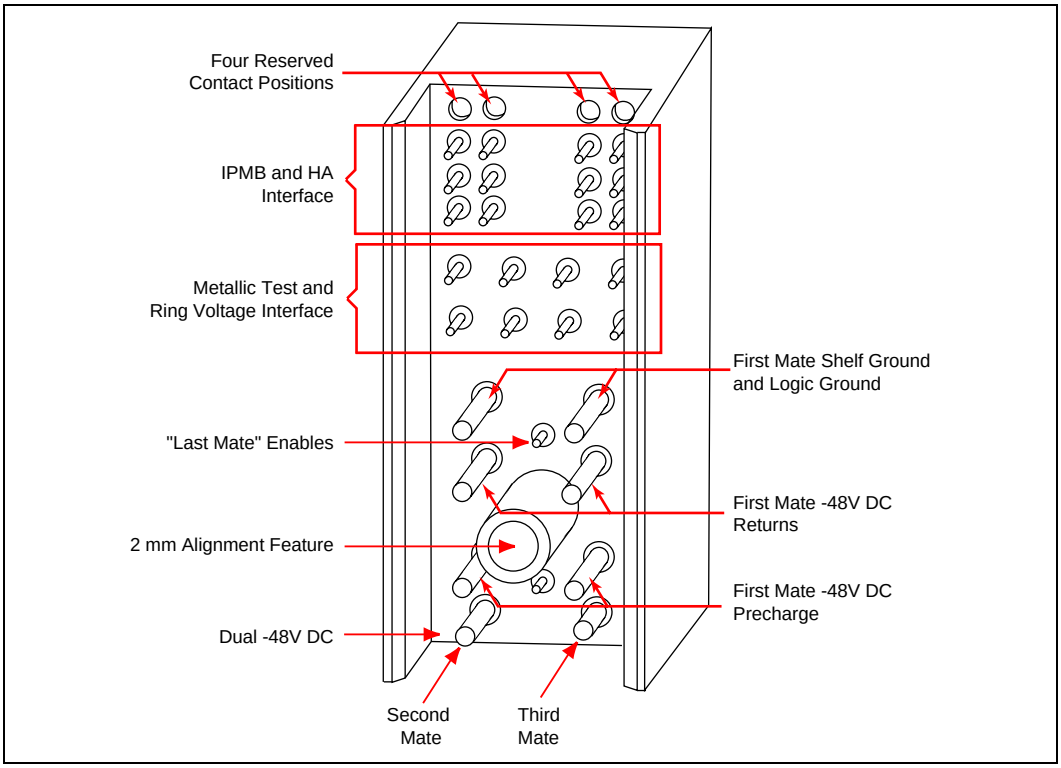


Table 14. Power Distribution Connector (Zone 1) P10 Pin Assignments

Pin	Signal	Description
1	Reserved	No Connect
2	Reserved	No Connect
3	Reserved	No Connect
4	Reserved	No Connect
5	GA0	Geographic Addr Bit 0
6	GA1	Geographic Addr Bit 1
7	GA2	Geographic Addr Bit 2
8	GA3	Geographic Addr Bit 3
9	GA4	Geographic Addr Bit 4
10	GA5	Geographic Addr Bit 5
11	GA6	Geographic Addr Bit 6
12	GA7/P	Geo Adr Bit 7 (Odd Parity)
13	IPMB_CLK_A	IPMB Bus A Clock
14	IPMB_DAT_A	IPMB Bus A Data
15	IPMB_CLK_B	IPMB Bus B Clock
16	IPMB_DAT_B	IPMB Bus B Data
17	Unused	No Connect
18	Unused	No Connect
19	Unused	No Connect
20	Unused	No Connect
21	Unused	No Connect
22	Unused	No Connect
23	Unused	No Connect
24	Unused	No Connect
25	EMI_GND	EMI Chassis Ground
26	LOGIC_GND	Gnd Ref for Card Logic
27	ENABLE_B	Enb DC-DC conv, B Feed
28	VRTN_A	-48 V Return, Feed A
29	VRTN_B	-48 V Return, Feed B
30	-48V_EARLY_A	No Connect
31	-48V_EARLY_B	No Connect
32	ENABLE_A	Enb DC-DC conv, A Feed
33	-48V_A	-48 V Input, Feed A
34	-48V_B	-48 V Input, Feed B

6.1.2 AdvancedTCA Data Transport Connector (J23)

Zone 2 consists of one 120-pin HM-Zd connector with 40 differential pairs. This data transport connector provides the following signals:

- Two 10/100/1000BASE-T/TX Ethernet Base channels (4 differential signal pairs each, 16 signals total)

- Two 1000BASE-BX Ethernet Fabric channels (4 differential signal pairs each, 8 signals total)

The connector used is an AMP*/Tyco* part number 1469001-1. Figure 11 shows a face view of the connector and Table 15 gives the connector pin assignments. The BG, DG, FG, and HG (G for Ground) columns contain the ground shields for the four columns of differential pairs. They have been omitted from the pinout table below for simplification. All pins in the BG, DG, FG, and HG columns are connected to logic ground.

Figure 11. Data Transport Connector (Zone 2) J23

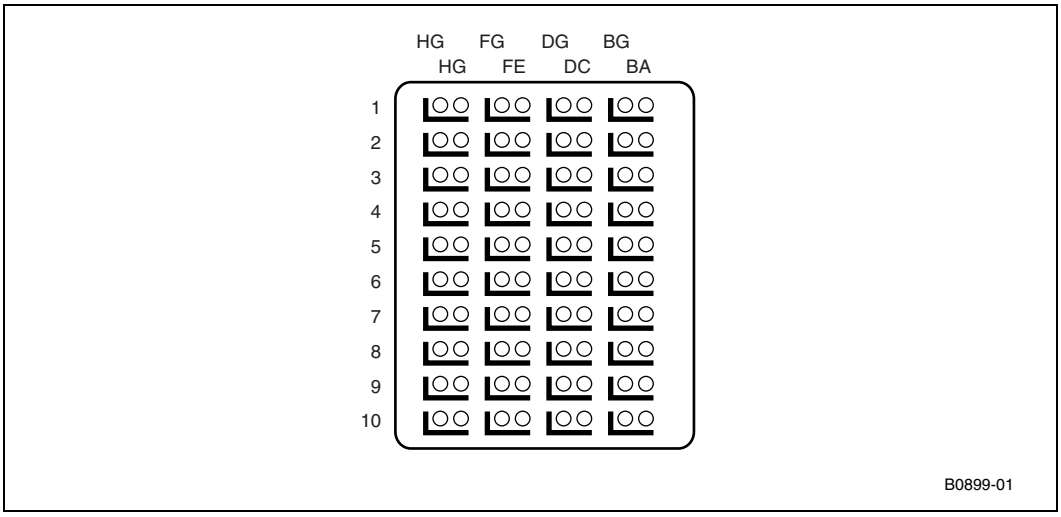


Table 15. AdvancedTCA Data Transport Connector (Zone 2) J23 Pin Assignments

Pin	A	B	C	D	E	F	G	H
1	No Connect	No Connect	Terminated	Terminated	No Connect	No Connect	Terminated	Terminated
2	F[2]Tx0+	F[2]Tx0-	F[2]Rx0+	F[2]Rx0-	F[2]Tx1+	F[2]Tx1-	F[2]Rx1+	F[2]Rx1-
3	No Connect	No Connect	Terminated	Terminated	No Connect	No Connect	Terminated	Terminated
4	F[1]Tx0+	F[1]Tx0-	F[1]Rx0+	F[1]Rx0-	F[1]Tx1+	F[1]Tx1-	F[1]Rx1+	F[1]Rx1-
5	BI_DA1+ (Tx1+)	BI_DA1- (Tx1-)	BI_DB1+ (Rx1+)	BI_DB1- (Rx1+)	BI_DC1+	BI_DC1-	BI_DD1+	BI_DD1-
6	BI_DA2 + (Tx2+)	BI_DA2- (Tx2-)	BI_DB1+ (Rx2+)	BI_DB1- (Rx2-)	BI_DC2+	BI_DC2-	BI_DD2+	BI_DD2-
7	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect
8	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect
9	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect
10	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect	No Connect

Fabric interface (Gigabit Ethernet) ports
Base interface (Gigabit Ethernet) ports

The following naming convention describes the signals on this connector. Signal direction is defined from the perspective of the Multimedia Blade.

For the Base interface, the bi-directional 10/100/1000BASE-T data signals have the following conventions:

BI_Dr[c]p
- r = differential pair (A, B, C, or D)
- c = channel (1, 2)
- p = polarity (+, -)

For the Fabric interface, the 1000BASE-BX data signals have the following conventions:

F[c]dnp
- c = channel (1, 2)
- d = direction (Tx = Transmit, Rx = Receive)
- n = port number (0, 1)
- p = polarity (+, -)

A port is two differential pairs; one Tx and one Rx.

6.1.3 Zone 3 Rear Transition Module Connectors (J30, J31, J32)

The Multimedia Blade main board implementation includes RTM connectors that mate directly to the RTM without connecting through the backplane. The J31 connector consists of one 120-pin HM-Zd connector with 40 differential pairs, which allows high-speed signals to be passed between the boards. In addition, the J30 connector carries management signals and power, and the J32 connector is used to route signals from the AdvancedMC slot to the RTM.

6.1.4 Alignment Blocks

The Multimedia Blade implements the K1 and K2 alignment blocks at the top of Zone 2 and Zone 3, as required in Section 2.4.4 of the PICMG 3.0 Specification. The Zone 2 alignment block (K1) is assigned a keying value of 1-1, and uses Tyco 1-1469373-1 (or equivalent). The Zone 3 alignment block (K2) is assigned a keying value of 7-3, and uses Tyco 7-1469373-3 (or equivalent).

6.2 Front Panel

The following sections provide information about the Multimedia Blade front panel:

- Front Panel Connectors
- Front Panel Reset Button
- Front Panel LEDs

6.2.1 Front Panel Connectors

Figure 12 shows the location of the front panel connectors on the board. Table 16 gives information about the cables that can be used with the front panel ports on the board. Each connector is then described in a separate section:

- USB Connector (J3)
- Serial Port Connector (J4)
- Front Panel SAS Connector (J8)

Note: The front panel Ethernet connectors are not used. During normal (run time) operation, connections to the PSTN and Ethernet are made via connectors on the Rear Transition Module (RTM). See Section 6.4, “Rear Transition Module” on page 77.

Figure 12. Front Panel Connectors

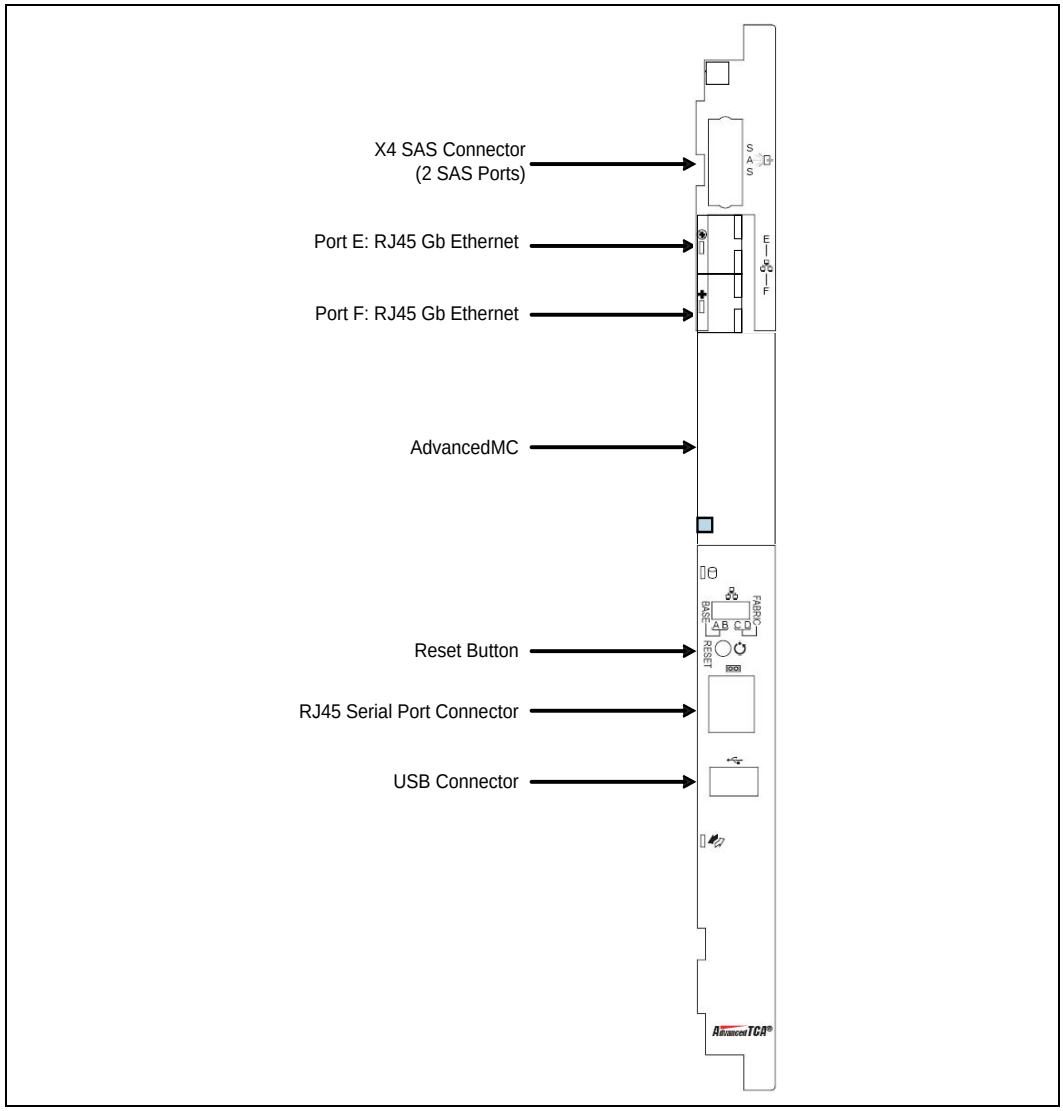


Table 16. Cable Information

Type	Cable Description	Max Length
SAS port	4X external SAS cable	7m
Ethernet ports	Do not use	
Serial port	Unshielded Category 5e cable with DB-9 to RJ45 converter	8m
USB port	External USB cable	5m

Notes: The SAS cable should be the Molex* 74527-3008 or equivalent.
See http://www.molex.com/cgi-bin/bv/molex/home_init.jsp.
The mating end of the SAS connector should be an SFF-8470 connector (or equivalent).
See http://www.cs-electronics.com/sas-cables.htm.

6.2.1.1 USB Connector (J3)

The Multimedia Blade has one USB connector, J3, that supports USB 2.0 and USB 1.1. Figure 13 shows the connector and Table 17 gives the pin assignments.

Figure 13. USB Connector (J3)

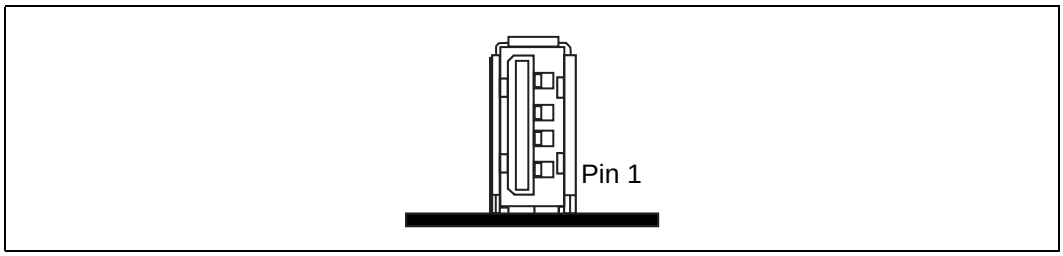


Table 17. USB Connector (J3) Pin Assignments

Pin	Signal
1	+5 V
2	-DATA
3	+DATA
4	GND

6.2.1.2 Serial Port Connector (J4)

A single serial port interface is provided on the front edge of the board using an RJ-45 style shielded connector. The connector is an 8-pin RJ-45. Figure 14 shows the connector and Table 18 gives the pin assignments. Figure 15 shows the RJ-45 to DB-9 translation.

Figure 14. Serial Port Connector (J4)

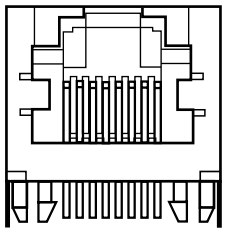


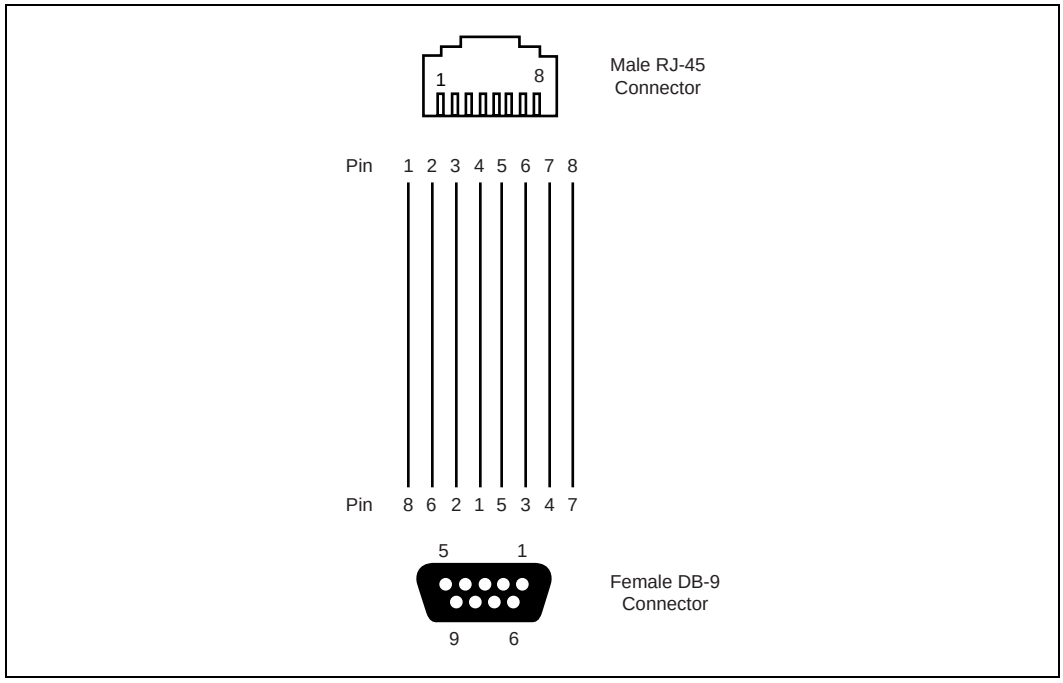
Table 18. Serial Port Connector (J4) Pin Assignments (Sheet 1 of 2)

Pin	Signal
1	RTS
2	DTR
3	TXD
4	GND
5	GND

Table 18. Serial Port Connector (J4) Pin Assignments (Sheet 2 of 2)

Pin	Signal
6	RXD
7	DSR
8	CTS

Figure 15. DB-9 to RJ-45 Pin Translation



6.2.1.3 Front Panel SAS Connector (J8)

Table 19 shows the pinout definition for the front panel SAS connector. There are two SAS ports available from the front panel connector.

Table 19. Front Panel SAS Connector Pin Assignments (Sheet 1 of 2)

Pin	Signal	Pin	Signal
1	GND	14	No Connect
2	SAS_RX2_P	15	No Connect
3	SAS_RX2_N	16	GND
4	GND	17	No Connect
5	SAS_RX3_P	18	No Connect
6	SAS_RX3_N	19	GND
7	GND	20	SAS_TX3_N
8	No Connect	21	SAS_TX3_P
9	No Connect	22	GND
10	GND	23	SAS_TX2_N

Table 19. Front Panel SAS Connector Pin Assignments (Sheet 2 of 2)

Pin	Signal	Pin	Signal
11	No Connect	24	SAS_TX2_P
12	No Connect	25	GND
13	GND		

6.2.2 Front Panel Reset Button

The front panel has a reset button, shown in Figure 12, “Front Panel Connectors” on page 71. The reset button is located in a small recessed hole below the Base and Fabric interface LEDs. The reset button is an input to the IPMC to request a payload soft reset. There are IPMI commands to reset the board and change power states through the software, and this is the recommended method to perform a reset (via ShMC).

6.2.3 Front Panel LEDs

The Multimedia Blade has LEDs to indicate status as shown in Figure 16. When these LEDs are lit, they indicate a status as defined in Table 20.

Figure 16. Front Panel LEDs

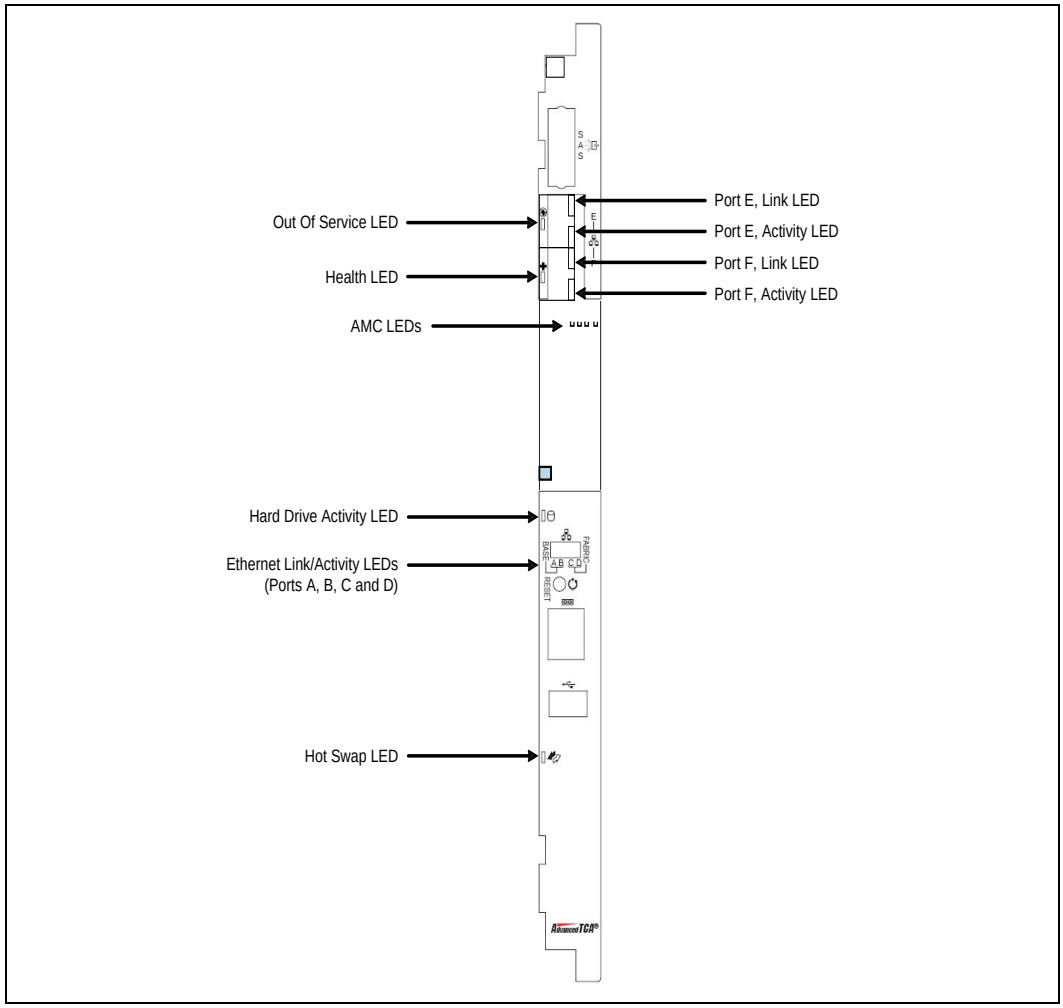


Table 20. Front Panel LED Descriptions (Sheet 1 of 2)

LED	Function
Hot Swap 	**Function:** Hot Swap as defined in the AdvancedTCA 3.0 specification **Possible States:** OFF: Normal operation or board not powered up BLUE: Safe to extract SHORT BLINK: Hot swap removal in process. LONG BLINK: Hot swap insertion in process. **Warning:** Removing the Multimedia Blade before the Hot Swap LED is solid blue can lead to device corruption or failure. It is possible for a user to override the default behavior of the LED using AdvancedTCA FRU LED Control commands.
Out of Service 	**Function:** Out of Service (AdvancedTCA LED 1) **Possible States:** OFF: The board is running. RED or AMBER: The system is not fully operational. It is possible for a user to override the default IPMC behavior of the LED using AdvancedTCA FRU LED Control commands.

Table 20. Front Panel LED Descriptions (Sheet 2 of 2)

LED	Function
Health 	**Function:** Health (AdvancedTCA LED 2). The Multimedia Blade health is based on an aggregation of IPMI sensors, like board temperature and voltage. **Possible States:** OFF / GREEN / RED / AMBER GREEN: The board is healthy. RED: The board is not healthy. **Note:** If only one -48V power supply feed is connected to the chassis, this LED will stay Red. It is possible for the user to override the default IPMC behavior of the LED using AdvancedTCA FRU LED Control commands.
Ports A, B, C, D Link / Activity	There is one LED that indicates link and activity for the following ports: • A: Base Interface: Port 0, Channel 1 • B: Base Interface: Port 0, Channel 2 • C: Fabric Interface, Port 0, Channel 1 • D: Fabric Interface, Port 0, Channel 2 **Function:** Gigabit Ethernet Base/Fabric Interface Link and Activity **Possible States:** OFF: No Link GREEN: Link GREEN-BLINK: Link and Activity
Ports E & F Link Speed 	There is one LED that indicates link speed for the following ports: • E: Fabric Interface: Port 1, Channel 1 • F: Fabric Interface: Port 1, Channel 2 **Function:** Gigabit Ethernet Fabric Interface Link Speed **Possible States:** OFF: 10 Mb/s GREEN: 100 Mb/s AMBER: 1000 Mb/s **Note:** This LED will be active even if the Ethernet port is connected to the backplane Fabric interface.
Ports E & F Activity 	There is one LED that indicates activity for the following ports: • E: Fabric Interface: Port 1, Channel 1 • F: Fabric Interface: Port 1, Channel 2 **Function:** Gigabit Ethernet Fabric Interface Link Activity: **Possible States:** GREEN-NO BLINKING: No traffic GREEN-BLINKING: Transmit or receive traffic is coming across the link **Note:** This LED will be active even if the Ethernet port is connected to the backplane Fabric interface.
Hard Drive Activity 	**Function:** Hard Drive Activity **Possible States:** OFF: No disk access BLINKING GREEN: Disk access (read/write activity)

6.3 AdvancedMC

The AdvancedMC LEDs are described in Table 21. For location of the AdvancedMC LEDs on the front panel, see Figure 16, “Front Panel LEDs” on page 75.

Table 21. AdvancedMC LED Descriptions

LED	Function
Health 	**Function:** Health (AdvancedTCA LED 2). The health is based on an aggregation of IPMI sensors, like board temperature and voltage. **Possible States:** OFF / GREEN / RED / AMBER GREEN: The board is healthy. RED: The board is not healthy. **Note:** If only one -48V power supply feed is connected to the chassis, this LED will stay Red. It is possible for the user to override the default IPMC behavior of the LED using AdvancedTCA FRU LED Control commands.
Out of Service 	**Function:** Out of Service (AdvancedTCA LED 1) **Possible States:** OFF: The board is running. RED or AMBER: The system is not fully operational. It is possible for a user to override the default IPMC behavior of the LED using AdvancedTCA FRU LED Control commands.
Note: The other AdvancedMC LEDs are not used.	

6.4 Rear Transition Module

During normal operation, Multimedia Blade connections to the PSTN and Ethernet are provided via the Rear Transition Module (RTM).

- Two Ethernet connectors are provided (Eth6 and Eth7) for Ethernet connections.
- Two RJ-21X connectors are provided for PSTN connections, such as for connections to the optional breakout box.

Table 22 shows the pinout definitions for the RJ-21X connectors. Each port (E1/T1 trunk) has four signals: a transmit pair and a receive pair.

The RTM also has LEDs, which are described in Table 23.

Table 22. Rear Transition Module RJ-21X Pin Assignments (Sheet 1 of 2)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A-1	STA1 RCV RING	A-38	STA5 RCV TIP	B-1	STA9 RCV RING	B-38	STA13 RCV TIP
A-26	STA1 RCV TIP	A-14	STA5 XMIT RING	B-26	STA9 RCV TIP	B-14	STA13 XMIT RING
A-2	STA1 XMIT RING	A-39	STA5 XMIT TIP	B-2	STA9 XMIT RING	B-39	STA13 XMIT TIP
A-27	STA1 SMIT TIP	A-15		B-27	STA9 SMIT TIP	B-15	
A-3		A-40		B-3		B-40	
A-28		A-16	STA6 RCV RING	B-28		B-16	STA14 RCV RING
A-4	STA2 RCV RING	A-41	STA6 RCV TIP	B-4	STA10 RCV RING	B-41	STA14 RCV TIP
A-29	STA2 RCV TIP	A-17	STA6 XMIT RING	B-29	STA10 RCV TIP	B-17	STA14 XMIT RING
A-5	STA2 XMIT RING	A-42	STA6 XMIT TIP	B-5	STA10 XMIT RING	B-42	STA14 XMIT TIP
A-30	STA2 XMIT TIP	A-18		B-30	STA10 XMIT TIP	B-18	
A-6		A-43		B-6		B-43	
A-31		A-19	STA7 RCV RING	B-31		B-19	STA15 RCV RING
A-7	STA3 RCV RING	A-44	STA7 RCV TIP	B-7	STA11 RCV RING	B-44	STA15 RCV TIP
A-32	STA3 RCV TIP	A-20	STA7 XMIT RING	B-32	STA11 RCV TIP	B-20	STA15 XMIT RING
A-8	STA3 XMIT RING	A-45	STA7 XMIT TIP	B-8	STA11 XMIT RING	B-45	STA15 XMIT TIP

Table 22. Rear Transition Module RJ-21X Pin Assignments (Sheet 2 of 2)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A-33	STA3 XMIT TIP	A-21		B-33	STA11 XMIT TIP	B-21	
A-9		A-46		B-9		B-46	
A-34		A-22	STA8 RCV RING	B-34		B-22	STA16 RCV RING
A-10	STA4 RCV RING	A-47	STA8 RCV TIP	B-10	STA12 RCV RING	B-47	STA16 RCV TIP
A-35	STA4 RCV TIP	A-23	STA8 XMIT RING	B-35	STA12 RCV TIP	B-23	STA16 XMIT RING
A-11	STA4 XMIT RING	A-48	STA8 XMIT TIP	B-11	STA12 XMIT RING	B-48	STA16 SMIT TIP
A-36	STA4 SMIT TIP	A-24		B-36	STA12 SMIT TIP	B-24	
A-12		A-49		B-12		B-49	
A-37		A-25		B-37		B-25	
A-13	STA5 RCV RING	A-50		B-13	STA13 RCV RING	B-50	

Table 23. Rear Transition Module LED Descriptions

LED	Function
Health 	**Function:** Health (AdvancedTCA LED 2). The health is based on an aggregation of IPMI sensors, like board temperature and voltage. **Possible States:** OFF / GREEN / RED / AMBER GREEN: The board is healthy. RED: The board is not healthy. **Note:** If only one -48V power supply feed is connected to the chassis, this LED will stay Red. It is possible for the user to override the default IPMC behavior of the LED using AdvancedTCA FRU LED Control commands.
Out of Service 	**Function:** Out of Service (AdvancedTCA LED 1) **Possible States:** OFF: The board is running. RED or AMBER: The system is not fully operational. It is possible for a user to override the default IPMC behavior of the LED using AdvancedTCA FRU LED Control commands.
Hot Swap 	**Function:** Hot Swap as defined in the AdvancedTCA 3.0 specification **Possible States:** OFF: Normal operation or board not powered up BLUE: Safe to extract SHORT BLINK: Hot swap removal in process. LONG BLINK: Hot swap insertion in process. It is also possible for a user to override the default behavior of the LED using AdvancedTCA FRU LED Control commands.
Note: The other RTM LED is not used.	

6.5 Ejector Handles

Note: The ejector handles described in this section are on the main board and Rear Transition Module (RTM). The AdvancedMC also has an ejector handle that should not be used. **Do not remove the AdvancedMC from the main board.**

The Multimedia Blade and RTM each have two ejector handles (top and bottom) to help insert and eject the board from a chassis.

To insert the board into the chassis, open the ejector handles and push the board into the chassis card slot. Once the ejector handles slide into the top and bottom notches in the chassis, push both ejector handles toward the faceplate of the board until the handles click into place. The ejector handles provide a positive cam action, which ensures that the blade is properly seated. Once the bottom ejector handle is closed, the Hot Swap switch is engaged and this starts the normal power-on sequence.

To eject the board from the chassis, slide the bottom ejector handle mechanism so it releases from the faceplate and then gently pull the bottom ejector handle away from the faceplate. When the lower ejector handle is disengaged from the faceplate, the Hot Swap switch is released and this starts the normal board shutdown process. This power down process is identified by the blinking blue Hot Swap LED. Once the Hot Swap LED turns solid blue, slide the top ejector handle mechanism so it releases from the faceplate, and use both ejector handles to disengage from the backplane.

Warning: Removing the Multimedia Blade before the Hot Swap LED is solid blue can lead to device corruption or failure.

7.0 Hardware Management

This chapter provides a high-level overview of the Intelligent Platform Management Interface (IPMI) implementation for the Dialogic® Multimedia Blade for AdvancedTCA® (hereinafter referred to as the Multimedia Blade), based on the PICMG* 3.0 and IPMI 2.0 specifications. The following topics are discussed:

- Overview
- Sensor Data Record (SDR)
- System Event Log (SEL)
- Field Replaceable Unit (FRU) Information
- IPMC Platform Event Filtering (PEF)
- Ejector Mechanism (Baseboard Only)
- Hot Swap LED (Baseboard Only)
- Resets
- Watchdog Timers (WDTs)
- FRU Payload Control
- OEM IPMI Commands

7.1 Overview

The Intelligent Platform Management Controller (IPMC) is based on the Renesas* microcontroller, model HD64F2166. It supports up to six I^2C buses (master/slave), three serial-ports, a low-pin count (LPC) interface, and an A/D and DAC. This microcontroller is also capable of addressing up to 16 MBytes and has external access to 2 MBytes of flash memory; 512 KBytes of SRAM and a 64 KByte serial EEPROM.

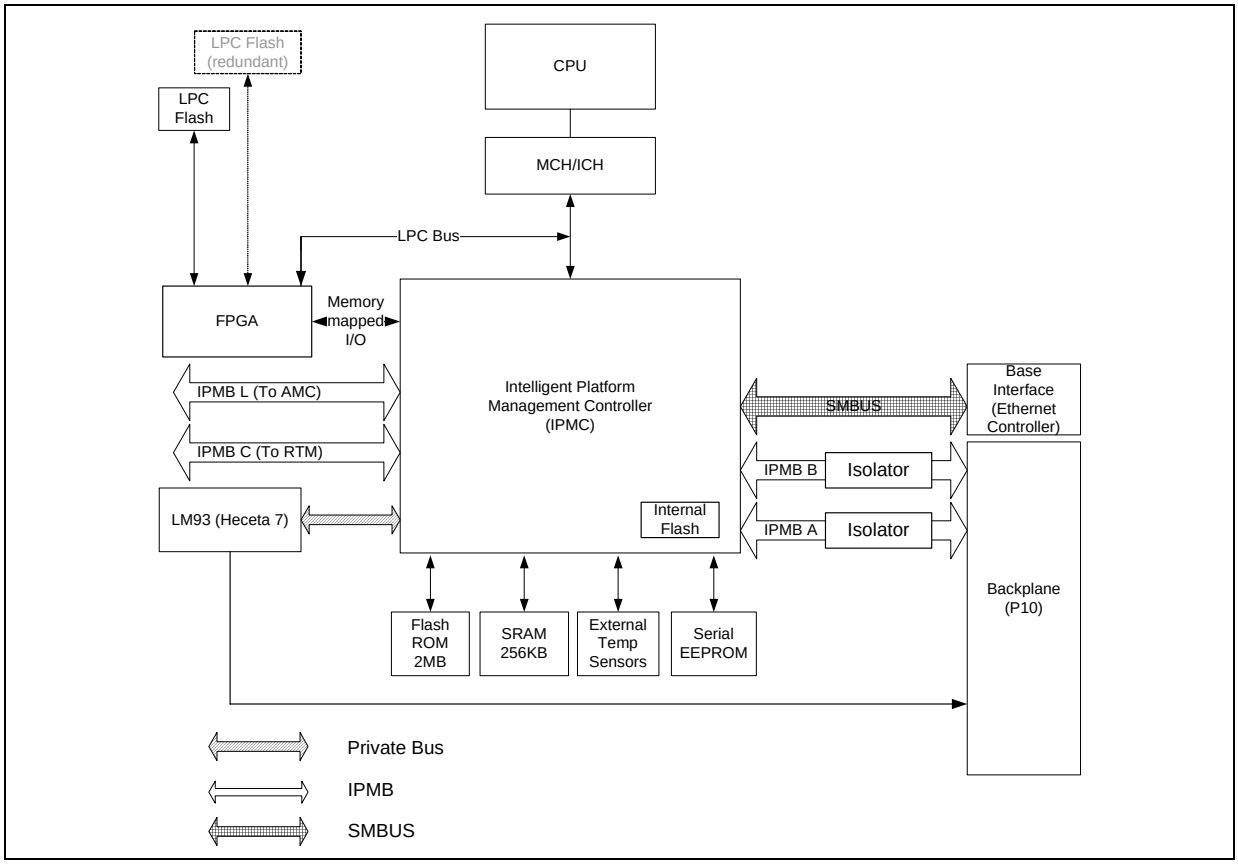
The IPMC implementation on the Multimedia Blade conforms to PICMG 3.0 R1.0 ECN001 specifications. The manageability framework for the Multimedia Blade is developed per the IPMI v2.0 Specification, which includes the following:

- Intelligent Platform Management Controller (IPMC)
- IPMI messaging, commands, and abstractions
- IPMI channels and sessions
- Sensor Data Records (SDRs) and SDR Repository
- FRU information
- Autonomous event logging
- System Event Log (SEL), holding at least 5,000 entries
- IPMI Hardware Watchdog Timer
- Platform Event Filtering (PEF)
- Serial Over LAN (SOL)— ability to redirect the system serial controller over LAN to a remote console

- Proprietary management features including:
  - — Fault Resilient Booting
  - — BIOS logging of power-on self-test (POST) progress and POST errors
  - — Snooping of port 80h POST code
  - — Advanced Configuration and Power Interface (ACPI)
  - — Serial port buffering (at least one screen worth of data)
  - — Rear Transition Module (RTM) management support

The high-level architecture of the baseboard management for the Multimedia Blade is represented in Figure 17.

Figure 17. IPMC Block Diagram



The main processors communicate with the IPMC using the Keyboard Controller Style (KCS) interface. Two KCS interfaces are available for the BIOS to communicate with the IPMC. The BIOS uses the SMS interface for normal communication and the SMM interface when executing code under Systems Management Mode (SMM). The base address of the LPC interface for SMS is 0xCA2/CA3 for the SMS and 0xCA8/CAC for SMM operation. The BIOS is also able to communicate with the IPMC via the KCS interface for POST error logging purposes, fault resilient purposes, and critical interrupts.

The Field Replaceable Unit (FRU) inventory information, SEL events, and SDR information is stored in an external Serial EEPROM. Having the SEL and logging functions managed by the IPMC helps ensure that "post-mortem" logging information is available even if the system processor becomes disabled.

IPMB isolators on both IPMB buses are used to switch and isolate a faulty IPMB bus on a board from the backplane IPMB bus connections. Where possible, the IPMC activates the redundant IPMB bus to re-establish system management communication to report the fault.

The on-board DC voltages are monitored by the LM93 device, manufactured by National Semiconductor*. The IPMC queries the LM93 over a local system management I^2C bus. External CPU thermal diodes and PROCHOT signals are connected to this device to ensure report thermal events.

The FPGA controls the enabling and monitoring of power good signals from all on-board power converters. It also controls power sequencing to ensure that all of the converters power up in the correct order to prevent any latch-up or damage to a device. The FPGA itself is also used to expand the GPIO capabilities of the IPMC management circuitry due to the limited number of GPIO's supported by the IPMC. The LPC interface between the FPGA and ICH is used to monitor the Port 80 codes during power up. In the event of a board failing to power up, a user can query the last five Port 80 codes stored in the FPGA registers using an Intel OEM IPMI command.

To increase the reliability of the Multimedia Blade, a watchdog timer is implemented.

Table 24 lists the main components that perform hardware monitoring of voltages and timers.

Table 24. Hardware Monitoring Components

Component	Function	Monitors
Intelligent Platform Management Controller	WDT #1	IPMI watchdog timer. If the timer expires (times out), it executes a pre-determined action (hard reset, power down, power cycle, or do nothing) and generates an IPMI SEL event.
Intelligent Platform Management Controller	WDT #2	IPMI hardware watchdog timer. This WDT is strobed by IPMC firmware. It has a 1 second timeout with a 500 ms strobe. If the WDT expires, it isolates the Multimedia Blade from the backplane IPMB buses, and resets the IPMC.
LM93	Voltage/ Temperature	On-board voltages/temperature, processor thermal diodes, CPU "PROCHOT", and processor VID.
Various Devices	Temperature	Monitor on-board temperature. The board is equipped with seven on-board temperature sensors. See Figure 18, "On-board Temperature Sensor Locations" on page 102.

7.2 Sensor Data Record (SDR)

Sensor Data Records (SDRs) contain information about the type and number of sensors in the baseboard, sensor threshold support, event generation capabilities, and the types of sensor readings handled by system management firmware. SDRs can be queried using Device SDR commands to the firmware.

Table 25, "Baseboard IPMC Hardware Sensor and Events" on page 86 lists the sensor identification numbers and information regarding the sensor type, name, supported thresholds, assertion and deassertion information, and a brief description of the sensor purpose. Following Table 25, information about the RTM and AdvancedMC is given in Table 26 and Table 27. The following sections describe the information contained in these tables.

7.2.1 Sensor Type

The sensor type references the values enumerated in the Sensor Type Codes table in the IPMI 2.0 specification. It provides the context in which to interpret the sensor, such as the physical entity or characteristic that is represented by this sensor.

7.2.2 Event/Reading Type

The Event/Reading Type references values from the Event/Reading Type Code Ranges and Generic Event/Reading Type Codes tables in the IPMI 2.0 specification. Note that digital sensors are a specific type of discrete sensors, which have only two states.

7.2.3 Event Thresholds/Triggers

Event thresholds are supported event generating thresholds for threshold type sensors.

- [u,l][nr,c,nc]: upper nonrecoverable, upper critical, upper noncritical, lower nonrecoverable, lower critical, lower noncritical
- uc, lc: upper critical, lower critical

Event triggers are supported event generating offsets for discrete type sensors. The offsets can be found in the Generic Event/Reading Type Codes or Sensor Type Codes tables in the IPMI specification, depending on whether the sensor event/reading type is generic or a sensor specific response

7.2.4 Assertion/De-assertion Enables

Assertion and de-assertion indicators reveal what type of events this sensor can generate:

- As: Assertion
- De: De-assertion

7.2.5 Readable Value/Offsets

Readable value indicates the type of value returned for threshold and other non-discrete type sensors.

Readable offsets indicate the offsets for discrete sensors that are readable via the Get Sensor Reading command. Unless otherwise indicated, all event triggers are readable, In other words, readable offsets consist of the reading type offsets that do not generate events.

7.2.6 Event Data

This is the data that is included in an event message generated by the associated sensor. For threshold-based sensors, the following abbreviations are used:

- R: Reading value
- T: Threshold value

7.2.7 Health LED On/Off

This indicates events that turn the Health LED on or off. The following abbreviations are used to indicate how the LED is affected:

- +: Event turns on the LED. Payload power cycle is required to turn it off.
- C/D: critical event assertion turns LED on. Deassertion turns it off.

7.2.8 Rearm Sensors

The “rearm” is a request for the event status for a sensor to be rechecked and updated upon a transition between good and bad states. Rearming the sensors can be done manually or automatically. This column indicates the type supported by the sensor. The following abbreviations are used in the Rearm column of Table 25 to describe a sensor:

- A: Auto rearm
- M: Manual rearm

7.2.9 Standby

Some sensors operate on standby power. These sensors may be accessed and/or generate events when the payload power is off, but DC power (48 Volt) is present.

7.2.10 Polling Time

The polling time is the interval time in seconds that the IPMC controller reads the sensor to determine the value of the sensor. For example, if the polling time is 5 seconds, the value of that sensor is read every 5 seconds and compared against the SDR thresholds. If the sensor value exceeds the threshold limits, then a SEL event is generated. Some sensors generate events asynchronously and do not need to be polled. Typically, only analog sensors require polling.

7.2.11 Managing Device

The managing device is the hardware device that a sensor is connected to.

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 1 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data Byte 2	Event Data Byte 3	Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
Power Unit Status	01h	Power Unit 09h	Sensor Specific 6Fh	00h	Power Fault Reg #1	Power Fault Reg #2			Power Off	As	–	A	X		
				01h					Power Cycle						
				05h					Soft Power Control Fault						
				06h					Power Unit Failure						
IPMC Watchdog	03h	Watchdog 2 23h	Sensor Specific 6Fh	00h	As per IPMI Spec.	FFh			Timer Expired	As & De	–	A	X		
				01h					Hard Reset						
				02h					Power Down						
				03h					Power Cycle						
				08h					Timer Interrupt						
Platform Security Violation	04h	Platform Security Violation Attempt 06h	Sensor Specific 6Fh	00h	FFh	FFh			Secure Mode Violation attempt	As	–	A	X		
				05h					Out-of-band access password violation						

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 2 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data Byte 2	Event Data Byte 3	Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
POST Error	06h	System Firmware Progress (formerly POST error) 0Fh	Sensor Specific 6Fh	00h (POST error)	99h	99h			BIOS Checksum	As	–	A	–		
					FEh	00h			Timer Count Read/Write						
					FEh	01h			CMOS Battery						
					FEh	02h			CMOS Diagnosis Status						
					FEh	03h			CMOS Checksum						
					FEh	04h			CMOS Memory Size						
					FEh	05h	+		RAM Read/Write Test						
					FEh	06h			CMOS Date/Time						
					FEh	07h	+		Clear CMOS Jumper						
					FEh	08h	+		Clear Password Jumper						
					FEh	09h	+		Manufacturing Jumper						
					FEh	0Ah	+		BMC in update						
					FEh	0Bh	+		BMC Response Fail						
					FEh	0Ch	+		Event Log Full						
					FEh	10h	+		Configuration error on DIMM pair 0						
					FEh	11h	+		N/A						
					FEh	12h	+		No system memory is physically installed or fails to access any DIMMs SPD data.						

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 3 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data Byte 2	Event Data Byte 3	Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
Critical Int	07h	Critical Interrupt 13h	Sensor Specific 6Fh	04h	FFh	FFh	+			As	-	A	-		
				05h			+								
				08h			+								
				0ah			+								
Memory	08h	Memory 0Ch	Sensor Specific 6Fh	00h	FFh	FFh			Correctable ECC	As	-	A	-		
				01h			+		Uncorrectable ECC						
Logging Disabled	09h	Event Logging Disabled 10h	Sensor Specific 6Fh	00h	DIMM id	FFh			Correctable Memory Error Logging Disabled	As	-	A	-		
				02h	FFh	FFh			Log Area Reset/Cleared						
Session Audit	0Ah	Session Audit 2Ah	Sensor Specific 6Fh	00h	FFh As per IPMI Spec.	FFh As per IPMI Spec.			Session Activation	As	-	A	X		
				01h					Session Deactivation						

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 4 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data		Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
					Byte 2	Byte 3									
FW Update	0Fh	OEM C0h	OEM 70h	00h	Major FW Version	Minor FW Version			Roll-Back FW Image Captured	As	–	A	X		
				01h	FFh	FFh			Roll-Back SDR Captured						
				02h	Major FW Version	Minor FW Version			Staged Image Registered						
				03h	FFh	FFh			Staged SDR Registered						
				04h	Major FW Version	Minor FW Version			Roll-Back via Switch						
				05h					Roll-Back via Command						
				06h					FW Roll-Back due to operation check failure						
				07h					Roll-Back Complete						
				08h					Staged FW Update Completed						
				09h					Direct FW Update Completed						
SBC +1.05V Vtt (DDR2 Vtt)	10h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
SBC +1.1V (NIC Core)	11h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
SBC +1.2V SAS (SAS I/O)	12h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
SBC +1.5V (Chipset Core)	13h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 5 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data		Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
					Byte 2	Byte 3									
SBC +1.5V SUS (Suspend Pwr)	14h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	5	LM93
SBC +1.8V LAN (NIC)	15h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
SBC +1.8V MEM (DDR2 Core)	16h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
SBC +3.3V	17h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	IPMC
SBC +3.3V SUS (Suspend Pwr)	18h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	5	LM93
SBC +5.0V	19h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
SBC +5.0V USB	1Ah	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
SBC +12.0V	1Bh	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
AdvancedMC +12.0V	1Ch	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
AdvancedMC CurSense	1Dh	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
SBC +1.1V SUS (NIC SUS)	20h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	IPMC
SBC +1.8V (NIC SUS)	21h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	IPMC
SBC RTC Vcc	22h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 6 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data		Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
					Byte 2	Byte 3									
SBC +5V SUS (Suspend Power)	23h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	IPMC
SBC Temp 1 (Internal LM93, between CPUs)	30h	Temp 01h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	0.5	LM93
SBC Temp 2 (Near MCH)	31h	Temp 01h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	0.5	ADT7483
SBC Temp 3 (Near DIMM 1)	32h	Temp 01h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	0.5	ADT7483
SBC Temp 4 (Near DIMM 2)	33h	Temp 01h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	0.5	ADT7483
SBC Temp 5 (Under HDD)	34h	Temp 01h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	0.5	ADT7483
SBC Temp 6 (Near SAS)	35h	Temp 01h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	0.5	ADT7483
SBC Temp 7 (Near NIC)	36h	Temp 01h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	0.5	ADT7483
BIOS FWH0 Flash	54h	OEM C0h	Digital Discrete 03h	00h	FFh	FFh			State Deasserted	As	–	A	–		
				01h					State Asserted						
BIOS FWH1 Flash	55h	OEM C0h	Digital Discrete 03h	00h	FFh	FFh			State Deasserted	As	–	A	–		
				01h					State Asserted						

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 7 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data Byte 2	Event Data Byte 3	Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
ACPI State	82h	System ACPI Power State 22h	Sensor Specific 6Fh	00h					S0 / G0						
				01h					S1						
				04h					S4						
				05h					S5 / G2						
				0Bh					Legacy On						
				0Ch					Legacy Off						
System Event	83h	System Event 12h	Sensor Specific 6Fh	01h	FFh As per IPMI Spec.	FFh			OEM System Boot Event	As	–	A	–		
				04h		FFh			PEF Action						
Button	84h	Button 14h	Sensor Specific 6Fh	00h	FFh	FFh			Reserved	As	–	A	X		
				01h					Reserved						
				02h					Reserved						
SMI Timeout	85h	SMI Timeout F3h	Digital Discrete 03h	00h	FFh	FFh			State Deasserted	As	–	A	–		
				01h					State Asserted						
NMI State	87h	OEM C0h	Digital Discrete 03h	00h	FFh	FFh			State Deasserted	–	–	–	–		
				01h					State Asserted						
SMI State	88h	OEM C0h	Digital Discrete 03h	00h	FFh	FFh			State Deasserted	–	–	–	–		
				01h			+		State Asserted						

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 8 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data Byte 2	Event Data Byte 3	Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
IPMC Watchdog Overflow	89h	OEM C0h	Digital Discrete 03h	01h	FFh	FFh			State Asserted	–	–	–	–		
SBC FRU Hot Swap	8Ah	PICMG Hot Swap Event F0h	Sensor Specific 6Fh	00h					M0 – FRU not installed						
				01h					M1 – FRU inactive						
				02h					M2 – FRU activation request						
				03h					M3 - FRU activation in progress						
				04h					M4 - FRU active						
				05h					M5 – FRU deactivation request						
				06h					M6 - FRU deactivation in Progress						
				07h					M7 - Communication lost						

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 9 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data Byte 2	Event Data Byte 3	Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
RTM FRU Hot Swap	8Bh	PICMG Hot Swap Event F0h	Sensor Specific 6Fh	00h	FFh	FFh			M0 – FRU not installed	As	–	A	X		
				01h					M1 – FRU inactive						
				02h					M2 – FRU activation request						
				03h					M3 - FRU activation in progress						
				04h					M4 - FRU active						
				05h					M5 – FRU deactivation request						
				06h					M6 - FRU deactivation in Progress						
				07h					M7 - Communication lost						
AdvancedMC FRU Hot Swap	8Ch	PICMG Hot Swap Event F0h	Sensor Specific 6Fh	00h	FFh	FFh			M0 – FRU not installed	As	–	A	X		
				01h					M1 – FRU inactive						
				02h					M2 – FRU activation request						
				03h					M3 - FRU activation in progress						
				04h					M4 - FRU active						
				05h					M5 – FRU deactivation request						
				06h					M6 - FRU deactivation in Progress						
				07h					M7 - Communication lost						

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 10 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data Byte 2	Event Data Byte 3	Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
IPMB-0 Link State	8Dh	PICMG Physical IPMB-0 Link F1h	Sensor Specific 6Fh	00h	FFh	FFh			IPMB A & B disabled	As	–	A	X		
				01h					IPBM A enabled IPMB B disabled						
				02h					IPMB A disabled IPMB B enabled						
				03h					IPMB A & B enabled						
CPU 1 Status	90h	Processor 07h	Sensor Specific 6Fh	00h	FFh	FFh	+		IERR	As & De	–	M	X	2	IPMC
				01h			+		Thermal Trip						
				02h					FRB1						
				03h					FRB2						
				04h					FRB3						
				05h					Config Error						
				07h				+	Presence						
				08h					Disabled						

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 11 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data Byte 2	Event Data Byte 3	Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
CPU 2 Status	91h	Processor 07h	Sensor Specific 6Fh	00h	FFh	FFh	+		IERR	As & De	–	M	X	2	IPMC
				01h			+		Thermal Trip						
				02h					FRB1						
				03h					FRB2						
				04h					FRB3						
				05h					Config Error						
				07h				+	Presence						
				08h					Disabled						
CPU 1 Temp	92h	Temp 01h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	0.5	LM93
CPU 2 Temp	93h	Temp 01h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	X	0.5	LM93
CPU 1 Vcc	94h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
CPU 2 Vcc	95h	Voltage 02h	Threshold 01h	R, T	–	–	C	D	[u,l][nr,c,nc]	As & De	Analog	A	–	5	LM93
CPU 1 ThermCtrl	96h	Temp 01h	Discrete 07h	00h					Transitioned to OK	As & De	–	M	–	2	LM93
				01h					Transitioned to Non-Critical from OK						
CPU 2 ThermCtrl	97h	Temp 01h	Discrete 07h	00h					Transitioned to OK	As & De	–	M	–	2	LM93
				01h					Transitioned to Non-Critical from OK						

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 12 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data Byte 2	Event Data Byte 3	Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
CPU 1 VRD Hot	98h	Temp 01h	Discrete 07h	00h			D		Transitioned to OK	As & De	Analog	A	–	2	LM93
				01h			+		Transitioned to Non-Critical from OK						
CPU 2 VRD Hot	99h	Temp 01h	Discrete 07h	00h			D		Transitioned to OK	As & De	Analog	A	–	2	LM93
				01h			+		Transitioned to Non-Critical from OK						
CPU Config Error	9Ah	Processor 07h	Generic 03h	01h	FFh	FFh			State Asserted	As & De	Discrete	A	–		
DIMM 1	E0h	Slot Connector 21h	Sensor Specific 6Fh	00h	07h	Slot Number			Fault Status Asserted	As	–	A	–		
				02h					Device Installed						
				08h					Disabled						
DIMM 2	E1h	Slot Connector 21h	Sensor Specific 6Fh	00h	07h	Slot Number			Fault Status Asserted	As	–	A	–		
				02h					Device Installed						
				08h					Disabled						
Feed Fail Status	E2h	OEM C1h	Digital Discrete 03h	00h	FFh	FFh	D		State Deasserted	–	–	–	X		
				01h			+		State Asserted						
Base Link A Sts	E3h	LAN 27h	Sensor Specific 6Fh	00h	FFh	FFh			Heartbeat Lost (Link Down)	As & De	–	M	X	1	82571EB
				01h					Heartbeat (Link Up)						
Base Link B Sts	E4h	LAN 27h	Sensor Specific 6Fh	00h	FFh	FFh			Heartbeat Lost (Link Down)	As & De	–	M	X	1	82571EB
				01h					Heartbeat (Link Up)						

Table 25. Baseboard IPMC Hardware Sensor and Events (Sheet 13 of 13)

Sensor Name	Sensor No.	Sensor Type	Event / Reading	Event Offset ED1 [3:0]	Event Data		Health Led On	Health Led Off	Event	Assert / De-assert	Readable Value / Offsets	Rearm	Standby	Polling Time, (seconds)	Managing Device
					Byte 2	Byte 3									
Fabric Lnk C St	E5h	LAN 27h	Sensor Specific 6Fh	00h	FFh	FFh			Heartbeat Lost (Link Down)	As & De	–	M	X	1	82571EB
				01h					Heartbeat (Link Up)						
Fabric Lnk D St	E6h	LAN 27h	Sensor Specific 6Fh	00h	FFh	FFh			Heartbeat Lost (Link Down)	As & De	–	M	X	1	82571EB
				01h					Heartbeat (Link Up)						
Fabric Lnk E St	E7h	LAN 27h	Sensor Specific 6Fh	00h	FFh	FFh			Heartbeat Lost (Link Down)	As & De	–	M	X	1	82571EB
				01h					Heartbeat (Link Up)						
Fabric Lnk F St	E8h	LAN 27h	Sensor Specific 6Fh	00h	FFh	FFh			Heartbeat Lost (Link Down)	As & De	–	M	X	1	82571EB
				01h					Heartbeat (Link Up)						

Table 26. RTM MMC Hardware Sensor and Events

Sensor Name	Sensor No.	Sensor Type	Event / Reading Type	Generated Event Offsets	Rearm	Poll Period
IPMI Watchdog Timer	03h	Watchdog 2 23h	Sensor Specific 6Fh	00h-Timer expired 01h-Hard Reset 02h-Power Down 03h-Power Cycle 08h-Timer Interrupt	Auto	N/A
Power Unit Status	05h	Power Unit 09h	Sensor Specific 6Fh	04h-AC Lost 05h-Soft Power Control Failure 06h-Power Unit Failure Detected	Auto	N/A
1.0V	10h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
1.2V (not present in the IP-only RTM)	11h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
1.5V (not present in the IP-only RTM)	12h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
1.8V (not present in the IP-only RTM)	13h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
2.5V	14h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
RTM Core (not present in the IP-only RTM)	15h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
3.3V (not present in the IP-only RTM)	16h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
3.3V Management Power	17h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
Temperature Sensor 1	30h	Temp. 01h	Threshold 01h	See Table 28	Auto	0.5 Second
Temperature Sensor 2	31h	Temp. 01h	Threshold 01h	See Table 28	Auto	0.5 Second
Temperature Sensor 3	32h	Temp. 01h	Threshold 01h	See Table 28	Auto	0.5 Second
Temperature Sensor 4	33h	Temp. 01h	Threshold 01h	See Table 28	Auto	0.5 Second
Module Hot Swap	8Eh	OEM-AMC.0 F2h	Sensor Specific 6Fh	00h-Module Handle Closed 01h-Module Handle Opened 02h-Quiesced	Auto	2.5 Seconds

Table 27. AdvancedMC MMC Hardware Sensor and Events

Sensor Name	Sensor No.	Sensor Type	Event / Reading Type	Generated Event Offsets	Rearm	Poll Period
Watchdog Timer	03h	Watchdog 2 23h	Sensor Specific 6Fh	00h-Timer expired 01h-Hard Reset 02h-Power Down 03h-Power Cycle 08h-Timer Interrupt	Auto	N/A
Power Unit Status	05h	Power Unit 09h	Sensor Specific 6Fh	04h-AC Lost 05h-Soft Power Control Failure 06h-Power Unit Failure Detected	Auto	N/A
1.0V	10h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
1.1V	11h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
1.2V	12h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
1.5V	13h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
1.8V	14h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
1.8B	15h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
2.5V	16h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
3.3V Management Power	17h	Voltage 02h	Threshold 01h	See Table 28	Auto	5 Seconds
Temperature Sensor 1	30h	Temp. 01h	Threshold 01h	See Table 28	Auto	0.5 Second
Temperature Sensor 2	31h	Temp. 01h	Threshold 01h	See Table 28	Auto	0.5 Second
Temperature Sensor 3	32h	Temp. 01h	Threshold 01h	See Table 28	Auto	0.5 Second
Temperature Sensor 4	33h	Temp. 01h	Threshold 01h	See Table 28	Auto	0.5 Second
Module Hot Swap	8Eh	OEM-AMC.0 F2h	Sensor Specific 6Fh	00h-Module Handle Closed 01h-Module Handle Opened 02h-Quiesced	Auto	2.5 Seconds

Table 28. Threshold Sensor Offsets Generated by MMC (Sheet 1 of 2)

Offset	Meaning
00h	Lower Non-Critical - going low
02h	Lower Critical - going low
04h	Lower Non-Recoverable - going low

Table 28. Threshold Sensor Offsets Generated by MMC (Sheet 2 of 2)

07h	Upper Non-Critical - going high
09h	Upper Critical - going high
0Bh	Upper Non-Recoverable - going high

7.3 System Event Log (SEL)

The System Event Log (SEL) is the collection of events that are generated by the IPMC. Event logs are stored in non-volatile memory (Serial EEPROM). This resides on the board and allows better tracking of error conditions on the baseboard when it is moved from chassis to chassis. Having the SEL and logging functions managed by the IPMC helps ensure that post-mortem logging information is available should a failure occur that disables the system's processor(s). In the Multimedia Blade, the 128kB of flash memory dedicated for SEL entries can store 6,552 entries. (128KB / 20 Bytes per SEL = 6553.6 entries less the SEL Cleared Timestamp overhead, brings it down to 6,552 entries.)

The list of all IPMC generated SEL events can be found in Table 25, Table 26, and Table 27.

Events are normally forwarded to the shelf manager and logged to SEL on the board. If SEL storage on the board is full, new events are forwarded to the shelf manager, but are not logged in SEL on the board. The user needs to ensure on-board SEL log events are backed up (if desired) and SEL log cleared if the log is full. If the on-board SEL is full, new events coming into the IPMC SEL will only be forwarded to the shelf manager. There are two ways to clear the on-board SEL. One method is to use the Event Log Configuration in the BIOS (see Chapter 9.0, "BIOS Settings"). The second method is to use Issue Reserve SEL and Issue Clear SEL IPMI commands.

Events may be received while the SEL is being cleared. The IPMC implements an event message queue to avoid messages being lost. Messages are not overwritten once they are stored in the queue.

To get all the SEL events stored on the board, issue a Get SEL Info IPMI command to find out the number of IPMC SEL entries. Then loop on the Get SEL Entry IPMI command for number of SEL entries obtained by the Get SEL Info command.

A set of IPMI commands allows the SEL to be read and cleared and allows events to be added to the SEL. The IPMI commands used for adding events to the SEL are Platform Event Message, Add SEL Entry, and Partial Add Entry. See Appendix A, "Supported IPMI Commands".

7.3.1 Analog Sensors

If a lower non-critical or upper non-critical threshold is exceeded, it raises a minor alarm. If a lower critical or upper critical threshold is exceeded, it raises a major alarm. The Multimedia Blade doesn't support any Upper/Lower Non-recoverable thresholds.

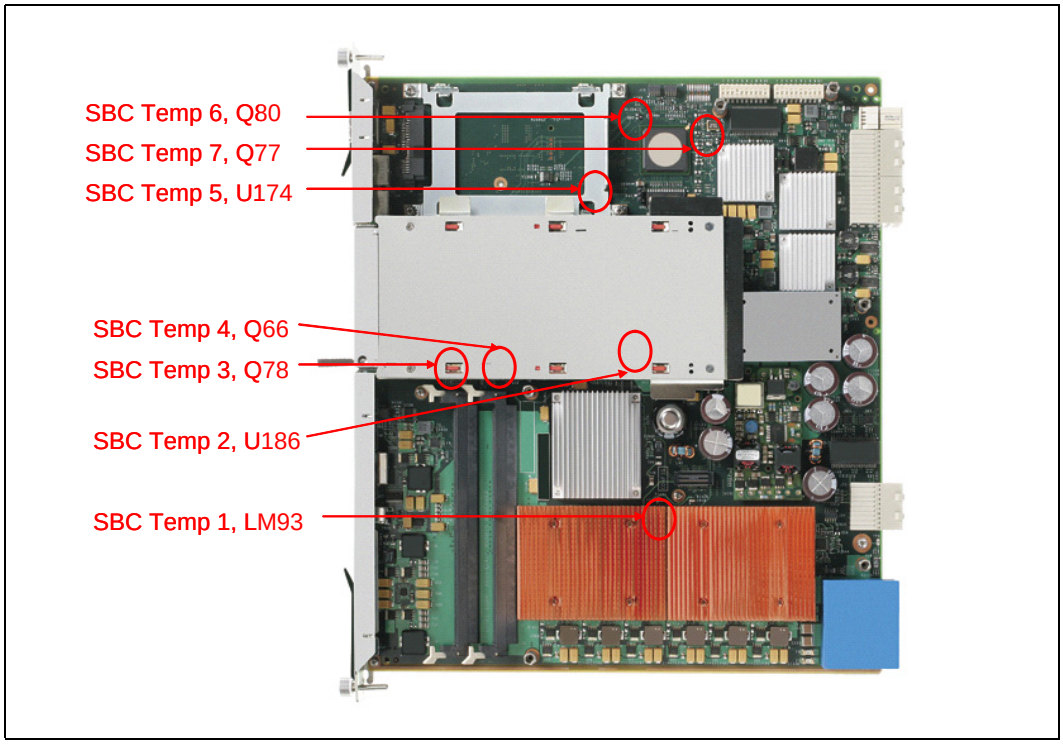
The health LED is turned solid red only when critical or non-recoverable thresholds (major alarm) are exceeded. However, for any of the categories above, the IPMC forwards the events to the shelf manager to log in the shelf manager's SEL.

7.3.2 Temperature Sensor Locations

The Multimedia Blade baseboard is equipped with seven temperature sensors. Refer to Figure 18 for the location of all the on-board temperature sensors.

Note: In Figure 18, the "SBC Temp *n*" is the name of the on-board temperature sensor followed by the reference designator on the board.

Figure 18. On-board Temperature Sensor Locations



7.3.3 Processor Events

The following processor events are supported:

- **Processor presence** - Indicates if the processor is present in the socket.
- **PROCHOT** - Indicates if the processor has entered automatic thermal throttling mode (50% frequency duty cycle) due to high CPU die temperature.
- **IERR** - The processor asserts IERR as the result of an internal error. Assertion of IERR# is usually accompanied by a SHUTDOWN transaction on the processor system bus.
- **Thermal trip -** A thermal trip error indicates that the processor junction temperature has reached a level where permanent silicon damage may occur. Upon THERMTRIP assertion, the IPMC powers down the boards. PROCHOT is asserted before THERMTRIP asserts.
- **FRB-3 Timeout** - The FRB-3 algorithm is used to detect whether the boot strap processor is healthy and can run the BIOS. The default FRB-3 timer is 10 seconds. The assumed initial condition is that both processors are enabled. The basic algorithm is followed on each power up or system reset as follows:

1. At power up/reset, the IPMC starts an internal FRB-3 timer.
2. In a good system, the BIOS issues the IPMI Set Watchdog Timer (WDT #1) command with the "timer use" byte configured for the BIOS. When the IPMC receives this command, the IPMC starts the IPMI Watchdog Timer, and then stops the internal FRB3 timer. At this point, the FRB-2 phase starts.

3. In a failing system, the BIOS does not issue the IPMI Set WDT command to the IPMC, and the IPMC FRB-3 timer expires.
4. The IPMC logs an FRB-3 failure event against the failing processor sensor, logs a processor disable event against that processor sensor, then causes the payload power to reset.

7.3.4 DIMM Memory Events

The Memory Controller Hub (MCH) has logic built into the hardware to detect and correct correctable errors and detect uncorrectable errors. For either type of error, an SMI is generated to the processor, so that the BIOS can take the appropriate actions. On the Multimedia Blade, such action includes detecting the error and sending a memory error event message to the IPMC firmware over the KCS interface.

7.3.4.1 Correctable Errors

MCH detects and corrects all 4-bit errors (S4EC) and generates an SMI to BIOS. The BIOS detects the cause of the error and sends an event to the IPMC. DIMM pair information is also sent to the IPMC as part of the event.

7.3.4.2 Single-Bit ECC Errors

The system detects, corrects, and logs correctable errors as long as these errors occur infrequently (the system should continue to operate without a problem). Occasionally, correctable errors are caused by a persistent failure of a single component. Although these errors are correctable, continual calls to the error logger can affect system performance, preventing further useful work.

For this reason, the system counts certain types of correctable errors and disables reporting if errors occur too frequently. Error correction remains enabled, but calls to the error handler are disabled. This allows the system to continue running, despite a persistent correctable failure.

The BIOS adds an entry to the event log to indicate that logging for that type of error has been disabled. The system BIOS implements this feature for correctable bus errors. If 10 errors occur within an hour, the corresponding error handler disables further reporting of that type of error. The BIOS re-enables logging and SMIs the next time the system is rebooted.

7.3.4.3 Uncorrectable Errors

The MCH detects 8-bit errors (D4ED) and generates SMI to the BIOS. If the error is in a data area, the BIOS generates an event to the IPMC. If the error is in a code area, the BIOS itself may be hung and it may not be possible to detect whether an uncorrectable error has occurred.

If the OS or application executing on the processor has set up a watchdog timer with IPMC and it times out, a WDT timeout event is sent out and the IPMC takes the action that has been set up in the WDT (hard reset, power down, power cycle or do nothing).

7.3.5 System Firmware Progress (POST Error)

The BIOS performs a power-on self-test (POST) at initialization and logs errors detected to the SEL by generating a System Firmware Progress (sensor type 0Fh) event. The following is a list of POST errors for which events are generated:

- Timer count read/write error
- CMOS battery error

- CMOS diagnostics status error
- CMOS checksum error
- CMOS memory size error
- RAM read/write test error
- CMOS date/time error
- Clear CMOS jumper
- Clear password jumper
- Event log full error
- No physical memory installed or access to the DIMM's SPD data failed

7.3.6 Port 80h POST Codes

As the BIOS goes through its initialization process, it sends progress codes to port 80h. These codes are useful for test debug purposes. For remote management purposes, the IPMC firmware provides the capability to snoop and capture up to five consecutive codes written to port 80h. These codes are captured automatically by the IPMC firmware when the board goes through either cold or warm reset or by an explicit IPMI command received by the IPMC via the available interfaces (IPMB or KCS).

7.3.7 IPMB Link Sensor

The Multimedia Blade provides two IPMB links to increase communication reliability with the shelf manager and other IPM devices on the IPMB bus. In the event of any link state changes, the events are written to the SEL. The IPMC monitors the bus for any link failure and isolates itself from the bus if it detects that it is causing errors on the bus. Events are sent to signify the failure of a bus or, conversely, the recovery of a bus.

7.3.8 FRU Hot Swap

The Hot Swap event message conveys the current state of the FRU, the previous state, and a cause of the state change as can be determined by the IPMC. Refer to the PICMG 3.0 specifications for further details on the hot swap state.

7.3.9 Ethernet Link Status

The IPMC firmware monitors the Ethernet link status (present/absent) of all available ports on the baseboard. In all, six Ethernet ports are available (two Base interface, four Fabric interface). One sensor is provided for each of the six Ethernet links. Events are generated by the IPMC firmware when the link status of the ports change.

7.3.10 Power Feeds

Each blade has redundant -48V feeds, from which local on-board voltages are generated. The IPMC firmware has the ability to detect errors on either of the feeds and generate an event. Only feed failure is detected; power feed number is not detected because such detection is assumed to be made at the chassis level.

7.3.11 IPMC Watchdog Timer Reset

As per PICMG 3.0 requirements (section 3.2.4.6.3), the Multimedia Blade provides a watchdog timer that can reset the IPMC if the firmware hangs. After a reset, when IPMC restarts, an event is generated indicating that the IPMC was reset due to watchdog timer expiration.

7.4 Field Replaceable Unit (FRU) Information

The ATCA Multimedia Platform includes the Multimedia Blade (main board), Rear Transition Module (RTM), and AdvancedMC. Each has its own Field Replaceable Unit (FRU) information. To access the FRU information for a specific board, use the following FRU IDs:

- FRU ID 0: main board
- FRU ID 1: RTM
- FRU ID 2: AdvancedMC

The following sections are definitions for the multirecords implemented by the firmware as part of the FRU data:

- Common Header
- Board Area
- Product Area
- Multirecord Area

7.4.1 Common Header

A common header is mandatory for all FRU Inventory Device implementations. It contains version information for the overall information and offsets to other information areas.

7.4.2 Board Area

The board area provides access to the following information, which is written when the board is manufactured:

- Manufacturing date and time
- Board manufacturer name
- Board product name
- Board serial number
- Board part number

7.4.3 Product Area

The product area provides access to following information, which is written when the board is manufactured:

- Manufacturer name (same as board manufacturer name)
- Product name (same as board product name)
- Product part/model number
- Product version
- Product serial number (same as board serial number)
- Asset tag

7.4.4 Multirecord Area

The multirecord area for the **main board** provides the following information:

- Board Point-to-Point Connectivity Record for E-Keying

- Carrier Information Table Record
- Carrier Activation and Current Management Record
- Carrier Point-to-Point Connectivity Record for E-Keying (AdvancedMC)
- Carrier Point-to-Point Connectivity Record for E-Keying (Rear Transition Module)
- AdvancedMC Point-to-Point Interface Record for E-Keying
- Rear Transition Module Point-to-Point Interface Record for E-Keying
- Record Type ID with version info for BIOS, Firmware, CPU and RAM Information
- Rear Transition MOdule Utilities Info Record

The multirecord area for the **RTM** provides the following information:

- RTM Info OEM Record
- Module Current Requirement Record
- AMC Point-to-Point Connectivity Record
- Utilities Info OEM Record

The multirecord area for the **AdvancedMC** provides the following information:

- AMC Info OEM Record
- Module Current Requirement Record
- AMC Point-to-Point Connectivity Record
- Utilities Info OEM Record

7.5 IPMC Platform Event Filtering (PEF)

The IPMC supports a maximum of 20 entries. Table 29 describes the factory pre-configured event filters and the unused entries that are software configurable. The only action available is to turn the Health LED to red.

Table 29. Factory Default Event Filters

Event Filter #	Offset Mask	Events
0	PCI PERR and SERR	Critical Interrupt
1	(any)	POST Memory resize
2	Clear CMOS jumper installed	System FW Progress
3	Various other jumpers installed	System FW Progress
4	Memory configuration errors	System FW Progress
5	Memory Read/Write test	System FW Progress
6	Config error, Uncorrectable ECC, or correctable ECC limit	Memory Sensor
7 to 19	Unused	

7.6 Ejector Mechanism (Baseboard Only)

In addition to captive retaining screws, the Multimedia Blade baseboard has two ejector mechanisms to provide a positive cam action, which ensures that the blade is properly seated and helps assist during blade extraction. The bottom ejector handle also has a micro switch that is connected to the IPMC to determine if the board has been properly inserted.

7.7 Hot Swap LED (Baseboard Only)

The Multimedia Blade baseboard supports one blue Hot Swap LED, mounted on the front panel. See Figure 16, "Front Panel LEDs" on page 75 for its location. This LED indicates when it is safe to remove the Multimedia Blade from the chassis. The on-board IPMC drives this LED to indicate the Hot Swap state. See Table 30.

When the lower ejector handle is disengaged from the faceplate, the Hot Swap switch on the board asserts a signal to the IPMC, and the IPMC moves from the M4 state to the M5 state. At the M5 state, the IPMC asks the shelf manager for permission to move to the M6 state. The Hot Swap LED indicates this state by blinking on for about 100 milliseconds, followed by 900 milliseconds in the off state. This occurs as long as the Multimedia Blade remains in the M5 state. Once permission is received from the shelf manager or higher level software, the Multimedia Blade moves to the M6 state.

Warning: Removing the Multimedia Blade prematurely can lead to corruption of files on the hard drive.

Table 30. Hot Swap LED

LED Status	Meaning
Off	Normal (active) status
Blinking Blue	Preparing for removal/insertion: Long blink indicates activation is in progress, short blink when deactivation is in progress.
Solid Blue	Ready for hot swap (extraction)

7.8 Resets

The two types of reset requests available on the Multimedia Blade are:

- Hard reset request (always results in a cold boot)
- Soft reset request (can result in either a warm or cold boot)

A hard reset request occurs whenever the processor Reset line is asserted, then deasserted. A soft reset occurs whenever an assertion occurs on the processor Init line. When a soft reset request occurs, the BIOS determines whether to initiate a warm boot while leaving main memory intact or a cold boot that clears memory. Table 31 summarizes hard and soft reset parameters.

Table 31. Reset Requests

Reset Request	Signal Activated	Type	Description
Hard	Reset	Full reboot	The payload is reset via the SYSRESET* input signal of the I/O Controller Hub (ICH). This reset results in a PCI reset, and thus the entire memory region is tested and the lower 8 MB of RAM is initialized. References to a cold reset imply a hard reset.
Soft	Init	Partial reboot	The payload is reset via the RCIN* input signal to the ICH. The ICH then maps this onto the INIT signal to the FW-Hub and CPUs and is asserted for 16 PCI clocks. A PCI reset is not generated in this scenario and the contents of the lower 8MB of RAM is maintained. References to a warm reset imply a soft reset.

Whenever the BIOS detects that the reset is either a hard reset or a cold boot, it specifically clears the memory location 40h:72h so it does not contain a 1234h. Under warm boot conditions, this memory location contains a 1234h (the developer's application writes this value in this location using /dev/mem when it is started). If a hard reset occurs, it is certain that the 40h:72h location contains a non-1234h value.

7.8.1 Reset Control Sources

Table 32 shows the sources for reset requests and the corresponding type of reset. Table 33 shows MMC payload reset sources, and Table 34 shows MMC reset sources.

Table 32. System Reset Sources and Actions

#	Reset Source	Payload Reset		IPMC Reset
		Soft	Hard	
1	Standby Power On Reset (SPOR)	--	--	Yes
2	IPMC Hardware Watchdog Timer	No	No	Yes
3	IPMC Exit Firmware Update Mode	No	No	Yes
4	IPMI BMC Cold Reset command	No	No	Yes
5	IPMI Chassis Control command (hard reset)	No	Yes	No
6	IPMI Watchdog timer expiration (w/ action configured for payload reset)	No	Yes	No
7	IPMI PEF action	No	Yes	No
8	Fault Resilient Booting (FRB3 failure)	No	Yes	No
9	Set Processor State command (upon disabling a processor)	No	Yes	No
10	Reset BIOS Flash Type	No	Yes	No
11	Front Panel Reset button	Yes	No	No
12	OS Warm Boot (restart)	Yes	No	No
13	PICMG Payload Activation (M4)	No	Yes	No
14	PICMG FRU Control command	Yes	Yes	No
15	Payload Power Off	--	Yes	No

Table 33. MMC Payload Reset Sources

Reset Source	Payload Reset	MMC Reset
Management power comes up	No (no input payload power)	Yes
MMC turns on board power	Yes	No
IPMI Chassis Control reset command	No	No
IPMI watchdog timer configured for reset	No	No
ATCA FRU Control (Cold Reset) command	Yes	No

Table 34. MMC Reset Sources

Reset Source	Payload Reset	MMC Reset
Management power comes up and ENABLE# asserted	No (no payload power)	Hard
Exit MMC firmware update mode	No	Soft
MMC hardware watchdog timer expiration	No	Hard
IPMI Cold Reset command	No	Hard

7.8.1.1 Front Panel Payload Reset

The Reset button is a momentary contact button on the front panel. Its signal is routed through the front panel connector to the IPMC. The Front Panel reset is mapped to generate a soft reset to the payload.

7.8.1.2 IPMI Commanded Reset

The IPMI Chassis Control command is supported and can be used to generate a hard reset to the payload.

The Intel OEM Set Processor State command, which is used by the BIOS during POST, also generates a hard reset when used to disable a processor as part of the Fault Resilient Booting (FRB) algorithm.

The PICMG FRU Control command can be used to generate either a hard or soft reset to the payload.

7.8.1.3 Watchdog Timer Expiration

The Watchdog Timer can be configured to cause a hard reset to the payload upon its expiration. Timeout and action can be configured in BIOS. For more information, see the *Intelligent Platform Management Interface Specification*, Version 2.0.

7.8.1.4 FRB3 Failure

Simultaneous with resetting the payload, the IPMC starts an internal FRB3 timer. If the BIOS does not start the IPMI Watchdog Timer with the usage set for BIOS/FRB2, the IPMC resets the system when the internal FRB3 timer expires.

7.8.2 IPMC Reset Control

Table 35 shows all the sources of IPMC resets and the actions by the system and the IPMC. The payload will not be reset by an IPMC reset except after a combined IPMC boot block and operational code update. In all other cases, the payload is not reset or powered down. The IPMC re-synchronizes itself to the state of the processor and power control signals it finds when it initializes.

Table 35. IPMC Reset Sources and Actions

#	Reset Source	System Reset	IPMC Reset
1	Standby Power On Reset	No (payload not up yet)	Yes
2	IPMC Hardware WDT Expiration	No	Yes
3	IPMC Exit Firmware Update mode	No	Yes

Table 35. IPMC Reset Sources and Actions (Continued)

#	Reset Source	System Reset	IPMC Reset
4	IPMI BMC Cold Reset command	No	Yes
5	IPMC boot block update followed by IPMC operational code update	Yes	Yes
6	IPMC operational code update (without updating boot block)	No	Yes

7.8.2.1 Standby Power On Reset

When the Multimedia Blade is inserted into a chassis, the IPMC is reset via the reset signal to the microcontroller. This is referred to as a Standby Power On Reset (SPOR).

7.8.2.2 IPMC Exit Firmware Update Mode

The IPMC firmware can be updated using firmware transfer commands through the LPC or IPMB interface. The IPMC automatically enters Firmware Transfer Mode if it detects that the Force Update signal is asserted during initialization or if the operation code checksum fails. Upon exit from Firmware Transfer Mode, the IPMC resets itself.

7.8.2.3 IPMI IPML Cold Reset Command

The IPMC firmware supports the ability to reset the IPMC via the IPMI Cold Reset command. For more information, see the *Intelligent Platform Management Interface Specification*, Version 2.0.

7.9 Watchdog Timers (WDTs)

There are two watchdog timers (WDTs) on the Multimedia Blade.

WDT #1 operates as per the IPMI version 2.0 Specification as IPMI Watchdog Timer.

The IPMC firmware must reset the internal timer (programmed to do this every 500 ms) before the timer expires. If the firmware fails to reset the hardware timer in 1 second, the IPMC core resets, and the firmware restarts.

Any reset of the IPMC is completely transparent to the host processor with the possible exception that system management software attempting to communicate with the IPMC might time out while the reset is in progress. There is no method for the processor to be explicitly notified that the IPMC is reset, but a SEL event will be logged upon next IPMC initialization cycle.

7.10 FRU Payload Control

The Multimedia Blade implements the FRU Control command as specified in the PICMG 3.0 Specification. Through this command, the payload can be reset, rebooted, or have its diagnostics initiated.

The following commands are supported by the Multimedia Blade. Refer to the appropriate documentation for third party shelf managers.

- Cold Reset
- Warm Reset (baseboard only)
- Graceful Reboot (baseboard only)

7.10.1 Cold Reset

When this command is initiated, the board performs a hard reset as described in Table 31, “Reset Requests” on page 107.

7.10.2 Warm Reset

When this command is initiated, the board performs a soft reset as described in Table 31, “Reset Requests” on page 107.

7.10.3 Graceful Reboot

This specific payload control command is implemented using the system interface messaging capability and the SMS_ATN bit of the KCS Status registers.

The Receive Message Queue is used to hold message data for system software until the system software can collect it, while the SMS_ATN bit is used to indicate that the IPMC requires attention from the system software.

7.11 OEM IPMI Commands

This section documents the OEM style IPMI commands implemented and supported on the Multimedia Blade. The commands are described in the following tables:

- Table 36, “Intel OEM Commands (Net Function 0x06h)” on page 112
- Table 37, “Intel OEM Commands (Net Function 0x30h)” on page 113
- Table 38, “Intel OEM Commands (Net Function 0x32h)” on page 121
- Table 39, “Intel OEM Net Function Commands Understood by the MMC” on page 122

Table 36. Intel OEM Commands (Net Function 0x06h)

Net Function = Application (0x06), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
01h	Get Device ID	Per the IPMI 2.0 specification	Platform-specific response fields: Byte 2 (Device ID) – 0x20 Byte 3 (Device revision) – 0x01 Byte 4 (Firmware revision 1) • bit 7 – Device available: 0 – Normal operation 1 – Update mode • bits 6:0 – Major firmware revision Byte 5 (Firmware revision 2) Byte 6 (IPMI version) – 0x02 Byte 7 (Additional device support) – 0xBF • bit 7: Chassis device • bit 6: Bridge • bit 5: IPMB event generator • bit 4: IPMB event receiver • bit 3: FRU inventory device • bit 2: SEL device • bit 1: SDR repository device • bit 0: Sensor device Bytes 8:10 (Manufacturer ID) – 343 (57h, 01h, 00h) Byte 11: (Product ID 1) – 0x0C (MPCBL0040) Byte 12: (Product ID 2) – 0x08 (EID/MCPD Block) Byte 13: (Boot block revision 1) Byte 14: (Boot block revision 2) Byte 15: (Firmware build number) Byte 16: (Reserved) – 0x00

Table 37. Intel OEM Commands (Net Function 0x30h) (Sheet 1 of 8)

Net Function = Intel® General Application (0x30), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
01h	Reset BIOS Flash Type	**Request**: Byte 1 – BIOS checksum success/failure indication • 00h – Checksum success (selects main flash) • 01h – Checksum failure (resets and selects other flash bank) **Response**: Byte 1 – Completion code	Switch the boot BIOS flash bank
02h	Set Fabric Interface Requested Selection	**Request**: Byte 1 – 57h (Internet Assigned Numbers Authority [IANA] number for Intel, 0x000157) Byte 2 – 01h Byte 3 – 00h Byte 4 – Fabric Ethernet interface setting: • 00h – Disabled • 01h – Front panel • 02h – Backplane • FFh – Don't change setting • 03h...FEh – Reserved **Response**: Byte 1 – Completion code Byte 2 – 57h (IANA number for Intel, 0x000157) Byte 3 – 01h Byte 4 – 00h	This command sets the Ethernet port "routing" as specified in the request data bytes. The command is available over KCS and IPMB interfaces. Settings made by this command are persistent until explicitly cleared, including board removal from system. The interface selection is independent of the E-keying status.
03h	Get Fabric Interface Requested Selection	**Request**: Byte 1 – 57h (IANA number for Intel, 0x000157) Byte 2 – 01h Byte 3 – 00h **Response**: Byte 1 – Completion code Byte 2 – 57h (Intel IANA 0x000157) Byte 3 – 01h Byte 4 – 00h Byte 5 – Fabric Ethernet interface setting • 00h – Disabled • 01h – Front panel • 02h – Backplane • 03h...FFh – Reserved	This command returns the "requested" Ethernet channel port "routing" selection. The command is available over KCS and IPMB interface. Note that due to E-keying, it may not be possible to connect to the backplane fabric; in that case, the interface will be routed to the front panel.

Table 37. Intel OEM Commands (Net Function 0x30h) (Sheet 2 of 8)

Net Function = Intel® General Application (0x30), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
04h	Get Fabric Interface Actual Selection	**Request**: Byte 1 – 57h (IANA number for Intel, 0x000157) Byte 2 – 01h Byte 3 – 00h **Response**: Byte 1 – Completion code Byte 2 – 57h (IANA number for Intel, 0x000157) Byte 3 – 01h Byte 4 – 00h Byte 5 – Fabric Ethernet interface setting: • 00h – Disabled • 01h – Front panel • 02h – Backplane • 03h…FFh – Reserved	This command returns the Ethernet channel port "routing" selection as actually routed. Due to E-keying selection of the backplane, it may not be permissible. The command is available over the KCS and IPMB interfaces.
05h	Set Control State	**Request**: Byte 1 – Control number: • 00h – FWH hub (for BIOS bank information) • 01h – Reserved • 02h – Reserved • 03h – Reserved • 04h – Reserved Byte 2 – Control state: • 00h – Deasserted • 01h – Asserted **Response**: Byte 1 – Completion code	This command sets the state of a control signal. This command overrides the AUTO-state of the control. These control states are persistent through payload reset, but not through IPMC reset.
06h	Get Control State	**Request**: n Byte 1 – Control Number • 00h – FWH hub (for BIOS bank information) • 01h – Reserved • 02h – Reserved • 03h – Reserved • 04h – Reserved **Response**: Byte 1 – Completion code Byte 2 – Control state • 00h – Deasserted • 01h – Asserted	This command gets the state of a control signal.

Table 37. Intel OEM Commands (Net Function 0x30h) (Sheet 3 of 8)

Net Function = Intel® General Application (0x30), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
07h	Get Version Data	**Request**: Byte 1 – None **Response**: Byte 1 – Completion code Byte 2 – Type/length • 02h – Two bytes Byte 3 – IPMC boot block revision 1 (Binary) Byte 4 – IPMC boot block revision 2 (BCD) Byte 5 – Type/length • 03h – Three bytes Byte 6 – IPMC firmware (FW) revision 1 (binary) Byte 7 – IPMC firmware revision 2 (BCD) Byte 8 – IPMC firmware build (binary) Byte 9 – Type/length • 01h – One byte Byte 10 – FPGA version/revision • bits 7:4 – Version bits • bits 3:0 – Revision bits Byte 11 – Type/length • 01h – One byte Byte 12 – Board version • bit 7 – SBC 0b – MPCBL0020 board 1b – MPCBL0040 board • bits 6:4 – Version bits • bits 3:0 – Revision bits Byte 13 – Type/length • 03h – Three bytes Byte 14 – Staged IPMC FW revision 1 (Binary) Byte 15 – Staged IPMC FW revision 2 (BCD) Byte 16 – Staged IPMC FW build (binary) Byte 17 – Type/length • 03h – Three bytes Byte 18 – Rollback IPMC FW revision 1 (binary) Byte 19 – Rollback IPMC FW revision 2 (BCD) Byte 20 – Rollback IPMC FW build (binary)	This command returns the boot and operational, staged, rollback firmware versions and the FPGA version information in the format required by the FRU OEM MRA definition for this platform. Type/Length Format: Bits Value Meaning 7:6 00 Binary/unspecified 01 BCD 10 6 bit ASCII, packed 11 8 bit ASCII
0Ah	Reserved	N/A	
0Bh	Reserved	N/A	
0Ch	Reserved	N/A	
0Dh	Reserved	N/A	
0Eh	Reserved	N/A	
0Fh	Reserved	N/A	
18h	Reserved	N/A	
19h	Reserved	N/A	
1Ah	Reserved	N/A	
1Bh	Reserved	N/A	

Table 37. Intel OEM Commands (Net Function 0x30h) (Sheet 4 of 8)

Net Function = Intel® General Application (0x30), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
1Eh	Reserved	N/A	
1Fh	Reserved	N/A	
20h	Reserved	N/A	
21h	Get DIMM State	**Request**: Byte 1 – DIMM group ID **Response**: Byte 1 – Completion code Byte 2 – DIMM group presence • bits 7:1 – Reserved • bit 0 – Presence (1 = group present) Byte 3 – Bitmap of DIMM slot existence Byte 4 – Bitmap of DIMM failure state Byte 5 – Bitmap of DIMM disabled state Byte 6 – Reserved Byte 7 – Bitmap of DIMM presence state	This command allows the *presence*, *disabled*, *failure*, and *spared* state of a set of DIMMs to be obtained without having to know the IPMC IPMI sensor numbers of the associated DIMM sensors. The state is returned as bitmaps. A bit being set in a bitmap indicates assertion of the state. The DIMM that a bit offset refers to is platform dependent. The DIMM group selector value is platform dependent. Platforms with 8 or fewer DIMMs should always use 0 as the group selector value. On platforms with memory boards, the group selector maps to a board number that is 1 based.
22h	Set DIMM State	**Request**: Byte 1 – DIMM group selector • bits 7:1 – Group ID • bit 0: – Presence (1 = group present) Byte 2 – Bitmap of DIMM slot existence Byte 3 – Bitmap of DIMM failure state Byte 4 – Bitmap of DIMM disabled state Byte 5 – Reserved Byte 6 – Bitmap of DIMM presence state **Response**: Byte 1 – Completion code	This command allows the failure state of a set of DIMMs to be set. Executing this command causes the associated sensor offsets for affected DIMM sensors. The DIMM Group ID is defined the same as for the Get DIMM State command. The IPMC will accept a variable number of request bytes: as few as 1 bytes of this command (group selector) up to the fully defined set. Note that the group/board presence flag is ignored on platforms that don't implement removable groups (memory boards). Example of the slot existence bitmap: Byte 2 = 0Fh and byte 6 = 0Dh means DIMM 1,3, and 4 are present, the DIMM 2 slot is empty. DIMM slots 5,6,7, and 8 do not exist on this platform.
23h	ReArm DIMMs	**Request**: N/A **Response**: Byte 1 – Completion code	This command causes the DIMM failure and disabled state of all DIMM sensors to be reset.
25h	SyncSMBus	**Request**: Byte 1 – Action • 0 – Release SMBus ownership • 1 – Acquire SMBus ownership **Response**: Byte 1 – Completion code	This command is used to synchronize IPMC and BIOS access to the SMBus. When the BIOS has SMBus ownership, the IPMC will not access the bus. The BIOS will not access the bus unless it has acquired ownership using this command. The BIOS will need to acquire ownership to access SMBus devices such as DIMM SPD FRUs, and clock chips.

Table 37. Intel OEM Commands (Net Function 0x30h) (Sheet 5 of 8)

Net Function = Intel® General Application (0x30), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
28h	Set Processor State	**Request**: Byte 1 – Processor ID Valid values are 0:N-1, where N is the number of processors supported by the platform. Byte 2,3 – Processor state to set This is a bit mask identifying the sensor offsets to set in the associated processor status sensor maintained by the IPMC. The offsets are as defined in the IPMI 1.5 specification. The following offsets are supported: • 0 – IERR • 1 – Thermal trip • 2 – FRB1/BIST failure • 3 – FRB2/POST hang failure • 4 – FRB3/Processor startup failure • 5 – Configuration error • 6 – SMBIOS uncorrectable CPU-complex error • 8 – Processor disabled Byte 4 – Action This byte specifies the action to take after setting the processor status sensor state as requested. It is a bit-mask and multiple actions may be set. • bits 7:1 – Reserved • bit 0 – Reset system **Response**: Byte 1 – Completion code	This command allows processor fault state to be asserted and an action to be taken afterwards. Asserting some fault states may cause the IPMC to generate SEL events (depending on the SDR configuration). Processor disabling will not take effect until the next reset.
29h	Get Processor State	**Request**: Byte 1 – Processor ID Valid values are 0:N-1, where N is the number of processors supported by the platform. **Response**: Byte 1 – Completion code Byte 2,3 – Processor state These bytes contain the associated processor status sensor's 2-byte event assertion status as defined in the IPMI 1.5 specification.	This command returns the current processor status sensor event assertion status for the requested processor. This command allows the caller to get processor state without having to know the IPMI sensor numbers of the processor status sensors.
2Ah	ReArm Processors	**Request**: N/A **Response**: Byte 1 – Completion code	This command clears all error and disabled state for all processors. Processor/terminator presence is not affected. Disabled processors are not actually run until the next system reset.
2Bh	Disable FRB2 Action	**Request**: N/A **Response**: Byte 1 – Completion code	This command disables resets associated with the watchdog timer expiring with an FRB2 reason.
41h	Set System GUID	**Request**: Bytes 1-16 – System GUID **Response**: Byte 1 – Completion code	This command sets the system GUID retrieved per the format of the IPMI 1.5 Get System GUID command. This value is stored persistently.

Table 37. Intel OEM Commands (Net Function 0x30h) (Sheet 6 of 8)

Net Function = Intel® General Application (0x30), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
48h	Internal Platform Event Message	**Request**: Byte 1 – Generator ID Byte 2 – Event message revision (04h) Byte 3 – Sensor type Byte 4 – Sensor number Byte 5 – Event dir / event type Byte 6 – Event data 1 Byte 7 – Event data 2 Byte 8 – Event data 3 **Response**: Byte 1 – Completion code Byte 2 – Reserved Byte 3:4 – Record ID Byte 5:8 – Timestamp	This command provides a mechanism for the BIOS to log event messages and retrieve the record ID and timestamp of the event. This command can be used to correlate BIOS events with SEL events. The format of the request data is the same as for the Platform Event Message command (Sensor/Event net function, command number 02). See the IPMI 1.5 specification for details on the definition of the request data fields.
50h	Set SM Signal	**Request**: Byte 1 – Signal type: • 0Eh – Health LED • 0Fh – Out Of Service LED • 10h – Hot-Swap LED Byte 2 – Signal instance, zero based Byte 3 – Action: • 0 – Force de-asserted • 1 – Force asserted • 2 – Revert Byte 4 – Optional value used by multi-value signals **Response**: Byte 1 – Completion code	This command allows the real-time state of certain output signals (for example, LEDs) to be set without losing the IPMC internal state associated with the signals. The command also allows the output signals to revert to their normal behavior. Certain signals take analog values or complex states. The fourth byte of the request is supported for these kinds of signals, for example, Fan Speed Control.
51h	Get SM Signal	**Request**: Byte 1 – Signal type: • 13h – Reset button • 14h – Hot-Swap handle switch Byte 2 – Signal instance, zero based Byte 3 – Action: • 0 – Sample • 1 – Ignore • 2 – Revert **Response**: Byte 1 – Completion code Byte 2 – Signal value: • 0 – De-asserted • 1 – Asserted	This command allows the real-time state of certain input signals to be polled without changes in the signals to be acted on by the IPMC. The command also allows the input signals to revert to their normal behavior. Not all signals are supported by all platforms.
52h	Get Self Test History	**Request**: Byte 1 – Action: • 0 – Get first • 1 – Get next **Response**: Byte 1 – Completion code Byte 2 – First byte of test result • FFh – No more results Byte 3 – Second byte of test result	This command retrieves stored self-test failures. Normal use is to first call with action of “get first” then use “get next” for subsequent calls. If byte 2 of the response is ever 0xFF, all the results have been retrieved.

Table 37. Intel OEM Commands (Net Function 0x30h) (Sheet 7 of 8)

Net Function = Intel® General Application (0x30), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
70h	Graceful OS Shutdown	**Request**: Byte 1 • bits 7:3 – Reserved • bits 2:0 – Shutdown operation: 00h – No change (used to read shutdown command status) 01h – Power off using the OS agent 02h – Reset using the OS agent 03h-07h – Reserved **Response**: Byte 1 – Completion code Byte 2 – Only for command status • bits 7:2 – Reserved • bits 1:0 – Shutdown command status 00h – Operation successful 01h – Operation failed (no OS agent) 02h – Operation in progress 03h – Reserved	This command performs graceful shutdown using OS agent.
82h	Get ACPI Configuration Mode	**Request**: N/A **Response**: Byte 1 – Completion code Byte 2 – ACPI configuration state • bits 7:1 – Reserved • bit 0 – ACPI mode: 0 = IPMC is in legacy mode 1 = IPMC is in ACPI mode	
83h	Set ACPI Configuration Mode	**Request**: Byte 1 – ACPI configuration state Byte 2– ACPI configuration mask • bits 7:1 – Reserved • bit 0 – ACPI mode mask 1 = set ACPI mode **Response**: Byte 1 – Completion code	
86h thru 8Ch	Reserved		Reserved for internal IPMC use
A0h thru A8h	Reserved		Reserved for internal IPMC use

Table 37. Intel OEM Commands (Net Function 0x30h) (Sheet 8 of 8)

Net Function = Intel® General Application (0x30), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
E6h	Get NMI/ INIT Source	**Request**: N/A **Response**: Byte 1 – Completion code Byte 2 – NMI/INIT source 1: • bits 7:6 – Reserved • bit 5 – Processor thermal trip [1, 2] • bit 4 – Processor IERR [1, 2] • bit 3 – Chassis control command • bit 2 – Event (PEF) • bit 1 – Watchdog NMI/diagnostic interrupt • bit 0 – Diagnostic interrupt (FP NMI) button Byte 3 – NMI/INIT source 2: • bits 7:4 – Reserved • bit 3 – Chipset NMI [1] • bit 2 – South Bridge NMI [1] • bit 1 – PCI SERR/PERR [1] • bit 0 – Multi-bit memory error [1]	This command returns the IPMC's understanding of the source of the latest NMI/INIT assertion. The source information is a composite of IPMC detected sources (source 1) and externally detected sources (source 2). Although multi-bit memory error is in the external source byte, on some platforms, this may be detected by the IPMC. The source 1 and 2 values are cleared when read and when the system is reset or powered off. [1] Only supported on IA32 systems [2] Support is platform specific
EDh	Set NMI/ INIT Source	**Request**: Byte 1 – • bits 7:4 – Reserved • bit 3 – Chipset NMI [1] • bit 2 – South Bridge NMI [1] • bit 1 – PCI SERR/PERR [1] • bit 0 – Multi-bit memory error [1] **Response**: Byte 1 – Completion code	This command merges the given values in with any NMI/INIT sources detected by the IPMC. The values given here will be read by the next Get NMI/INIT Source command. This command also causes the IPMC to generate an NMI/ INIT pulse for a supported source. [1] Only supported on IA32 systems
F7h	NMI/INIT Enable / Disable	**Request**: Byte 1 – NMI/INIT enable state • 0 = Disable IPMC NMI/INIT generation • 1 = Enable IPMC NMI/INIT generation **Response**: Byte 1 – Completion code	This command is the master control for the IPMC NMI/INIT generation. The default state for NMI/INIT generation is enabled. The state set by this command is volatile, that is, it is not saved across AC power cycles.
FAh	Get Latest Post Code	**Request**: None **Response**: Byte 1 – Completion code Byte 2 – Pre-reset port80 (byte n-4) Byte 3 – Pre-reset port80 (byte n-3) Byte 4 – Pre-reset port80 (byte n-2) Byte 5 – Pre-reset port80 (byte n-1) Byte 6 – Pre-reset port80 (last byte n) Byte 7 – Post-reset port80 (byte n-4) Byte 8 – Post-reset port80 (byte n-3) Byte 9 – Post-reset port80 (byte n-2) Byte 10 – Post-reset port80 (byte n-1) Byte 11 – Post-reset port80 (last byte n)	Returns two port80 signatures from before the last reset, and afterwards. The first signature contains the last 5 port80 bytes prior to the last payload reset event. The second signature contains the most current port80 bytes after the payload was reset.

Table 38. Intel OEM Commands (Net Function 0x32h)

Net Function = Intel® Platform Specific (0x32), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
01h	Get HW Info	**Request:** N/A **Response**: Byte 1 – Board version • bit 7 – Board type 1 = MPCLB0040 2 = MPCBL0020 • bits 6:4 – Fab version • bits 3:2 – Fab revision • bits 1:0 – Reserved Byte 2 – FPGA version • bits 7:4 – Version • bits 3:0 – Revision	Provides the SBC board and FPGA versions from the FPGA itself.
02h	Get Power Unit Status	**Request:** N/A **Response:** Byte 1 – Power state • 00h – Power is ON (ACPI state S0) • 05h – Power is OFF (ACPI state S5) • 20h – Legacy ON • 21h – Legacy OFF • [xx] – All others reserved Byte 2 – Power status bits • bit 0 – Power cycle • bit 1 – Control fault • bits 7:2 – Reserved Byte 3 – Power fault 1 bits • bit 0 – 1.1 V standby power fault • bit 1 – 1.5 V standby power fault • bit 2 – 1.8 V standby power fault • bit 3 – 3.3 V standby power fault • bit 4 – 5 V standby power fault • bit 5 – 12 V power fault • bit 6 – 5 V power fault • bit 7 – 3.3 V power fault Byte 4 – Power fault 2 bits • bit 0 – 1.2 V power fault • bit 1 – 1.5 V power fault • bit 2 – 1.8 V power fault • bit 3 – 1.8 V LAN power fault • bit 4 – 1.1 V power fault • bit 5 – 1.05 V power fault • bit 6 – CPU 0 power fault • bit 7 – CPU 1 power fault	Provides the status of the payload power and corresponding power unit faults from the FPGA.
55h	Reserved		
57h	Reserved		

Table 39. Intel OEM Net Function Commands Understood by the MMC (Sheet 1 of 2)

Net Function = Intel® General Application (30h), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
41h	Set System GUID	**Request:** Bytes 1-16 – System GUID **Response**: Byte 1 – completion code	This command sets the System GUID gotten per the format of the IPMI 1.5 Get System GUID command. This value is stored persistently.
43h	Set/Get SDR Translation Channels	**Request:** Byte 1 – Operation code • 0 –Get Bitmask • 1 –Set Bitmask • 2-FFh –Reserved Byte 2 – MS byte of channel bitmap, channels 8-15 Byte 3 – LS byte of channel bitmap, channels 0-7 **Response:** Byte 1 – completion code Byte 2 – MS byte of bitmask Byte 3 – LS byte of bitmask	Bitmask that enables translation on a per channel basis. Bit 0 of the LS byte is channel 0, bit 7 of the MS byte is channel 15. A 1 enables translation for the channel. Translation status for channel 0Eh (the IPMI “self” channel) is ignored. Bytes 2 and 3 of the request may be omitted for Get operations; if present they will be ignored. Bytes 2 and 3 of the response are returned only for Get Bitmask operation.
50h	Force Watchdog Timer Reset	**Request:** N/A **Response:** Byte 1 – completion code	This command allows the PICMG watchdog timer to time out and reset the MMC. Although the command will attempt to return a completion code before the reset occurs, there is no guarantee that this will happen.
51h	Get SM Signal	**Request:** Byte 1 – signal ID **Response:** Byte 1 – completion code Byte 2 – signal value • 0 – De-asserted • 1 – Asserted	This command allows the real-time state of certain input signals to be polled. Note that this implementation of this command is unique to the Multimedia Blade.
52h	Read Self Test	**Request:** Byte 1 – action • 0 – Get First • 1 – Get Next **Response:** Byte 1 – completion code Byte 2 – First byte of test result • FFh = no more results Byte 3 – Second byte of test result	This command allows all stored self-test results to be retrieved. Normal use is to first call with action of ‘Get First’ then use ‘Get Next’ for subsequent calls. If byte 2 of the response is ever FFh, all the results have been retrieved. Unlike the Get Self Test Results command, this command will never return code 55h (No Error) for the first response byte.

Table 39. Intel OEM Net Function Commands Understood by the MMC (Sheet 2 of 2)

Net Function = Intel® General Application (30h), LUN = 00			
Code	**Command**	**Request, Response Data**	**Description**
53h	Dump Link State	**Request:** N/A **Response:** Byte 1 – completion code Byte 2 – Link 0 Status Byte 3-(N+1) – Link 'n' Status Byte (N+2) – All Links Visited Status	Dumps the link status for the eKeying links. The RTM has 3 links, the AMC has 6. Each link status byte uses only the least significant 3 bits, the upper 5 bits are zeros: Bit 2: 1=Port Enabled Bit 1: 1=Port Visited Bit 0: 1=Link is Visible "Visited" means a Set AMC Port State command has been received for that link. "Visible" means that this link is available in this configuration, only the AMC will have invisible links. The "All Links Visited" status is non-zero if the MMC thinks that all visible links have received a Set AMC PORT State command, zero indicates that not all visible ports have received a command.
C0h	Set/Get LED Red Color Override	**Request:** Byte 1: • 0 = Get override state • 1 = Set override state Byte 2: • Red color not allowed • 1 = Red color allowed **Response:** Byte 1 – completion code Byte 2 – current override state	This command selects whether MMC LEDs can display the color red.

8.0 BIOS Features

This chapter provides information about the BIOS features of the Dialogic® Multimedia Blade for AdvancedTCA® (hereinafter referred to as the Multimedia Blade). The following topics are discussed:

- BIOS Flash Memory Organization
- Complementary Metal-Oxide Semiconductor (CMOS)
- Language Support
- Recovering BIOS Data
- BIOS Security Features
- Remote Access Configuration

For information about BIOS setup, see Chapter 9.0, "BIOS Settings."

8.1 BIOS Flash Memory Organization

The Multimedia Blade uses a BIOS developed by Intel and AMI*, which is stored in flash memory and can be updated remotely or locally. In addition to the BIOS and BIOS Setup program, the flash memory contains POST and Plug and Play support. The BIOS displays a message during POST identifying the type of BIOS and a revision code.

The Multimedia Blade hardware has two flash devices for BIOS where redundant copies are stored. Logic to select the active BIOS device is connected to the IPMC. IPMC firmware selects the BIOS device to boot from.

By default, the firmware selects the first BIOS device. The BIOS executes code from this flash and performs checksum validation of its operational code. This checksum occurs in the boot block of the BIOS. If the boot block detects a checksum failure in the remainder of the BIOS, it notifies the IPMC of the failure. In the event of failure, the IPMC firmware:

1. Asserts the RESET pin on the processor.
2. Switches the flash device.
3. Deasserts the RESET pin on the processor, allowing BIOS to execute off the second flash device.
4. Logs a SEL event.

8.2 Complementary Metal-Oxide Semiconductor (CMOS)

CMOS RAM is a nonvolatile storage that stores data needed by the BIOS. The data consists of certain on-board configurable settings, including time and date. CMOS resides in the Intel® 6300ESB ICH and is powered by the SuperCap when the blade is power off. The settings in the BIOS Setup menu are stored in the CMOS RAM and are often called CMOS settings.

Note: If the board is powered down or out of the chassis for greater than 2 hours, the system time and date revert to the factory defaults and will have to be reset. Ensure that the time and date are set before starting the Dialogic Services.

8.3 Language Support

English is the only supported language.

8.4 Recovering BIOS Data

Some types of failure can destroy the BIOS. For example, the data can be lost if a power outage occurs while the BIOS is being updated in flash memory. The BIOS can be recovered from the backup BIOS. Recovery mode is active when BIOS checksum fails and notifies the IPMC to failover to the backup BIOS.

8.5 BIOS Security Features

The BIOS includes security features that restrict access to the BIOS Setup program and booting the computer. A supervisor password and a user password can be set for the BIOS Setup program and for booting the computer, with the following restrictions:

- The supervisor password gives unrestricted access to view and change all the Setup options in the BIOS Setup program. This is the supervisor mode.
- The user password gives restricted access to view and change Setup options in the BIOS Setup program. This is the user mode.
- If only the supervisor password is set, pressing the “Enter” key at the password prompt of the BIOS Setup program allows the user restricted access to Setup.
- If both the supervisor and user passwords are set, users can enter either the supervisor password or the user password to access Setup. Users have access to Setup respective to which password is entered.
- Setting the user password restricts who can boot the computer. The password prompt is displayed before the computer is booted. If only the supervisor password is set, the computer boots without asking for a password. If both passwords are set, the user can enter either password to boot the computer.

Table 40 shows the effects of setting the supervisor password and user password. This table is for reference only and is not displayed on the screen.

Table 40. Supervisor and User Password Functions (Sheet 1 of 2)

Password Set	Supervisor Mode	User Mode	Password to Enter Setup	Password During Boot
None	Any user can change all options	Any user can change all options	None	None
Supervisor and User	Can change all options	Based on user access level: • No access • View only • Limited • Full access	Supervisor or User	If the password check option is set to “Setup”, no password is required. Otherwise, a supervisor or a user password is required.
Supervisor only	Can change all options	Based on user access level: • No access • View only • Limited • Full access	Supervisor (for supervisor mode) or enter only (for user mode)	If the password check option is set to “Setup”, no password is required. Otherwise, a supervisor or a user password is required.

Table 40. Supervisor and User Password Functions (Sheet 2 of 2)

Password Set	Supervisor Mode	User Mode	Password to Enter Setup	Password During Boot
User only	Can not get into supervisor mode until user password is cleared	Can change all options	User	If the password check option is set to “Setup”, no password is required. Otherwise, a supervisor or a user password is required.

8.6 Remote Access Configuration

Remote access using serial console redirection allows users to monitor the Multimedia Blade boot process and run the Multimedia Blade BIOS Setup from a remote serial terminal. Connection is made directly through the front panel serial port.

The console redirection feature is useful in cases where it is necessary to communicate with the Multimedia Blade in an embedded application without video support. This is the recommended method when using the Multimedia Blade.

Note: The default settings used for console redirection to the serial port are 9600, n, 8, 1, and no flow control.

Table 41 shows the escape code sequences that may be useful for things like BIOS Setup if function keys cannot be directly sent from a terminal application.

Table 41. Function Key Escape Code Equivalents

Key	Escape Sequence	Notes
F1	ESC OP	
F2	ESC OQ	
F3	ESC OR	
F4	ESC OS	To enter BIOS Setup
F5	ESC OT	
F6	ESC OU	
F7	ESC OV	
F8	ESC OW	
F9	ESC OX	
F10	ESC OY	To save and exit BIOS Setup
F11	ESC OZ	
F12	ESC OI	PXE boot

9.0 BIOS Settings

This chapter describes the BIOS settings for the Dialogic® Multimedia Blade for AdvancedTCA® (hereinafter referred to as the Multimedia Blade). Information is presented as follows:

- Introduction
- Main Menu
- Advanced Menu
- PCIPnP Menu
- Boot Menu
- Security Menu
- Chipset Menu
- Exit Menu

Note: In most cases, the BIOS defaults provide the correct configuration for board use. However, it is necessary to set the following:

- On the Main Menu, set the system time and system date.
- On the Advanced Menu, IDE Configuration Submenu, disable flash drives.
- On the Advanced Menu, Ethernet Ports Direction Configuration Submenu, set the Ethernet Ports C&D Direction.
- On the Advanced Menu, Remote Access Configuration Submenu, change the Serial Port Mode settings.
- On the Boot Menu, Boot Device Priority Submenus, set the first boot device.

Default values for all other BIOS settings are given in this chapter for reference.

9.1 Introduction

The BIOS Setup program can be used to view and change the BIOS settings for the Multimedia Blade. The BIOS Setup program is accessed by pressing the F4 key (when connected via serial port). Table 42 lists the BIOS Setup program menu items.

Table 42. BIOS Setup Program Menu Bar

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit
Allocates resources for hardware components	Configures advanced chipset features	Configures PCI Plug and Play devices	Selects boot options	Sets passwords and security features	Configures chipset	Saves or discards changes to Setup program options

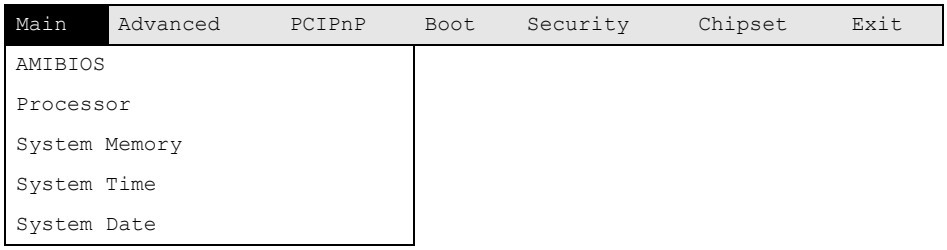
Table 43 lists the function keys available for menu screens.

Table 43. BIOS Setup Program Function Keys

BIOS Setup Program Function Key	Description
<↔> or <→>	Selects a different menu screen (moves the cursor left or right)
<↑ > or <↓>	Selects an item (moves the cursor up or down)
<Tab>	Selects a field (not implemented)
<Enter>	Executes command or selects the submenu
<F9>	Loads the default configuration values for the current menu
<F10>	Saves the current values and exits the BIOS Setup program
<Esc>	Exits the menu

9.2 Main Menu

To access this menu, select **Main** on the menu bar at the top of the screen. The Main menu options are described in Table 44.



Note: The Main menu is used to set the system time and date. All other options are for display only.

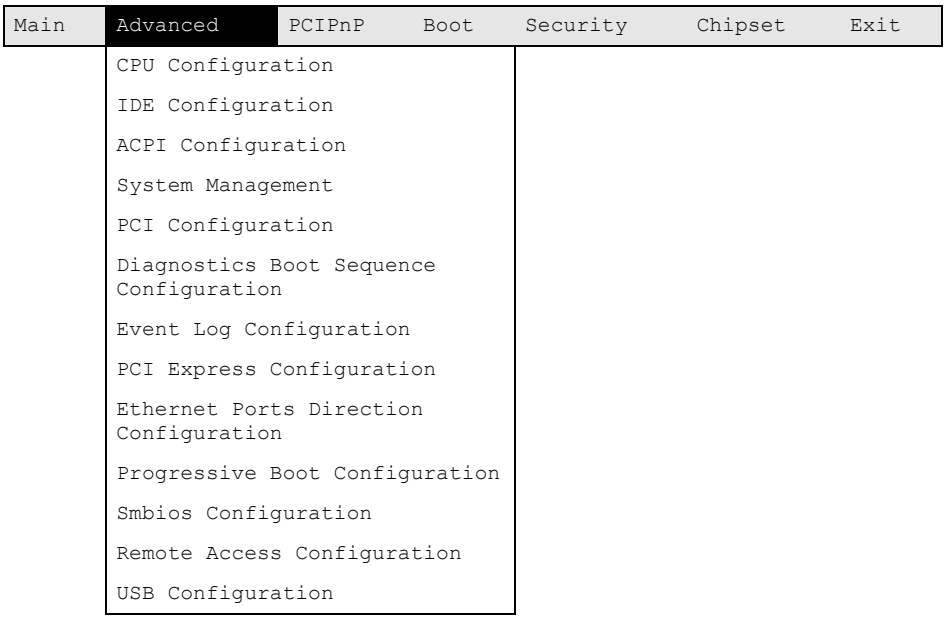
Note: If the board is powered down or out of the chassis for greater than 2 hours, the system time and date revert to the factory defaults and will have to be reset. Ensure that the time and date are set before starting the Dialogic Services.

Table 44. Main Menu Options

Feature	Options	Description
BIOS ID	AMIBIOS version Build date ID	Display BIOS ID
Processor	Type Speed Count	Display processor type, speed, and count
System Memory	Size	Display system memory size of recognized DIMMs
System Time	Hour/minute/second	Specify the current time
System Date	Day of week Month/day/year	Specify the current date

9.3 Advanced Menu

To access this menu, select **Advanced** on the menu bar at the top of the screen. The submenus for the Advanced menu are displayed.



Under the Advanced menu, the following warning message appears:

"WARNING: Setting the wrong values in the sections that follow may cause system to malfunction."

This is a warning message to users to not modify the settings unless they are familiar with the items. To restore factory defaults, select **Exit**, then **Load Optimal Defaults**.

The submenus for the Advanced menu are described below.

9.3.1 CPU Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **CPU Configuration**. The CPU Configuration options are described in Table 45.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 45. CPU Configuration Submenu Options (Sheet 1 of 2)

Feature	Options	Description
Manufacturer		Display CPU manufacturer
Brand String		Display CPU brand string
Frequency		Display CPU frequency
FSB Speed		Display front side bus speed
Cache L1		Display L1 cache size
Cache L2		Display L2 cache size
Note: **Bold** text indicates default setting.		

Table 45. CPU Configuration Submenu Options (Sheet 2 of 2)

Feature	Options	Description
Execute Disable Bit	**Enabled**	
Core Multi-Processing	**Enabled**	
CPU TM Function	**Enabled**	Thermal monitoring
Virtualization Technology	**Disabled**	
Intel SpeedStep® Technology	**Automatic**	CPU speed controlled by OS
C-State Technology from Intel C1 Enable	**Enabled** **Standard**	Enable specific C-state: • Enabled = C-state enabled • Standard(Std) = Conventional C-state
Note: **Bold** text indicates default setting.		

9.3.2 IDE Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **IDE Configuration**. The IDE Configuration options are described in Table 46.

Note: When using the Multimedia Blade, the setting for the IDE Configuration feature should be set to **Disabled**. Do not change any of the other default settings.

Table 46. IDE Configuration Submenu Options

Feature	Options	Description
IDE Configuration	Must be set to Disabled	IDE Mode configuration for the 128 MByte flash memory. **Must be set to Disabled** for the Multimedia Blade. (Default setting is P-ATA Only.)
Primary IDE Master	**Hard Disk**	Display the primary IDE master drive. While entering Setup, the BIOS auto-detects the presence of IDE devices. This displays the status of the auto detection of IDE devices. See Table 47.
Secondary IDE Master	**Hard Disk**	Display the secondary IDE master drive. While entering Setup, the BIOS auto-detects the presence of IDE devices. This displays the status of the auto-detection of IDE devices. See Table 47.
Hard Disk Write Protect	**Disabled**	
IDE Detect Time Out	**35**	Time-out value in seconds for detecting ATA/ATAPI device(s)
Note: **Bold** text indicates default setting.		

Table 47. Primary/Secondary IDE Master Submenu Options (Sheet 1 of 2)

Feature	Options	Description
Device		Display IDE device
Vendor		Display IDE vendor name
Size		Display IDE device size
LBA Mode		Display IDE LBA mode status
Block Mode		Display IDE block mode status
PIO Mode		Display PIO mode status
Async DMA		Display async DMA status
Ultra DMA		Display ultra DMA-5 status
Note: **Bold** text indicates default setting.		

Table 47. Primary/Secondary IDE Master Submenu Options (Sheet 2 of 2)

Feature	Options	Description
S.M.A.R.T		Display S.M.A.R.T status (SMART stands for self-monitoring, analysis, and reporting technology.)
Type	**Auto**	Type of IDE device connected
LBA/Large Mode	**Auto**	Enable the LBA mode if the device supports it and the device is not already formatted with LBA mode disable.
Block (Multi-Sector Transfer)	**Auto**	The data transfer from and to the device occurs multiple sectors at a time if the device supports it.
PIO Mode	**Auto**	PIO mode
DMA Mode	**Auto**	Auto detected
S.M.A.R.T	**Auto**	Enable S.M.A.R.T if the device supports it
32 Data Transfer	**Disabled**	Disable 32-bit data transfer
Note: **Bold** text indicates default setting.		

9.3.3 ACPI Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **ACPI Configuration**. There are two ACPI configuration submenus:

- Advanced ACPI Configuration submenu, described in Table 48
- Chipset ACPI Configuration submenu, described in Table 49

Warning: When using the Multimedia Blade, do not change the default settings.

Table 48. Advanced ACPI Configuration Submenu Options

Feature	Options	Description
ACPI Version Features	**ACPI v1.0**	Enable RSDP pointers to 64-bit fixed system description tables
ACPI APIC Support	**Enabled**	Include ACPI APIC table pointer to RSDT pointer list
AMI OEMB Table	**Enabled**	Include OEMB table pointer to R(x)SDT pointer lists
Headless Mode	**Disabled**	Disable headless operation mode through ACPI
Note: **Bold** text indicates default setting.		

c

Table 49. Chipset ACPI Configuration Submenu Options

Feature	Options	Description
APIC ACPI SCI IRQ	**Disabled**	Disable APIC ACPI SCI IRQ
Note: **Bold** text indicates default setting.		

9.3.4 System Management Submenu

To access this submenu, select **Advanced** on the menu bar, then **System Management**. The System Management options are described in Table 50.

Table 50. System Management Submenu Options

Feature	Options	Description
FRU Board Information Area		Display FRU board information: • Board Product Name • Board Serial Number • Board Part Number
FRU Product Information Area		Display FRU product information • Product Name • Product Part/Model • Product Version Number • Product Serial Number
BMC Device and Firmware Information		Display BMC device and firmware information • BMC Device ID • BMC Firmware Revision • BMC Revision • SDR Revision

9.3.5 PCI Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **PCI Configuration**. The PCI configuration options are described in Table 51.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 51. PCI Configuration Submenu Options

Feature	Options	Description
SAS Controller	**Enabled**	Enable Option ROM for on-board SAS controller
Ethernet Controller (Ports A&B)	**Enabled**	Enable Option ROM for Gigabit Ethernet LAN Controller Ports A & B (IBA GE Slot 0200 and 0201)
Ethernet Controller (Ports C&D)	**Enabled**	Enable Option ROM for Gigabit Ethernet LAN Controller Ports C & D (IBA GE Slot 0300 and 0301)
Ethernet Controller (Ports E&F)	**Enabled**	Enable Option ROM for Gigabit Ethernet LAN Controller Ports E & F (IBA GE Slot 0400 and 0401)
Note: **Bold** text indicates default setting.		

9.3.6 Diagnostics Boot Sequence Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **Diagnostics Boot Sequence Configuration**. The Diagnostics Boot Sequence Configuration options are described in Table 52.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 52. Diagnostics Boot Sequence Configuration Submenu Options

Feature	Options	Description
Default PXE	**Network: IBA GE Slot 0201 v0006**	Default PXE boot location for board diagnostic tests

Table 52. Diagnostics Boot Sequence Configuration Submenu Options (Continued)

Default Hard Drive	**SCSI: #508 ID00 LUN0**	Default hard drive location for board diagnostic tests
Default Removable Drive	**Disabled**	Default removable drive location for board diagnostic tests (for example, USB drive). If the USB drive is installed, it becomes the default.
Note: **Bold** text indicates default setting.		

9.3.7 Event Log Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **Event Log Configuration**. The Event Log Configuration options are described in Table 53.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 53. Event Log Configuration Submenu Options

Feature	Options	Description
View Event Log	**Enabled**	Enable NMI reporting on fatal error events
Clear Event Log		Discard all events in the event log. This event log is the local SEL (System Event Log) for the board.
ECC Event Logging	**Enabled**	Enable ECC event logging
Hub Interface Event Logging	**Enabled**	Enable hub interface event logging
System Bus Event Logging	**Enabled**	Enable system bus error logging
Memory Buffer Event Logging	**Enabled**	Enable memory buffer event logging
PCI Error logging	**Enabled**	Enable PCI error logging
Note: **Bold** text indicates default setting.		

9.3.8 PCI Express* Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **PCI Express Configuration**. The PCI Express Configuration options are described in Table 54.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 54. PCI Express Configuration Submenu Options

Feature	Options	Description
Active State Power Management	**Disabled**	Disable PCI Express L0s and L1 link power state
PCI Express Port A0	**Enabled**	Always visible
PCI Express Port A1	**Enabled**	Always visible
PCI Express Port B1	**Enabled**	Always visible
PCI Express Port C0	**Enabled**	Always visible
PCIe Jitter Tolerance	**Enabled**	Enable PCI Express jitter tolerance
PCIe Compliance Mode	**Disabled**	Disable MCH entering PCI Express compliance mode
PIC Express Hot Plug on AdvancedMC	**Enabled**	Enable PCI Express Hot Plug on AdvancedMC slot (EXPC0 of MCH)
Note: **Bold** text indicates default setting.		

9.3.9 Ethernet Ports Direction Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **Ethernet Ports Direction Configuration**. The Ethernet Ports Direction Configuration options are described in Table 55.

Note: When using the Multimedia Blade, select Yes to change Ethernet Ports C&D Direction (default is No). The BIOS screen should look like this:

```
************************************************
* Ethernet Ports Direction Configuration Menu
* ********************************************
* Current Eth Ports E&F Direction :Front Panel
* Change Eth Ports E&F Direction [No]
*
* Current Eth Ports C&D Direction :Onboard GbEth
* Change Eth Ports C&D Direction [Yes]
*
```

Table 55. Ethernet Ports Direction Configuration Submenu Options

Feature	Options	Description
Current Eth Ports E&F Direction: Front Panel	**No**	
Current Eth Ports C&D Direction: Onboard GbEth	Must be set to Yes	Change Eth Ports C&D Direction; default setting is No, must be set to Yes.
Note: **Bold** text indicates default setting.		

9.3.10 Progressive Boot Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **Progressive Boot Configuration**. The Progressive Boot Configuration options are described in Table 56.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 56. Progressive Boot Configuration Submenu Options

Feature	Options	Description
Progressive Boot Support	**Disabled**	Disable progressive boot support
Note: **Bold** text indicates default setting.		

9.3.11 Smbios Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **Smbios Configuration**. The Smbios Configuration options are described in Table 57.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 57. Smbios Configuration Submenu Options

Feature	Options	Description
Smbios SMI Support	**Enabled**	Smbios SMI wrapper support for Plug and Play function 50h-54h
Note: **Bold** text indicates default setting.		

9.3.12 Remote Access Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **Remote Access Configuration**. The Remote Access Configuration options are described in Table 58.

Note: When using the Multimedia Blade, the serial port settings should be changed to **115200,8,n,1**. Do not change any of the other default settings.

Table 58. Remote Access Configuration Submenu Options

Feature	Options	Description
Remote Access	**Enabled**	
Serial Port Number	**COM1**	Displays serial port number, base address, and IRQ
Serial Port Mode	Must be set to 115200,8,n,1	Serial port settings. **Must be set to 115200,8,n,1** for the Multimedia Blade. (Default is 9600 8, n, 1.)
Flow Control	**None**	
Redirection After BIOS POST	**Always**	
Terminal Type	**ANSI**	Target terminal type
VT-UTF8 Combo Key Support	**Enabled**	Enable VT-UTF8 combination key support for ANSI/VT100 terminals
Sredir Memory Display Delay	**No Delay**	
Note: **Bold** text indicates default setting.		

9.3.13 USB Configuration Submenu

To access this submenu, select **Advanced** on the menu bar, then **USB Configuration**. The USB Configuration options are described in Table 59.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 59. USB Configuration Submenu Options

Feature	Options	Description
USB Devices Enabled		Display the number of USB devices detected by BIOS
USB Function	**2 USB Ports**	
Legacy USB Support	**Enabled**	
USB 2.0 Controller	**FullSpeed**	Configures the USB 2.0 controller to 12 Mbps, USB 1.1
BIOS EHCI Hand Off	**Enabled**	
Note: **Bold** text indicates default setting.		

9.4 PCIPnP Menu

To access this menu, select **PCIPnP** on the menu bar at the top of the screen. The PCIPnP menu options are described in Table 60.

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

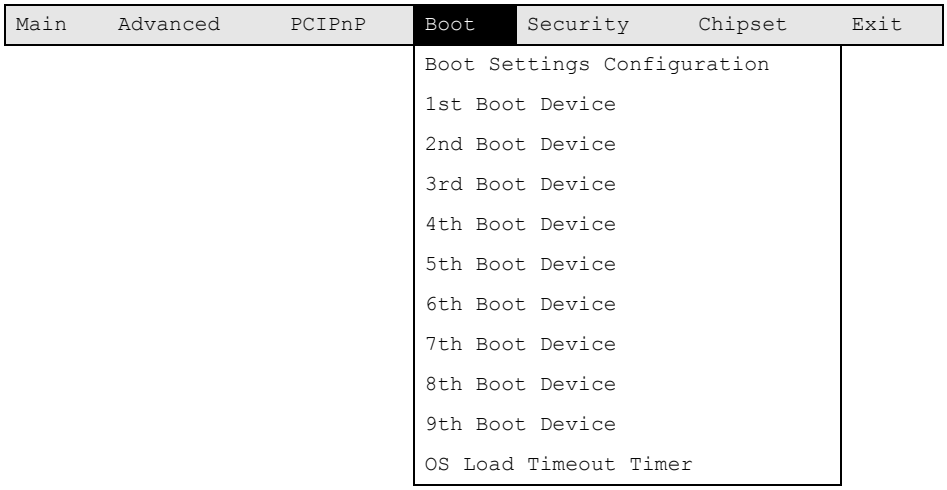
Warning: When using the Multimedia Blade, do not change the default settings.

Table 60. PCIPnP Menu Options

Feature	Options	Description
Clear NVRAM	**No**	Do not clear NVRAM during system boot
Plug & Play OS	**No**	BIOS configures all devices in the system
PCI Latency Timer	**32**	Value in units of PCI clocks for PCI device latency timer register
Allocate IRQ to PCI VGA	**Yes**	Assign IRQ to PCI VGA card if card requests IRQ
PCI IDE BusMaster	**Disabled**	
Note: **Bold** text indicates default setting.		

9.5 Boot Menu

To access this menu, select **Boot** on the menu bar at the top of the screen. The submenus for the Boot menu are displayed.



The submenus for the Boot menu are described below.

9.5.1 Boot Settings Configuration Submenu

To access this submenu, select **Boot** on the menu bar, then **Boot Settings Configuration**. The Boot Settings Configuration options are described in Table 61.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 61. Boot Settings Configuration Submenu Options (Sheet 1 of 2)

Feature	Options	Description
Quick Boot	**Enabled**	Enable the BIOS to skip certain tests while booting, to decrease the time needed to boot the system.
Quiet Boot	**Disabled**	Display normal POST message/OEM logo
AddOn ROM Display Mode	**Force BIOS**	
Note: **Bold** text indicates default setting.		

Table 61. Boot Settings Configuration Submenu Options (Sheet 2 of 2)

Feature	Options	Description
Bootup Num-Lock	**On**	
Wait For "F1" If Error	**Enabled**	Enable waiting for F1 key to be pressed if error occurs
Hit "DEL" Message Display	**Enabled**	Display "Press DEL to run Setup" in POST
Interrupt 19 Capture	**Disabled**	Disable the ability for option ROMs to trap interrupt 19
Soft Reset	**Disabled**	Disable soft reset feature
Note: **Bold** text indicates default setting.		

9.5.2 Boot Device Priority Submenus

To access these submenus, select **Boot** on the menu bar, then **<n> Boot Device**, where "n" is the order of priority. The Boot Device Priority options are described in Table 62.

Note: When using the Multimedia Blade, set the first boot device to the device where you will be installing the Linux operating system from (for example, USB disk or SCSI device). After setting the boot device, confirm that the setting took effect; reset it if necessary.

Table 62. Boot Device Priority Submenu Options

Feature	Options	Description
1st Boot Device	**USB: USB Disk †** SCSI: #xxx IDxx† HDD: PM- 128MB ATA Flash Disk HDD: SM- 128MB ATA Flash Disk Network: IBA GE Slot 0400 v0006 Network: IBA GE Slot 0401 v0006 Network: IBA GE Slot 0300 v0006 Network: IBA GE Slot 0301 v0006 Network: IBA GE Slot 0200 v0006 Network: IBA GE Slot 0201 v0006	Set the first boot device to the device where you will be installing the Linux operating system from. † This option will only appear if a USB disk or SAS drive is connected. When USB drive is installed, it will always default to 1st boot device. When there is no USB disk installed and a SAS hard drive is installed, the SAS hard drive becomes the 1st boot device. Booting from anything other than SAS or USB is not supported.
2nd Boot Device	**HDD: PM- 128 MB ATA Flash Disk**	
3rd Boot Device	**HDD: SM- 128MB ATA Flash Disk**	
4th Boot Device	**Network: IBA GE Slot 0400 v0006**	
5th Boot Device	**Network: IBA GE Slot 0401 v0006**	
6th Boot Device	**Network: IBA GE Slot 0300 v0006**	
7th Boot Device	**Network: IBA GE Slot 0301 v0006**	
8th Boot Device	**Network: IBA GE Slot 0200 v0006**	
9th Boot Device	**Network: IBA GE Slot 0201 v0006**	
Note: **Bold** text indicates default setting.		

9.5.3 OS Load Timeout Timer

To access this submenu, select **Boot** on the menu bar, then **OS Load Timeout Timer**. The OS Load Timeout Timer options are described in Table 63.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 63. OS Load Timeout Timer Submenu Options

Feature	Options	Description
OS Load Timeout	**Disabled**	
OS Load Action	**Reset**	Action upon timeout is hard reset
Note: **Bold** text indicates default setting.		

9.6 Security Menu

To access this menu, select **Security** on the menu bar at the top of the screen. The Security menu options are described in Table 64.

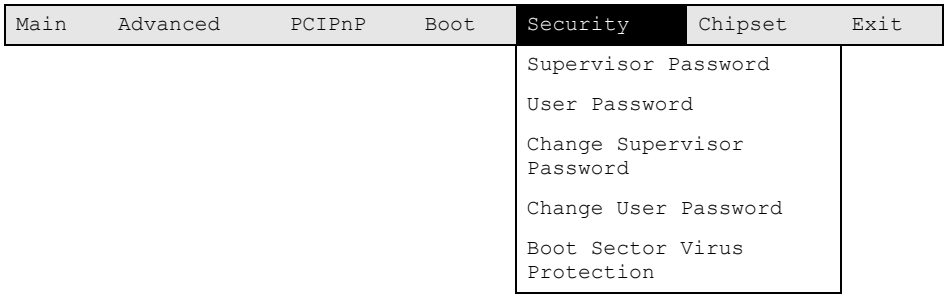
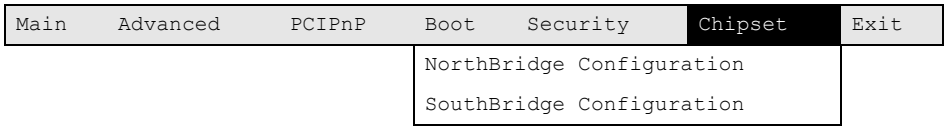


Table 64. Security Menu Options

Feature	Options	Description
Supervisor Password	Not Installed	Display the supervisor password status
User Password	Not Installed	Display the user password status
Change Supervisor Password		Set the supervisor password
Change User Password		Set the user password
Boot Sector Virus Protection	**Disabled**	Disable boot sector virus protection
Note: **Bold** text indicates default setting.		

9.7 Chipset Menu

To access this menu, select **Chipset** on the menu bar at the top of the screen. The submenus for the Chipset menu are displayed.



The submenus for the Chipset menu are described below.

9.7.1 NorthBridge Configuration Submenu

To access this submenu, select **Chipset** on the menu bar, then **NorthBridge Configuration**. The NorthBridge Configuration options are described in Table 65.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 65. NorthBridge Configuration Submenu Options

Feature	Options	Description
Memory remap feature	**Enabled**	Allow remapping of overlapped PCI memory above the total physical memory
Note: **Bold** text indicates default setting.		

9.7.2 SouthBridge Configuration Submenu

To access this submenu, select **Chipset** on the menu bar, then **SouthBridge Configuration**. The SouthBridge Configuration options are given in Table 66.

Warning: When using the Multimedia Blade, do not change the default settings.

Table 66. SouthBridge Configuration Submenu Options

Feature	Options	Description
ICH SIO Serial Port1 Address	**3F8 / IRQ4**	ICH serial I/O unit serial port 1 base address
ICH SIO Serial Port2 Address	**2F8 / IRQ3**	ICH serial I/O unit serial port 2 base address
Note: **Bold** text indicates default setting.		

9.8 Exit Menu

To access this menu, select **Exit** on the menu bar at the top of the screen. The Exit menu options are described in Table 67.

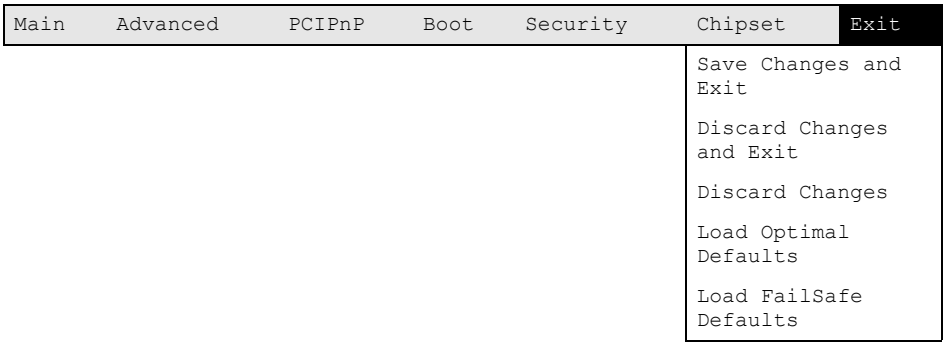


Table 67. Exit Menu Options

Feature	Options	Description
Save Changes and Exit		Exit system Setup after saving changes. Use this to save your configured settings to the CMOS and flash. Same as pressing F10
Discard Changes and Exit		Exit system Setup without saving changes Same as pressing ESC
Discard Changes		Discard changes without exiting Same as pressing F7
Load Optimal Defaults		Load optimal default values Same as pressing F9
Load FailSafe Defaults		Load failsafe default values Same as pressing F8

10.0 Specifications

This chapter provides the following Multimedia Blade specifications:

- Operation
- Connectivity
- Operating System
- Power
- Environmental
- Connections
- Storage
- Physical
- Specification Compliance
- Regulatory Compliance

10.1 Operation

Number of ports	250 to 500 of G.711 (densities for some other coders may be lower)
Form factor	AdvancedTCA* with AdvancedMC* and RTM
Media processing	Voice, T.38 fax, audio conferencing, audio transcoding, and video messaging (play/record)
Voice processing	Play, record, tone generation and detection: DTMF (inband and RFC2833), call progress analysis tones, and custom tones
Audio conferencing	Active talker detection, n-way summing (configurable 1 to 6 speakers), DTMF clamping, coach-pupil mode, per party volume control
Play/record coders	Video: H.263 Profile 0 Level 1 (native mode) Audio: OKI ADPCM (24K, 32K); G.711 Alaw, µlaw 48K, 64K; Linear PCM 8b 8K
Network coders	Video: H.263 Profile 0 Level 30 Audio: G.711 (10 ms, 20 ms, 30 ms frames), G.723, G.726, G.729a/b, AMR-NB, EVRC
Echo cancellation	G.168-2004, echo tail up to 128 ms
File I/O	Remote (NFS) Local storage (up to 35 GB)
Application model	Local and remote
Local APIs	R4/Global Call with video extensions
Remote media control interface	MSML
Manageability	Full remote capability via Command Line Interface (CLI) and SNMP v2 and v3. Supports IPMI v2.0

10.2 Connectivity

Network interfaces	IP TDM
TDM connectivity	16 E1/T1 (up to 496 time slots) via optional RTM (2 RJ-21 connectors)
TDM signaling	Clear channel, which requires a separate SS7 signaling module
IP signaling	SIP-based connection control
Ethernet connectivity	Gigabit Ethernet Base (signaling, file I/O) Fabric (RTP media streaming) RTM (signaling, media streaming, signaling, file I/O)
Ethernet redundancy	Dual port for each connection

10.3 Operating System

Linux	Red Hat Linux RHEL* 4.0U2 SUSE Linux SLES* Release 9 SP3

10.4 Power

Supported voltage (normal)	-38 VDC to -72 VDC
Maximum power draw	Baseboard and AMC: 175W RTM: 12W

10.5 Environmental

Ambient temperature	Operating (normal): 5°C to 40°C (board intake) Operating (short term): -5°C to 55°C Storage: -40°C to 70°C
Airflow	Operating: 30 CFM per minute minimum
Humidity	Operating: 15% to 90% non-condensing at 55°C Storage: 5% to 95% non-condensing at 40°C
Vibration	Operating: 5 to 100Hz: 1G @ 0.25 octave/minute; 100 to 500Hz: 1G @ 1 octave/minute
Shock (unpackaged)	Operating: 30 G, 11 ms half sine Non-operating: 50 G, 170 inches/second trapezoidal

10.6 Connections

Front panel I/O	One USB 2.0 port One serial port (RJ-45) One x4 SAS connector with support for 2 SAS ports Two 10/100/1000 Ethernet ports LEDs for hot swap, out of service, health, hard drive activity, and Ethernet ports
Backplane	Dual Gigabit Ethernet (AdvancedTCA Base Interface) Quad Gigabit Ethernet (AdvancedTCA Fabric Interface; PICMG 3.1, option 2) Dual IPMB connections (Zone 1) Support for Rear Transition Module (Zone 3)
Rear Transition Module ports	16 E1/T1 ports (2 RJ-21 connectors, 8 spans each) Two Gigabit Ethernet ports

10.7 Storage

Hard drive capacity	37 GB SAS
DRAM capacity	2 GB SDRAM

10.8 Physical

Height	8U, 14 in. (35.56 cm)
Width	1.2 in. (3.048 cm)
Depth	11.02 in. (28 cm)
Weight (main board)	7.76 lbs. (3.52 kg) — without packaging 9.58 lbs. (4.3 kg) — with packaging
Weight (RTM)	1.79 lbs. (0.8 kg) — without packaging 3.5 lbs. (1.5 kg) — with packaging

10.9 Specification Compliance

AdvancedTCA	AdvancedTCA 3.0 R1.0 and ECN001

10.10 Regulatory Compliance

Electromagnetic compatibility for telecom network equipment	EN 300 386 V1.3.1 (2001-09)
Emissions	CISPR22:1997 & 2003/EN55022:1998 & EN55022 A1:2000 & A2:2003 Class A EN 300 386 V1.3.2:2003
EMC (additional standard)	FCC 47 CFR Ch1, part 15, ES201 468:03.00
Safety	EN60950:2000, UL 60950:2000
Climate	EN 300 019-2-2, 09/1999/T2.3 EN 300 019-2-3, 09/1999/T3.1E
Earthquake zone	EN 300 019-2-3, 09/1999/T3.1-T3.5
Transportation	EN 300 019-2-2, 09/1999/T2.3
NEBS compliance	Ready for NEBS Level 3 system certification when packaged or installed: All materials UL 94V-1 or better fire resistant Tolerates airborne contaminants Meets shock and vibration requirements

11.0 Troubleshooting

This chapter provides information about debugging your system.

If you suspect a problem with the Multimedia Platform, monitor var/log/messages and look for any hardware or service errors. Keep a copy of var/log/messages from the time you start the service and run your application to the time you stop the service. Make this file available to your Dialogic support representative upon request. For more information on CLI logging, see Section 4.10.2, “CLI Logging” on page 45.

Appendix A Supported IPMI Commands

Table 68 shows the IPMI commands supported by the Dialogic® Multimedia Blade for AdvancedTCA®.

Table 68. Supported IPMI Commands (Sheet 1 of 6)

Command	NetFn	Cmd
IPM Device "Global" Commands		
Get Device ID	App	1
Cold Reset	App	2
Get Self Test Results	App	4
Set ACPI Power State	App	6
Get ACPI Power State	App	7
BMC Watchdog Timer Commands		
Reset Watchdog Timer	App	22h
Set Watchdog Timer	App	24h
Get Watchdog Timer	App	25h
BMC Device and Messaging Commands		
Set BMC Global Enables	App	2Eh
Get BMC Global Enables	App	2Fh
Clear Message Flags	App	30h
Get Message Flags	App	31h
Get Message	App	33h
Send Message	App	34h
Read Event Message Buffer	App	35h
Get System GUID	App	37h
Get Channel Authentication Capabilities	App	38h
Get Session Challenge (through LAN interface only)	App	39h
Activate Session (through LAN interface only)	App	3Ah
Set Session Privilege Level (through LAN interface only)	App	3Bh
Close Session (through LAN interface only)	App	3Ch
Get Session Info (through LAN interface only)	App	3Dh
Get AuthCode	App	3Fh
Set Channel Access	App	40h
Get Channel Access	App	41h
Get Channel Info	App	42h

Table 68. Supported IPMI Commands (Sheet 2 of 6)

Command	NetFn	Cmd
Set User Access	App	43h
Get User Access	App	44h
Set User Name	App	45h
Get User Name	App	46h
Set User Password	App	47h
Activate Payload	App	48h
Deactivate Payload	App	49h
Get Payload Activation Status	App	4Ah
Get Payload Instance Info	App	4Bh
Set User Payload Access	App	4Ch
Get User Payload Access	App	4Dh
Get Chan Payload Support	App	4Eh
Get Chan Payload Version	App	4Fh
Get Chan OEM Payload Info	App	50h
Master Write Read I2C	App	52h
Get Chan Cipher Suites	App	54h
Suspend Payload Encryption	App	55h
Set Channel Security Keys	App	56h
Reserved	App	E0h
Chassis Device Commands – 00h		
Get Chassis Status	Chassis	01h
Chassis Control	Chassis	02h
Get System Restart Cause	Chassis	07h
Set System Boot Options	Chassis	08h
Get System Boot Options	Chassis	09h
Event Commands – 04h		
Set Event Receiver	S/E	00h
Get Event Receiver	S/E	01h
Platform Event (a.k.a. “Event Message”)	S/E	02h
PEF and Alerting Commands – 04h		
Get PEF Capabilities	S/E	10h
Arm PEF Postpone Timer	S/E	11h
Set PEF Configuration Parameters	S/E	12h
Get PEF Configuration Parameters	S/E	13h
Set Last Processed Event ID	S/E	14h
Get Last Processed Event ID	S/E	15h
Sensor Device Commands – 04h		
Get Device SDR Info	S/E	20h
Get Device SDR	S/E	21h

Table 68. Supported IPMI Commands (Sheet 3 of 6)

Command	NetFn	Cmd
Reserve Device SDR Repository	S/E	22h
Set Sensor Hysteresis	S/E	24h
Get Sensor Hysteresis	S/E	25h
Set Sensor Threshold	S/E	26h
Get Sensor Threshold	S/E	27h
Set Sensor Event Enable	S/E	28h
Get Sensor Event Enable	S/E	29h
Re-arm Sensor Events	S/E	2Ah
Get Sensor Event Status	S/E	2Bh
Get Sensor Reading	S/E	2Dh
FRU Device Commands – 0Ah		
Get FRU Inventory Area Info	Storage	10h
Read FRU Data	Storage	11h
Write FRU Data	Storage	12h
SDR Repository Commands – 0Ah		
Get SDR Repository Info	Storage	20h
Get SDR Repository Allocation Info	Storage	21h
Reserve SDR Repository	Storage	22h
Get SDR	Storage	23h
Add SDR	Storage	24h
Partial Add SDR	Storage	25h
Delete SDR	Storage	26h
Clear SDR Repository	Storage	27h
Get SDR Repository Time	Storage	28h
Set SDR Repository Time	Storage	29h
Enter SDR Repository Update Mode	Storage	2Ah
Exit SDR Repository Update Mode	Storage	2Bh
Run Initialization Agent	Storage	2Ch
SEL Device Commands – 0Ah		
Get SEL Info	Storage	40h
Get SEL Allocation Info	Storage	41h
Reserve SEL	Storage	42h
Get SEL Entry	Storage	43h
Add SEL Entry	Storage	44h
Partial Add SEL Entry	Storage	45h
Delete SEL Entry	Storage	46h
Clear SEL	Storage	47h
Get SEL Time	Storage	48h
Set SEL Time	Storage	49h

Table 68. Supported IPMI Commands (Sheet 4 of 6)

Command	NetFn	Cmd
LAN Device Commands – 0Ch		
Set LAN Configuration Parameters	Transport	01h
Get LAN Configuration Parameters	Transport	02h
Suspend BMC ARPs	Transport	03h
Get IP/UDP/RMCP Statistics	Transport	04h
Serial/Modem Device Commands – 0Ch		
Set Serial/Modem Configuration	Transport	10h
Get Serial/Modem Configuration	Transport	11h
Set Serial/Modem Mux	Transport	12h
Serial/Modem Connection Active	Transport	18h
Callback	Transport	19h
Set SOL 2.0 Configuration	Transport	21h
Get SOL 2.0 Configuration	Transport	22h
AdvancedTCA - 2Ch		
Get PICMG Properties	PICMG	00h
Get Address Info	PICMG	01h
FRU Control	PICMG	04h
Get FRU LED Properties	PICMG	05h
Get LED Color Capabilities	PICMG	06h
Set FRU LED State	PICMG	07h
Get FRU LED State	PICMG	08h
Set IPMB State	PICMG	09h
Set FRU Activation Policy	PICMG	0Ah
Get FRU Activation Policy	PICMG	0Bh
Set FRU Activation	PICMG	0Ch
Get Device Locator Record ID	PICMG	0Dh
Set Port State	PICMG	0Eh
Get Port State	PICMG	0Fh
Compute Power Properties	PICMG	10h
Set Power Level	PICMG	11h
Get Power Level	PICMG	12h
Get Fan Speed Properties	PICMG	14h
Get IPMB Link Info	PICMG	18h
Set AMC Port State (AMC.0)	PICMG	19h
Get AMC Port State (AMC.0)	PICMG	1Ah
Intel OEM Commands (INTEL = 30h)		
Reset BIOS Flash Type	INTEL	01h
Set Fabric Interface Selection	INTEL	02h
Get Fabric Interface Selection	INTEL	03h

Table 68. Supported IPMI Commands (Sheet 5 of 6)

Command	NetFn	Cmd
Get Fabric Interface Actual Selection	INTEL	04h
Set Control State	INTEL	05h
Get Control State	INTEL	06h
Get Version Data	INTEL	07h
Reserved	INTEL	0Ah thru 0Fh
Reserved	INTEL	18h thru 1Fh
Reserved	INTEL	20h
Get DIMM State	INTEL	21h
Set DIMM State	INTEL	22h
ReArm DIMMs	INTEL	23h
Sync SMBus Arbitration	INTEL	25h
Set Processor Status	INTEL	28h
Get Processor Status	INTEL	29h
ReArm Processors	INTEL	2Ah
Disable FRB Action	INTEL	2Bh
Set System GUID	INTEL	41h
SetAuxChannelInfo	INTEL	42h
SetGetSDRTransChans	INTEL	43h
Log Post Code	INTEL	47h
SEL Internal Platform Event	INTEL	48h
Set SM Signal	INTEL	50h
Get SM Signal	INTEL	51h
Get Self Test History	INTEL	52h
Manufacturing Test Mode	INTEL	54h
Graceful OS Shutdown	INTEL	70h
Init Agent Started	INTEL	80h
Init Agent End	INTEL	81h
Get ACPI Configuration	INTEL	82h
Set ACPI Configuration	INTEL	83h
Reserved	INTEL	86h thru 8Ch
Reserved	INTEL	A0h
Reserved	INTEL	A1h
Reserved	INTEL	A2h
Reserved	INTEL	A3h
Reserved	INTEL	A4h
Reserved	INTEL	A5h
Reserved	INTEL	A6h
Reserved	INTEL	A7h
Reserved	INTEL	A8h
Get NMI Source	INTEL	E6h

Table 68. Supported IPMI Commands (Sheet 6 of 6)

Command	NetFn	Cmd
Set NMI Source	INTEL	EDh
NMI Enable Disable	INTEL	F7h
Get Latest Port80 Codes	INTEL	FAh
Other Intel OEM Commands: PLAT=32h		
Get HW Info	PLAT	01h
Get Power Unit Status	PLAT	02h
Reserved	PLAT	55h
Reserved	PLAT	57h

Appendix B BIOS Error Messages

Table 69 lists the BIOS error messages supported by the Multimedia Blade.

Table 69. BIOS Error Messages

Error Message	Explanation of Error Message
Timer Error	This timer is based on the 8254 that resides in ICH-3. The error message indicates an error while programming the count register of the timer. This may indicate a problem with the timer in ICH-3.
CMOS Battery Low	The BIOS reports this error message when the status bit (RTC_REGD.Bit7) in the ICH3 is low. This bit is hard wired to RTC power, so it is low when the voltage in SuperCAP is low.
CMOS Settings Wrong	The BIOS loads default values when it detects CMOS corruption. This error message is triggered if the BIOS fails to load the default value to CMOS.
CMOS Checksum Bad	CMOS contents failed the checksum check. Indicates that the CMOS data has been changed by a program other than the BIOS, or the CMOS is not retaining its data due to hardware malfunction.
RAM R/W test failed	Indicates a BIOS fail to read/write to memory content during RAM R/W test. The RAM R/W test is executed during POST.
CMOS Date/ Time Not Set	Indicates that the BIOS has detected an invalid value in the date & time register, for example, invalid date = 50h or invalid month = 13h.
Clear CMOS Jumper installed	Indicates that switch (SW 5-2) is set to OFF.
Clear Password Jumper installed	Indicates that switch (SW 5-1) is set to OFF.
MFG Jumper installed	Indicates that switch (SW 5-3) is set to OFF (for manufacturing test use only)
BMC in Update Mode	Indicates that switch (SW 5-8) is set.
BMC does not respond to BIOS IPMI command	Occurs when the BIOS issues an IPMI command to the IPMC, but the IPMC does not respond to the command and does not return a successful completion code to the BIOS.
System Event Log is Full	Indicates that System Event Log (SEL) storage is full.
Refresh timer test failed	This timer is a counter based on the 82C54, which provides a memory refresh request signal periodically. The memory content needs to be refreshed to compensate for the gradual leakage of charge from the capacitors that store the data.
KBC BAT Test failed	Indicates that the keyboard controller BAT test has failed.

Appendix C POST Checkpoints

Table 70, Table 71, and Table 72 describe the Power On Self Test (POST) codes generated by the BIOS. During POST, the BIOS generates diagnostic progress codes (POST codes) to I/O port 80h. If the POST fails, execution stops and the last POST code generated is left at port 80h. This code is useful for determining the point where an error occurred.

Port 80h POST codes can be retrieved from IPMC with an OEM IPMI command from the Shelf Management Controller (ShMC). Refer to Appendix A, "Supported IPMI Commands."

Table 70. Boot Block Initialization Code Checkpoints

Checkpoint	Description
Before D0	If the boot block debugger is enabled, CPU cache-as-RAM functionality is enabled at this checkpoint. The stack is enabled at this checkpoint.
D0	Early Boot Strap Processor (BSP) initialization like microcode update, frequency, and other CPU critical initialization. Early chipset initialization is done at this checkpoint.
D1	Early super I/O initialization is done including RTC and keyboard controller. The serial port is enabled at this point if needed for debugging. NMI is disabled. Performs keyboard controller BAT test. Saves power-on CPUID value in scratch CMOS. Goes to flat mode with 4 GByte limit and GA20 enabled.
D2	Verifies the boot block checksum. The system hangs here if the checksum is bad.
D3	Disables cache before memory detection. Executes full memory sizing module. If the memory sizing module is not executed, start memory refresh and do memory sizing in boot block code. Does additional chipset initialization. Re-enables cache. Verify that flat mode is enabled.
D4	Tests base 512 KBytes memory. Adjusts policies and caches first 8 MBytes. Sets stack.
D5	Bootblock code is copied from ROM to lower system memory and control is given to it. The BIOS now executes from RAM. Copies compressed boot block code to memory in right segments. Copies BIOS from ROM to RAM for faster access. Performs main BIOS checksum and updates recovery status accordingly.
D6	Both key sequence and OEM-specific methods are checked to determine if BIOS recovery is forced. If BIOS recovery is necessary, control flows to checkpoint E0.
D7	Restores the CPUID value back into the register. The Bootblock-Runtime interface module is moved to system memory and control is given to it. Determines whether to execute serial flash.
D8	The Runtime module is uncompressed into memory. CPUID information is stored in memory.
D9	Stores the uncompressed pointer for future use in PMM. Copies the main BIOS into memory. Leaves all RAM below 1 MByte read-write including E000 and F000 shadow areas, but closing SMRAM.
DA	Restore the CPUID value back into the register. Give control to BIOS POST (ExecutePOSTKernel). See Table 71 for more information.
DC	System wakes from ACPI S3 state.
E1-E8 EC-EE	OEM memory detection/configuration error. This range is reserved for chipset vendors and system manufacturers. The error associated with this value may be different depending on the platform.

Table 71. POST Code Checkpoints (Sheet 1 of 2)

Checkpoint	Description
03	Disables NMI, parity, video for EGA, and DMA controllers. Initializes BIOS, POST, runtime data area. Also initializes BIOS modules on POST entry and GPNV area. Initializes CMOS as mentioned in the kernel variable “wCMOSFlags”.
04	Checks CMOS diagnostic byte to determine if battery power is OK and CMOS checksum is OK. Verifies CMOS checksum manually by reading storage area. If the CMOS checksum is bad, updates the CMOS with power-on default values and clear passwords. Initializes the status register A. Initializes data variables that are based on CMOS Setup questions. Initializes both the 8259-compatible PICs in the system.
05	Initializes the interrupt controlling hardware (generally PIC) and interrupt vector table.
06	Does a R/W test to CH-2 count register. Initializes CH-0 as the system timer. Installs the POSTINT1Ch handler. Enables IRQ-0 in PIC for system timer interrupt. Traps INT1Ch vector to “POSTINT1ChHandlerBlock”.
07	Fixes CPU POST interface calling pointer.
08	Initializes the CPU. The BAT test is being done on the KBC. Program the keyboard controller command byte is being done after auto detection of the KB/MS using AMI KB-5.
C0	Early CPU Init Start. Disable cache. Init local APIC.
C1	Sets up bootstrap processor information.
C2	Sets up bootstrap processor for POST.
C5	Enumerates and sets up application predecessors.
C6	Re-enables cache for bootstrap processor.
C7	Early CPU Init Exit.
0A	Initializes the 8042-compatible keyboard controller.
0B	Detects the presence of a PS/2 mouse.
0C	Detects the presence of a keyboard in the KBC port.
0E	Tests and initializes different input devices. Also, updates the kernel variables. Traps the INT09h vector, so that the POST INT09h handler gets control for IRQ1. Uncompresses all available language, BIOS logo, and silent logo modules.
13	Early POST initialization of chipset registers.
20	Relocates the system management interrupt vector for all CPUs in the system.
24	Uncompresses and initializes any platform-specific BIOS modules. GPNV is initialized at this checkpoint.
2A	Initializes different devices through DIM. See Table 72, “Device Initialization Manager (DIM) Code Checkpoints” on page 160 for more information.
2C	Initializes different devices. Detects and initializes the video adapter installed in the system that has optional ROMs.
2E	Initializes all the output devices.
31	Allocates memory for ADM module and uncompresses it. Gives control to the ADM module for initialization. Initializes language and font modules for ADM. Activates the ADM module.
33	Initializes the silent boot module. Sets the window for displaying text information.
37	Displays the sign-on message, CPU information, setup key message, and any OEM-specific information.
38	Initializes different devices through DIM. See Table 72, “Device Initialization Manager (DIM) Code Checkpoints” on page 160 for more information. USB controllers are initialized at this point.
39	Initializes DMAC-1 and DMAC-2.
3A	Initializes RTC date/time.
3B	Tests for total memory installed in the system. Also, checks for DEL or ESC keys to limit memory test. Displays total memory in the system.

Table 71. POST Code Checkpoints (Sheet 2 of 2)

Checkpoint	Description
3C	Mid-POST initialization of chipset registers.
40	Detects different devices successfully installed in the system (for example, parallel ports, serial ports, and coprocessor in the CPU) and updates the BDA, EBDA, etc.
52	Updates CMOS memory size from memory found in memory test. Allocates memory for the extended BIOS data area from base memory. Programs the memory hole or any kind of implementation that needs an adjustment in system RAM size.
60	Initializes NUM-LOCK status and programs the KBD typematic rate.
75	Initializes Int-13 and prepares for IPL detection.
78	Initializes IPL devices controlled by BIOS and option ROMs.
7C	Generates and writes content of ESCD in NVRam.
84	Logs errors encountered during POST.
85	Displays errors to the user and gets the user response to error.
87	Executes BIOS Setup if needed/requested. Checks boot password if installed.
8C	Late POST initialization of chipset registers.
8D	Builds ACPI tables (if ACPI is supported).
8E	Programs the peripheral parameters. Enable/disable NMI as selected.
90	Initializes system management interrupt by invoking all handlers. ***Note:*** This checkpoint comes immediately after checkpoint 20h.
A1	Clean-up work needed before booting to OS.
A2	Takes care of runtime image preparation for different BIOS modules. Fills the free area in F000h segment with 0FFh. Initializes the Microsoft IRQ Routing Table. Prepares the runtime language module. Disables the system configuration display if needed.
A4	Initializes runtime language module. Displays the boot option popup menu.
A7	Displays the system configuration screen if enabled. Initializes the CPUs before boot, which includes the programming of the MTRRs.
A9	Waits for user input at configuration display if needed.
AA	Uninstalls the POST INT1Ch vector and the INT09h vector.
AB	Prepares the BBS for Int 19 boot. Initializes MP tables.
AC	End of POST initialization of chipset registers. De-initializes the ADM module.
B1	Saves the system context for ACPI. Prepares the CPU for OS boot including final MTRR values.
00	Passes control to the OS loader (typically INT19h).

Table 72. Device Initialization Manager (DIM) Code Checkpoints

Checkpoint	Description
2A	Initializes different buses and performs the following functions: • **Function 0: Reset, Detect, and Disable** - Disables all device nodes, PCI devices, and PnP ISA cards. Assigns PCI bus numbers. • **Function 1: Static Device Initialization** - Initializes all static devices that include manual configured on-board peripherals, memory, and I/O decode windows in PCI-PCI bridges, and noncompliant PCI devices. Reserves static resources. • **Function 2: Boot Output Device Initialization** - Searches for and initializes any PnP, PCI, or AGP video devices.
38	Initializes different buses and performs the following functions: • **Function 3: Boot Input Device Initialization** - Searches for and configures PCI input devices and detects if system has a standard keyboard controller. • **Function 4: IPL Device Initialization** - Searches for and configures all PnP and PCI boot devices. • **Function 5: General Device Initialization** - Configures all on-board peripherals that are set to automatic configuration and configures all remaining PnP and PCI devices.

Appendix D Using Shelf Managers with the Multimedia Platform

This appendix provides information about using various shelf managers with the Dialogic® Multimedia Platform for AdvancedTCA® (hereinafter referred to as the Multimedia Platform). Compatibility issues may arise if incorrect firmware versions of the shelf manager are used.

This appendix provides a procedure for upgrading the Schroff Shelf Manager firmware. This appendix also explains how to reprogram the Multimedia Platform RTM FRU file, in case it was overwritten.

The following topics are discussed:

- Using Intel NetStructure® MPCMM001 Chassis Management Module
- Upgrading Schroff Shelf Manager Firmware
- Programming the RTM FRU File

D.1 Using Intel NetStructure® MPCMM001 Chassis Management Module

When using the Intel NetStructure® MPCMM001 Chassis Management Module with the Multimedia Platform, the CMM firmware must be upgraded to Version 6.1.2.315 or later. If an older version is used, the board remains in hot swap mode and the blue LEDs on the board blink.

To download the latest firmware version, see:

http://downloadfinder.intel.com/scripts-df-external/detail_desc.aspx?agr=Y&ProductID=1662&DwnldID=10994

For instructions on installing the new firmware version, see the Software Specification Update:

http://download.intel.com/design/network/specupdt/309201_v6_1_2.pdf

D.2 Upgrading Schroff Shelf Manager Firmware

The Multimedia Platform has a compatibility issue with Schroff Shelf Manager firmware prior to Version 2.2.0. If an older version is used, the Field Replaceable Unit (FRU) file on the Rear Transition Module (RTM) will be overwritten, making the RTM unusable until the RTM is reprogrammed. Before installing the RTM in a Schroff chassis, the Shelf Manager firmware must be upgraded to v2.2.0.

This procedure describes how to upgrade the Schroff Shelf Manager firmware from an older version to Version 2.2.0. The firmware is transferred to the Shelf Manager using TFTP protocol. Once the firmware is on Shelf Manager, it is programmed into the Shelf Manager's flash area.

There are two main parts to this procedure:

- Environment Setup
- Firmware Upgrade

D.2.1 Environment Setup

The environment setup includes:

- Required Components
- Hardware Setup
- Installing Schroff Firmware Package
- Installing TFTP Server
- Setting up the Terminal Emulator
- Assigning the IP Addresses
- Installing sbcupdate Utility

D.2.1.1 Required Components

The following components are required during this procedure:

- TFTP server
- Schroff firmware v2.2.0
- ATCA Shelf Manager ShMM-300 V2.2.0 Firmware Update Procedure document (available at http://www.schroff.us/home.asp)
- Desktop or laptop PC
- Schroff serial cable
- Ethernet crossover cable or Ethernet cable and Ethernet hub
- Schroff 5-slot chassis
- Dialogic board assembly
- Dialogic RTM FRU file

D.2.1.2 Hardware Setup

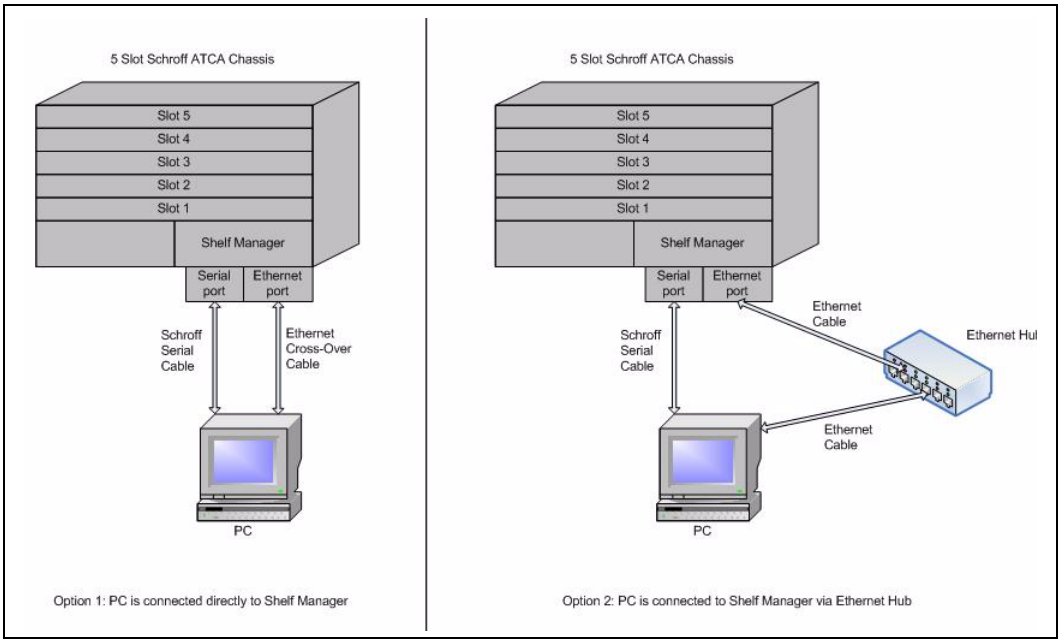
The Shelf Manager needs to be connected to the TFTP server via Ethernet and RS232 serial connections. The connection is illustrated in Figure 19.

The Ethernet connection can be done through a hub or with a crossover cable.

The serial connection is done by connecting the PC running the terminal emulator to the console port of the Shelf Manager. The connection must be done using a special console cable for Schroff Shelves. The console cable is part number CBL000001. Refer to the ATCA Shelf Manager ShMM-300 V2.2.0 Firmware Update Procedure document for a pinout description of the cable.

The jumpers on the Shelf Manager card must be set to route the Ethernet connection to the front connector. Refer to the ATCA Shelf Manager ShMM-300 V2.2.0 Firmware Update Procedure document for instructions on how to set the jumpers.

Figure 19. Connecting PC to the Shelf Manager for Schroff Firmware Upgrade



D.2.1.3 Installing Schroff Firmware Package

Download the Schroff Shelf Manager Firmware Version V2.2.0 to a PC using the following steps:

1. Go to the web site http://www.schroff.us/home.asp
2. Download the file 63998-05154.zip, which contains Schroff Shelf Manager Firmware Version V2.2.0 for ShMM-300.
   a. Choose AdvancedTCA -> Shelf Manager.
   b. Choose Support Information -> Firmware.
   c. Download Schroff Shelf Manager Firmware Version V2.2.0 for ShMM-300.

Create a directory C:\shmm2.2 on your local drive. Unzip the files located in 63998-05154.zip into that directory. The file list includes the following:

- 63998-05154.kernel
- 63998-05154.rfs
- Firmware_Update_Notice_2.2.0.pdf
- Firmware_Update_Procedure_ShMM-300_2.2.0.pdf

The Firmware_Update_Procedure_ShMM-300_2.2.0.pdf document contains the details of the upgrade procedure.

D.2.1.4 Installing TFTP Server

The Schroff firmware upgrade procedure uses TFTP server to transfer the firmware files from the PC to the Shelf Manager. Any available TFTP server can be installed on the PC for that purpose.

This document describes the use of the freely available SolarWinds TFTP server (v8.2). The software can be downloaded from http://www.solarwinds.net

Install SolarWinds TFTP server on the PC. Activate the TFTP server. Configure the TFTP server by doing the following:

1. Select File -> Configure
2. Under TFTP Root Directory tab, select c:\shmm2.2 directory.
3. Configure the TFTP server into transmit and receive file mode by selecting the following:

   File -> Configure -> Security -> Transmit and Receive files -> OK

Leave TFTP server running.

D.2.1.5 Setting up the Terminal Emulator

The Schroff firmware upgrade procedure uses an RS232 terminal emulator program to issue commands to the Shelf Manager. Any available terminal emulator can be used. Set the terminal emulator to 9600, N, 8, 1 (9600 baud, no parity, 8 bits per character, 1 stop bit).

D.2.1.6 Assigning the IP Addresses

The Dialogic baseboard and Shelf Manager need to be assigned the IP addresses. See Figure 20 for an example. Refer to Section D.2.2, “Firmware Upgrade” on page 164 for a description on how to assign the IP addresses to the Shelf Manager.

D.2.1.7 Installing sbcupdate Utility

The Intel sbcupdate utility allows the user to update the FRU and SDR files on the Dialogic baseboard, AdvancedMC, and RTM cards. The latest version of the utility can be downloaded from the technical support web site using the following instructions:

1. Go to the web site http://www.intel.com/support/telecom/computeboards/mpcbl0040/software.htm
2. Under Software and Application Development Tools category, select SBC Utilities.
3. Download MPCBL0020_MPCBL0040_SBC_Utilities_1.3.1-1.zip file to the PC.
4. Uncompress the zip file on the PC.
5. FTP the file “sbcutils-1.3.1-1.i386-rhel4.rpm” to the Dialogic baseboard.

On the Dialogic baseboard, install SBC utilities as follows:

1. If the sbcutils package is already present, you must remove it before attempting to install the latest version with the command:

   # rpm -e sbcutils

2. Once older packages have been removed from the host system, use the following command to install the sbcutils package:

   # rpm -ivh XXXX.rpm (where XXXX represents the name of the RPM file)

D.2.2 Firmware Upgrade

The Firmware_Update_Procedure_ShMM-300_2.2.0.pdf document provides the detailed steps to upgrade the firmware. This section provides supplementary information to that document.

a. Follow steps 1 and 2 described in Firmware_Update_Procedure_ShMM-300_2.2.0.pdf.

b. Before performing step 3, determine or assign the IP addresses to the Shelf Manager and PC. Based on that information, set up the environment variables in Shelf Manager shell as shown in step 3 in Firmware_Update_Procedure_ShMM-300_2.2.0.pdf. Refer to Figure 20 for an example.

— "serverip" is the IP address of the computer that contains TFTP server

— "ipaddr" is the IP address of the Schroff Shelf Manager

— "netmask" is the IP mask of the network that Shelf Manager resides on

— "gateway" is the IP address of the gateway machine on the network where Shelf Manager resides

c. When performing step 3 in Firmware_Update_Procedure_ShMM-300_2.2.0.pdf, define an additional environment variable "rmcpaddr". The value of "rmcpaddr" should be the same as "ipaddr" above. Do not forget to issue "saveenv" command.

d. Follow steps 4 to 9 described in Firmware_Update_Procedure_ShMM-300_2.2.0.pdf.

e. Before performing Step 10 in Firmware_Update_Procedure_ShMM-300_2.2.0.pdf, type the following on the Shelf Manager console:

— Login as root.

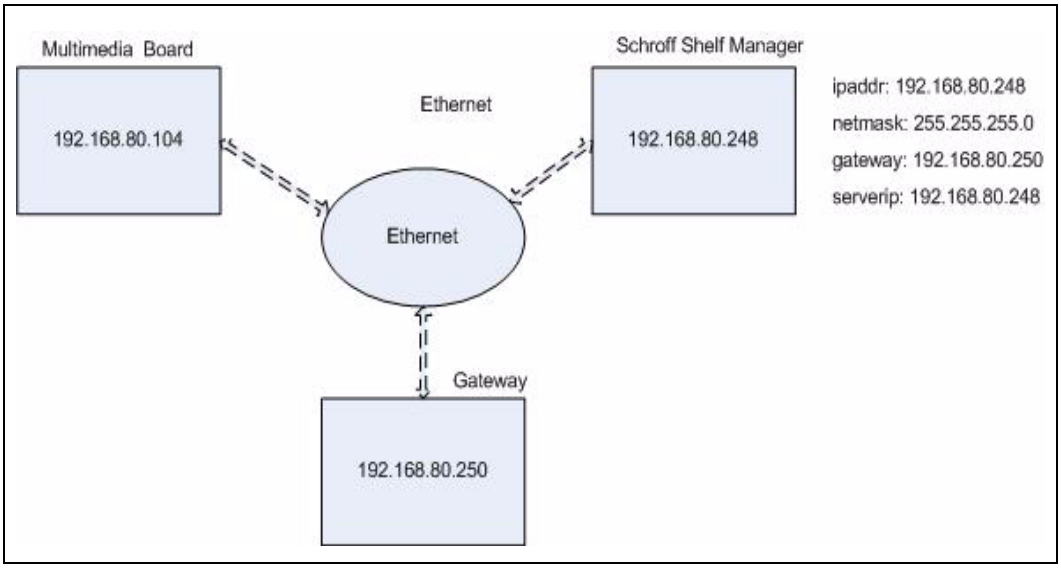
— Type the following: setenv rc2 /etc/rc.acb2

— Toggle the power on Shelf Manager.

f. Perform step 10 in Firmware_Update_Procedure_ShMM-300_2.2.0.pdf, with the following modification. Instead of using "version" command, use "clia version".

g. Verify that the firmware version is 2.2.0.

Figure 20. Example of Initializing Shelf Manager IP Environment Variables



D.3 Programming the RTM FRU File

This section explains how to reprogram the Multimedia Platform RTM FRU file, in case it was overwritten.

D.3.1 Required Components

First, acquire the version of FRU file that needs to be programmed into the RTM. In this section, the name TVL2RTM.FRU is used as an example. Refer to Table 73 for the list of the FRU files.

Table 73. List of FRU Files

Part Number	Description	FRU File Name
MMB496RTMTE01AQ	PSTN RTM	MMB496RTMTE01AQ_200.fru
MMBRTMIP011AQ	IP Only RTM	MMBRTMIP01AQ_200.fru
MMB250BA01AQ	250 port density AdvancedMC	MMB250BA01AQ_200.fru
MMB500BA01AQ	500 port density AdvancedMC	MMB500BA01AQ_200.fru
MPCBL0040	Dialogic baseboard	MPCBL040.FRU

Second, make sure that the Dialogic baseboard can communicate to the Shelf Manager over the Ethernet. The communication can be verified by using the Linux “ping” command.

D.3.2 RTM FRU Programming Procedure

The RTM FRU can be programmed using the following command (all on one line):

```
sbcupdate
-H shmmaddr -Ilan -Uopenhpi -Popenhpi -tbladeX:rtm
--force-fru-update --force-id TVL2RTM.FRU
```

where:

shmmaddr is the RMCP IP address of the Shelf Manager.

bladeX is the number of the slot, in Schroff chassis, in which the Dialogic baseboard is inserted. The possible values for the 5-slot chassis are: blade1, blade2, blade3, blade4, and blade5.

The version of the FRU file can be displayed using the following command:

```
sbcupdate
-H shmmaddr -Ilan -Uopenhpi -Popenhpi -tbladeX:rtm
-Minfo
```

The output will look similar to the example below (the versions of the modules may differ):

```
sbcupdate - Dialogic SBC Utilities 1.3.0.3
Running with the invocation: -H 146.152.80.248 -Ilan -Uopenhpi -Popenhpi -
tblade5:rtm -Minfo
Current firmware version: 0.02.01
Current bootblock version: 1.08

Manufacturer ID: 343 (0x00000157)
Product ID: 2225 (0x08b1)
Device type: RTM
FPGA revision: <Not Available>
```

```
Board revision: <Not Available>

FRU device ID: 1
FRU version: 0.06
FRU Asset Tag: <Blank>

SDR version: SDR File 0.1
SDR Package version: SDR Package 0.0.1

Successful retrieval of data from the target.
Program exiting!
```

D.3.3 AdvancedMC FRU Programming Procedure

The AdvancedMC FRU can be programmed using the following command (all on one line):

```
sbcupdate
-H shmmaddr -Ilan -Uopenhpi -Popenhpi -tbladeX:amc
--force-fru-update --force-id TVL2AMC.FRU
```

where:

shmmaddr is the RMCP IP address of the Shelf Manager.

bladeX is the number of the slot, in Schroff chassis, in which the Dialogic baseboard is inserted. The possible values for the 5-slot chassis are: blade1, blade2, blade3, blade4, and blade5.

The version of the FRU file can be displayed using the following command:

```
sbcupdate
-H shmmaddr -Ilan -Uopenhpi -Popenhpi -tbladeX:amc
-Minfo
```

D.3.4 Baseboard FRU Programming Procedure

The baseboard FRU can be programmed using the following command (all on one line):

```
sbcupdate
-H shmmaddr -Ilan -Uopenhpi -Popenhpi -tbladeX
--force-fru-update --force-id MPCBL040.FRU
```

where:

shmmaddr is the RMCP IP address of the Shelf Manager.

bladeX is the number of the slot, in Schroff chassis, in which the Dialogic baseboard is inserted. The possible values for the 5-slot chassis are: blade1, blade2, blade3, blade4, and blade5.

The version of the FRU file can be displayed using the following command:

```
sbcupdate
-H shmmaddr -Ilan -Uopenhpi -Popenhpi -tbladeX
-Minfo
```