» User Guide «

CP308 uEFI BIOS

Doc. ID: 1030-4232, Rev. 3.0
December 21, 2012

Revision History

Publication Title:	CP308 uEFI BIOS	
Doc. ID:	1030-4232	
Rev.	Brief Description of Changes	Date of Issue
1.0	Initial issue	3-Apr-2009
2.0	General update based on the uEFI BIOS revision 13	22-Jul-2009
3.0	General update based on the uEFI BIOS revision 22	21-Dec-2012

Imprint

Kontron Europe GmbH may be contacted via the following:

MAILING ADDRESS

Kontron Europe GmbH

Sudetenstraße 7

D - 87600 Kaufbeuren Germany

TELEPHONE AND E-MAIL

+49 (0) 800-SALESKONTRON

sales@kontron.com

For further information about other Kontron products, please visit our Internet web site: www.kontron.com.

Disclaimer

Copyright © 2012 Kontron AG. All rights reserved. All data is for information purposes only and not guaranteed for legal purposes. Information has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Kontron and the Kontron logo and all other trademarks or registered trademarks are the property of their respective owners and are recognized. Specifications are subject to change without notice.

Table of Contents

Revision History ..... *ii*
Imprint ..... *ii*
Disclaimer ..... *ii*
Table of Contents ..... *iii*

1. Starting uEFI BIOS Setup 3
1.1 Main Setup Menu 4
1.2 Navigation 5

2. Main Setup 9
2.1 BIOS Information 9
2.2 Memory Information 9
2.3 System Language 9
2.4 System Date 10
2.5 System Time 10
2.6 Access Level 10

3. Advanced Setup 13
3.1 Legacy OpROM 13
3.1.1 Launch PXE OpROM 13
3.1.2 Use Device for PXE 14
3.1.3 Launch Storage OpROM 14
3.2 Trusted Computing 15
3.2.1 TPM Configuration 15
3.2.1.1 TPM Support 15
3.2.1.2 TPM State 16
3.2.2 Pending TPM Operation 16
3.2.3 Current TPM Status Information 16
3.3 CPU Configuration 17
3.3.1 CPU Configuration 18
3.3.2 Intel® SpeedStep™ 18

3.3.3 Execute Disable Bit 18
3.3.4 Limit CPUID Maximum 19
3.3.5 Intel Virtualization Technology 19
3.3.6 TM Support 19
3.3.7 C-States 20
3.3.8 Enhanced C1 20
3.4 SATA Configuration 21
3.4.1 SATA Configuration 21
3.4.2 SATA Mode 21
3.4.3 SATA-ATA Controller 0 22
3.4.4 SATA-ATA Controller 1 22
3.4.5 SATA Speed Limit 22
3.4.6 SATA Hot Plug Support 23
3.4.7 Serial-ATA Ports 0..5 23
3.5 USB Configuration 24
3.5.1 USB Configuration 24
3.5.2 Legacy USB Support 24
3.5.3 USB Hardware Delays and Time-outs 25
3.5.3.1 USB Transfer Timeout 25
3.5.3.2 Device Reset Timeout 25
3.5.3.3 Device Power-up Delay 25
3.5.4 Mass Storage Devices 26
3.6 Serial Port Console Redirection 27
3.6.1 COM0 27
3.6.1.1 Console Redirection 27
3.6.1.2 Console Redirection Settings 27
3.6.2 COM1 28
3.6.2.1 Console Redirection 28
3.6.2.2 Console Redirection Settings 28
3.6.3 Serial Port for Out-of-Band Management/Windows EMS 28
3.6.3.1 Console Redirection 28
3.6.3.2 Out-of-Band Mgmt Port 28
3.6.3.3 Data Bits 28
3.6.3.4 Parity 28

3.6.3.5 Stop Bits 29
3.6.3.6 Terminal Type 29
3.6.4 Console Redirection Settings 29
3.6.4.1 Terminal Type 30
3.6.4.2 Bits per second 30
3.6.4.3 Data Bits 30
3.6.4.4 Parity 30
3.6.4.5 Stop Bits 30
3.6.4.6 Flow Control 31
3.6.4.7 Terminal Resolution 31

4. Chipset Setup 35
4.1 NorthBridge Configuration 36
4.1.1 Primary Display 36
4.1.2 Always Enable PEG 36
4.2 SouthBridge Configuration 37
4.2.1 ICH9 Device Configuration 37
4.2.1.1 Azalia HD Audio 37
4.2.2 High Precision Event Timer Configuration 38
4.2.2.1 High Precision Event Timer 38
4.2.3 USB Configuration 38
4.2.3.1 UHCI Controller 1..3 38
4.2.3.2 UHCI Controller 4 38
4.2.3.3 USB 2.0 (EHCI 1) Support 39
4.2.3.4 USB 2.0 (EHCI 2) Support 39

5. Boot Setup 43
5.1 Boot Configuration 43
5.1.1 Quiet Boot 43
5.1.2 Fast Boot 43
5.1.3 Setup Prompt Timeout 44
5.1.4 Bootup NumLock State 44
5.1.5 GateA20 Active 44

5.1.6 Option ROM Messages 44
5.1.7 Interrupt 19 Capture 45
5.2 Boot Option Priorities 45
5.2.1 Boot Option #1..2 45
5.2.2 Hard Drive/Network Device/CD/DVD ROM Drive/Floppy Drive/etc. ... 45

6. Security Setup 49
6.1 Administrator Password 50
6.2 User Password 50
6.3 Remember the Password 50

7. Save & Exit 53
7.1 Save Changes and Exit 53
7.2 Discard Changes and Exit 53
7.3 Save Changes and Reset 53
7.4 Discard Changes and Reset 54
7.5 Save Changes (Save Options) 54
7.6 Discard Changes (Save Options) 54
7.7 Restore Defaults (Save Options) 54
7.8 Save as User Defaults (Save Options) 54
7.9 Restore User Defaults (Save Options) 54
7.10 Boot Override 54

8. uEFI Shell 57
8.1 Introduction, Basic Operation 57
8.1.1 Shell Startup 57
8.2 Kontron Shell Commands 58
8.2.1 kboardconfig uEFI Shell Command 59
8.2.2 kboardinfo uEFI Shell Command 60
8.2.3 kboot uEFI Shell Command 62
8.2.4 kbootnsh uEFI Shell Command 63
8.2.5 kclearnvram uEFI Shell Command 64

8.2.6 kclsp uEFI Shell Command 64
8.2.7 kflash uEFI Shell Command 65
8.2.8 kipmi uEFI Shell Command 66
8.2.9 kmkramdisk uEFI Shell Command 67
8.2.10 kpassword uEFI Shell Command 68
8.2.11 kresetconfig uEFI Shell Command 69
8.2.12 kwdt uEFI Shell Command 70
8.3 uEFI Shell Scripting 71
8.3.1 Startup Scripting 71
8.3.2 Create a Startup Script 71
8.3.3 Examples of Startup Scripts 71
8.3.3.1 Automatic Booting from USB Flash Drive 71
8.3.3.2 Execute Shell Script on Other Harddrive 71
8.3.3.3 Enable Watchdog and Control PXE Boot 71
8.3.3.4 Handling the Startup Script in the Flash Bank 73

8. Updating the uEFI BIOS 77

8.1 uEFI BIOS Fail-Over Mechanism 77
8.2 Updating Strategy 77
8.3 Fallback Mechanism 77
8.4 Determining the Active Flash 77

This page has been intentionally left blank.

Chapter 1

Starting uEFI BIOS Setup

This page has been intentionally left blank.

1. Starting uEFI BIOS Setup

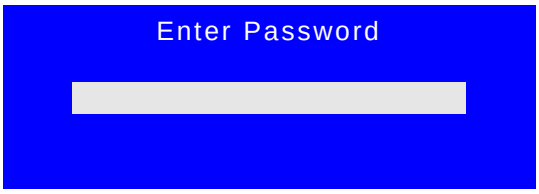
The CP308 is provided with a Kontron-customized, pre-installed and configured version of Aptio® (referred to as uEFI BIOS in this manual), AMI's next generation BIOS firmware based on the Unified Extensible Firmware Interface (uEFI) specification and the Intel® Platform Innovation Framework for EFI. This uEFI BIOS provides a variety of new and enhanced functions specifically tailored to the hardware features of the CP308.

To take advantage of these functions, the uEFI BIOS comes with a Setup program which provides quick and easy access to the individual function settings for control or modification of the uEFI BIOS configuration.

The Setup program allows the accessing of various menus which provide functions or access to sub-menus with more specific functions of their own. The individual menus and the configurable functions are described in this guide.

To start the uEFI BIOS Setup program, follow the steps below:

1. Power on the board.
2. Wait until the first characters appear on the screen (POST messages or splash screen).
3. Press the <DEL> or the <F2> key.
4. If the uEFI BIOS is password-protected, a window such as the one below will appear:



   Enter either the User password or the Administrator password (refer to chapter 6, Security Setup, for further information), press <RETURN>, and proceed with step 2.

5. A Setup menu with the following token attributes will appear.
   The currently active menu and the currently active uEFI BIOS Setup item are highlighted in white.

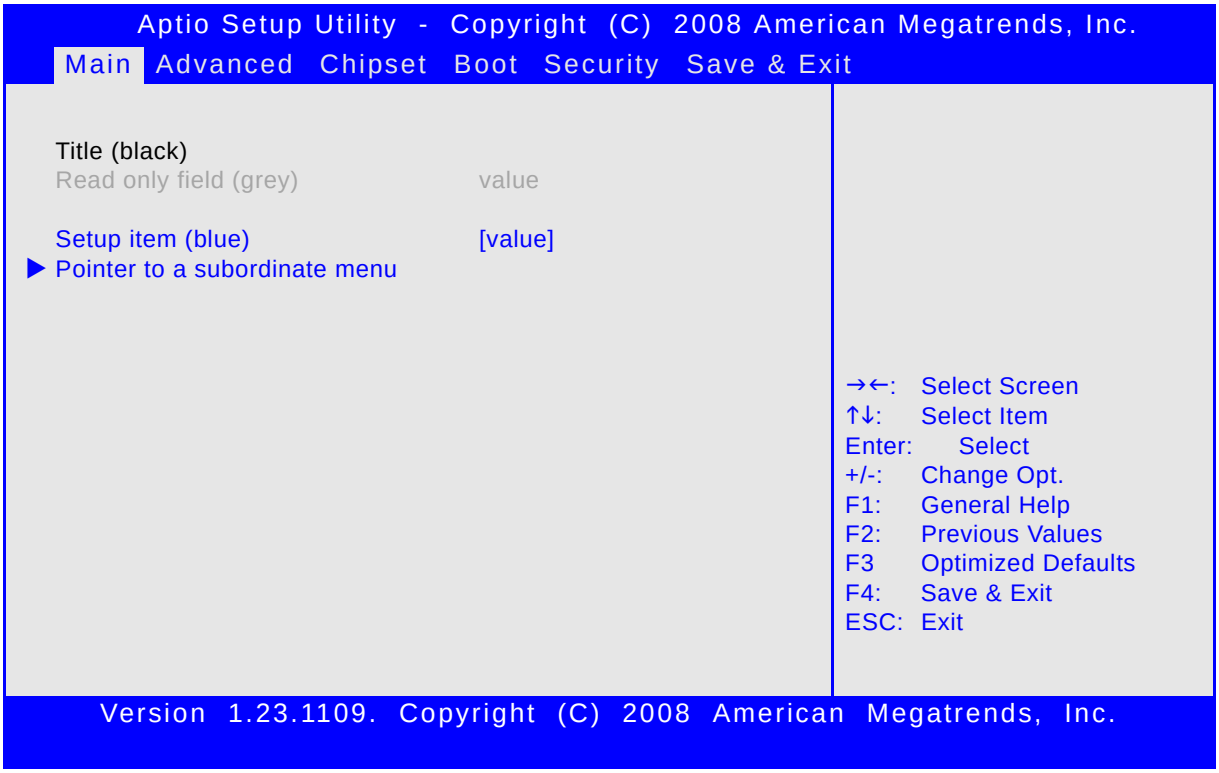
1.1 Main Setup Menu

The Main setup menu is the first screen that appears after starting the Setup program.

At the top of this screen and all of the other major screens, there is a setup menu selection bar, which permits access to all of the other major setup menus. These menus are selected via the left-right arrow keys.

All setup menu screens have two main frames. The left frame displays all the functions that can be configured. They are displayed in blue. Functions displayed in gray provide information about the status or the operational configuration.

The right frame displays the key legend. Above the key legend there is an area reserved for a text message. When a function is selected in the left frame, it is displayed in white. Often a text message will accompany it.



1.2 Navigation

The CP308 uEFI BIOS setup program uses a hot key-based navigation system. A hot key legend is located in the right frame on most setup screens.The following table provides information concerning the usage of these hot keys.

HOT KEY	DESCRIPTION
<F1>	The <F1> key is used to invoke the General Help window.
<F2>	The <F2> key is used to restore the previous values.
<F3>	The <F3> key is used to load the defaults.
<F4>	The <F4> key is used to save the current settings and exit the uEFI BIOS Setup.
→ ← Right/Left	The *Right and Left* <Arrow> keys are used to select a major Setup screen. For example: Main Screen, Advanced Screen, Chipset Screen, etc.
↑ ↓ Up/Down	The *Up and Down* <Arrow> keys are used to select a Setup function or a sub-screen.
+ - Plus/Minus	The *Plus and Minus* <Arrow> keys are used to change the field value of a particular Setup function, for example, system date and time.
<ESC>	The <ESC> key is used to exit a menu or the uEFI BIOS Setup. Pressing the <ESC> key in a sub-menu causes the next higher menu level to be displayed. When the <ESC> key is pressed in a major Setup menu, the uEFI BIOS Setup is terminated without saving any changes made.
<Enter>	The <Enter> key is used to execute a command or select a menu.

This page has been intentionally left blank.

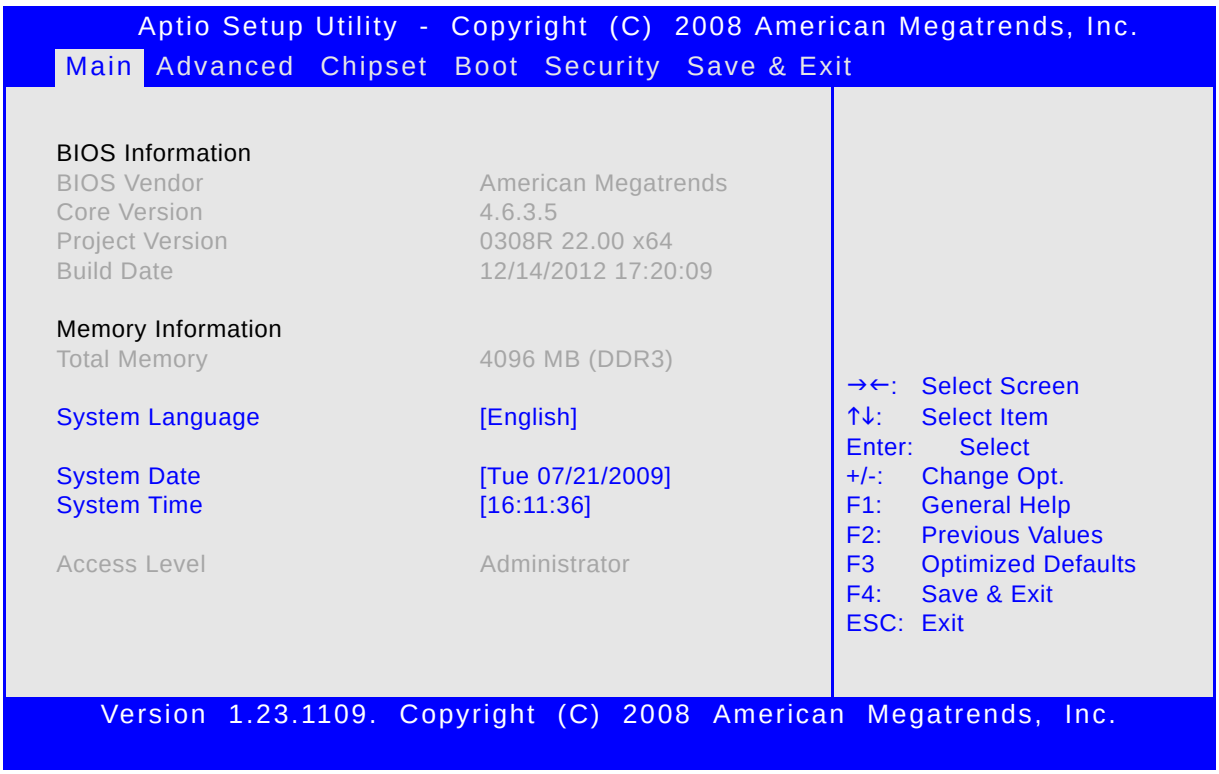
Chapter 2

Main Setup

This page has been intentionally left blank.

2. Main Setup

Upon entering the uEFI BIOS Setup program, the Main setup screen is displayed. This screen provides very basic system information as well as functions for setting the system time and date. In addition, the remaining major setup menus can be accessed from this screen. This screen can also be selected from any other major setup screen by using the Main tab.



2.1 BIOS Information

This function provides display-only information concerning the uEFI BIOS.

2.2 Memory Information

This function provides display-only information concerning the system memory.

2.3 System Language

SETTING	DESCRIPTION
English	Use this function to select the system language. Currently, only English is supported.

2.4 System Date

SETTING	DESCRIPTION
<WD MM/DD/YYYY>	Use this function to change the system date. Select System Date using the Up and Down <Arrow> keys. Enter the new values through the keyboard or press +/- to increment/decrement values. Use “Tab” to switch between date elements.

2.5 System Time

SETTING	DESCRIPTION
<HH:MM:SS>	Use this function to change the system time. Select System Time using the Up and Down <Arrow> keys. Enter the new values through the keyboard or press +/- to increment/decrement values. Use “Tab” to switch between time elements.

Note: The time is in 24-hour format. For example, 5:30 A.M. appears as 05:30:00, and 5:30 P.M. as 17:30:00.

2.6 Access Level

This function provides display-only information concerning the uEFI BIOS Setup accessibility for the current Setup session. Depending on the type of password protection used, one of the following settings is displayed:

SETTING	DESCRIPTION
Administrator	This setting indicates that read/write access to all setup options is available.
User	This setting indicates that only a limited subset of all setup options is modifiable.

Note: If no password is set, the access setup is Administrator.

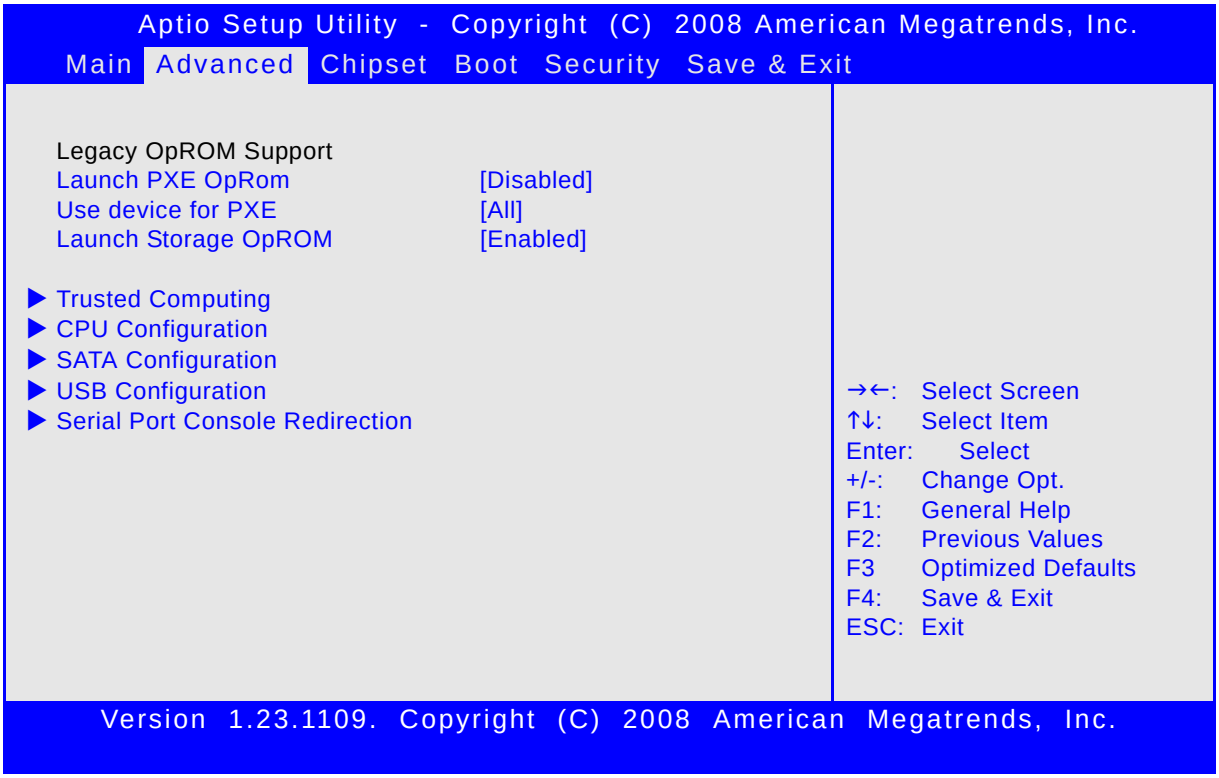
Chapter 3

Advanced Setup

This page has been intentionally left blank.

3. Advanced Setup

Select the Advanced tab to enter the Advanced Setup screen. This screen lists the advanced configuration sub-screens.



3.1 Legacy OpROM

3.1.1 Launch PXE OpROM

SETTING	DESCRIPTION
Disabled	Use this setting to ignore all PXE Option ROMs.
Enabled	Use this setting to load PXE Option ROMs. To limit the PXE support to particular devices, use the function Use device for PXE.

Default setting: Disabled

3.1.2 Use Device for PXE

This function is used to select the Ethernet devices for booting via PXE. This reduces the boot time if network boot is needed only on one particular device, not on all devices.

Note: This function is available only when the function Launch PXE OpROM is set to Enabled.

SETTING	DESCRIPTION
All	Use this setting to provide PXE support for the Ethernet connectors GbE A and GbE B.
GbE A	Use this setting to provide PXE support only for the Ethernet connector GbE A.
GbE B	Use this setting to provide PXE support only for the Ethernet connector GbE B.

Default setting: All

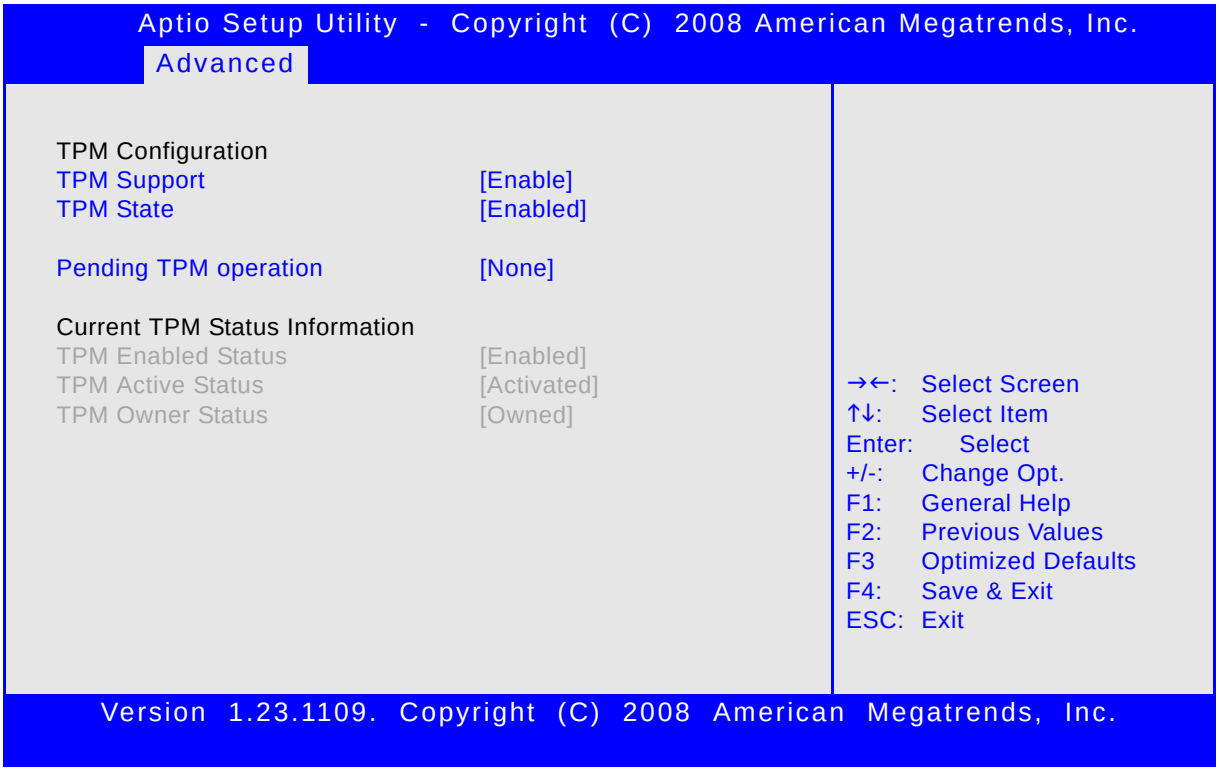
3.1.3 Launch Storage OpROM

SETTING	DESCRIPTION
Disabled	Use this setting to disable this function.
Enabled	Use this setting to specify that legacy PCI option ROMs for PCI storage devices are to be loaded and executed, if found. Typical examples of PCI storage devices include SCSI or similar devices.

Default setting: Enabled

3.2 Trusted Computing

This screen provides functions for specifying the TPM configuration settings and TPM displaying status information.



3.2.1 TPM Configuration

3.2.1.1 TPM Support

This function is used to provide the Trusted Platform Module (TPM) functionality to the OS.

SETTING	DESCRIPTION
Disable	Use this setting to disable the TPM support. If this setting is used, the TPM is not present for the OS, regardless whether the function TPM State is enabled or not.
Enable	Use this setting to enable the TPM support.

Default setting: Disable

3.2.1.2 TPM State

This function is used to select the TPM State command to be issued to the TPM after POST.

Note: This function is available only when the function TPM Support is set to Enable.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the TPM after POST. If this setting is used, the TPM is present for the OS but its functionality is locked.
Enabled	Use this setting to enable the TPM after POST.

Default setting: Disabled

3.2.2 Pending TPM Operation

This function is used to select a TPM command to be issued once against the TPM during the next boot.

Note: This function is available only when the function TPM Support is set to Enable.

SETTING	DESCRIPTION
None	Use this setting to prevent the system from issuing any TPM commands.
Enable Take Ownership	Use this setting to allow the system to issue an Enable Take Ownership command during the next boot. If this setting is used, the Take Ownership command is enabled, which allows the OS to take ownership of the TPM.
Disable Take Ownership	Use this setting to allow the system to issue a Disable Take Ownership command during the next boot. If this setting is used, the Take Ownership command is disabled, which prevents the OS from taking ownership of the TPM.
TPM Clear	Use this setting to allow the system to issue a TPM Clear command during the next boot. If this setting is used, the TPM is reset to the factory default. Warning: Use of this setting also deletes any keys and passwords stored within the TPM. Always ensure that encryption software such as Microsoft BitLocker, etc. are deactivated prior to selecting this setting.

Default setting: None

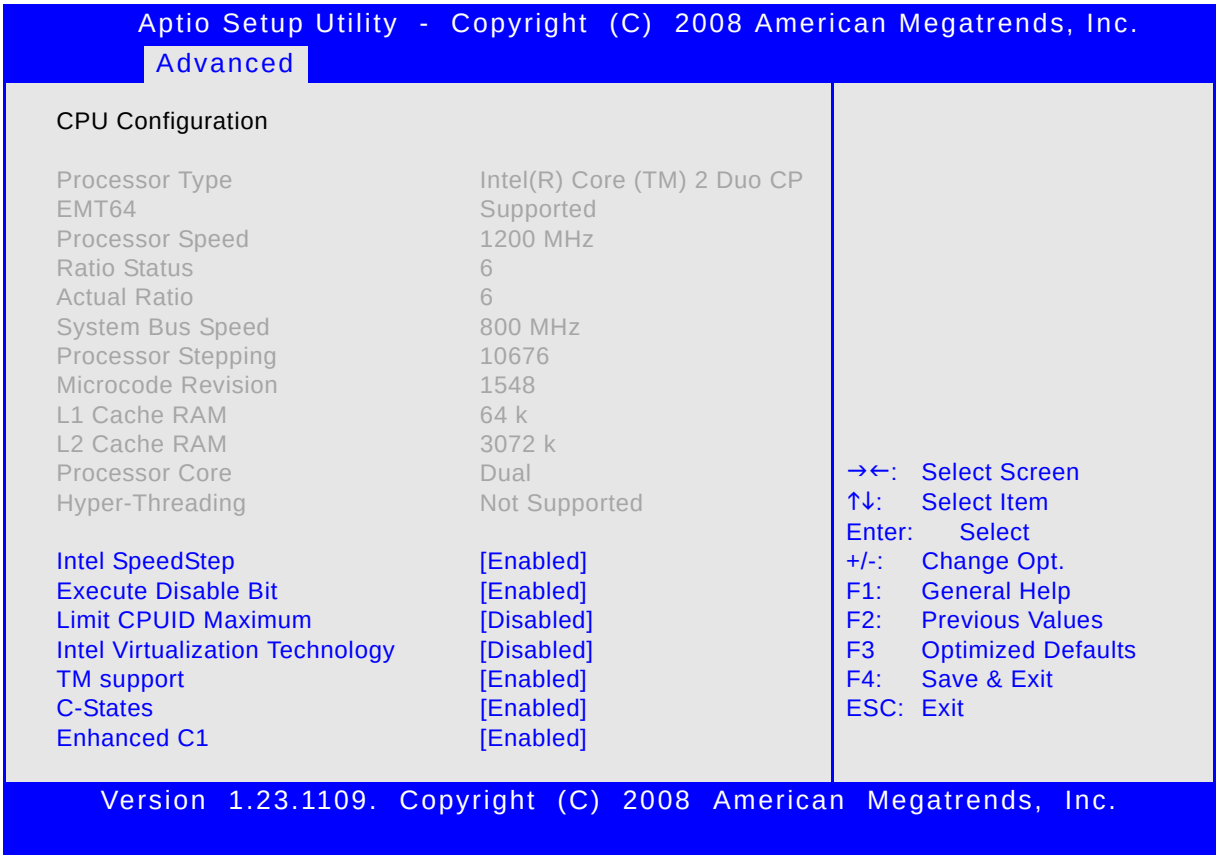
3.2.3 Current TPM Status Information

This is a display-only function providing status information about the TPM.

FUNCTION	DESCRIPTION
TPM Enabled Status	Displays if the TPM device is enabled.
TPM Active	Displays if the TPM has been activated by the OS.
TPM Owner Status	Displays if the OS has taken ownership of the TPM device.

3.3 CPU Configuration

This screen provides basic information about the CPU and functions for specifying CPU configuration settings.



3.3.1 CPU Configuration

This is a display-only function indicating general information about the installed CPU.

FUNCTION	DESCRIPTION
Processor Type	CPU brand string provided by the CPU
EMT64	Displays if the CPU supports the EMT64 extension to the x86 feature set. This is also known as x86_64 or AMD64 extension.
Processor Speed	Current CPU frequency in MHz
Ratio Status	Maximum FSB divisor Processor Speed == ((System Bus Speed) /4) * Ratio
Actual Ratio	Current FSB divisor Processor Speed == ((System Bus Speed) /4) * Ratio
System Bus Speed	Current speed of front side bus in MHz
Processor Stepping	CPU ID and stepping in hexadecimal
Microcode Revision	Microcode revision in hexadecimal
L1 Cache RAM	Amount of L1 cache present in KiB
L2 Cache RAM	Amount of L2 cache present in KiB
Processor Core	Number of CPU cores present
Hyper-Threading	Displays if the current CPU supports Hyper-Threading

3.3.2 Intel® SpeedStep™

SETTING	DESCRIPTION
Disabled	Use this setting to prevent the system from controlling the CPU frequency via the Intel® SpeedStep™ function. The CPU will run at maximum speed.
Enabled	Use this setting to allow the operating system to control the CPU speed via the Intel® SpeedStep™ function. The BIOS will start with high CPU speed.

Default setting: Enabled

3.3.3 Execute Disable Bit

The Execute Disable Bit feature is also known as NX feature or XD feature.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the Execute Disable Bit function.
Enabled	Use this setting to prevent malicious buffer overflow attacks when combined with a supporting OS.

Default setting: Enabled

3.3.4 Limit CPUID Maximum

When the computer is booted up, the operating system queries the processor to find out the highest input value CPUID recognized. Older processors until the Pentium IV returned 3 here, newer CPUs support more information. Some older operating systems such as Windows 98 fail for CPUID levels greater than 3. This function is used to limit the return value for the maximum CPUID input value to 03h when queried.

SETTING	DESCRIPTION
Disabled	Use this setting to obtain the processor's current CPUID input value when queried.
Enabled	Use this setting to limit the processor's maximum CPUID input value to 03h when queried, even if the processor supports a higher CPUID input value.

Default setting: Disabled

3.3.5 Intel Virtualization Technology

This function is used to enable a Virtual Machine Manager (VMM) to utilize the additional hardware capabilities provided by the Vanderpool Technology. To change the state of this function, a hardware reset is necessary.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the Intel Virtualization Technology.
Enabled	Use this setting to enable the Intel Virtualization Technology. A VMM can utilize the additional hardware capabilities provided by the Vanderpool Technology.

Default setting: Disabled

3.3.6 TM Support

This function is used to specify the Thermal Monitor feature. When Enabled is selected, the CPU speed will be automatically reduced if the maximum die temperature of 105°C is reached. It is highly recommended to leave this function enabled.

When Disabled is selected, the CPU temperature may exceed 105°C in case of insufficient cooling. This will reduce the CPU's life time, and may cause system instabilities.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the CPU Thermal Monitor function.
Enabled	Use this setting to enable the CPU Thermal Monitor function.

Default Setting: Enabled

3.3.7 C-States

Modern CPUs can be put into a deeper sleeping state when idle at the cost of longer wake-up times. This function is used to prevent the OS from putting the CPU in a deeper sleep state than C1.

SETTING	DESCRIPTION
Disabled	Use this setting to prevent the OS from using any C-State except C1.
Enabled	Use this setting to allow all supported C-States.

Default setting: Enabled

3.3.8 Enhanced C1

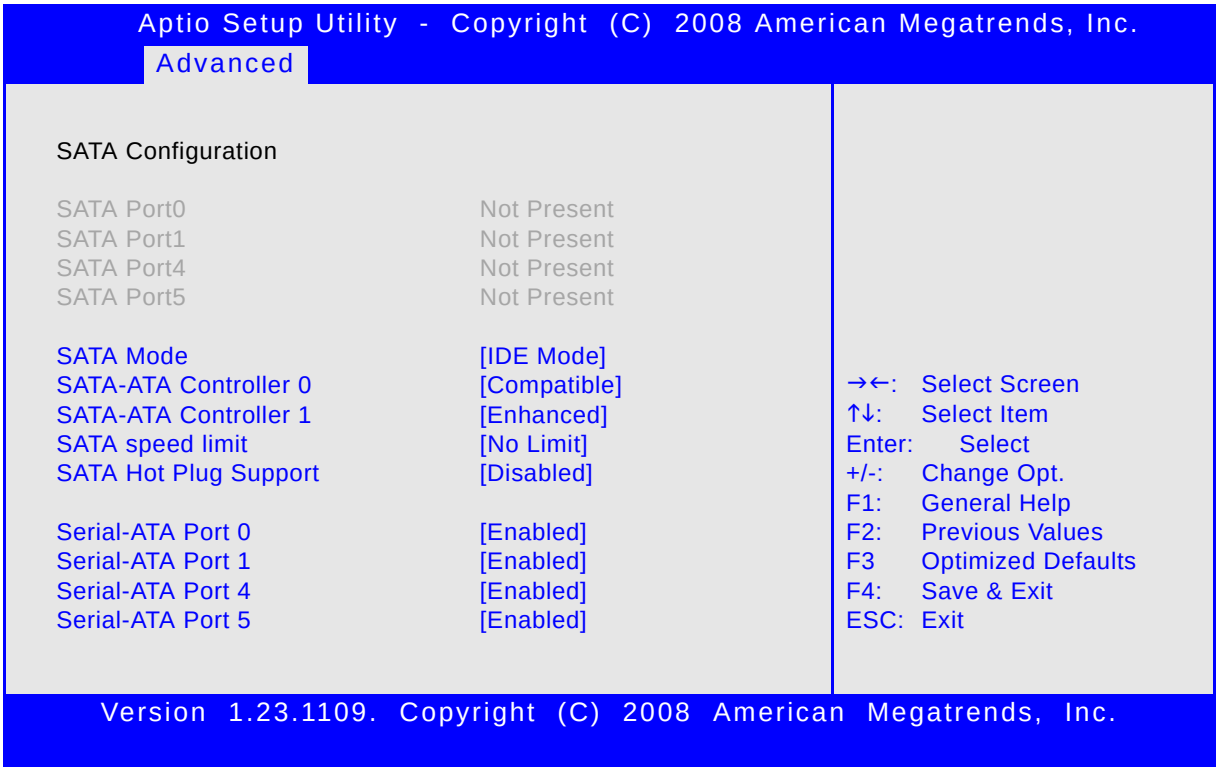
When the CPU is idle, the OS can put the CPU into a halt state. This function is used to exchange the normal halt state by the extended one that also lowers the CPU voltage .

SETTING	DESCRIPTION
Enabled	Use this setting to enable the enhanced C1 state.
Disabled	Use this setting to disable the enhanced C1 state.

Default setting: Enabled

3.4 SATA Configuration

This screen provides information about SATA ports installed as well as functions for specifying the SATA configuration settings.



3.4.1 SATA Configuration

This is a display-only function indicating if a device is connected to the corresponding port (SATA Ports 0..5). If no device is connected, the functions indicate Not Present. Otherwise, they show the device's built-in reference name followed by the device's size if the device is a hard drive. The functions are available only if the SATA controller is not in RAID mode.

3.4.2 SATA Mode

This function is used to select the modes in which the four SATA channels can be operated in.

SETTING	DESCRIPTION
Disabled	Use this setting to disable all SATA channels.
IDE Mode	When this setting is used, the following applies: The SATA channels 0 and 1 are logically attached to one legacy ATA compatible PCI device where channel 0 forms the primary master and channel 1 the primary slave. The remaining channels 4 and 5 are attached to another legacy ATA-compatible PCI device. For further information about the Serial ATA Ports 0..5, refer to Chapter 3.4.6 in this manual.
AHCI Mode	Use this setting to attach all four SATA channels to one AHCI-compatible PCI mass storage controller.
RAID Mode	Use this setting to specify that all four channels are attached to one RAID PCI controller.

Default setting: IDE Mode

3.4.3 SATA-ATA Controller 0

This function controls the resource of the SATA-ATA controller 0.

Note: This function is available only when the function SATA Mode is set to IDE Mode.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the SATA-ATA controller 0.
Enhanced	Use this setting to specify that the IDE controller is to behave like any other PCI device. When this setting is used, the IDE controller gets its resources assigned by the normal PCI initialization process.
Compatible	Use this setting to specify that the IDE controller is to behave like a standard ISA IDE controller requiring dedicated IO and IRQ resources.

Default setting: Compatible

3.4.4 SATA-ATA Controller 1

This function controls the resource of the SATA-ATA controller 1.

Note: This function is available only when the function SATA Mode is set to IDE Mode.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the SATA-ATA controller 1.
Enhanced	Use this setting to specify that the IDE controller is to behave like any other PCI device. When this setting is used, the IDE controller gets its resources assigned by the normal PCI initialization process.

Default setting: Enhanced

3.4.5 SATA Speed Limit

This function specifies the SATA speed limit. The ICH9M-SFF features four SATA channels capable of SATA GEN2 operating speed. Due to the bad signal quality when using long cables or rear I/O, it may be required to limit the speed to SATA GEN1.

SETTING	DESCRIPTION
No Limit	Use this setting to specify that all four SATA channels are allowed to operate up to SATA GEN2 speed if supported by the connected device.
GEN1 Rate	Use this setting to specify that all four channels are limited to SATA GEN1 speed regardless of the connected device.

Default setting: No Limit

3.4.6 SATA Hot Plug Support

The ICH9M-SFF features SATA hot plugging if the function SATA Mode is set to AHCI or RAID Mode.

SETTING	DESCRIPTION
Disabled	Use this setting to disable SATA hot plug support.
Enabled	Use this setting to enable SATA hot plug support.

Default setting: Enabled

3.4.7 Serial-ATA Ports 0..5

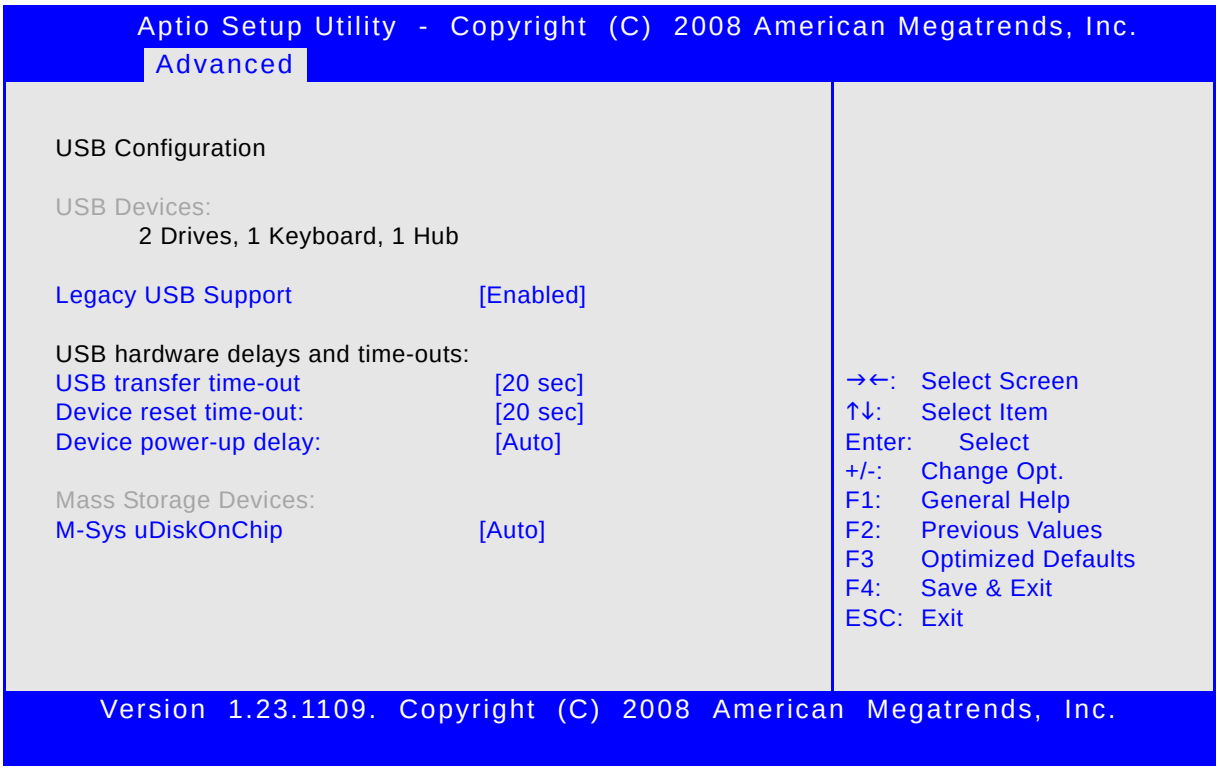
Use one of these functions to enable or disable the SATA ports on a port-by-port basis.

FUNCTION	ROUTING	SETTING
Serial-ATA Port 0	Onboard SATA connector	Disabled / Enabled
Serial-ATA Port 1	Onboard SATA connector	Disabled / Enabled
Serial-ATA Port 4	Rear I/O	Disabled / Enabled
Serial-ATA Port 5	Rear I/O	Disabled / Enabled

Default setting: Enabled

3.5 USB Configuration

This screen provides information about support for USB devices as well as functions for specifying the USB configuration settings.



3.5.1 USB Configuration

This is a display-only function providing general information about the USB devices detected.

SETTING	DESCRIPTION
USB Devices	Shows a summary of detected USB devices.

3.5.2 Legacy USB Support

This function is required for booting from USB devices and for operating systems which do not support USB themselves (mainly DOS and some BootLoaders).

SETTING	DESCRIPTION
Disabled	Use this setting to disable legacy USB support.
Enabled	Use this setting to enable legacy USB support.
Auto	Use this setting to enable legacy USB support if there are USB devices present.

Default setting: Enabled

3.5.3 USB Hardware Delays and Time-outs

3.5.3.1 USB Transfer Timeout

This function selects the timeout in seconds that the USB core will wait for Control, Bulk, and Interrupt transfers.

SETTING	DESCRIPTION
1 sec 5 sec 10 sec 20 sec	Use one of these settings to specify how long the USB core is to wait for Control, Bulk, and Interrupt transfers.

Default setting: 20 sec

3.5.3.2 Device Reset Timeout

This function selects the timeout in seconds that the POST will wait for a USB mass storage device to become ready after start unit command.

SETTING	DESCRIPTION
10 sec 20 sec 30 sec 40 sec	Use one of these settings to specify how long the POST will wait for a USB mass storage device to become ready after the start unit command.

Default setting: 20 sec

3.5.3.3 Device Power-up Delay

This function determines the maximum time the device will take before it properly reports itself to the Host Controller. “Auto” uses a default value: for a Root port it is 100 ms, for a Hub port the delay is taken from Hub descriptor.

If the “Manual” option is chosen, the device power up delay in seconds field will be enabled to accept a delay ranging from 1 to 40 seconds.

SETTING	DESCRIPTION
Auto	Use this setting to specify a default delay time for a Root or Hub port. (root port = 100ms; hub port = value in hub descriptor)
Manual	Use this setting to specify a delay time from 1 to 40 seconds. (contents of seconds field)

Default setting: Auto

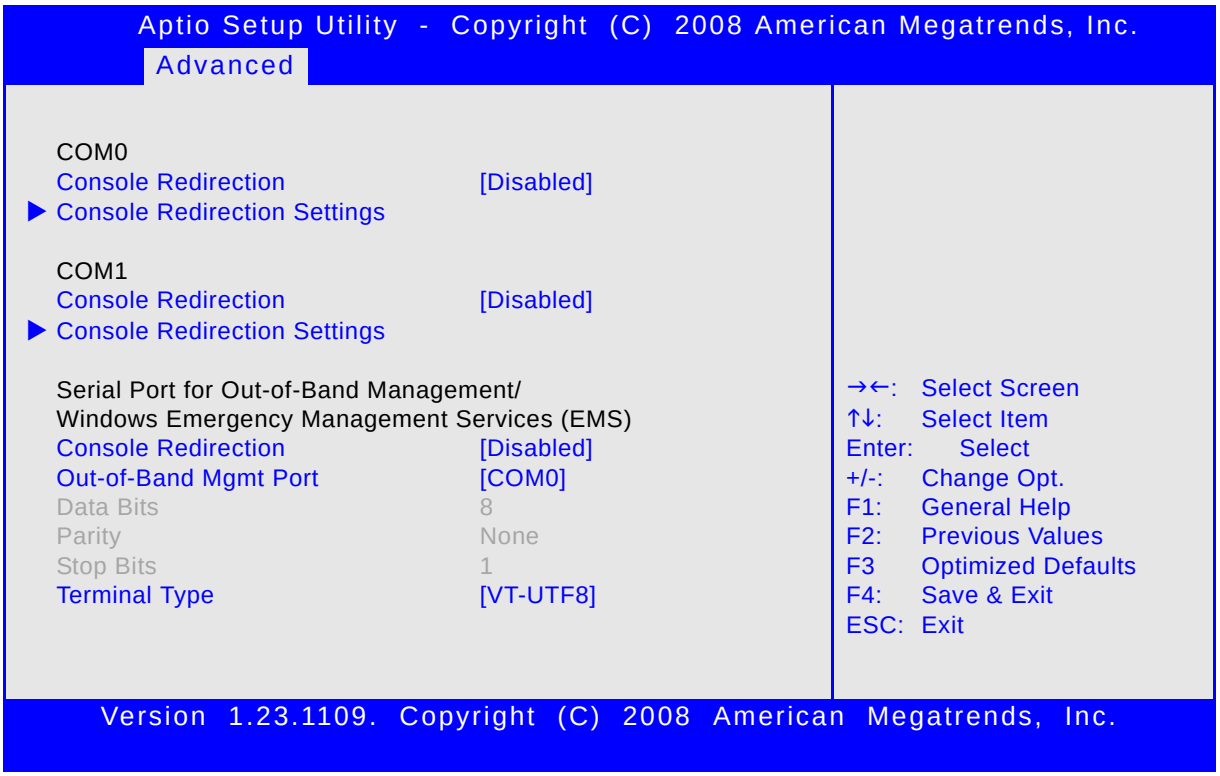
3.5.4 Mass Storage Devices

This function shows a list of connected USB mass storage devices and allows the user to select how the respective device is to be treated.

SETTING	DESCRIPTION
Auto	Use this setting to specify that devices with a capacity smaller than 128 MB will be treated as USB floppy, larger ones as USB hard drive.
Floppy	Use this setting to specify that the device is treated as floppy, regardless of its size.
Forced FDD	Use this setting to force the device to be booted as floppy, even if it is formatted as hard drive.
Hard Disk	Use this setting to treat the device as hard disk, regardless of its size.
CD-ROM	Use this setting to treat the device as CD-ROM even if its class code does not identify the device as such.

3.6 Serial Port Console Redirection

This screen provides information about functions for specifying the Serial Port Console Redirection configuration settings. Console redirection can be used to remotely operate system settings and the EFI console.



3.6.1 COM0

3.6.1.1 Console Redirection

SETTING	DESCRIPTION
Disabled	Use this setting to disable console redirection for the serial port 0.
Enabled	Use this setting to enable console redirection for the serial port 0.

Default setting: Disabled

3.6.1.2 Console Redirection Settings

For information about this function, refer to Chapter 3.6.4 in this manual.

3.6.2 COM1

3.6.2.1 Console Redirection

SETTING	DESCRIPTION
Disabled	Use this setting to disable console redirection for the serial port 1.
Enabled	Use this setting to enable console redirection for the serial port 1.

Default setting: Disabled

3.6.2.2 Console Redirection Settings

For information about this function, refer to Chapter 3.6.4 in this manual.

3.6.3 Serial Port for Out-of-Band Management/Windows Emergency Management Services (EMS)

The following functions control the presence and content of the ACPI serial port redirection table (SPCR). This table is mainly used by the Windows server variants to provide Windows Emergency Management Services (EMS). This functionality is totally independent from serial redirection of other console output.

3.6.3.1 Console Redirection

SETTING	DESCRIPTION
Disabled	Use this setting to prevent the system from adding the SPCR table to the ACPI tables.
Enabled	Use this setting to add the SPCR table to the ACPI tables. The OS can further use the information provided for serial redirection services.

Default setting: Disabled

3.6.3.2 Out-of-Band Mgmt Port

This function is used to select the serial port intended for use with Out-of-Band Management. This functionality is independent from serial redirection of other console output.

SETTING	DESCRIPTION
COM0	Use this setting to specify that the serial port 0 is to be used with Out-of-Band Management.
COM1	Use this setting to specify that the serial port 1 is to be used with Out-of-Band Management.

Default setting: COM0

3.6.3.3 Data Bits

This is a display-only function providing information about the frame width for the Out-of-Band Management.

3.6.3.4 Parity

This is a display-only function providing information about the parity for Out-of-Band Management.

3.6.3.5 Stop Bits

This is a display-only function providing information about the number of stop bits for Out-of-Band Management.

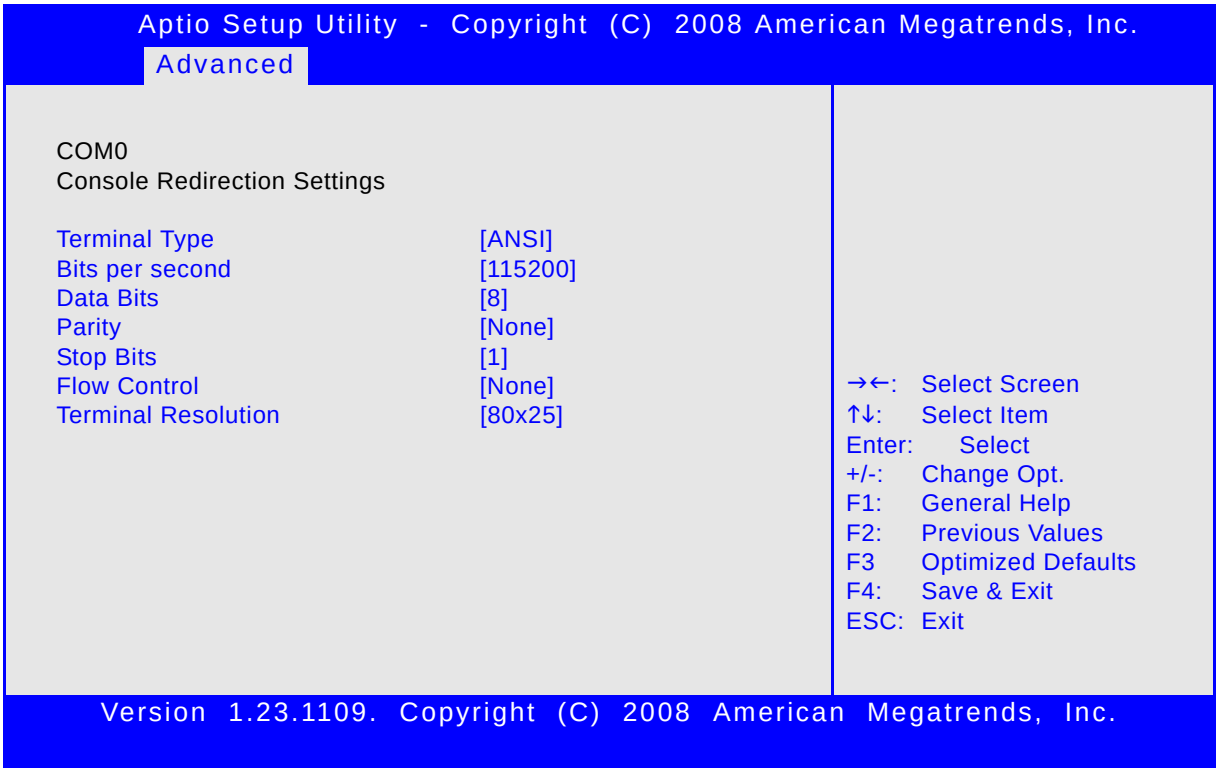
3.6.3.6 Terminal Type

SETTING	DESCRIPTION
VT100	Use one of these settings to select the terminal type for out-of-band management.
VT100+	
VT-UTF8	
ANSI	

Default setting: VT-UTF8

3.6.4 Console Redirection Settings

This screen provides information about functions for specifying the Console Redirection configuration settings for the serial port 0 and the serial port 1. Each serial port can be independently configured.



3.6.4.1 Terminal Type

SETTING	DESCRIPTION
VT100	Use one of these settings to select the terminal type to be emulated.
VT100+	
VT-UTF8	
ANSI	

Default setting: ANSI

3.6.4.2 Bits per second

SETTING	DESCRIPTION
9600	Use one of these settings to select the baud rate of the serial port.
19200	
57600	
115200	

Default setting: 115200

3.6.4.3 Data Bits

SETTING	DESCRIPTION
7	Use one of these settings to specify the number of data bits per frame.
8	

Default setting: 8

3.6.4.4 Parity

SETTING	DESCRIPTION
None	Use one of these settings to select the parity for the serial port.
Even	
Odd	
Mark	
Space	

Default setting: None

3.6.4.5 Stop Bits

SETTING	DESCRIPTION
1	Use one of these settings to specify the number of stop bits for the serial port.
2	

Default setting: 1

3.6.4.6 Flow Control

SETTING	DESCRIPTION
None	Use one of these settings to specify the type of flow control to be used for this serial port.
Hardware RTS/CTS	
Software Xon/Xoff	

Default setting: None

3.6.4.7 Terminal Resolution

SETTING	DESCRIPTION
80x25	Use one of these settings to select the resolution of the emulated terminal for serial console redirection.
100x31	

Default setting: 80x25

This page has been intentionally left blank.

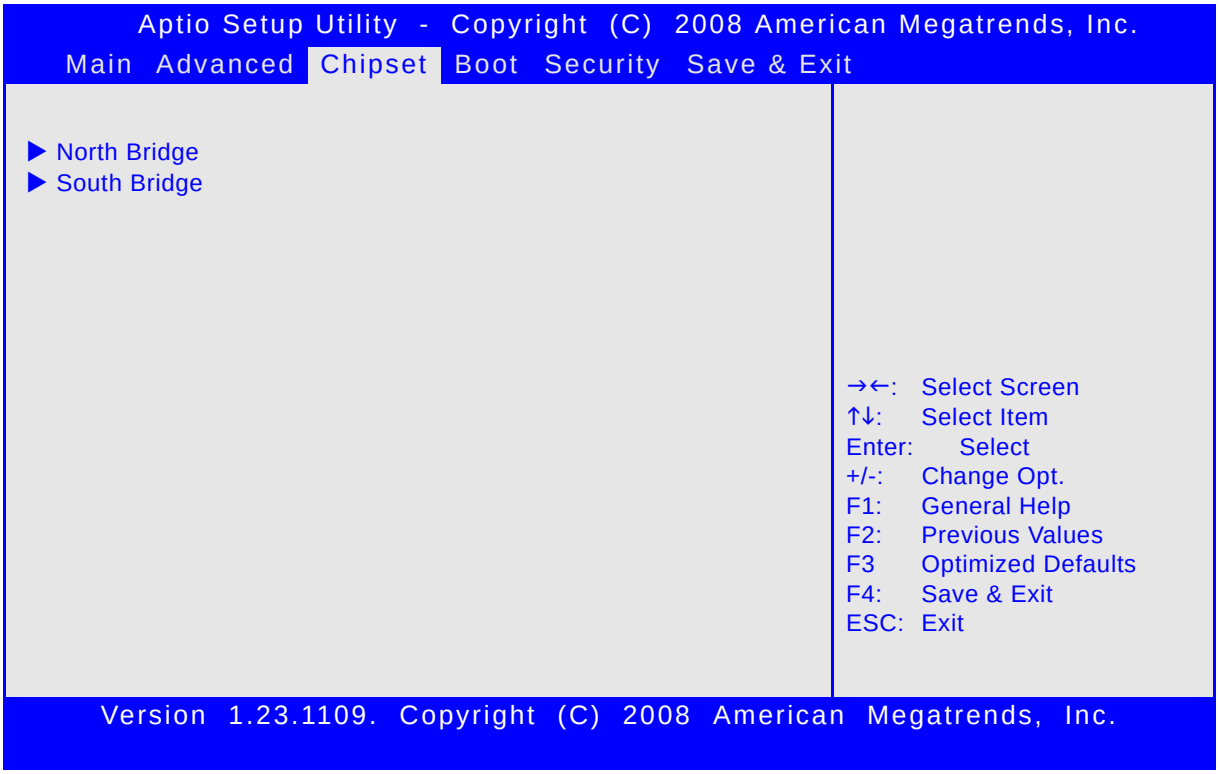
Chapter 4

Chipset Setup

This page has been intentionally left blank.

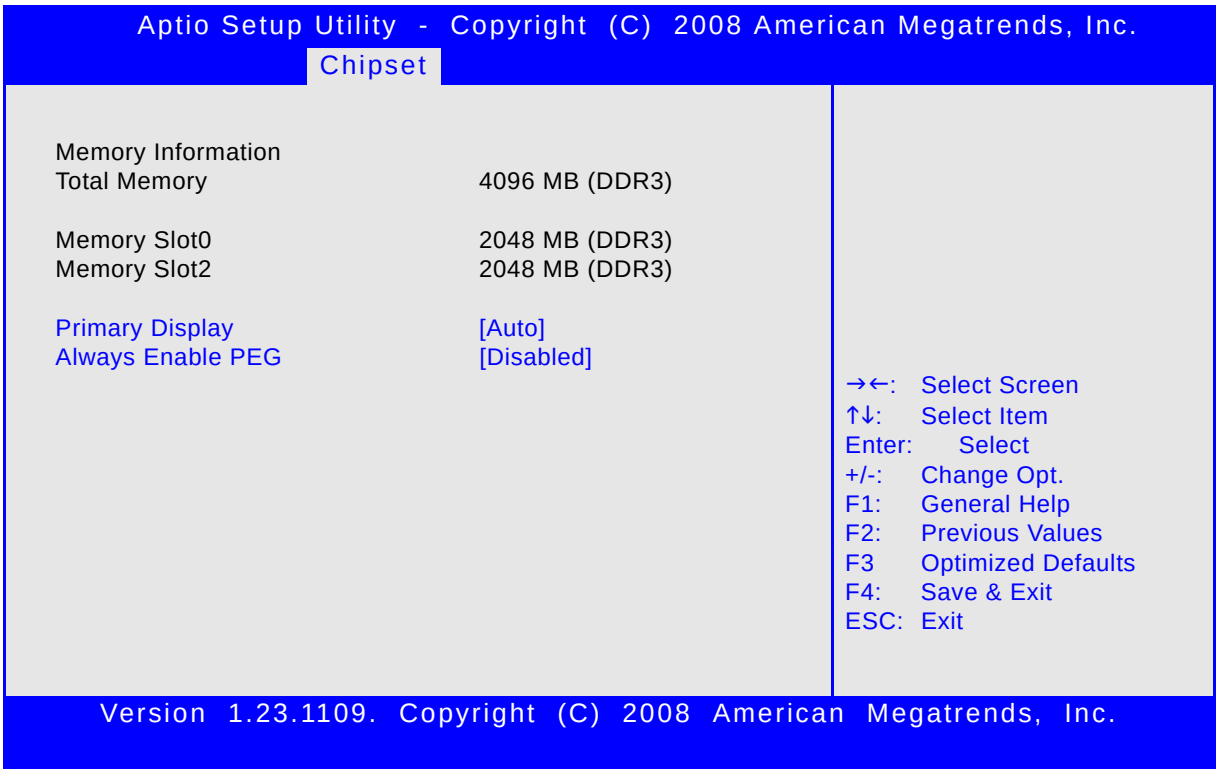
4. Chipset Setup

Select the Chipset tab to enter the Chipset Setup screen. This screen lists the chipset configuration sub-screens.



4.1 NorthBridge Configuration

This screen provides functions for specifying the NorthBridge configuration settings.



4.1.1 Primary Display

SETTING	DESCRIPTION
Auto	Use this setting to allow the system to automatically select the graphics device to be used for POST output.
IGD	Use this setting to specify that the onboard graphics device is used for POST output.
PEG	Use this setting to specify that the graphics device on PCIe x16 high-speed slot.
PCI	Use this setting to specify that the PCI graphics device is used for POST output.

Default setting: Auto

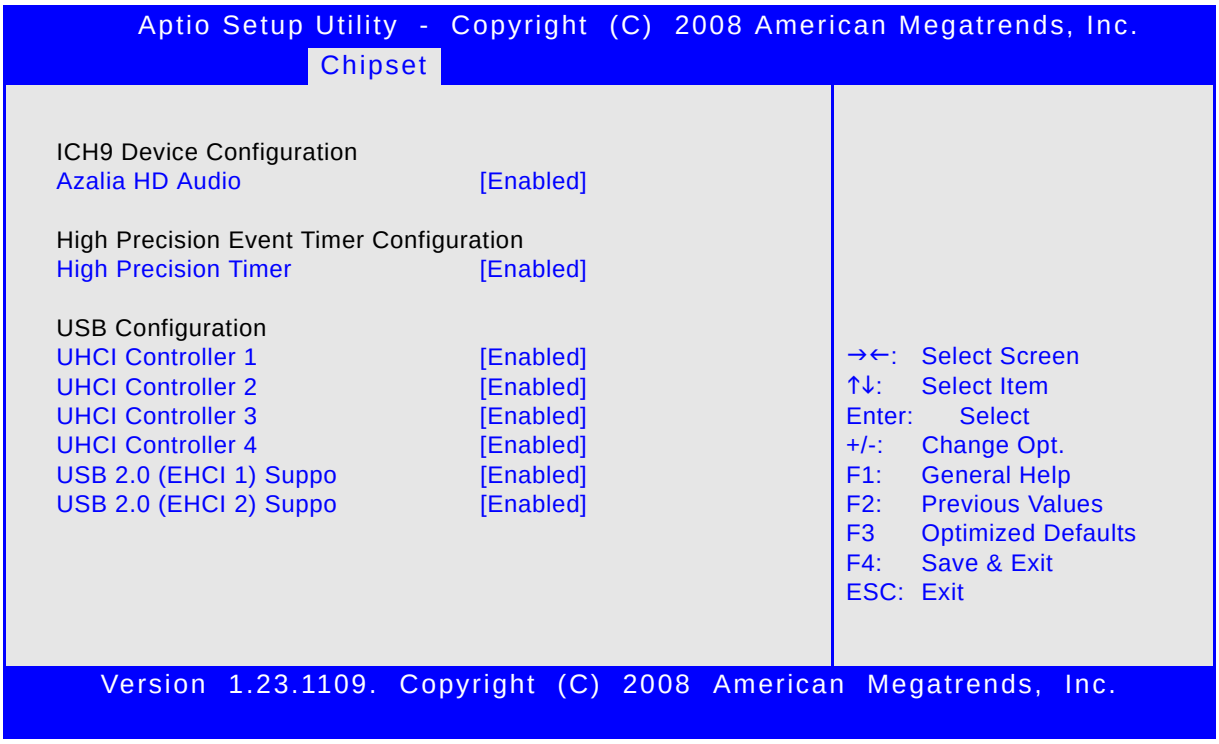
4.1.2 Always Enable PEG

SETTING	DESCRIPTION
Enabled	Use this setting to enable the PCI Express graphics port.
Disabled	Use this setting to disable the PCI Express graphics port.

Default setting: Disabled

4.2 SouthBridge Configuration

This screen provides functions for specifying the SouthBridge configuration settings.



4.2.1 ICH9 Device Configuration

4.2.1.1 Azalia HD Audio

This function is used to configure the presence of the chipset's internal HD audio controller. Currently, there is no HD audio codec present on the CP308. Therefore, this option is reserved for future use.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the Azalia HD Audio function.
Enabled	Use this setting to enable the Azalia HD Audio function.

Default setting: Enabled

4.2.2 High Precision Event Timer Configuration

4.2.2.1 High Precision Event Timer

This function is used to control the presence of the high-precision event timer ACPI description table (HPET). This table is required for a modern OS (Linux) to make use of the high-precision event timer.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the high-precision event timer.
Enabled	Use this setting to enable the high-precision event timer.

Default setting: Enabled

4.2.3 USB Configuration

4.2.3.1 UHCI Controller 1..3

Use these functions to configure the presence of the UHCI controllers 1..3 and their connected physical ports.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the UHCI controllers 1..3.
Enabled	Use this setting to enable the UHCI controllers 1..3.

Default setting: Enabled

Mapping of the UHCI Controllers 1..3 to the USB port:

UHCI CONTROLLER	USB PORT NUMBERS	CONNECTORS
1	0, 1	Front USB, Rear USB
2	2, 3	Front USB, Rear USB
3	4, 5	Rear I/O, Rear I/O

4.2.3.2 UHCI Controller 4

Use these functions to configure the presence of the UHCI controller 4 and its connected physical ports.

Note: This function is available only if USB 2.0 (EHCI 2) Support is set to Enabled.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the UHCI controller 4.
Enabled	Use this setting to enable the UHCI controller 4.

Default setting: Enabled

Mapping of the UHCI Controller 4 to the USB port:

UHCI CONTROLLER	PORT NUMBERS	CONNECTORS
4	6,7	Onboard NAND Flash, NAND Flash Extension

4.2.3.3 USB 2.0 (EHCI 1) Support

SETTING	DESCRIPTION
Disabled	Use this setting to disable the USB 2.0 EHCI 1 controller.
Enabled	Use this setting to enable the USB 2.0 EHCI 1 controller.

Default setting: Enabled

4.2.3.4 USB 2.0 (EHCI 2) Support

Note: Disabling the USB 2.0 (EHCI 2) controller also disables the UHCI controller 4.

SETTING	DESCRIPTION
Disabled	Use this setting to disable the USB 2.0 EHCI 2 controller.
Enabled	Use this setting to enablethe USB 2.0 EHCI 2 controller.

Default setting: Enabled

Note: If the USB2.0 (EHCI 2) support is set to Disabled, the configuration is always 4x0. This means that four physical ports on the EHCI controller 1 and none on the EHCI controller 2 are available since EHCI 2 is disabled.

This page has been intentionally left blank.

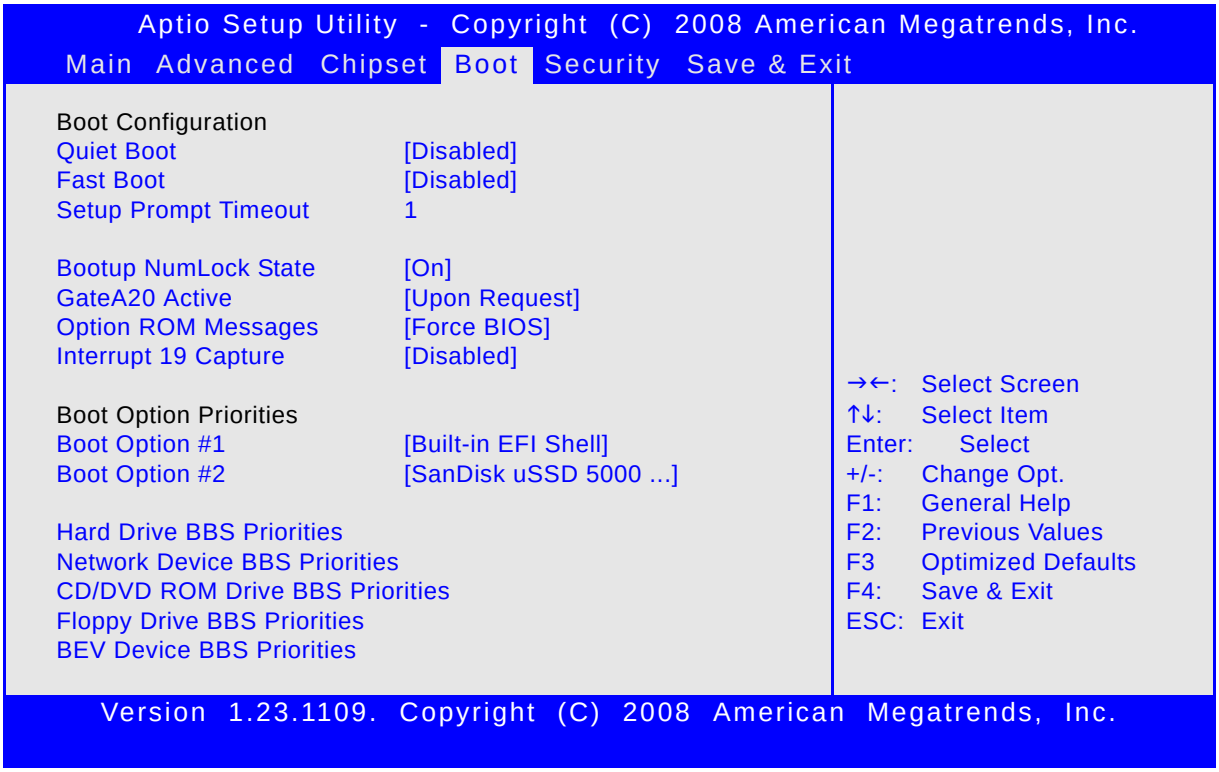
Chapter 5

Boot Setup

This page has been intentionally left blank.

5. Boot Setup

Select the Boot tab to enter the Boot Setup screen. This screen lists the sub-screens for boot configuration and boot device priority.



5.1 Boot Configuration

5.1.1 Quiet Boot

This function is used to display either POST output messages or a splash screen during boot-up.

SETTING	DESCRIPTION
Disabled	Use this setting to display POST output messages during boot-up.
Enabled	Use this setting to display a splash screen during boot-up.

Default setting: Disabled

5.1.2 Fast Boot

This function is used to enable or disable boot with initialization of a minimal set of devices required to launch active boot option..

SETTING	DESCRIPTION
Disabled	Use this setting to disable fast boot.
Enabled	Use this setting to enable fast boot.

Default setting: Disabled

5.1.3 Setup Prompt Timeout

This integer function is used to set an additional time the POST should wait for the operator to press the key to enter setup. The time is entered in seconds.

SETTING	DESCRIPTION
1 ⋮ 65535	Use one of these settings to specify the setup prompt timeout.

Default setting: 1

5.1.4 Bootup NumLock State

This function is used to set the state of the keyboard's numlock function after POST.

SETTING	DESCRIPTION
On	Use this setting to switch on the keyboard's numlock function after POST.
Off	Use this setting to switch off the keyboard's numlock function after POST.

Default setting: On

5.1.5 GateA20 Active

This function is used to enable or disable GateA20.

SETTING	DESCRIPTION
Upon Request	Use this setting to disable GA20 in the uEFI BIOS.
Always	Use this setting to prevent the system from disabling GA20.

Default setting: Upon Request

5.1.6 Option ROM Messages

This function is used to control the messages of the loaded PCI option ROMs.

SETTING	DESCRIPTION
Force BIOS	Use this setting to force to a BIOS-compatible output. This will show the option ROM messages.
Keep Current	Use this setting to keep the current video mode. This will suppress option ROM messages. Option ROMs requiring interactive inputs may not work properly in this mode.

Default setting: Force BIOS

5.1.7 Interrupt 19 Capture

This function is used to specify if legacy PCI option ROMs are allowed to capture software interrupt 19h.

SETTING	DESCRIPTION
Disabled	Use this setting to prevent legacy PCI option ROMs from capturing software interrupt 19h.
Enabled	Use this setting to allow legacy PCI option ROMs to capture software interrupt 19h.

Default setting: Disabled

5.2 Boot Option Priorities

5.2.1 Boot Option #1..2

These functions are used to form the boot order and are dynamically generated. They represent either a legacy BBS (BIOS Boot Specification) class of devices or a native EFI boot entry. Press Return on each option to select the BBS class / EFI boot entry desired.

5.2.2 Hard Drive/Network Device/CD/DVD ROM Drive/Floppy Drive/BEV Device BBS Priorities

These functions lead to sub-menus that allow configuring the boot order for a specific device class. These options are only visible if at least one device for this class is present. These functions are dynamically generated.

This page has been intentionally left blank.

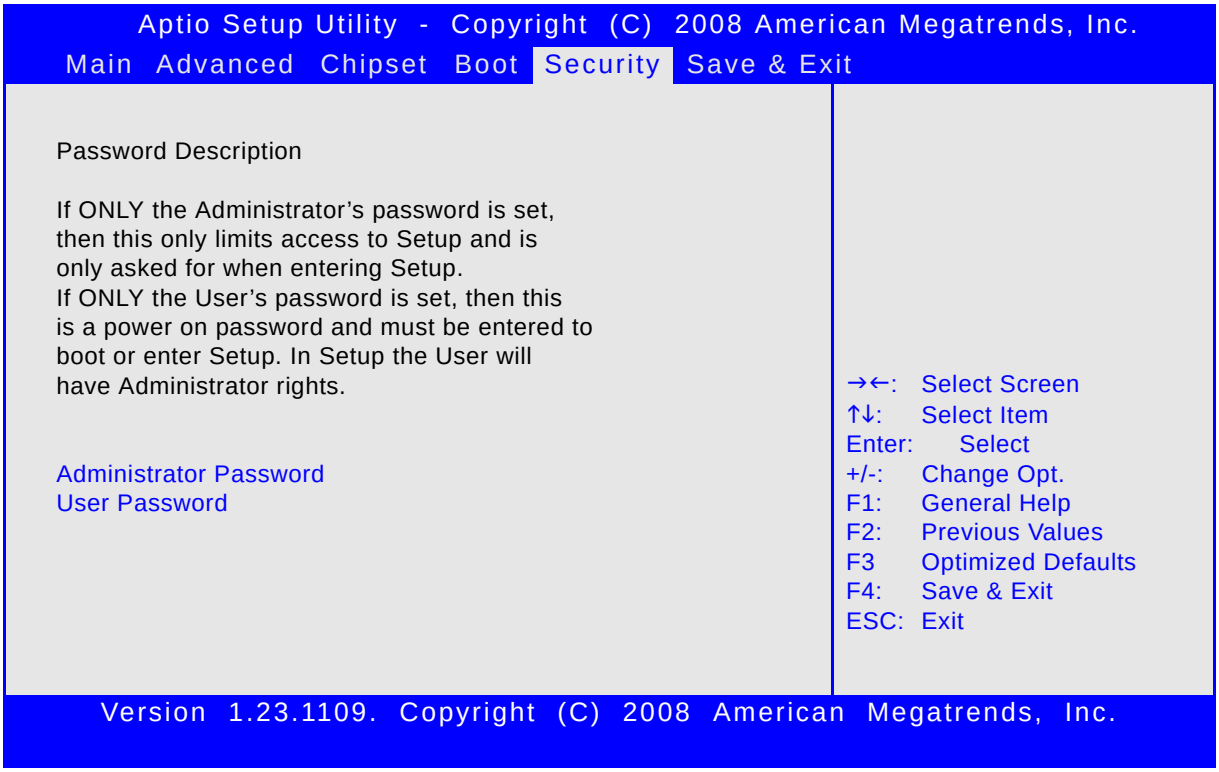
Chapter 6

Security Setup

This page has been intentionally left blank.

6. Security Setup

Select the Security tab to enter the Security Setup screen. This screen provides information about the passwords and functions for specifying the security settings.



The following modes of security are provided:.

SETTING	DESCRIPTION
No password is set	Booting the system as well as entering the Setup is unsecured.
Only Administrator password is set	Booting the system is unsecured. For entering the Setup, the Administrator password is required.
Only User password is set	The password is required for booting the system as well as for entering the Setup menu. On every startup, the user will be asked for the password.
Both User and Administrator passwords are set	Booting the system is unsecured. For entering the Setup, a password is required. If the User password is entered here, most of the Setup entries are read only; only entries related to the boot sequence can be modified. Entering the Administrator password provides full access to all Setup entries.

6.1 Administrator Password

This function is used to set, change or delete the Administrator password. If there is already a password installed, the system asks for this first. To clear a password, simply enter nothing and acknowledge by pressing Return. To set a password, enter it twice and acknowledge by pressing Return.

Note: The password is case sensitive.

6.2 User Password

This function is used to set, change or delete the User password. If there is already a password installed, the system asks for this first. To clear a password, simply enter nothing and acknowledge by pressing Return. To set a password, enter it twice and acknowledge by pressing Return.

Note: The password is case sensitive.

6.3 Remember the Password

It is highly recommended to keep a record of all passwords in a safe place. Forgotten passwords may lead to being completely locked out of the system. Booting may not be possible, and in worst case the uEFI BIOS Setup program will also not be accessible.

If the system cannot be booted because neither the User password nor the Administrator password are known, refer to Chapter 4.1 and Chapter 4.2.1 in the CP308 User Guide for information about clearing the uEFI BIOS settings, or contact Kontron for further assistance.

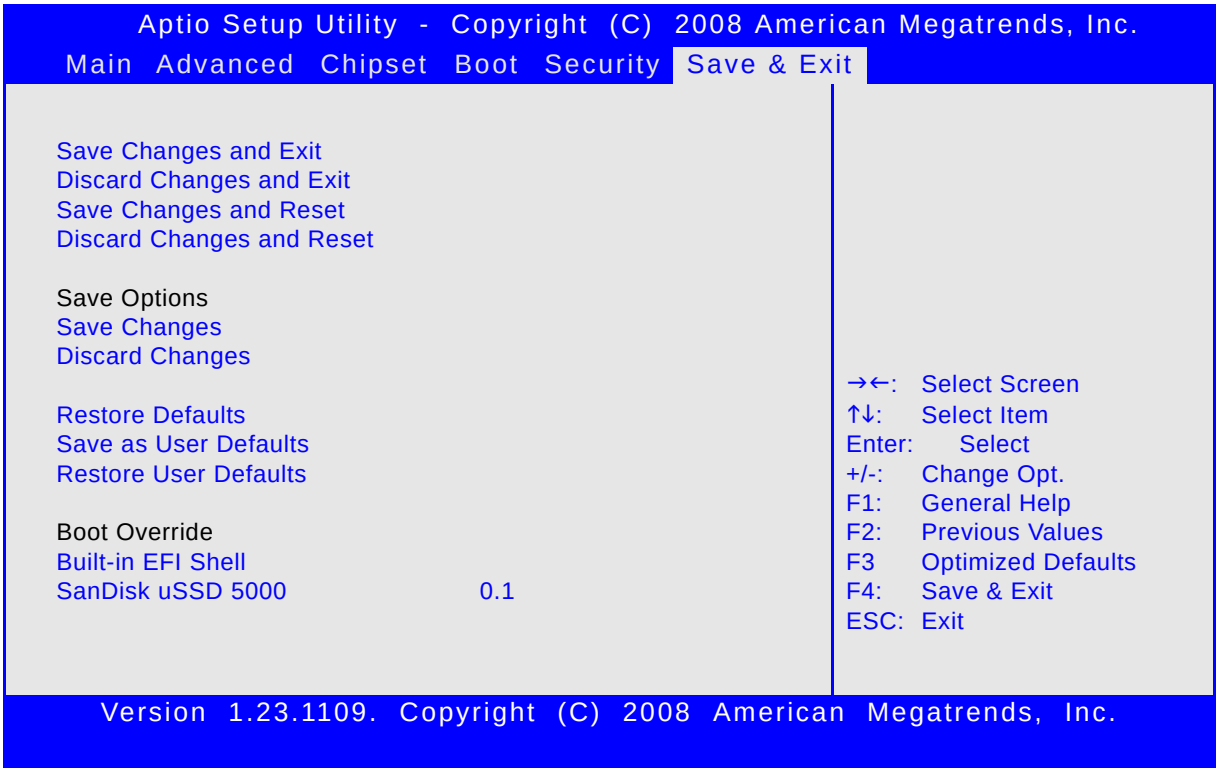
Chapter 7

Save & Exit

This page has been intentionally left blank.

7. Save & Exit

Select the Save & Exit tab to enter the Save & Exit menu screen. This screen provides functions for handling changes made to the uEFI BIOS settings and the exiting of the Setup program.



7.1 Save Changes and Exit

This function is used to save all changes made within the Setup to Flash. This function continues the boot process as long as no option was altered that requires a reboot.

Note: The Setup will ask for confirmation prior to executing this command.

7.2 Discard Changes and Exit

This function is used to discard all changes made within the Setup. This function continues the boot process.

Note: The Setup will ask for confirmation prior to executing this command.

7.3 Save Changes and Reset

This function is used to save all changes made within the Setup to Flash. This function performs a reboot afterwards.

Note: The Setup will ask for confirmation prior to executing this command.

7.4 Discard Changes and Reset

This function is used to discard all changes made within the Setup. This function performs a reboot afterwards.

Note: The Setup will ask for confirmation prior to executing this command.

7.5 Save Changes (Save Options)

This function is used to save all changes made within the Setup to Flash. This function returns to Setup.

Note: The Setup will ask for confirmation prior to executing this command.

7.6 Discard Changes (Save Options)

This function is used to discard all changes made within the Setup. This function returns to Setup.

Note: The Setup will ask for confirmation prior to executing this command.

7.7 Restore Defaults (Save Options)

This function is used to restore all tokens to factory default.

Note: The Setup will ask for confirmation prior to executing this command.

7.8 Save as User Defaults (Save Options)

This function is used to save all current settings as user default. The current setup state can later be restored using Restore User Defaults.

Note: The Setup will ask for confirmation prior to executing this command.

7.9 Restore User Defaults (Save Options)

This function is used to restore all tokens to settings previously stored by Save as User Defaults.

Note: The Setup will ask for confirmation prior to executing this command.

7.10 Boot Override

This group of functions includes a list of tokens, each of them corresponding to one device within the boot order. Select a drive to immediately boot that device regardless of the current boot order. If booting to EFI Shell this way, an exit from the shell returns to Setup.

Chapter 8

uEFI Shell

This page has been intentionally left blank.

8. uEFI Shell

The Kontron uEFI BIOS features a built-in and enhanced version of the uEFI Shell. For a detailed description of the available standard shell scripting refer to the EFI Shell User's Guide. For a detailed description of the available standard shell commands, refer to the Shell Command Manual 1.0. Both documents can be downloaded from the EFI and Framework Open Source Community homepage (https://efi-shell.tianocore.org) under the "Documents and Files" section.

Please note that not all shell commands described in the Shell Command Manual 1.0 are provided by the Kontron uEFI BIOS.

8.1 Introduction, Basic Operation

The uEFI shell forms an entry into the uEFI boot order and is the first boot option by default. It is simply started by putting the uEFI shell first in boot and running the board as usual.

8.1.1 Shell Startup

If the shell is executed, it displays its signon message followed by a list of detected devices. The output produced by the device mapping table can vary depending on the board's configuration.

```
EFI Shell version 2.00 [4.631]
Current running mode 1.1.2
Device mapping table
  fs0     :Removable HardDisk - Alias hd33b0b0b blk0
           Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(1, 0)/HD(Part1,Sig17731773)
  fs1     :Removable BlockDevice - Alias f33b0c0 blk1
           Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(2, 0)
  blk0    :Removable HardDisk - Alias hd33b0b0b fs0
           Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(1, 0)/HD(Part1,Sig17731773)
  blk1    :Removable BlockDevice - Alias f33b0c0 fs1
           Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(2, 0)
  blk2    :HardDisk - Alias (null)
           Acpi(PNP0A03,0)/Pci(1F|2)/Ata(Primary,Master)/HD(Part1,SigC811D18D)
  blk3    :BlockDevice - Alias (null)
           Acpi(PNP0A03,0)/Pci(1F|2)/Ata(Primary,Master)
  blk4    :Removable BlockDevice - Alias (null)
           Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(1, 0)

Press the ESC key within 5 seconds to skip startup.nsh, and any other key to
continue.
```

If the ESC key is pressed before the 5-second timeout has elapsed, the shell prompt is shown:

```
Shell>
```

8.2 Kontron Shell Commands

The Kontron uEFI implementation provides a number of additional commands that are related to the specific HW features of the Kontron system.

- **kboardconfig**
- **kboardinfo**
- **kboot**
- **kbootnsh**
- **kclearnvram**
- **kclsp**
- **kflash**
- **kipmi**
- **kmkramdisk**
- **kpassword**
- **kresetconfig**
- **kwdt**

The following tables provide information concerning these Kontron-specific commands. Where “RESPONSE” information is provided in “USAGE”, the value indicated in brackets is the currently selected setting. Where “SETTINGS” information is provided, the value indicated in brackets is the current setting.

8.2.1 kboardconfig uEFI Shell Command

kboardconfig

FUNCTION:	Configure the non-volatile board settings
SYNTAX:	**kboardconfig** **kboardconfig [-?|<device>|<setting>]** where: ? Show online help <device> Specify device from list <setting> Select configuration type
DESCRIPTION:	The **kboardconfig** command enables the PXE feature or sets the front/ rear I/O configuration of the dedicated device.
USAGE:	Show all possible configurations COMMAND / RESPONSE EXAMPLE: **Shell> kboardconfig** **com_0: COM 0 port configuration ([rear] extension)** **gbe_a: GbE A port configuration ([front] rear)** **gbe_b: GbE B port configuration ([front] rear)** **pxe: Select PXE boot network adapter ([disabled] all gbe_a gbe_b)** **sata: SATA port configuration ([rear] extension)** **vga: VGA port configuration (auto [front] rear disabled)** **pdisplay: Primary Display configuration (igd peg pci [auto])** **peg: Always Enable PEG Port ([disabled] enabled)** **sleep: Support for ACPI sleep function ([disable] enable)**
	Show allowed settings e.g. for “vga”: **Shell> kboardconfig vga** **vga: VGA Port configuration** **vga==front** **Allowed options: auto, front, rear, disabled**
	Set configuration settings e.g. for “gbe_a” : **Shell> kboardconfig gbe_a rear** **Setting gbe_a to rear ... Done** **Reset is required for this parameter to take effect.**
	Check configuration e.g. of ”sata”: **Shell> kboardconfig sata** **sata: SATA Port configuration** **sata==rear** **Allowed options: rear, extension**

8.2.2 kboardinfo uEFI Shell Command

kboardinfo

FUNCTION:	Show board identification data
SYNTAX:	**kboardinfo**
DESCRIPTION:	The **kboardinfo** command shows a summary of board-specific identification data. It is especially useful for support queries because it contains this data in a concentrated form.
USAGE:	Show board identification data COMMAND / RESPONSE EXAMPLE: **Shell> kboardinfo** **KOMaOEMF rev.: 1** **Board ID: 0xB1** **Hardware rev.: 0x0** **Logic rev.: 0x9** **Boot flash: Boot flash 0** **In system slot: No** **Geographic Address: 3** **Material number: 1234-5678** **Hardware index:** **Serial number: 1234567001** **EFI article name: SK-EFI-CP308** **EFI material number: 1234-5678** **EFI index: 22, standard** **EFI build time: 17:20:09** **EFI build date: 12/14/2012** **NorthBridge.: 0x90** **SouthBridge.: 0x2** **Microcode: 0x60C** **CPU ID: 0x10676** **CPU Branding: Intel(R) Core(TM)2 Duo CPU L9400 @ 1.86 GHz** **RIO Module: 001**

kboardinfo (continued)

KOMaOEMF rev.:	Revision of KOMaOEMF protocol
Board ID:	Kontron board identification value (should be 0xB1 for the CP308)
Hardware rev.:	Hardware revision of this board
Logic rev.:	Logic revision of this board
Boot flash:	Current boot flash: either “Boot flash 0” or “Boot flash 1”
In system slot:	“Yes” if the board is plugged into the system slot, otherwise “No”.
Geographic Address:	Geographic address of the CPCI slot the board is currently plugged into
Material number:	Kontron hardware reference number
Hardware index:	Kontron hardware index
Serial number:	This board’s unique serial number
EFI article name:	Kontron uEFI reference name
EFI material number:	Kontron uEFI reference number
EFI index:	Version of this uEFI BIOS
EFI build time:	Build time of this uEFI BIOS
EFI build date:	Build date of this uEFI BIOS
NorthBridge.:	Chip revision of the NorthBridge
SouthBridge.:	Chip revision of the SouthBridge
Microcode:	Currently loaded microcode
CPU ID:	CPUID
CPU Branding:	CPU identification string
RIO Module:	Type of attached RIO module

8.2.3 kboot uEFI Shell Command

kboot

FUNCTION:	Boot a legacy OS Not to be used for uEFI BootLoaders!
SYNTAX:	**kboot [-?|-d|-p|-p <path>|-n <name>|-t <type>]** where: -? Show online help -d Boot default order -p <path> Specify the path to the device to boot from -n <name> Specify the device name to boot from -t <type> Specify the device type to boot from Available types are: floppy harddrive cdrom network usb-floppy usb-harddrive usb-cdrom
DESCRIPTION:	The **kboot** command boots a legacy OS. Boot device can be selected in a very flexible way. If the requested device is not present, boot returns to shell. The **kboot** command cannot boot native uEFI operating systems. But since these are bootable from shell by calling their bootloader, this is not necessary either. If a requested device is present but not bootable, uEFI continues to boot with the next bootable device in the boot order.
USAGE:	Show all connected devices: COMMAND / RESPONSE EXAMPLE: **fs0:\> kboot** **____BBS_TABLE____** **00002 network "IBA GE Slot 0100 v1300"** **00003 network "IBA GE Slot 0101 v1300"** **00004 network "IBA GE Slot 0200 v1300"** **00005 network "IBA GE Slot 0201 v1300"** **00002 usb-harddrive "SanDisk uSSD 5000 0.1"** **Device path: Acpi(PNP0A03,0)/Pci(1A|7)/Usb(1,0)** **0001 usb-harddrive "KingstonDataTraveler 2.04.10"** **Device path: Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1,0)** Boot from device containing the string "Kingston": **fs0:\> kboot -n Kingston** Boot from the first device found that is of type floppy: **fs0:\> kboot -t floppy**

8.2.4 kbootnsh uEFI Shell Command

kbootnsh

FUNCTION:	Manage the startup script stored in the flash
SYNTAX:	**`kbootnsh [-?|-g <filename>|-p <filename>|-d]`** where: -? Show online help -g <filename> Store the current boot script to disk. If there is no physical disk drive present, the **kmkramdisk** command may be used. -p <filename> Store the shell script pointed to by filename to flash. Note: The shell script cannot be larger then 400 bytes. -d Delete the current startup script from flash.
DESCRIPTION:	The **kbootnsh** command manages the flash stored startup script. If the shell is launched by the boot process, it executes a shell script stored in the flash. If the shell script terminates, the shell executes a **kboot -d** command to continue the boot process. However, the shell script can of course contain any other boot command.
USAGE:	Get current startup script to file named boot.nsh **`kbootnsh -g boot.nsh`** Store file named boot.nsh to flash: **`kbootnsh -p boot.nsh`** Delete startup script: **`kbootnsh -d`**

8.2.5 kclearnvram uEFI Shell Command

kclearnvram

FUNCTION:	Clear the NVRAM to restore the system's default settings
SYNTAX:	**`kclearnvram [-?] [-q]`** where: -? Show online help -q Silent mode operation (for use of this command in shell scripts)
DESCRIPTION:	Invoking the **kclearnvram** command clears the system NVRAM. Since all uEFI settings are stored inside the NVRAM, the default settings are loaded afterwards. When invoked without the "-q" option, this command must be confirmed by pressing "c". If invoked with the "-q" option, no confirmation is requested.

8.2.6 kclsp uEFI Shell Command

kclsp

FUNCTION:	Configure clock spreading
SYNTAX:	**`kclsp [-?|-d|-e]`** where: -? show help -d disable clock spreading -e enable clock spreading
DESCRIPTION:	The **kclsp** command enables or disables clock spreading on the onboard core clock generator. Clock spreading can be used to reduce system EMI. The default setting is enabled.
USAGE:	Get help: COMMAND / RESPONSE EXAMPLE: **`Shell> kclsp -?`** **`Kontron Clock Spreading Configuration for ICS9LPRS365`** **`-d disable clock spreading`** **`-e enable clock spreading`** This parameter is volatile, and at the next startup, it is set to enabled.

8.2.7 kflash uEFI Shell Command

kflash

FUNCTION:	Manage uEFI BIOS update
SYNTAX:	**`kflash [-?] [-h] [-q] [-c] [-i]`** **`[[-p] [-v] [-s] [-r] [-f] <file>]`** where: -? Show online help -h Show this help -p Program flash -i Show information string and check CRC -v Verify flashed image -s Save current ROM image to file -c Clone flash content to second flash -f Force write -q Silent mode, no user interaction required (for use of this command in shell scripts) -r Raw image mode (.bin, .rom) (expert function, not recommended for standard usage) <file> uEFI BIOS binary file
DESCRIPTION:	The **kflash** command is used to program and verify the flash banks holding the uEFI BIOS code. uEFI BIOS binary files must be available from connected mass storage devices, such as USB flash drive or harddisk.
USAGE:	Get help: COMMAND / RESPONSE EXAMPLE: **`Shell> kflash -?`**
	Get help: COMMAND / RESPONSE EXAMPLE: **`Shell> kflash -h`**
	Program the uEFI BIOS into the standard SPI boot flash: COMMAND / RESPONSE EXAMPLE: **`Shell> kflash -p BIOS_file.kfl`**
USAGE: (continued)	Copy the currently running uEFI BIOS into the inactive SPI boot flash: COMMAND / RESPONSE EXAMPLE: **`Shell> kflash -c`** **Note:** Using this function will overwrite the inactive SPI boot flash. Failures during the process will make the inactive SPI boot flash invalid. In such cases, it will be necessary to execute the function again until the process completes successfully.

8.2.8 kipmi uEFI Shell Command

kipmi

FUNCTION:	Read or configure available Board Management Controller parameters
SYNTAX:	**kipmi [-?|-b|parameters]** where: -? show online help -b display output page by page parameters fru -- display fru data ipmb -- ipmb bus settings: ipmb [redundant/single] irq -- irq [number]: get/set KCS IRQ mode -- set ipmi controller mode: mode [bmc/smc] net -- display and change mode SOL network settings sel -- handle system event log raw -- execute raw ipmi command rawsendmessage -- execute raw SendMessage ipmi cmd info -- show information about the device and firmware
DESCRIPTION:	The **kipmi** command can read event logs or can set the Board Management Controller IRQ configuration. This shell application can also be used to set up raw command to the Board Managment Controller.
USAGE:	Read or configure available Board Management Controller parameters COMMAND / RESPONSE EXAMPLE: **Clear all sel entries** **kipmi sel clear** **Execute raw command. Ex: Get selftest results** **kipmi raw 0x06 0x00 0x04** **Change IRQ** **kipmi irq 10** **Show IRQ configuration** **kipmi irq**

8.2.9 kmkramdisk uEFI Shell Command

kmkramdisk

FUNCTION:	Create RAMdisk drives
SYNTAX:	**kmkramdisk [-?|-s <size> <name>]** where: -? show help -s <size> <name> create a RAMdisk of given size in Megabytes with the mount point name <name>
DESCRIPTION:	Creates a RAMdisk of variable size. Can be very useful to perform file operations when no real filesystem is connected to the system. Note: The RAMdisk loses its mount point name after all drives are remapped by the **map -r** command. The RAMdisk will then be enumerated as any other connected drive and gain a mount point name like “fs0”. This is not a bug of the **kmkramdisk** command but a normal function of the uEFI framework.
USAGE:	Create RAMdisk: COMMAND / RESPONSE EXAMPLE: **rd:\> kmkramdisk -s 5 myramdisk** **Device mapping table** **myramdisk :BlockDevice - Alias (null)** **VenMsg’(93B5F448-127A-4B29-B306-** **5BE8AAC4826E)** **Success - Force file system to mount** **rd:\> myramdisk:** **myramdisk:\> echo testfile > testfile** **myramdisk:\> ls** **Directory of: myramdisk:** **05/24/08 04:39a 22 testfile** **1 File(s) 22 bytes** **0 Dir(s)**

8.2.10 kpassword uEFI Shell Command

kpassword

FUNCTION:	Control uEFI setup and shell passwords
SYNTAX:	**kpassword [-u [-n <password>] [-o <password>]]** **[-s [-n <password>] [-o <password>]]** where: -u Install or change user password -s Install or change superuser password Additional options for automated scripting -n <password> New password to be set -o <password> Password to be overwritten if one is already set When used without option “-n” the password is cleared
DESCRIPTION:	The **kpassword** command is used to determine the status of both passwords (set or not set) and to set or clear the uEFI shell and setup passwords. Both user and superuser (Administrator) passwords can be controlled with this command. Call without options to get current password status If a password has been previously entered, it must be re-entered to validate the command (-o <*old-password*>). Entering an empty password clears the password.
USAGE:	Set User password for uEFI setup and shell COMMAND / RESPONSE EXAMPLE: **kpassword -u** **No password is installed!** **Enter new USER password** **-->** **Retype password** **-->** **Done.**

8.2.11 kresetconfig uEFI Shell Command

kresetconfig

FUNCTION:	Control the board reset behavior
SYNTAX:	**`kresetconfig [-?|<parameter>]`** where: -? Show help <parameter> One of pcislave [on|off] Controls if the board shall react on a PCI backplane reset if it is used as slave board in a peripheral slot. It has no effect if the board is located within a PCI master slot.
DESCRIPTION:	The **kresetconfig** command controls the board's reset behavior.
USAGE:	Enable PCI backplane reset: COMMAND / RESPONSE EXAMPLE: **`Shell> kresetconfig pcislave on`** **`Reset from system master is enabled`**
	Disable PCI backplane reset: COMMAND / RESPONSE EXAMPLE: **`Shell> kresetconfig pcislave off`** **`Reset from system master is disabled`**

8.2.12 kwdt uEFI Shell Command

kwdt

FUNCTION:	Configure the Kontron onboard Watchdog
SYNTAX:	**kwdt [-?|-t <timeindex>]** where: -? Show help -t <timeindex> Configure the Watchdog with the time related to timeindex and activate it with reset routing Call kwdt -h to obtain a list of timeindex values and related times
DESCRIPTION:	The **kwdt** command allows to enable the Kontron onboard Watchdog with reset target before OS boot. This can be used to detect if the OS fails to boot and react by reset. The OS Watchdog driver is required for this functionality to operate.
USAGE:	Get help: COMMAND / RESPONSE EXAMPLE: **Shell> kwdt -?** **-t [time] - set Timer** **value 0 = 125ms** **value 1 = 250ms** **value 2 = 500ms** **value 3 = 1s** **value 4 = 2s** **value 5 = 4s** **value 6 = 8s** **value 7 = 16s** **value 8 = 32s** **value 9 = 64s** **value 10 = 128s** **value 11 = 256s**
	Set Watchdog to 16 seconds and activate it COMMAND / RESPONSE EXAMPLE (none): **Shell> kwdt -t 7** Note: Because there is no application which triggers the Watchdog, the system will be reset after 16 seconds in this case. This command should be invoked from a script, followed by an operating system boot, and the OS then has to start triggering the Watchdog.

8.3 uEFI Shell Scripting

8.3.1 Startup Scripting

If the ESC key is not pressed and the timeout is run out, the uEFI Shell tries to execute some startup scripts automatically. It searches for scripts and executes them in the following order:

1. Kontron flash-stored startup script
2. If there is no Kontron flash-stored startup script present, the uEFI-specified **`startup.nsh`** script is used. This script must be located on any of the attached FAT formatted disk drives under **`\efi\boot\startup.nsh.`**
3. If none of the startup scripts is present or the startup script terminates, the default boot order is continued.

8.3.2 Create a Startup Script

Startup scripts can be created using the uEFI Shell built-in editor **edit** or under any OS with a plain text editor of your choice. To create a startup shell script, simply save the script on any FAT-formatted drive attached to the system under the file name **`\efi\boot\startup.nsh.`** To copy the startup script to the flash use the **kbootnsh** uEFI Shell command.

In case there is no mass storage device attached, the startup script can be generated in a RAM disk and stored in the flash bank.

8.3.3 Examples of Startup Scripts

8.3.3.1 Automatic Booting from USB Flash Drive

Automatic booting is made from a USB flash drive, if present, otherwise the boot is made from the harddrive.

```
kboot -t usb-harddrive
kboot -t harddrive
```

If neither a USB flash drive nor a harddrive is present, the boot order is continued.

8.3.3.2 Execute Shell Script on Other Harddrive

This example executes the shell script named **`bootme.nsh`** located in the root of the first detected disc drive (**`fs0`**).

```
fs0:
bootme.nsh
```

8.3.3.3 Enable Watchdog and Control PXE Boot

The uEFI Shell provides environment variables used to control the execution flow.

The following sample start-up script shows two uEFI Shell environment variables, **wdt_enable** and **pxe_first**, used to control the boot process and the Watchdog.

```
echo -off
echo “Executing sample startup.nsh...”
if %wdt_enable% == “on” then
   kwdt -t 15
   echo “Watchdog enabled”
endif
if %pxe_first% == “on” then
   echo “forced booting from network”
   kboot -t network
endif
```

To create uEFI Shell environment variables, use the **set** uEFI Shell command as shown below:

```
Shell> set wdt_enable on
Shell> set pxe_first on
Shell> set
   pxe_first : on
   wdt_enable : on
Shell> reset
```

8.3.3.4 Handling the Startup Script in the Flash Bank

In case there is no mass storage device attached, the startup script can be generated in a RAM disk and stored in the flash bank using the following instructions:

1. Press <ESC> during power-up to log into the uEFI Shell.
2. Create a RAM disk and set the proper working directory as shown below:

```
Shell> kmkramdisk -s 3 myramdisk
Shell> myramdisk:
```

3. Enter the sample start-up script mentioned above in this section using the **edit** uEFI Shell command.

```
myramdisk:\> edit boot.nsh
```

4. Save the start-up script to the uEFI flash bank using the **kbootnsh** uEFI Shell command.

```
myramdisk:\> kbootnsh -p boot.nsh
```

5. Reset the board to execute the newly installed script using the **reset** uEFI Shell command.

```
myramdisk:\> reset
```

6. If a script is already installed, it can be edited using the following **kbootnsh** uEFI Shell commands.

```
myramdisk:\> kbootnsh -g boot.nsh
myramdisk:\> edit boot.nsh
```

This page has been intentionally left blank.

Chapter 7

Updating the uEFI BIOS

This page has been intentionally left blank.

8. Updating the uEFI BIOS

BIOS updates are typically delivered as an update CD ISO image. This ISO image needs just to be burned to a CD and booted. Follow the menu for updating the uEFI BIOS. For further information refer to the update CD documentation.

8.1 uEFI BIOS Fail-Over Mechanism

The CP308 has two SPI boot flashes programmed with the uEFI BIOS, a primary SPI boot flash and a secondary SPI boot flash. The basic idea behind that is to always have at least one working uEFI BIOS flash available regardless if there have been any flashing errors or not.

8.2 Updating Strategy

To always maintain at least one EFI flash correct, the update CD uses the following update procedure:

1. Switch to the second flash.
   Since the update CD always changes the flash chip prior to doing any updates, the uEFI BIOS that was used to actually boot the board and is therefore known to be good is preserved for backup.
2. Update the second flash.
   This flash is now selected as active boot flash.

The update CD will not allow to flash both chips at a time. Flashing both chips would destroy the backup version and therefore break the redundancy.

If you want to have the same BIOS version on both flash chips, then simply run the update CD twice.

8.3 Fallback Mechanism

In case of one EFI being corrupted and therefore the board not starting up, the IPMI controller automatically switches to the other flash and resets the board. The board should now come up successfully from the other not corrupted image. The flashing procedure can now be restarted to restore the broken image.

8.4 Determining the Active Flash

Sometimes it may be necessary to check which flash is active. On the AMI Aptio-based uEFI BIOS, the information is available using the **kboardinfo** uEFI Shell command. For further information, refer to Chapter 6.2.3, kboardinfo uEFI Shell Command.

This page has been intentionally left blank.