Advanced Power Management (APM)

BIOS Interface Specification

Revision 1.2

February 1996

Intel Corporation

Microsoft Corporation

For additional copies of this specification please contact:

Intel Corporation
http://www.intel.com/IAL/powermgm

OR

Microsoft Corporation
http://www.microsoft.com/windows/thirdparty/hardware/pcfuture.htm

Questions regarding this specification and requests for OEM IDs may be forwarded to:

APM Support Desk
Intel Corporation
TEL: (800) 628-8686
FAX: (916) 356-6100

Revision Record		
Edition	**Date Published**	**Revised Comments**
1.0	January 1992	Original Release.
1.1	September 1993	Editorial Cleanup. Restructured the document. Expanded all sections. Added add-in device support. Added glitch protection support. Added new power management events. Added minor enhancements and fixes. Added the APM Driver Version function. Added the Engage/Disengage Power Management function. Added the Battery flag (CH) return status field to the Get Power Status function. Added Network Adapter and PCMCIA Socket Device IDs. Added Last Request Processing Notification and Last Request Rejected parameters to the Set Power State function. Added 5 second response limit for Standby Request, Suspend Request and Critical Suspend Notification events. Added requirements to the APM Power Driver to poll the APM BIOS with the Get PM Event Function every second with a one second grace period.
1.2	February 1996	Added support for Get/Set/Disable Resume Timer. Added Get Capabilities function. Added Enable/Disable Resume on Ring Indicator function. Added a PCMCIA socket on/off indicator returned with a Normal Resume System Notification PM Event from the Get PM Event function. Added Capabilities Change Notification Added Enable/Disable Timer Based Requests function. New error codes defined for the new functions. Added multiple battery support to Get Power Status call. Power Status Change event must be posted when number of batteries installed changes. Changes to 32-bit Interface Connect return parameters. Clarifications to Engage/Disengage Power Management, Enable/Disable Power Management, Restore APM BIOS Power-On Defaults, and Set Power State functions. Changes to Enable/Disable Power Management function error codes. Changes to the CPU Usage Functions section. Added the requirement for 16-bit and 32-bit APM interface support. Clarifications to Battery Low Notification and Update Time Notification. Updated APM System State table Reordered function sections numerically.

Table of Contents

1. Introduction

1.1 Scope

This document defines the APM software interface. It contains information for software developers to implement an APM BIOS or APM Driver.

1.2 Overview

Chapter 1 is a general discussion of APM concepts.

Chapter 2 describes the APM model for power managed systems and devices. The advanced power management states and the transitions between those states are defined.

Chapter 3 discusses the APM software layers.

Chapter 4 presents the APM software interfaces. APM functions are described in detail, including their purpose and function arguments.

The appendices summarize APM functions and return codes. Considerations for APM BIOS, APM Driver, and APM-aware applications also appear in the appendix.

1.3 Concept

Advanced Power Management (APM) consists of one or more layers of software that support power management in computers with power manageable hardware. APM defines the hardware independent software interface between hardware-specific power management software and an operating system power management policy driver. It masks the details of the hardware, allowing higher-level software to use APM without any knowledge of the hardware interface.

The APM software interface specification defines a layered cooperative environment in which applications, operating systems, device drivers and the APM BIOS work together to reduce power consumption. APM extends the life of system batteries and thereby increases productivity and system availability. APM is also useful in environments and applications such as portables in docking stations, desktops on a network, and energy conscious power managed desktops.

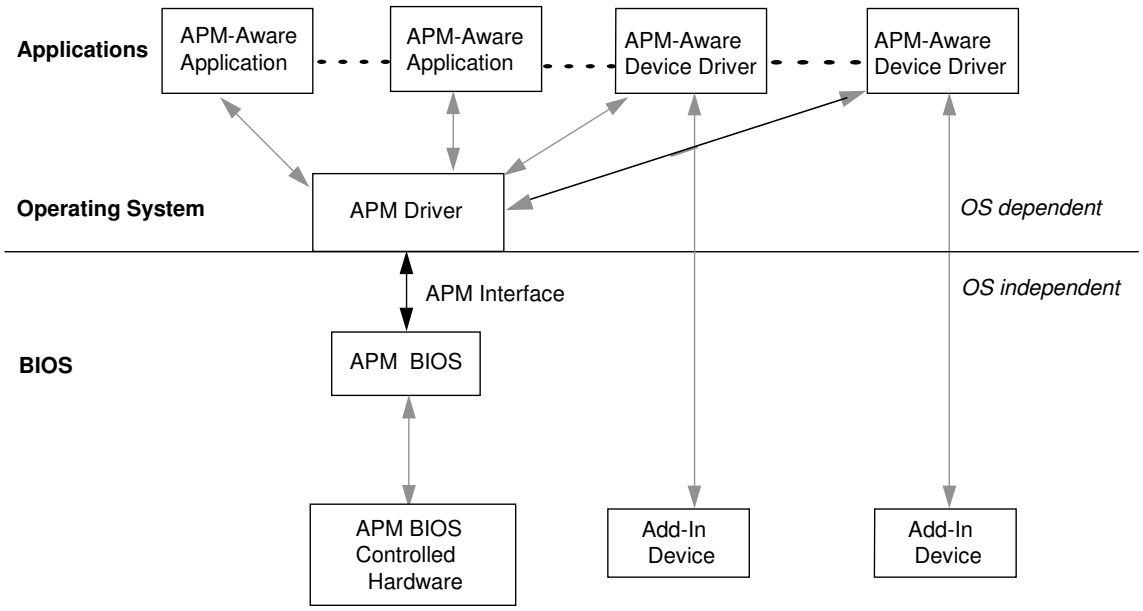

Figure 1. Advanced Power Management System

APM partitions power management functionality into a hierarchy of cooperating layers and standardizes the flow of information and control across the layers. Figure 1 shows the software components in an APM system.

- The **APM BIOS** is the software interface to the motherboard and its power managed devices and components.
- The **APM Interface** is the interaction between the **APM Driver** and the **APM BIOS**.
- The **APM Driver** module connects to the **APM BIOS** and controls power management policy. The APM Driver communicates with **APM-aware Applications**.
- **APM-aware Applications** interface with the **APM Driver** to monitor and/or control power management.
- The **APM-aware Device Driver** modules provide power management software interface for add-in devices.

1.4 Terms

Table 1. Terms Defined

Add-in device	A device which is not on the Motherboard. This device may be a permanent part of the system like an add-in card, or it may be insertable and removable by the operator, like a PCMCIA card. APM-aware device drivers provide power management software support for add-in devices.
APM BIOS	The system BIOS module that provides power management functionality for motherboard resident hardware adhering to the APM Interface specification.
APM Driver	The software module connected to the APM BIOS Interface. The APM Driver and the APM BIOS coordinate APM system power management through the software connection.
APM-aware Application	An application that participates in system power management through an operating system dependent APM Driver interface.
APM-aware Device Driver	An APM-aware application that provides power management for add-in devices.
APM connection	The protocol and mechanism by which an APM Driver and an APM BIOS intercommunicate.
CPU Core	The hardware of the system that must be powered on to perform any computation. The CPU core includes the CPU clock, cache, system bus, and system timers.
Resume Timer	This is a system timer used to wake up the system from a low power state at a specified time.
System	The entire computer. It includes the CPU core, I/O devices and peripherals, the power supply, and all devices and components.
System Idle	The system is inactive. The CPU is not processing and there is no I/O activity.

2. Advanced Power Management Model

The objective of APM is to control the power usage of the system based on system activity. As system activity decreases, APM reduces power to unused system resources until the system is brought into a suspend state.

There are two methods of power level control.

1. The APM BIOS manages power in the background based on device activity. The APM BIOS is supplied by the OEM and is specific to the hardware platform.
2. An APM Driver participates in managing power levels via function calls to the APM software interface. See the section *APM Functions* on page 26 for a complete description of these calls.

Both methods cause transitions from one APM-defined power management state to another. These states are defined by the power levels of the controlled devices and CPU.

2.1 System Power States

An APM system has five general system power states: Full On, APM Enabled, APM Standby, APM Suspend, and Off. The main difference between the states is the latency needed for the system to power up to a working state. Power consumption and performance are greatest in the Full On State and decrease with each state. APM States are defined in general terms and allow for implementation variations. Refer to Table 2 for a detailed description of each state.

- The Full On State is the default mode when the system is not doing power management.
- In the APM Enabled State the system is doing work but some unused devices may not be powered.
- The system enters the APM Standby State after a short period of inactivity. Recovery from the APM Standby State to the APM Enabled state appears instantaneous.
- The system enters the APM Suspend State after a relatively long period of inactivity. It takes a relatively long time to recover from the APM Suspend State to the APM Enabled State.

Table 2. APM System States

System States	Characteristics
Full On	System is working. System is not power managed. All devices are on.
APM Enabled	System is working. System is power managed. The CPU clock is slowed or stopped as needed. Devices are power managed as needed.
APM Standby	System may not be working. System is in a low power state with some power savings. Most devices are in a low power mode. The CPU clock is slowed or stopped. Operational parameters are retained. System returns quickly to the APM Enabled State. The Resume Timer event must return the system to the APM Enabled State. User activity may be required to return the system to the APM Enabled State. The operating system is notified after the system transitions to the APM Enabled State. Prior operation resumes after returning to the APM Enabled State. Interrupts must still be processed normally. This may require waking up the CPU temporarily if it was stopped. The CPU may be stopped again when the APM Driver calls the CPU Idle function.
APM Suspend	System is not working. System is in a low power state with maximum power savings. Most power managed devices are not powered. The CPU clock is stopped. The CPU core is in its minimum power state. Operational parameters are saved to be restored later when resuming. System takes a relatively long time to return to the APM Enabled State. Wakeup events defined by the OEM return the system to the APM Enabled State. The Resume Timer event must be one of the wakeup events. The operating system is notified after the system transitions to the APM Enabled State. Prior operation resumes after returning to the APM Enabled State. System may go into hibernation state to save operational parameters and allow transition into and out of the off state. Hibernation is a special implementation of Suspend.
Off	System is not working. The power supply is off. Operational parameters are not saved. System resets and initializes when transitioning to the Full On State.

Note:

The saving of operational parameters is the responsibility of the BIOS and/or APM aware device driver that is specific to the hardware and power management features of a particular platform.

2.2 Device Control

Devices may be power-managed either by the system BIOS or by the operating system. Using the *Disable Device Power Management* function, the operating system may tell the BIOS to stop power managing a particular device, after which operating system-specific drivers for that device may directly take over power management of the device instead.

However, prior to entering the APM Suspend State, the operating system should assume that all peripheral devices will lose power under the control of the system BIOS. This makes it imperative that the system BIOS respect the operating system's ability to reject a timer- or user-initiated suspend so that all operating system device drivers may adequately prepare for the removal of power. New parameters have been added to the *Set Power State* function to assist in this process. After resuming from a critical suspend event, it is also crucial that the system be returned to its prior state as closely as possible so that device I/O which may have been pending just prior to the critical suspend may continue as transparently as possible.

Devices may have built-in automatic power management functions which are not software programmable. These devices are outside the scope of this APM specification.

Devices are controlled in the background by the APM BIOS or by the APM Driver using APM function calls.

Note: The following table is a list of device modes that one could choose to implement. Other modes could be implemented at the OEMs discretion. The relationship between device modes and APM states is defined by the OEM.

Table 3. APM Device Modes

Power Managed Device Mode	Characteristics
Device On	Device is fully powered and able to perform work. All device features are available.
Device Power Managed	Device is working, but some features may not be operational, or may be functioning at reduced performance levels. Power is maintained. Operational parameters are retained. Devices may have variable numbers of device power managed submodes.
Device Low Power	Device is not working. Power is maintained. Operational parameters are retained. Devices may have variable numbers of low power submodes.
Device Off	Device is not working. Device is powered off. Operational parameters are not retained.

Note:

The saving of operational parameters is the responsibility of the BIOS and/or APM-aware device driver that is specific to the hardware and power management features of a particular platform.

2.3 The CPU Core Control

The CPU core is managed differently than other devices. It is typically the last device to power off and the first device to power on.

The CPU core includes the CPU clock, cache, system bus, and system timers. The CPU core is required to perform system power state changes.

- In the APM Enabled State the CPU clock is controlled locally by being turned off and on as needed but the CPU core is never turned off.
- In the APM Standby State the CPU clock is stopped but the CPU core is never turned off.
- In the APM Suspend State the CPU clock is stopped and the CPU core is in its minimum power state.

The CPU core is controlled through the APM BIOS. An APM Driver may notify the APM BIOS about CPU usage but the APM BIOS determines the action to take. The APM function calls that notify the APM BIOS are *CPU Idle,* and *CPU Busy.*

Table 4. CPU Core Modes

CPU Core Modes	Characteristics
Full On	Full speed operation. High power consumption. High performance level.
Slow Clock	Reduced speed operation. Reduced power consumption. Reduced performance level. Interrupts restore to Full On Mode. Restoration to Full On appears instantaneous.
Stop	Only a hardware interrupt starts the clock. Restoration to Full On appears instantaneous.

Note:

The saving of operational parameters is the responsibility of the BIOS and/or APM-aware device driver that is specific to the hardware and power management features of a particular platform.

2.4 System Power State Transitions

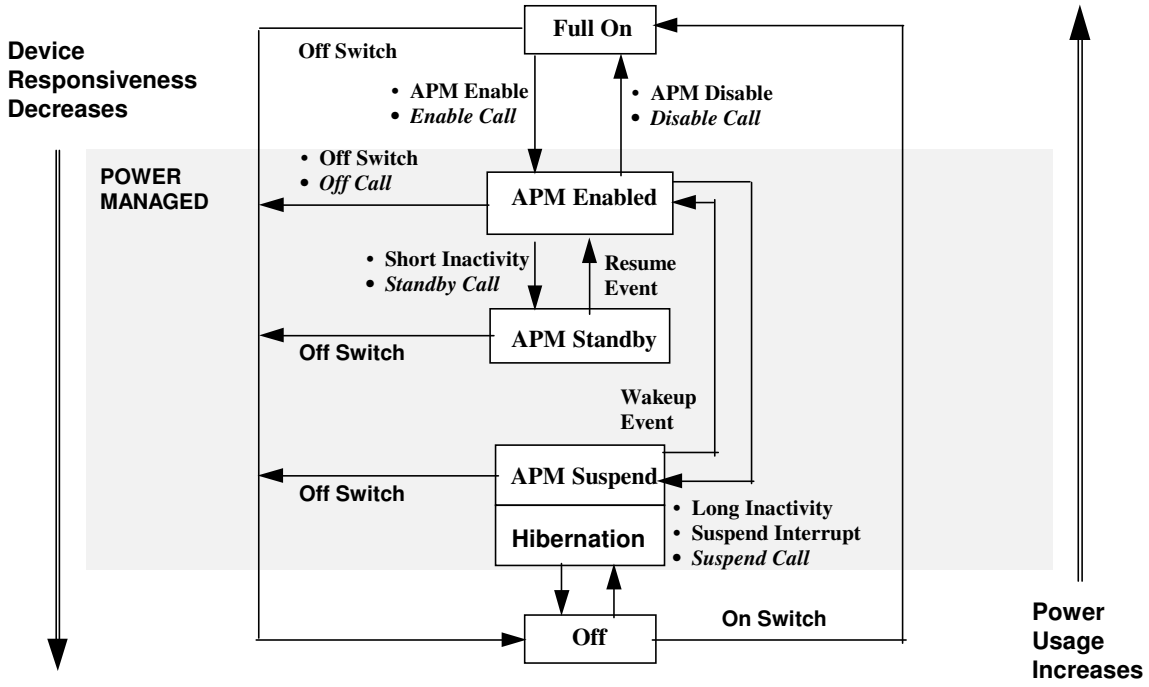

Figure 2. APM System State Transitions

Inactivity causes a gradual powering down of the system. The system state changes are caused by the APM BIOS due to system activity level, interrupts, and external events or due to software requests if an APM Driver is connected.

When an APM Driver is connected, the APM BIOS and the APM Driver cooperate to effect system state transitions. The system transitions caused by APM functions called by the APM Driver are in italics.

Off Switch or Off Call

Turning the on/off switch to off from any state puts the system in the Off state.

The Set Power State function puts the system into the Off state from the APM Enabled state.

If hibernation is activated before the system enters the Off state, the system state and memory contents will be saved through the Off period and can therefore be restored.

On Switch

Turning the on/off switch to on from off powers the system on. The system goes through a reset and initialization process and loads the operating system.

If hibernation was activated before the system entered the Off state, the system will restore the operational parameters and memory contents previously saved, and will place the system into the APM Enabled state.

APM Enable or Enable Call

APM Enable puts the system into the APM Enabled state from the Full On State. This transition happens after an APM Driver connects with the APM BIOS or by software that indicates that APM should be enabled.

The Enable Power Management call puts the system into the APM Enabled state from the Full On state.

APM Disable or Disable Call

APM Disable puts the system into the Full On state from the APM Enabled state.

The Disable Power Management call puts the system into the Full On state from the APM Enabled state.

Short Inactivity or Standby Call

System Inactivity for a specified short period of time puts the system into the APM Standby state from the APM Enabled state.

The Set Power State function called with BX=0001H (all devices) and CX=0001H (Standby) tells the APM BIOS to put the system into the APM Standby state from the APM Enabled state

Long Inactivity or Suspend Interrupt or Suspend Call

System Inactivity for a specified long period of time puts the system into the APM Suspend state from the APM Enabled state.

Certain interrupts such as a suspend button press, battery low alarm or a lid closure put the system into the APM Suspend state from the APM Enabled state.

The Set Power State function called with BX=0001H (all devices) and CX=0002H (Suspend) tells the APM BIOS to put the system into the APM Suspend state from the APM Enabled state.

Resume Event

An interrupt such as mouse movement, keyboard press, and any device access puts the system into the APM Enabled state from the APM Standby state.

Wakeup Event

Certain interrupts such as a resume button press, modem ring, or a real-time clock alarm put the system into the APM Enabled state from the APM Suspend state.

3. Advanced Power Management Software Layers

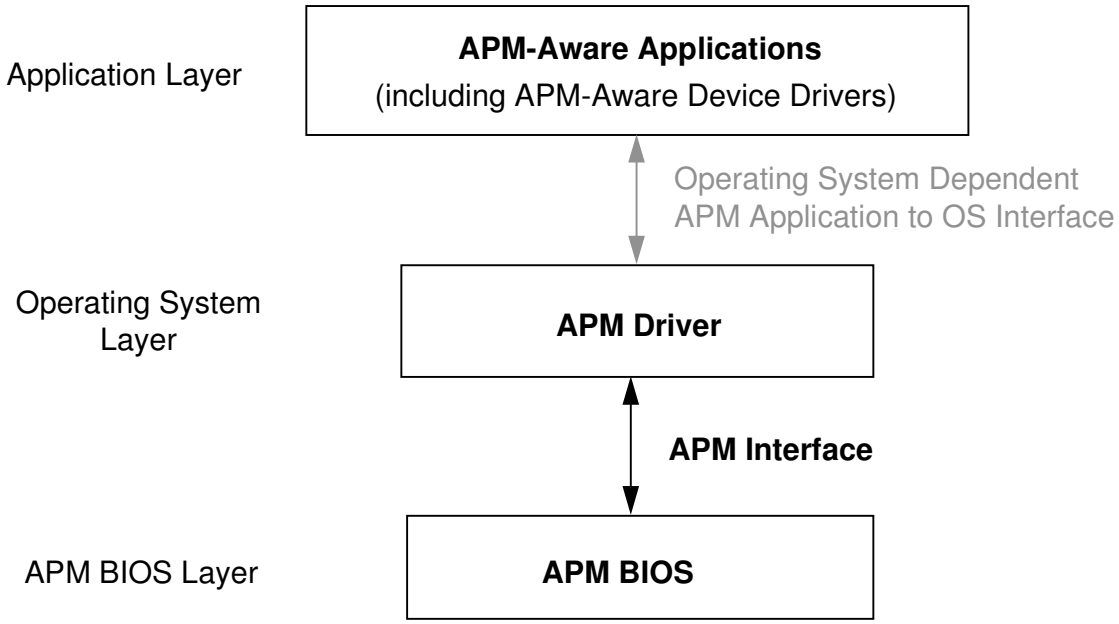

Figure 3. APM Software Layers

There are three APM software components.

APM BIOS	This is the motherboard hardware-specific power management module.
APM Driver	This operating system-dependent module cooperates with the APM BIOS to enforce the power management policy.
APM-aware Application	The APM Driver exchanges power management information with these applications. APM-aware device drivers are APM-aware applications that manage add-in devices.

The APM BIOS and the APM Driver communicate to effect cooperative power management.

- The APM Driver sends information to the APM BIOS via function calls.
- The APM Driver uses polling function calls to the APM BIOS to gather information about power management events.

The APM Driver exchanges power management information with its APM-aware applications.

3.1 The BIOS Layer

The APM BIOS interfaces directly to the power managed system motherboard hardware. The APM BIOS is the lowest level of power management software in the system. It is supplied by the OEM and is specific to the hardware platform. An APM BIOS may provide some degree of power management functionality without any support from operating system or application software. The OEM or BIOS provider determines the degree to which power management functionality is implemented by the APM BIOS in a stand-alone configuration.

The APM BIOS stand-alone power management functionality is enhanced once an APM Driver establishes a cooperative connection with the APM BIOS. Once made, this connection establishes a protocol that allows the firmware to communicate power management events to the APM Driver and to wait for APM Driver concurrence if necessary. Details of the APM BIOS operational changes are documented in the description of the interface.

The APM BIOS software interface is described in the Advanced Power Management Software Interface section.

3.2 The Operating System Layer

The APM Driver has three primary power management functions:

1) passing calls and information between the application and APM BIOS layers
2) arbitrating application power management calls in a multitasking environment
3) identifying power saving opportunities not apparent at the application or BIOS layer.

An APM connection must be established between the APM BIOS and an APM Driver for optimum system power management. By regularly polling the APM BIOS, the APM Driver will determine whether the APM BIOS wants a state transition to occur. In the case of a Standby or Suspend request, the APM Driver is expected to do the appropriate processing to prepare for the state change and then call the APM BIOS to perform the power state change.

Since only one APM Driver can exist in the system, and there may be many APM applications, the APM Driver must provide an interface between the APM BIOS and APM-aware applications. This interface should pass APM-aware application APM requests to the APM BIOS, and send the APM BIOS PM events up to the applications.

Each APM Driver should specify a power management application-to-OS interface. The details of such an interface are beyond the scope of this document. Refer to the OS' APM Driver documentation for this information.

3.3 The Application Layer

APM-aware applications assist in power management by providing information that only the application is in a position to know or easily ascertain. Similarly, device drivers for add-in devices which are not under direct control of the BIOS (such as PCMCIA cards or ISA video adapters) may be able to manage power on their device given suitable information about the overall system state or by monitoring usage of their own hardware. Applications and device drivers are not required to be APM-aware, but they can greatly increase APM effectiveness, particularly on less sophisticated operating systems. Under MS-DOS in particular, the application or device driver is often in the best position to know when the application or an add-in device is idle and awaiting further activity.

APM-aware applications register with the APM Driver using an operating system dependent mechanism. The APM Driver notifies registered applications and device drivers when system power management events occur, and the applications and device drivers then take suitable action. For information on how applications and device drivers should interface with the APM driver, reference the documentation for the operating system in question.

For instance, when an APM-aware device driver learns from the APM Driver that the system will be suspended, it saves device information to be restored when the system resumes. When an APM-aware device driver is notified that the system has resumed it restores the add-in device to its previous operating state.

Important Note:
Different operating systems may provide varying levels of support for APM-aware applications and device drivers, but note that such applications and device drivers must not call the APM interface directly using the functions described in this document. These calls are intended for use strictly by the operating system specific APM Driver.

4. Advanced Power Management Software Interface

This chapter describes the software interface between the APM BIOS and the APM Driver.

This chapter shows APM BIOS developers how to determine which functions must be implemented in the APM BIOS and how to perform cooperative power management with the APM Driver.

This chapter shows APM Driver developers how to call the APM BIOS and how to perform cooperative power management with the APM BIOS.

APM-aware application developers participate in power management of the system through an operating system dependent interface that is outside the scope of this specification. Refer to the OS' APM Driver documentation.

This chapter, particularly section 4.3, Power Management Events, describes a lower-level (transparent to the APM-aware application) APM software interface. The particular APM Driver that an APM-aware application is written for provides APM documentation for its specific APM-aware application interface.

- Section 4.1 describes the calling interface used by an APM Driver to access APM BIOS functions.
- Section 4.2 describes the establishment and use of a cooperative power management connection between an APM Driver and the APM BIOS.
- Section 4.3 describes the implementation and use of Power Management Event Notification from the APM BIOS to the APM Driver.
- Section 4.4 describes the use of the *CPU Idle* and *CPU Busy* functions by the APM Driver to indicate CPU usage to the APM BIOS.
- Section 4.5 presents an overview of the APM BIOS functions.
- Section 4.6 describes the APM BIOS function register interface.
- Section 4.7 describes the OEM defined APM function register interface.

4.1 APM Calling Interface

APM 1.2 requires support for real and protected mode calling interfaces.

A real mode APM calling interface is required for all implementations. The real mode APM BIOS interface is an extension to the existing PC/AT Int 15h BIOS interface. The APM BIOS Int 15h interface must operate in either real mode, or virtual-86 mode on 80386 and later processors.

Protected mode interface support for both the 16-bit protect mode and the 32-bit protect mode interface is required for APM 1.2. The protected mode APM BIOS interface is accessed by calling through protected mode entry points into the APM BIOS.

4.2 APM Connection

The APM Driver becomes a partner in the power management process by providing a cooperative interface between the APM BIOS and an APM Driver. When this interface is established, the APM BIOS informs the APM Driver of power related events, and relies on the APM Driver to actively participate in the power management process.

The *APM Installation Check* function and *APM Connection* functions are accessed through the real mode interface only. The *APM Installation Check* function determines if there is an APM BIOS module in the system, the version of APM the APM BIOS supports, and the attributes of the APM BIOS implementation.

The APM Driver must issue an APM Connection call, *APM Protected Mode 16-bit Interface Connect*, *APM Protected Mode 32-bit Interface Connect*, or *APM Real Mode Interface Connect* to establish the appropriate cooperative connection with the APM BIOS. The *APM Interface Disconnect* function disconnects the cooperative connection between the APM Driver and the APM BIOS. An alternative technique to temporarily disable certain aspects of power management functionality without having the APM Driver disconnect altogether is to use the *Enable/Disable Power Management* and the *Engage/Disengage Power Management* functions described later in the document.

A protected mode APM connection must be established for any APM BIOS functions to be accessible in protected mode. Note that the protected mode connection must be established by issuing a call in real mode. The protected mode connection function returns pointers to an entry point into the APM BIOS code and to the data segment of the APM BIOS. The APM Driver issues APM function calls to the APM BIOS using these pointers.

Only one APM connection can be active at a time.

4.3 Power Management Events

Power Management Events are communicated from the APM BIOS to the APM Driver through a polling mechanism. The APM Driver polls to determine if a Power Management Event has occurred by issuing the *Get PM Event* call. The APM Driver broadcasts Power Management Event Notifications to its APM-aware applications.

An APM 1.2 Operating System Driver must poll the system BIOS at a minimum rate of once per second, with a grace period of one second before the BIOS assumes that the operating system is malfunctioning. However, the APM BIOS should make this assumption only when an APM event has been posted and has not been picked up by the APM 1.2 Driver within the two second (one second plus one second grace period) time limit. Older drivers may not be able to conform to this requirement, however the BIOS should use the information provided by the *APM Driver Version* function to determine whether to expect this behavior or not.

Power Management Events 01H - 0CH are described individually in this section. The descriptions say what the APM Driver should do for each event notification. For the Reserved System Events, Reserved Device Events, and OEM-Defined APM Events, the APM Driver broadcasts the events to its APM-aware applications.

The APM BIOS provides notification of a Power Management Event only once. Events are cleared after notification. No additional notifications are posted until another event occurs.

4.3.1 Power Management Event Codes

This is a list of the defined power management events that the APM BIOS can report.

0001H	System Standby Request Notification
0002H	System Suspend Request Notification
0003H	Normal Resume System Notification
0004H	Critical Resume System Notification
0005H	Battery Low Notification
0006H	Power Status Change Notification
0007H	Update Time Notification
0008H	Critical System Suspend Notification
0009H	User System Standby Request Notification
000AH	User System Suspend Request Notification
000BH	System Standby Resume Notification
000CH	Capabilities Change Notification
000DH - 00FFH	Reserved System Events
0100H - 01FFH	Reserved Device Events
0200H - 02FFH	OEM-Defined APM Events
0300H - FFFFH	Reserved

4.3.2 System Standby Request Notification (0001H)

This notifies the APM Driver that the APM BIOS wishes to put the system into the Standby state. The APM Driver determines whether to place the system in the Standby state.

The APM Driver notifies its APM-aware applications of the System Standby Request. This notification allows the device drivers the opportunity to prepare for the standby state change or to reject the request. While processing the System Standby Request, the APM Driver must call the *Set Power State* function with BX=0001H (all devices) and CX=0004H (Request processing) a minimum of once every 5 seconds. When processing completes, the APM Driver calls the *Set Power State* function to set the system state to the Standby state or to reject the standby request. The BIOS may not enter the Standby state of its own accord unless the APM Driver fails to meet the requirement of calling *Set Power State* once every five seconds to indicate Request Processing.

After a system has entered a Standby state using the *Set Power State* function, and the system is resumed, a System Standby Resume Notification power management event will be posted by the APM BIOS to be polled by the APM Driver.

4.3.3 System Suspend Request Notification (0002H)

Notifies the APM Driver that the APM BIOS wishes to put the system into the Suspend State. It is the responsibility of the APM Driver to determine whether the system should be placed in the Suspend State.

The APM Driver should notify its APM-aware applications of the system suspend request. This notification allows the device drivers the opportunity to prepare for the

suspend state change or to reject the request. While processing the suspend request, the APM Driver must call the *Set Power State* function with BX=0001H (all devices) and CX=0004H (Request processing) a minimum of once every 5 seconds. When processing completes, the APM Driver calls the *Set Power State* function to set the system state to suspend or to reject the suspend request. The BIOS may not enter the suspend state of its own accord unless the APM Driver fails to meet the requirement of calling *Set Power State* once every five seconds to indicate Request Processing.

APM O/S Driver suggestion: If the suspend is being delayed due to application activity such as flushing open files, etc., positive feedback should be given to the user of the pending suspend if possible.

After a system has been suspended using the *Set Power State* function, and the system is resumed, a Normal Resume System Notification power management event will be posted by the APM BIOS to be polled by the APM Driver.

4.3.4 Normal Resume System Notification (0003H)

This indicates that a system resume following a coordinated normal system suspend has occurred. The APM Driver must update the time from the real-time clock when it receives this notification.

An indication of whether or not the PCMCIA socket was powered off during the suspend is returned with this notification. The indicator is in the low bit of CX, a value of one in that bit indicates that the PCMCIA socket was powered off.

The APM Driver must notify its APM-aware applications of the normal system resume power management event. The device drivers can then prepare their devices for use.

4.3.5 Critical Resume System Notification (0004H)

This indicates that a Critical Resume System operation occurred. This notification allows the APM Driver to recover from the suspension as best it can. The APM Driver updates the time from the real-time clock when it receives this notification.

An indication of whether or not the PCMCIA socket was powered off during the suspend is returned with this notification. The indicator is in the low bit of CX, a value of one in that bit indicates that the PCMCIA socket was powered off.

The APM Driver notifies its APM-aware applications of the critical resume power management event. The device drivers then prepare their devices for use.

4.3.6 Battery Low Notification (0005H)

Informs the APM Driver that the system's battery is running low. The APM BIOS must send this notification while the battery still contains enough power to suspend or shut down the computer. It is recommended that this notification be sent with at least 10 minutes of battery power remaining to provide the user sufficient advance warning to complete any tasks in progress.

4.3.7 Power Status Change Notification (0006H)

This informs the APM Driver that the system's power status has changed. The APM Driver must issue the *Get Power Status* call to determine the change to the power status.

An APM BIOS signals a Power Status Change Notification event when the system's AC line status or Battery status values (as returned in BH and BL from a *Get Power Status* call) change. The APM BIOS must also signal a Power Status Change Notification event when the number of currently installed battery units changes (as returned in SI from a *Get Power Status* call).

4.3.8 Update Time Notification (0007H)

This informs the APM Driver that the time must be updated by reading the real-time clock. The APM BIOS should send this notification after situations in which the timer ticks are not serviced. An example is when the APM BIOS keeps the clock tick from being serviced while the APM BIOS is handling a device.

The APM Driver is required to update the system time when it receives a System Standby Resume Notification, Normal Resume System Notification or Critical Resume System Notification. The APM BIOS should not send an Update Time event to the APM Driver in those cases.

4.3.9 Critical System Suspend Notification (0008H)

This notifies the APM Driver that the APM BIOS has detected a situation in which the system must be suspended without notification to APM-aware applications. This notification is intended to be used by the APM Driver to perform emergency shutdown actions. The APM Driver must recognize the power management event and then issue the *Set Power State* function call to set the system state to Suspend without delay, and without notifying its applications or device drivers first. However, once the APM Driver receives the Critical Suspend Notification, the BIOS must wait at least 5 seconds from the time that the APM Driver receives the Critical Suspend Notification for it to call *Set Power State* to suspend (BX=0001H, CX=0002H). This must be done because it may not be possible to properly resume the devices in the system if it is suspended before the operating system becomes aware of the impending shutdown.

The APM BIOS may only cause a Critical System Suspend to occur when a critical power condition forces the system to shut down. Critical System Suspends may not be used for non-emergency power situations.

After a system has been suspended and the system is resumed, a Critical Resume System Notification power management event will be posted by the APM BIOS.

The APM Driver must inform APM-aware applications and device drivers without delay when the system is resumed from a critical suspend. Otherwise, a device driver may execute before recovering from the suspend, possibly causing a system failure.

4.3.10 User System Standby Request Notification (0009H)

This notifies the APM Driver that the User wishes to put the system into the Standby state. The APM Driver determines whether to place the system in the Standby state.

The APM Driver notifies its APM-aware applications of the system standby request. This notification allows the device drivers the opportunity to prepare for the standby state change or to reject the request. While processing the standby request, the APM Driver must call the *Set Power State* function with BX=0001H (all devices) and CX=0004H (Request processing) a minimum of once every 5 seconds. When processing completes, the APM Driver calls the *Set Power State* function to set the system state to standby or to reject the standby request. The BIOS may not enter the standby state of its own accord unless the APM Driver fails to meet the requirement of calling *Set Power State* once every five seconds to indicate Request Processing.

After a system has entered the Standby state using the *Set Power State* function, and the system is resumed, a System Standby Resume Notification power management event will be posted by the APM BIOS to be polled by the APM Driver.

4.3.11 User System Suspend Request Notification (000AH)

Notifies the APM Driver that the User wishes to put the system into the Suspend State. It is the responsibility of the APM Driver to determine whether the system should be placed in the Suspend State.

The APM Driver should notify its APM-aware applications of the system suspend request. This notification allows the device drivers the opportunity to prepare for the suspend state change or to reject the request. While processing the suspend request, the APM Driver must call the *Set Power State* function with BX=0001H (all devices) and CX=0004H (Request processing) a minimum of once every 5 seconds. When processing completes, the APM Driver calls the *Set Power State* function to set the system state to suspend or to reject the suspend request. The BIOS may not enter the suspend state of its own accord unless the APM Driver fails to meet the requirement of calling *Set Power State* once every five seconds to indicate Request Processing.

After a system has been suspended using the *Set Power State* function, and the system is resumed, a Normal Resume System Notification power management event will be posted by the APM BIOS to be polled by the APM Driver.

4.3.12 System Standby Resume Notification (000BH)

This event is posted as soon as a resume event occurs.

This indicates that a system standby resume following a coordinated system standby has occurred. The APM Driver must update the time from the real-time clock when it receives this notification.

The APM Driver notifies its APM-aware applications of the normal standby resume power management event. The device drivers can then prepare their devices for use.

4.3.13 Capabilities Change Notification (000CH)

The power management capabilities of a system may change as a result of changes via a setup utility or by the arrival or removal of devices. If these capabilities change, the APM BIOS must post a Capabilities Change Notification. Upon receiving the Capabilities Change Notification, the APM Driver will call Get Capabilities to determine the new capabilities. The APM Driver should also call Get Power Status to determine any status changes caused by the capabilities changes.

4.4 CPU Usage Functions

APM's *CPU Idle and CPU Busy* functions are used by the APM Driver to indicate CPU usage to the APM BIOS.

- The *CPU Idle* function means that the APM Driver detected minimal processing at the application level.
- The *CPU Busy* function means that the APM Driver detected that the application level is no longer operating with minimal processing. Note that many multitasking operating systems are not capable of distinguishing this condition and do not call the *CPU Busy* function.

The APM BIOS takes platform-specific action when these calls are made.

There are two types of actions the BIOS may take when a *CPU Idle* function is executed:

- The APM BIOS stops the clock until the next timer tick interrupt.
- The APM BIOS slows the clock. The *CPU Busy* call can be executed to restore the clock to its normal speed, or alternatively selected forms of system activity such as mouse movement or keyboard activity can be used to restore normal speed as well. This is the only case in which the *CPU Busy* call should be invoked. Execute the *APM Installation Check* function to determine whether the APM BIOS slows the clock in response to a *CPU Idle* call. Bit 2 of the APM flags return parameter is set to 1 if the APM BIOS slows the clock when the *CPU Idle* function is called.

4.5 Function Overview

The register interfaces to the APM BIOS functions are described on the following pages. In real mode, the functions are accessed through INT 15h. In protected mode, the functions are accessed through the protect mode entry point returned from the protected mode connection call.

The carry (CY) flag is set and an error code is placed in the AH register when an error condition is detected during the processing of an APM BIOS function. The carry flag is reset to zero upon return from successful calls. The contents of the AH register depend on the particular call.

In general, all reserved bits must be set to 0 and may not be appropriated for uses not delineated in this specification.

This interface has a convention for identifying a device class and units within that class to allow for direct control of devices. For example, all disk devices are a class and the units are the physical unit numbers.

The power device ID parameter is passed to the APM BIOS in a word length register (BX) where the most significant byte is the device class and the least significant byte is the device unit. The impact of this parameter is described in each interface description. The intent for power device IDs is to have a fixed code for known and generally used peripherals and devices.

4.5.1 Power Device IDs

The APM Power Device Class/Subclass IDs are defined as follows:

BX = Power Device ID

00XXh	System
00	APM BIOS
01	All devices power managed by the APM BIOS
01XXh	Display
02XXh	Secondary Storage
03XXh	Parallel Ports
04XXh	Serial Ports
05XXh	Network Adapters
06XXh	PCMCIA sockets
0700h - 7FFFh	Reserved
80XXh	Batteries
81XXh - DFFFh	Reserved
E000h - EFFFh	OEM-defined Power Device IDs
F000h - FFFFh	Reserved

where:

XX h = Unit Number (0 based)
FF h = All devices in this class

OEM-defined Power Device IDs accommodate devices such as Coprocessors, Memory, Keyboard, Mice, other Handheld entry devices, PCMCIA cards, and Clocks.

4.5.2 APM Session Functions

These functions determine whether APM BIOS support exists, and manage APM sessions.

APM Driver Version	Indicates the APM Driver's level of APM support
APM Installation Check	Return APM information
APM Interface Disconnect	Disconnect an APM connection
APM Protected Mode 16-bit Interface Connect	Establish a 16-bit protect mode APM connection
APM Protected Mode 32-bit Interface Connect	Establish a 32-bit protect mode APM connection
APM Real Mode Interface Connect	Establish a real-mode APM connection
Get Capabilities	Return additional capabilities

4.5.3 APM Status and Control Functions

These functions query and/or control the power state of the system and/or devices.

CPU Busy	APM Driver detected CPU usage
CPU Idle	APM Driver detected CPU idleness
Enable/Disable Device Power Management	Enable or disable APM BIOS power management for a specified device
Enable/Disable Power Management	Enable or disable APM BIOS power management
Enable/Disable Resume on Ring Indicator	Manages the system’s resume on ring indicator functionality.
Enable/Disable Timer Based Requests	Manages the BIOS’ capability to request low power states based on system inactivity.
Engage/Disengage Power Management	Engage or disengage cooperative power management for the system or device
Get Power State	Return power state information
Get Power Status	Return the system's power status information
Get/Set/Disable Resume Timer	Manages the system global resume timer
Restore APM BIOS Power-On Defaults	Restore default APM BIOS power management
Set Power State	Set the power state for the system or a device

4.5.4 PM Event Polling

This function returns information about a power management event that the APM BIOS has detected.

Get Power Management Event	Poll for a power management event

4.5.5 OEM Defined Functions

These functions are reserved for OEM APM implementations.

OEM Defined Function	Perform OEM defined function
OEM Installation Check	Check if an OEM APM module exists

4.6 APM Functions

4.6.1 APM Installation Check (00H)

This call allows the APM Driver (caller) to determine if the system's BIOS supports the APM functionality and if so, which version of the specification it supports.

The APM version number returned from this call is the highest level of APM supported by the APM BIOS.

Call With

AH = 53H APM
AL = 00H Installation Check
BX = Power device ID
0000H APM BIOS
All other values reserved

Returns

If function successful:

Carry = 0 APM is supported by BIOS
AH = 01 APM major version number (in BCD format)
AL = 02 APM minor version number (in BCD format)
BH = ASCII "P" character
BL = ASCII "M" character
CX = APM flags

bit 0 = 1 16-bit protected mode interface supported (required in APM 1.2)

bit 1 = 1 32-bit protected mode interface supported (required in APM 1.2)

bit 2 = 1 *CPU Idle* call slows processor clock speed. This bit set indicates that the APM Driver must call the *CPU Busy* function to ensure that the system is restored to normal processing clock speed after calling the *CPU Idle* function to slow the processor clock speed.

bit 2 = 0 *CPU Idle* call does not slow the processor clock speed or *CPU Idle* call stops the clock. This bit clear indicates that the APM Driver does not need to call the *CPU Busy* function.

bit 3 = 1 APM BIOS Power Management disabled

bit 4 = 1 APM BIOS Power Management disengaged

Other bits Reserved (must be set to 0)

If function unsuccessful:

Carry = 1

AH = Error code
09H Unrecognized device ID
86H APM not present

Supported modes

Real mode

Comments

APM 1.2 provides a functional superset of APM 1.1. An APM 1.2 BIOS must implement the backward-compatible features of APM 1.0 and APM 1.1 to work with APM 1.0 and APM 1.1 APM Driver applications.

4.6.2 APM Real Mode Interface Connect (01H)

This call establishes the cooperative interface between the APM Driver (caller) and the APM BIOS. The APM BIOS provides OEM-defined power management functionality before the interface is established.

Once the interface is established, the APM BIOS and the APM Driver coordinate power management activities. The APM BIOS rejects an interface connect request if any real or protected mode connection already exists.

Call With

AH = 53H APM
AL = 01H Real mode interface connect
BX = Power device ID
0000H APM BIOS
All other values reserved

Returns

If function successful:
Carry = 0
If function unsuccessful:
Carry = 1
AH = Error code
02H Real mode interface connection already established
05H 16-bit protected mode interface already established
07H 32-bit protected mode interface already established
09H Unrecognized device ID
86H APM not present

Supported modes

Real mode

4.6.3 APM Protected Mode 16-bit Interface Connect (02H)

This call initializes the 16-bit protected mode interface between the APM Driver (caller) and the APM BIOS. This interface allows a protected mode caller to invoke the APM BIOS functions without first switching into real or virtual-86 mode. This function must be invoked in real mode.

Call With

AH = 53H APM
AL = 02H Protected mode 16-bit interface connect
BX = Power device ID

0000H APM BIOS

All other values reserved

Returns

If function successful:

Carry = 0
AX = APM 16-bit code segment (real mode segment base address)
BX = Offset of entry point into the APM BIOS
CX = APM 16-bit data segment (real mode segment base address)
SI = APM BIOS code segment length
DI = APM BIOS data segment length

If function unsuccessful:

Carry = 1
AH = Error code

02H Real mode interface connection already established

05H 16-bit protected mode interface connection already established

06H 16-bit protected mode interface not supported

07H 32-bit protected mode interface connection already established

09H Unrecognized device ID

86H APM not present

Supported modes

Real mode

Comments

The APM BIOS 16-bit protected mode interface requires two consecutive selector/segment descriptors for use as a 16-bit code and data segment, respectively. The caller must initialize these descriptors using the segment base and length information returned from this call to the APM BIOS. These selectors may either be in the GDT or LDT, and must be valid when the system APM BIOS is called in protected mode. The code segment descriptor must specify protection level 0, and the APM BIOS routines must be invoked with CPL = 0 so they can execute privileged instructions such as port inputs and outputs.

The APM caller invokes the APM BIOS routines using the 16-bit interface by making a far call to the code segment selector it initializes, and the offset returned in BX from this call. The caller must supply a stack large enough for use by the APM BIOS and potential interrupt handlers. The caller's stack will be active if or when interrupts are enabled in the APM BIOS routines; the BIOS will not switch stacks when interrupts are enabled, including NMI interrupts. The APM BIOS 16-bit protected mode interface must be called with a 16-bit stack.

When the APM BIOS routines are called in protected mode, the current I/O permission bit map must allow access to the I/O ports the BIOS may need to access in the process of performing the selected function.

4.6.4 APM Protected Mode 32-bit Interface Connect (03H)

This call initializes the 32-bit protected mode interface between the APM Driver (caller) and the APM BIOS. This interface allows a protected mode APM Driver to invoke the APM BIOS functions without the need to first switch into real or virtual-86 mode. This function must be invoked in real mode.

Call With

AH = 53H APM
AL = 03H Protected mode 32-bit interface connect
BX = Power device ID

0000H APM BIOS

All other values are reserved

Returns

If function successful:

Carry = 0
AX = PM 32-bit code segment (real mode segment base address)
EBX = Offset of the entry point into the APM BIOS
CX = APM 16-bit code segment (real mode segment base address)
DX = APM data segment (real mode segment base address)
ESI = APM BIOS 32-bit code segment length (low word of ESI)
APM BIOS 16-bit code
segment length (high word of ESI)
DI = APM BIOS data segment length

If function unsuccessful:

Carry = 1
AH = Error code

02H Real mode interface connection already established

05H 16-bit protected mode interface connection already established

07H 32-bit protected mode interface connection already established

08H 32-bit protected mode interface not supported

09H Unrecognized device ID

86H APM not present

Supported modes

Real mode

Comments:

The APM BIOS 32-bit protected mode interface requires 3 consecutive selector/segment descriptors for use as 32-bit code, 16-bit code, and data segments, respectively. Both 32-bit and 16-bit code segment descriptors are necessary so the APM BIOS 32-bit interface can call other BIOS routines in a 16-bit code segment if necessary. The caller must initialize these descriptors using the segment base and length information returned from this call to the APM BIOS. These selectors may either be in the GDT or LDT, but must be valid when the APM BIOS is called in protected mode. The code segment descriptors must specify protection level 0, and the APM BIOS routines must be invoked with CPL = 0 so they can execute privileged instructions such as port inputs and outputs.

The three segment lengths returned in DI and the high/low words of ESI must contain the lengths of their respective segments in bytes. The protected mode segment descriptors created for these segments will have a limit that is 1 less than the length. For example, a segment with length 0800H will be given a segment limit of 07FFH, allowing access to offsets 0 to 07FFH within the segment.

The APM caller invokes the APM BIOS routines using the 32-bit interface by making a far call to the 32-bit code segment selector it initializes, with the offset returned in EBX from this call. The caller must supply a stack large enough for use by the APM BIOS and potential interrupt handlers. The caller's stack will be active if or when interrupts are enabled in the APM BIOS routines; the BIOS will not switch stacks when interrupts are enabled, including NMI interrupts. The APM BIOS 32-bit protected mode interface must be called with a 32-bit stack.

When the APM BIOS routines are called in protected mode, the current I/O permission bit map must allow access to the I/O ports the BIOS may need to access in the process of performing the selected function.

4.6.5 APM Interface Disconnect (04H)

This call breaks the cooperative connection between the APM BIOS and the APM Driver (caller), and returns control of the power management policy to the APM BIOS. Power management parameter values (timer values, enable/disable settings, etc.) in effect at the time of the disconnect remain in effect.

Call With

AH = 53H APM
AL = 04H Interface disconnect
BX = Power device ID
0000H APM BIOS
All other values reserved

Returns

If function successful:
Carry = 0
If function unsuccessful:
Carry = 1
AH = Error code
03H Interface not connected
09H Unrecognized device ID

Supported modes

Real mode
Protected mode (16-bit and 32-bit)

4.6.6 CPU Idle (05H)

The APM Driver uses this call to tell APM BIOS that the system is idle. The APM BIOS suspends the system until the next system event (typically an interrupt) occurs. This function allows the APM BIOS to take some implementation-specific power saving action, such as a CPU HLT instruction or stopping the CPU clock.

When the APM Driver regains control from the APM BIOS idle routine, it should determine if there is any processing to be performed, and reissue the *CPU Idle* call if not. If the APM Driver is a multitasking supervisor, it may be necessary for it to dispatch its applications, allowing them to check for activity that they should now perform.

Call With

AH = 53H APM
AL = 05H CPU Idle

Returns

If function successful:

Carry = 0

If function unsuccessful:

Carry = 1

AH = Error code

03H Interface not connected

0BH Interface not engaged

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments:

Refer to section 4.4 CPU Usage Functions for information on the types of actions that the APM BIOS may take when a CPU Idle function is executed.

4.6.7 CPU Busy (06H)

This call informs the APM BIOS that the APM Driver has determined that the system is now busy. The APM BIOS restores the CPU clock rate to full speed.

This routine need only be called if the *CPU Idle* function was previously called, resulting in slowing of the CPU clock rate. To determine if an APM BIOS slows the clock rate during a *CPU Idle* call, check bit 2 of the CX register returned from the *APM Installation Check* call.

Calling *CPU Busy* when the system is already operating at full speed is discouraged due to the unnecessary call overhead, but the operation is allowed and has no unexpected side effects.

Call With

AH = 53H APM
AL = 06H CPU Busy

Returns

If function successful:

Carry = 0

If function unsuccessful:

Carry = 1

AH = Error code

03H Interface not connected

0BH Interface not engaged

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

4.6.8 Set Power State (07H)

This call sets the system or device specified in the power device ID into the requested power state.

Call With

AH = 53H APM

AL = 07H Set Power State

BX = Power device ID

0001H All devices power managed by the APM BIOS

01XXH Display

02XXH Secondary Storage

03XXH Parallel Ports

04XXH Serial Ports

05XXH Network Adapters

06XXH PCMCIA Sockets

E000H - EFFFH OEM-defined Power Device IDs

All other values reserved

where:

XX H = Unit Number (0 based)

Unit Number of FF H means all devices in this class

CX = Power state

*0000H	APM Enabled
0001H	Standby
0002H	Suspend
0003H	Off
**0004H	Last Request Processing Notification
**0005H	Last Request Rejected
0006H-001FH	Reserved system states
0020H-003FH	OEM-defined system states
0040H-007FH	OEM-defined device states
0080H-FFFFH	Reserved device states

* Not supported for Power Device ID 0001H

** Only supported for Power Device ID 0001H

Returns:

If function successful:

Carry = 0

If function unsuccessful:

Carry = 1

AH = Error code

01H Power management functionality disabled

03H Interface not connected

09H Unrecognized device ID

0AH Parameter value out of range

0BH Interface not engaged

60H Unable to enter requested state

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments:

If the *Set Power State* function is called to place the entire system in Standby mode (BX=0001h), the BIOS must take care not to take any actions which might inadvertently change the state of any devices which have had BIOS power management disabled using the *Disable Device Power Management* function. Otherwise an operating system driver doing power management of such a device might not be able to properly restore the device after resuming from the Standby.

However, when entering Suspend mode, the BIOS may take whatever actions are necessary to suspend the machine. It is assumed that an O/S driver doing device power management will be aware that power may have been completely removed from its device during a suspend operation.

When the *Set Power State* function is called with power device ID = 0001H to enter the global Standby or global Suspend states, the APM BIOS must not return to the APM Driver until one of the following conditions occurs:

1. The APM BIOS fails the set state request by returning with the Carry flag set and a valid error code in the AH register.
2. The system is leaving the requested state and is ready to resume normal processing. The APM BIOS must then return the appropriate power management resume event code to the next *Get Power Management Event* call.

3. The APM BIOS has entered the requested Standby State, and has slowed the clock. After a resume event has been detected, the APM BIOS must return the System Standby Resume Notification event code to the next *Get Power Management Event* call.

It is also important to note that APM BIOSes must **not** return the usual success sequence (carry clear followed by posting a resume event) if the requested power state was never actually entered. In this case they must return with carry set to indicate the failure to enter the low power state. Returning with the normal success sequence is likely to cause lengthy operations in the APM Driver and the layers above related to normal resume.

Set Power State entry codes of CX=0004h and CX=0005h are used by the APM Driver to respond to requests from the system BIOS for the global Standby and Suspend states. The APM Driver uses the Last Request Processing Notification (0004h) to indicate that it is currently in the process of determining whether or not to reject the request. This notification must be sent at least once every five seconds after the APM driver receives the request by calling *Get Power Management Event*. The APM driver must eventually end this "busy" state by accepting the request, (calling *Set Power State* with the appropriate state) or by rejecting the request using CX=0005h.

The APM Driver may not use either CX=0004H (Request processing) or CX=0005H (Request rejected) codes to respond to a Critical Suspend Notification.

It is possible that the BIOS may need to post a Critical Suspend Notification due to imminent power supply failure while the operating system is still working on a previous standby or suspend request. To this end, the APM Driver must continue to poll for power management events once per second even while processing a previous standby or suspend request. If a Critical Suspend Notification is received in this situation, the APM Driver has 5 seconds to call *Set Power State* to suspend the machine, and recover to the best extent possible when the system resumes. Note that the BIOS may not use Critical Suspend Notifications under any circumstances except for emergency low-power conditions.

The APM BIOS must allow the APM Driver to enter lower power states without the APM BIOS first requesting the state change. For example, the APM BIOS must allow the APM Driver to put the system in the global Suspend state even though the APM BIOS has not posted a System Suspend Request Notification based on system inactivity, hardware switches, or other logic.

Refer to Chapter 2. Advanced Power Management Model for more information on power states.

4.6.9 Enable/Disable Power Management (08H)

This call enables or disables all APM BIOS automatic power management. When disabled, the APM BIOS does not automatically power manage devices, enter the Standby State, enter the Suspend State, or take power saving steps in response to *CPU Idle* calls.

Call With

AH = 53H APM
AL = 08H Enable/disable power management
BX = Power device ID

0001H All devices power managed by the APM BIOS

*FFFFH All devices power managed by the APM BIOS

All other values reserved

* Must be implemented for compatibility with APM 1.0

CX = Function code

0000H Disable power management

0001H Enable power management

Returns

If function successful:
Carry = 0
If function unsuccessful:
Carry = 1
AH = Error code

03H Interface not connected

09H Unrecognized device ID

0AH Parameter value out or range (function code)

0BH Interface not engaged

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments:

The APM Driver must never cause the APM BIOS to be both Disabled and Disengaged at the same time. Any transition from “Disabled” to “Disengaged,” or vice versa, must pass through the Enabled and Engaged state first. If the APM driver attempts to Disable Power Management, and the interface is currently Disengaged, the

APM BIOS should return error code 0Bh to indicate that power management is disengaged. See section 4.6.16 for details.

4.6.10 Restore APM BIOS Power-On Defaults (09H)

This call reinitializes all power-on defaults.

Call With

AH = 53H APM
AL = 09H Restore Power-On Defaults
BX = Power device ID

0001H All devices power managed by the APM BIOS

*FFFFH All devices power managed by the APM BIOS

All other values reserved

* Must be implemented for compatibility with APM 1.0

Returns

If function successful:
Carry = 0

If function unsuccessful:

Carry = 1

AH = Error code

03H Interface not connected

09H Unrecognized device ID

0BH Interface not engaged

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments:

This function should set all devices to the disengaged state, and the system as a whole to the engaged state.

4.6.11 Get Power Status (0AH)

This call returns the system current power status.

Call With

AH = 53H APM
AL = 0AH Get Power Status
BX = Power device ID

0001H APM BIOS, or
80XXH Specific Battery Unit Number
where:
XX = Battery Unit Number (1 based)

All other values reserved

Returns

If function successful:

Carry = 0
BH = AC line status

00H Off-line
01H On-line
02H On backup power
FFH = Unknown
All other values reserved

BL = Battery status

00H High
01H Low
02H Critical
03H Charging
FFH = Unknown
All other values reserved

CH = Battery flag

bit 0 = 1 High
bit 1 = 1 Low
bit 2 = 1 Critical
bit 3 = 1 Charging
bit 4 = 1 Selected battery not present
bit 7 = 1 No system battery
All other bits reserved (must be set to 0)
FFH = Unknown
All other values reserved

CL = Remaining battery life - percentage of charge

0-100 = Percentage of full charge
FFH = Unknown
All other values reserved

DX = Remaining battery life - time units

Bit 15 = 0 Time units are seconds

1 Time units are minutes

Bits 14-0 = Number of seconds or minutes

0-7FFFH Valid number of seconds

0-7FFEH Valid number of minutes

FFFFH = Unknown

If BH (Power Device ID/Device Class) = 80H on entry, then return:

SI = Number of battery units currently installed in the machine (0 to *n*). Must be <= the maximum number of battery units supported by this machine (returned in BX by the *Get Capabilities* function.) There is no “unknown” return value for this parameter.

If function unsuccessful:

Carry = 1

AH = Error code

09H Unrecognized device ID

0AH Parameter value out of range

86H APM not present

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments:

The battery flag (CH) return status field communicates both the battery status level (low, high, critical) as well as whether or not the battery is currently being charged. The battery status field (BL) communicates the current status of the given battery.

APM versions 1.0 and 1.1 do not directly support multiple battery units. APM 1.0 and 1.1 compatible drivers and applications should always call the *Get Power Status* function with BX = 0001H because that was the only power device ID allowed by those specifications. On systems that support multiple battery units, the APM BIOS must return an aggregate of all currently installed battery units when called to report the status for power device ID = 0001H.

An APM 1.2-capable APM driver can use the *Get Capabilities* function to determine the number of battery units supported by the system. The driver can then request the status of each individual battery by calling the *Get Power Status* function with BX = 80XXH where XXH is the 1 based battery number. For example, a power device ID of 8001H returns the status of the first battery unit while a power device ID of 8002H returns the status of the second battery unit.

If the APM BIOS is called with a power device ID set to a valid battery unit number, but the corresponding battery is not currently installed in the machine, the BIOS must return battery status (BL) = FFH (unknown), battery flag (CH) = 10H (selected battery not present), CL = FFH (remaining % unknown), and DX = FFFFH (remaining time

unknown). The APM BIOS must not “shift” the position of battery units as batteries are added or removed from the system. For example, if a machine supports 2 battery units, and a battery is only present in position 2, the APM BIOS must report CH = 10H (selected battery not present) for power device ID 8001H and valid battery information for power device ID 8002H.

If the APM BIOS is called with a power device ID set to a battery unit number, but the corresponding battery is not supported (i.e. there is no such battery socket), the APM BIOS must return an error, setting the Carry flag and AH = 09H (unrecognized device ID).

When connected to an APM 1.1 or later driver, a 1.2-level APM BIOS must post a Power Status Change Notification event whenever the number of currently installed battery units changes. APM 1.1 drivers may want to update their statistics even though they are not directly aware of the multiple-battery capabilities of the particular system.

When called with power device ID = 0001H, the APM BIOS may return with BH=02H (On backup power) if the power status values apply to the system’s backup battery. Since APM 1.0 did not define BH=02H (On backup power), an APM 1.2 BIOS that is running in APM 1.0 BIOS mode (i.e. no APM Driver Version call was issued by the APM Driver to set the APM BIOS mode to 1.1 or 1.2) must return BH=01H (Off-line) instead of BH=02H when the power status values apply to the system’s backup battery.

For all *Get Power Status* calls that are not failed with an error code, the APM BIOS must return valid AC line status in BH, regardless of the specific power device ID selected.

When BH=80H, the APM BIOS must return the number of currently installed battery units in SI.

If the computer does not support batteries (e.g. desktops), and the APM Driver calls the BIOS with BX=0001, BIOS must return the AC line status (BH) = 01H (on-line), battery status (BL) = FFH (unknown), battery flag (CH) = 80H (no system battery), CL = FFH (remaining % unknown), DX = FFFFH (remaining time unknown), and the number of batteries (SI) = 0. Also, Get Capabilities should indicate that the number of batteries is 0.

4.6.12 Get PM Event (0BH)

Get PM Event returns the next pending PM event, or indicates if no PM events are pending. This function should be called until there are no more pending PM events, that is, an error is returned. Then, the APM Driver will poll for new events at least once per second. Events apply to the APM System or to a device.

Call With

AH = 53H APM
AL = 0BH Get PM Event

Returns

If function successful:

Carry = 0
BX = PM event code
Refer to the Section *Power Management Events*
CX = PM event information
When BX = 0003H or 0004H (Normal Resume System Notification or Critical Resume System Notification):
Bit 0 = 0 PCMCIA socket was powered on in the Suspend state
Bit 0 = 1 PCMCIA socket was powered off in the Suspend state
All other bits reserved (must be set to 0)

If function unsuccessful:

Carry = 1
AH = Error code
03H Interface not connected
0BH Interface not engaged
80H No power management events pending

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments:

When the returned event code is 0003H, Normal Resume System Notification, the low bit of the CX register indicates whether or not the PCMCIA socket was powered off during the Suspend state from which the system is resuming. A value of one in the low bit indicates that the PCMCIA socket was powered off.

Knowing whether the socket was powered on or off during the Suspend State allows the APM Driver or Operating System to optimize its resume handling of PCMCIA devices. This can be particularly important if a PCMCIA modem device is used to resume the system via the resume-on-ring-indicator feature.

4.6.13 Get Power State (0CH)

This call returns the device power state when a specific device ID is used. The power state value returned when the Power device ID specified indicates "all devices power managed by the APM BIOS" or "all devices in a class" is defined only when that Power device ID has been used in a call to *Set Power State*. In the case where the Power device ID (class value) has not been used in a call to *Set Power State*, the function will be unsuccessful, and will return error code 9 (Unrecognized device ID).

Call With

AH = 53H APM
AL = 0CH Get Power State
BX = Power device ID

0001H All devices power managed by the APM BIOS

01XXH Display

02XXH Secondary Storage

03XXH Parallel Ports

04XXH Serial Ports

05XXH Network Adapters

06XXH PCMCIA Sockets

E000H - EFFFH OEM-defined Power Device IDs

All other values reserved

where:

XX H = Unit Number (0 based)

Unit Number of FF H means all devices in this class.

Returns

If function successful:

Carry = 0
CX = Power state

0000H	APM Enabled
0001H	Standby
0002H	Suspend
0003H	Off
0004H-001FH	Reserved system states
0020H-003FH	OEM-defined system states
0040H-007FH	OEM-defined device states
0080H-FFFFH	Reserved device states

If function unsuccessful:

Carry	=	1	
AH	=	Error code	
		01H	Power management functionality disabled
		09H	Unrecognized device ID
		86H	APM not present

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments:

The *Get Power State* call can be used to determine whether BIOS power management is currently enabled for a device. The *Get Power State* call will return error code 1 (Power management functionality disabled) if BIOS power management is disabled for the specified device.

4.6.14 Enable/Disable Device Power Management (0DH)

This call enables or disables APM BIOS automatic power management for a specified device. When disabled, the APM BIOS does not automatically power manage the device.

Upon system reset, any supported device's enabled/disabled status should initially be enabled. These devices can also be returned to their initial states by using the Restore APM BIOS Power-On Defaults function.

Call With

AH = 53H APM
AL = 0DH Enable/disable device power management
BX = Power device ID
0001H All devices power managed by the APM BIOS
01XXH Display
02XXH Secondary Storage
03XXH Parallel Ports
04XXH Serial Ports
05XXH Network Adapters
06XXH PCMCIA Sockets
E000H - EFFFH OEM-defined Power Device IDs
All other values reserved
where:
XX H = Unit Number (0 based)
Unit Number of FF H means all devices in this class.

CX = Function code
0000H Disable power management
0001H Enable power management

Returns

If function successful:
Carry = 0
If function unsuccessful:
Carry = 1
AH = Error code
01H Power management functionality disabled
03H Interface not connected
09H Unrecognized device ID

0AH Parameter value out or range (function code)

0BH Interface not engaged

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

4.6.15 APM Driver Version (0EH)

The APM Driver uses this call to indicate its level of APM support to the APM BIOS. The APM BIOS returns the APM connection version number.

Call With

AH = 53H APM
AL = 0EH APM Driver Version
BX = 0000H APM BIOS

CH = APM Driver major version number (in BCD format)

CL = APM Driver minor version number (in BCD format)

Returns

If function successful:

Carry = 0
AH = APM Connection major version number (in BCD format)
AL = APM Connection minor version number (in BCD format)

If function unsuccessful:

Carry = 1
AH = Error code
03H Interface not connected
09H Unrecognized device ID
0BH Interface not engaged

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments

This call must be made by an APM version 1.1 (or greater) Driver after a connection has been established between an APM Driver and APM BIOS. An APM version 1.0 Driver will not make this call.

An APM version 1.0 BIOS should return Carry=1 from this function.

An APM version 1.2 Driver issues this call with CH = 1 and CL = 2 to select APM version 1.2 BIOS functionality. An APM version 1.2 BIOS will return Carry=0 from this function, and switch (from APM version 1.0) to APM version 1.2 functionality.

See Appendix C, *APM 1.0/1.1/1.2 Modal BIOS Behavior*, for discussion of how the BIOS should handle differences in APM Driver level.

4.6.16 Engage/Disengage Power Management (0FH)

This call engages or disengages cooperative power management of the system or device. When disengaged, the APM BIOS automatically power manages the system or device.

Any supported device's engaged/disengaged status should initially be set to disengaged upon system reset, or by using the *Restore APM BIOS Power-On Defaults* function. The system as a whole should be engaged.

Call With

AH = 53H APM
AL = 0FH Engage/disengage power management
BX = Power device ID
0001H All devices power managed by the APM BIOS
01XXH Display
02XXH Secondary Storage
03XXH Parallel Ports
04XXH Serial Ports
05XXH Network Adapters
06XXH PCMCIA Sockets
E000H - EFFFH OEM-defined Power Device IDs
All other values reserved
where:
XX H = Unit Number (0 based)
Unit Number of FF H means all devices in this class.
CX = Function code
0000H Disengage power management
0001H Engage power management

Returns

If function successful:
Carry = 0
If function unsuccessful:
Carry = 1
AH = Error code
01H Power management functionality disabled
03H Interface not connected
09H Unrecognized device ID

0AH Parameter value out or range (function code)

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments:

The Engage/Disengage Power Management function disables only the cooperative APM BIOS/APM Driver system power management. The Enable/Disable Power Management function disables the automatic system power management provided by the BIOS. When Power Management is disabled, the system is neither engaged nor disengaged, calls to Engage/Disengage power management should return error 01H (Power management functionality disabled).

The Engage/Disengage Power Management function provides the ability to stop the cooperative power management between the APM BIOS and APM Driver without having to disconnect APM (APM Interface Disconnect). This is useful in protected mode systems in which cooperative power management is selectable. The problem with disconnecting and then reconnecting in these systems is that APM connect calls must be issued in real mode.

The behavior of Enable/Disable and Engage/Disengage is shown in the following state diagram:

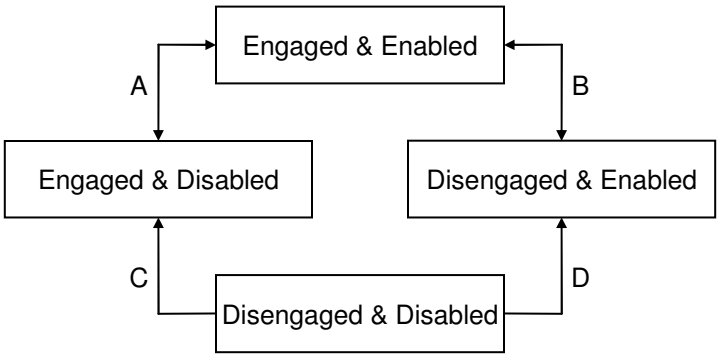

Figure 4. Engage/Disengage, Enable/Disable States

The Engage/Disengage Power Management function performs transitions B and C. The Enable/Disable Power Management function performs transitions A and D. Notice that in performing transitions A and D, the Enable/Disable Power Management function must maintain the state of engage vs disengage.

The APM Driver must never cause the APM BIOS to be both Disabled and Disengaged at the same time. As shown above, any transition from “Disabled” to “Disengaged,” or vice versa, must pass through the Enabled and Engaged state first.

If the APM driver attempts to Disengage and the interface is currently Disabled, the APM BIOS should return error code 01h to indicate that power management is disabled. Thus, transitions C and D above are one-way.

4.6.17 Get Capabilities (10H)

This call returns the features which this particular APM 1.2 BIOS implementation supports.

Call With

AH = 53H APM
AL = 10H Get Capabilities
BX = Power device ID
0000H APM BIOS
All other values reserved

Returns

If function successful:

Carry = 0 APM is supported by BIOS
BL = Number of battery units this machine supports. A value of 0 indicates this machine does not have any system batteries. Most systems will return 0, 1, or 2.
CX = Capability flags
Bit 0 = 1 System can enter global standby state. Indicates BIOS will post standby and standby-resume events.
Bit 1 = 1 System can enter global suspend state. Indicates BIOS will post suspend and suspend-resume events.
Bit 2 = 1 Resume timer will wake up from standby.
Bit 3 = 1 Resume timer will wake up from suspend.
Bit 4 = 1 Resume on ring indicator (internal COM or modem) will wake up from standby.
Bit 5 = 1 Resume on ring indicator (internal COM or modem) will wake up from suspend.
Bit 6 = 1 PCMCIA Ring indicator will wake up from standby.
Bit 7 = 1 PCMCIA Ring indicator will wake up from suspend.
Other bits Reserved (must be set to 0)

If function unsuccessful:

Carry = 1
AH = Error code
09H Unrecognized device ID
86H APM not present

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments

The *Get Capabilities* function does not require an APM connection.

Capabilities may change as a result of changes via a setup utility or by the arrival or removal of devices. If power capabilities change, the APM BIOS must post a Capabilities Change Notification. Upon receiving the Capabilities Change Notification, the APM Driver will call Get Capabilities and Get Power Status to determine the new capabilities. Calls to the *Get Capabilities* function must return the capabilities currently in effect, and not capabilities which have been selected but require a system restart before taking effect.

An APM BIOS that supports any of the global standby, or global suspend power states must return from the *Get Capabilities* call with the corresponding capability bit(s) set. The BIOS must post the appropriate request and resume events when entering and leaving the supported power states. In addition, the BIOS must allow the APM driver to enter any of the supported power states by use of the APM *Set Power State* function without the APM BIOS having first requested the state change via the Get Power Management Event function.

If a system does not support a resume timer, the APM BIOS must not set any of the resume timer capability bits (bits 2, 3,). Similarly, if the system does not support resume on ring indicator, the APM BIOS must not set any of the resume on ring indicator capability bits (bits 4, 5, 6, 7).

If the computer does not support batteries, it must return BL=0. Get Power Status must also return the appropriate values. See section 4.6.11 for details.

4.6.18 Get/Set/Disable Resume Timer (11H)

This call gets, sets, or disables the system resume timer.

Call With

AH = 53H
AL = 11H
BX = Power device ID
0000H APM BIOS
All other values reserved
CL = Function code
00H Disable Resume Timer
01H Get Resume Timer
02H Set Resume Timer
If CL = 02H: (Set resume timer values)
CH = Seconds (0 to 59) (in BCD format)
DH = Hours (0 to 23) (in BCD format)
DL = Minutes (0 to 59) (in BCD format)
SI = Month (high byte, 1 to 12 in BCD), Day (low byte, 1 to 31 in BCD)
DI = Year (1995 to ... in BCD format)

Returns:

If function successful:
Carry = 0
if CL = 01H (Get resume timer values)
CH = Seconds (0 to 59) (in BCD format)
DH = Hours (0 to 23) (in BCD format)
DL = Minutes (0 to 59) (in BCD format)
SI = Month (high byte, 1 to 12 in BCD), Day (low byte, 1 to 31 in BCD)
DI = Year (1995 to ... in BCD format)
If function unsuccessful:
Carry = 1
AH = Error code
03H Interface not connected
09H Unrecognized device ID
0AH Parameter value out of range
0BH Interface not engaged
0CH Function not supported
0DH Resume timer disabled (valid for get resume timer value only)

Supported modes

Real mode

Protected mode (16-bit and 32-bit)

Comments:

The APM Driver can use the resume timer to ensure the system is in an operational state at a specific date and time. The APM BIOS must resume from the global Standby or global Suspend state at the specified time provided the resume timer is supported for that state. The APM BIOS indicates which states support the resume timer via flags returned by the *Get Capabilities* function.

The APM BIOS must enable the resume timer on any acceptable *Get/Set/Disable Resume Timer* call to set the resume timer. If an APM Driver disables the resume timer via this function and then disconnects from the APM BIOS, the APM BIOS must not re-enable the resume timer simply because the APM driver has disconnected. Otherwise it would be possible that the machine would spontaneously power-on and reboot with no specific task to be executed, leading to unnecessary power consumption.

4.6.19 Enable/Disable Resume on Ring Indicator (12H)

This call enables or disables the system's resume on ring indicator functionality.

Call With

AH = 53H APM
AL = 12H Enable/Disable Resume on Ring Indicator
BX = Power device ID
0000H APM BIOS
All other values reserved
CX = Function code
0000H Disable Resume on Ring Indicator
0001H Enable Resume on Ring Indicator
0002H Get Enabled/Disabled status
All other values are reserved

Returns:

If function successful:
Carry = 0
CX = Enabled/Disabled status
0000H Disabled
0001H Enabled

If function unsuccessful:
Carry = 1
AH = Error code
03H Interface not connected
09H Unrecognized device ID
0AH Parameter value out of range
0BH Interface not engaged
0CH Function not supported

Supported modes

Real mode
Protected mode (16-bit and 32-bit)

Comments

The APM BIOS must disable the resume on ring indicator feature upon a call to *APM Driver Version* function which indicates a 1.2 or greater capable APM driver, a call to the *Enable/Disable Resume on Ring Indicator* function with CX=0000H, or a call to the *Restore APM BIOS Power-On Defaults* function. This function should enable or disable all resume-on-ring functionality, regardless of whether the ring indication

comes from motherboard serial ports or modems, modems inserted into PCMCIA slots, or any other modem ring which the system is wired to accept as a wakeup event.

If the APM Driver calls the BIOS with CX=0002, the BIOS should return the current state of whether this feature is enabled or disabled in CX.

4.6.20 Enable/Disable Timer Based Requests (13H)

This call enables or disables the APM BIOS's generation of global Standby and global Suspend requests based on inactivity timers.

Call With

AH = 53H APM
AL = 13H Enable/Disable Timer Based Requests
BX = Power device ID
0000H APM BIOS
All other values reserved
CX = Function code
0000H Disable Timer Based Requests
0001H Enable Timer Based Requests
0002H Get Enabled/Disabled status
All other values are reserved

Returns:

If function successful:
Carry = 0
CX = Enabled/Disabled status
0000H Disabled
0001H Enabled

If function unsuccessful:
Carry = 1
AH = Error code
03H Interface not connected
09H Unrecognized device ID
0AH Parameter value out of range
0BH Interface not engaged

Supported modes

Real mode
Protected mode (16-bit and 32-bit)

Comments

When the system is powered up, or when the *Restore APM BIOS Power-On Defaults* function is called, the BIOS is allowed to generate Standby and Suspend requests based on the BIOS' own determination of system activity. When this function is called with CX set to 00H, the APM BIOS must not generate any more Standby or Suspend requests based on inactivity timers. This is desirable for the operating system

to use its own techniques to determine inactivity and to generate such low power requests on its own.

Once disabled, the APM BIOS may still post normal System or User Standby or Suspend requests only in the event of a user-initiated action, such as lid closings, suspend button pushes, etc. Critical suspend requests due to imminent battery failure are also still allowed.

If the APM Driver calls the BIOS with CX=0002, the BIOS should return the current state of whether this feature is enabled or disabled in CX.

4.7 OEM-defined APM Functions

4.7.1 OEM APM Installation Check (80H)

This call checks to see if the APM BIOS supports OEM hardware-dependent functions. To get an OEM ID, refer to the information on page iii.

Call With

AH = 53H APM
AL = 80H OEM APM
BH = 7FH Installation check

Returns:

If function successful:

Carry = 0

BX = OEM ID

All other registers carry OEM-specific information

If function unsuccessful:

Carry = 1

AH = Error code (OEM-defined error codes)
= 03H Interface not connected

All other registers carry OEM-specific information

Supported modes

Real mode

Protected mode (16-bit and 32-bit) Optional

4.7.2 OEM APM Function (80H)

This function accesses OEM product-specific APM functions.

Call With:

AH	=	53H	APM
AL	=	80H	OEM APM
BH	=	00H-7EH, 80H-FFH	OEM-defined function code

All other registers carry OEM-specific information

Returns:

If function successful:

Carry = 0

All other registers carry OEM specific information

If function unsuccessful:

Carry = 1

AH = Error code (OEM-defined error codes)

03H Interface not connected

All other registers carry OEM-specific information

Supported modes

Real mode

Protected mode (16-bit and 32-bit) Optional

Appendix A - APM Function Summary

Function	Int 15	16 Bit	32 Bit	Requires Connection
APM Installation Check (00h)	XXX			No
APM Real Mode Interface Connect (01h)	XXX			No
APM Protected Mode Connect 16 bit (02h)	XXX			No
APM Protected Mode Connect 32 bit (03h)	XXX			No
APM Interface Disconnect (04h)	XXX	XXX	XXX	Yes
CPU Idle (05h)	XXX	XXX	XXX	Yes
CPU Busy (06h)	XXX	XXX	XXX	Yes
Set Power State (07h)	XXX	XXX	XXX	Yes
Enable/Disable Power Management (08h)	XXX	XXX	XXX	Yes
Restore Power on Defaults (09h)	XXX	XXX	XXX	Yes
Get Power Status (0Ah)	XXX	XXX	XXX	No
Get PM Event (0Bh)	XXX	XXX	XXX	Yes
Get Power State(0Ch)	XXX	XXX	XXX	No
Enable/Disable Device Power Management (0Dh)	XXX	XXX	XXX	Yes
APM Driver Version (0Eh)	XXX	XXX	XXX	Yes
Engage/Disengage Power Management (0Fh)	XXX	XXX	XXX	Yes
Get Capabilities (10H)	XXX	XXX	XXX	No
Get/Set/Disable Resume Timer (11H)	XXX	XXX	XXX	Yes
Enable/Disable Resume on Ring Indicator (12H)	XXX	XXX	XXX	Yes
Enable/Disable Timer Based Requests (13H)	XXX	XXX	XXX	Yes
OEM APM Function(80h)	XXX	Optional	Optional	Yes

Appendix B - Firmware Error Codes

Error Code Groupings:

Interface and general errors	00H - 1FH
CPU errors	20H - 3FH
Device errors	40H - 5FH
System errors	60H - 7FH
Power Management event errors	80H - 9FH
Reserved errors	A0H - FEH
Undefined function	FFH

Interface & General Errors

Error Number	APM Function Error Messages	Applicable APM Calls
01h	Power Management functionality disabled	Enable/Disable Device Power Management Engage/Disengage Power Management Get Power State Set Power State
02	Real Mode interface connection already established	APM Protect Mode 16-bit Interface Connect APM Protect Mode 32-bit Interface Connect APM Real Mode Interface Connect
03h	Interface not connected	APM Driver Version APM Interface Disconnect CPU Busy CPU Idle Enable/Disable Device Power Management Enable/Disable Power Management Enable/Disable Resume on Ring Indicator Enable/Disable Timer Based Requests Engage/Disengage Power Management Get PM Event Get/Set/Disable Resume Timer OEM APM Function Restore APM BIOS Power-on Defaults Set Power State
05h	16-bit protected mode interface already established	APM Protect Mode 16-bit Interface Connect APM Protect Mode 32-bit Interface Connect APM Real Mode Interface Connect

06h	16-bit protected mode interface not supported	APM Protect Mode 16-bit interface connect
07h	32-bit protected mode interface already established	APM Protect Mode 16-bit Interface Connect APM Protect Mode 32-bit Interface Connect APM Real Mode Interface Connect
08h	32-bit protected mode interface not supported	APM Protect Mode 32-bit Interface Connect
09h	Unrecognized Device ID	APM Driver Version APM Installation Check APM Interface Disconnect APM Protect Mode 16-bit Interface Connect APM Protect Mode 32-bit Interface Connect APM Real Mode Interface Connect Enable/Disable Device Power Management Enable/Disable Power Management Enable/Disable Resume on Ring Indicator Enable/Disable Timer Based Requests Engage/Disengage Power Management Get Capabilities Get Power State Get Power Status Get/Set/Disable Resume Timer Restore APM BIOS Power-On Defaults Set Power State
0Ah	Parameter value out of range	Enable/Disable Device Power Management Enable/Disable Power Management Enable/Disable Resume on Ring Indicator Enable/Disable Timer Based Requests Engage/Disengage Power Management Get Power Status Get/Set/Disable Resume Timer Set Power State
0Bh	Interface not engaged	APM Driver Version CPU Busy CPU Idle Enable/Disable Device Power Management Enable/Disable Power Management Enable/Disable Resume on Ring Indicator Enable/Disable Timer Based Requests Get PM Event Get/Set/Disable Resume Timer Restore APM BIOS Power-on Defaults Set Power State
0Ch	Function not supported	Enable/Disable Resume on Ring Indicator Get/Set/Disable Resume Timer
0Dh	Resume timer disabled	Get/Set/Disable Resume Timer

System Errors

Error Number	APM Function Error Messages	Applicable APM Calls
60h	Unable to enter requested state	Set Power State

Power Management Event Errors

Error Number	APM Function Error Messages	Applicable APM Calls
80h	No power management events pending	Get PM event
86h	Reserved - APM not present	APM Installation Check APM Protected Mode Connect 16 bit APM Protected Mode Connect 32 bit APM Real Mode Interface Connect Get Capabilities Get Power State Get Power Status

Appendix C - APM BIOS Considerations

APM 1.0 Compatibility

This is a list of functions that must be supplied for an APM 1.1 BIOS to be compatible with APM 1.0. APM 1.0 BIOS functionality is required if the BIOS is to be used with an APM Driver written for APM 1.0.

- *APM Installation Check*
- *APM Interface Disconnect*
- *APM Real Mode Interface Connect/APM Protected Mode 16-bit Interface Connect/APM Protected Mode 32-bit Interface Connect*
- *CPU Busy*
- *CPU Idle*
- *Enable/Disable Power Management*. The *Enable/Disable Power Management* function must accept FFFFh as the Power Device ID parameter.
- *Get PM Event*
- *Get Power Status*
- *Restore APM BIOS Power-On Defaults*. The *Restore APM BIOS Power-On Defaults* function must accept FFFFh as the Power Device ID parameter.
- *Set Power State*

Differences between APM 1.0 and APM 1.1

- The *APM Driver Version* function has been added.
- The *APM Installation Check* function has been updated to return 1 for the APM minor version number in AL.
- The *APM Protected Mode 16-bit Interface Connect* function has been updated to return the segment extent lengths for the APM BIOS code segment and data segment.
  The list of error codes that can be returned from the function has been updated to include 02H (real mode interface connection already established), and 07H (32-bit protected mode interface connection already established).
- The *APM Protected Mode 32-bit Interface Connect* function has been updated to return the segment extent lengths for the APM BIOS code segment and data segment.
  The list of error codes that can be returned from the function has been updated to include 02H (real mode interface connection already established), and 05H (16-bit protected mode interface connection already established).

- The list of error codes that can be returned from the *APM Real Mode Interface Connect* function has been updated to include 05H (16-bit protected mode interface already established) and 07H (32-bit protected mode interface already established).
- The list of error codes that can be returned from the *CPU Busy* function has been updated to include 03H (Interface not connected).
- The list of error codes that can be returned from the *CPU Idle* function has been updated to include 03H (Interface not connected).
- Added the *Enable/Disable Device Power Management* function.
- Updated the *Enable/Disable Power Management* function to take 0001H as its Power Device ID parameter.
- Added the *Engage/Disengage Power Management* function. Added the 0BH (Interface not engaged) error code.
- Updated the *Get PM Event* function to return Power Status Change Notification, Update Time Notification, Critical System Suspend Notification, User System Standby Request Notification, User System Suspend Request Notification, and System Standby Resume Notification power management events.
- Added the *Get Power State* function.
- Updated the *Get Power Status* function to return the remaining battery life in seconds or minutes.
  Added the return of battery flag information in CH.
  Added the AC line status (BH) code 02H, On backup power.
- Updated the *Restore APM BIOS Power-On Defaults* function to take 0001H as its Power Device ID parameter.
- Updated the *Set Power State* function to allow the setting of the system power state (Power Device ID 0001H) to Off.
- Updated the *Set Power State* function to create a positive feedback system whereby the APM Driver notifies the BIOS at least once every five seconds after receiving a standby or suspend request that the Driver is processing the request. The "busy" state is ended by setting the appropriate power state, or by rejecting the request. "Rejection" and "Busy" are now explictly defined as new entry parameters to *Set Power State*.
- Created a minimum polling requirement whereby the APM Driver must poll for power management events at least once per second under all circumstances.
- Added a restriction that the BIOS may not suspend or standby of its own accord unless the operating system fails to meet its minimum polling or "busy" notification requirements.
- Added the restriction that all reserved bits must be set to 0.
- Added the *OEM-defined APM function* definition.

Differences between APM 1.1 and 1.2

- *Get Capabilities* function added.
- *Get/Set/Disable Resume Timer* function added.
- *Enable/Disable Resume on Ring Indicator* function added.
- Added a new return parameter from the *Get PM Event* function, valid when the PM Event returned is Normal Resume System Notification, to indicate whether the PCMCIA sockets were powered on or off during the Suspend state.
- *Enable/Disable Timer Based Events* function added.

- *Get Power Status* function enhanced to support multiple system batteries.
- Added Power Device IDs for batteries.
- *Power Status Change* Notification must be sent when the number of currently installed battery units changes.
- Requirement added that APM BIOS must not return from *Set Power State* function until either the BIOS fails the attempt, or the complete suspend/resume cycle has been completed.
- Requirement added that APM BIOS must accept *Set Power State* calls from the APM Driver which were not initiated by the APM BIOS itself.
- Stopping the clock for multiple timer ticks upon receiving the *CPU Idle* call is no longer permitted. Only single clock tick stoppage or clock slowing is permitted.
- Both 16-bit and 32-bit protect mode interface implementation is now required.
- 32-bit Interface Connect return parameters changed. ESI indicates BIOS code segment lengths.
- Clarifications to intended behavior of the *Engage/Disengage Power Management* function.
- Updated Appendix A function summary.
- Updated Appendix B error codes. Added error codes 0Ch to indicate function not supported, and 0Dh to indicate resume timer disabled.
- Updated the “APM System State” table
- Renamed the “Interrupt” System Power State transition to “Resume Event”
- Updated “Battery Low Notification” and “Update Time Notification” PM event descriptions.
- Added “Capabilities Change Notification” PM event.
- Updated “APM Installation Check” and “APM Driver Version” functions for APM 1.2.
- Updated the “Restore APM BIOS Power-On Defaults” to include the description of initial engaged/disengaged state.
- Updated CPU Usage Functions section (4.4).

APM 1.2 Requirements

The following functions must be provided by an APM 1.2 BIOS.

- *APM Protected Mode 16-bit Interface Connect*
- *APM Protected Mode 32-bit Interface Connect*
- *APM Driver Version*
- *APM Installation Check*
- *APM Interface Disconnect*
- *APM Real Mode Interface Connect*
- *CPU Busy*
- *CPU Idle*
- *Enable/Disable Device Power Management*
- *Enable/Disable Power Management*

- *Engage/Disengage Power Management*
- *Get Capabilities*
- *Get PM Event*
- *Get Power State*
- *Get Power Status*
- *Restore APM BIOS Power-On Defaults*
- *Set Power State*

The following functions may optionally be provided by an APM 1.2 BIOS.

- *OEM-defined APM Functions*
- *Get/Set/Disable Resume Timer*
- *Enable/Disable Resume on Ring Indicator*
- *Enable/Disable Timer Based Requests*

APM 1.0/APM 1.1/APM 1.2 Modal BIOS Behavior

When an APM Driver connects with an APM 1.2 BIOS , the APM 1.2 BIOS will default to an APM 1.0 connection. After an APM Driver calls the APM Driver Version function, specifying that it supports APM 1.2, an APM 1.2 BIOS will change its behavior to an APM 1.2 connection. If the APM BIOS is an APM 1.0 or APM 1.1 BIOS, the *APM Driver Version* function call will fail, and the connection will remain an APM 1.0 connection. The APM Driver may in such cases revert to reporting that it is an APM 1.1 Driver.

If an APM 1.1 Driver connects with an APM 1.2 BIOS, the APM BIOS must provide only those functions, return values, and error codes provided in the APM 1.1 specification.

The following events should be posted by an APM BIOS only when at least an APM 1.1 connection exists:

Power Status Change
Update Time Notification
Critical System Suspend Notification
User System Standby Request Notification
User System Suspend Request Notification
System Standby Resume Notification

An APM Driver should make the following calls only when at least an APM 1.1 connection exists:

Get Power State
Enable/Disable Device Power Management
Engage/Disengage Power Management

The following events should be posted by an APM BIOS only when at least an APM 1.2 connection exists:

Capabilities Change Notification

An APM Driver should make the following calls only when an APM 1.2 connection exists:

Get Capabilities
Get/Set/Disable Resume Timer
Enable/Disable Resume on Ring Indicator
Enable/Disable Timer Based Requests

Appendix D - APM Driver Considerations

The APM Driver connects with the APM BIOS. After a connection is established, the APM Driver and APM BIOS cooperatively perform power management.

When an APM connection exists, the APM BIOS transitions into System Standby and System Suspend states only when directed to do so by a call from the APM Driver. The calls to change system states are invoked by the APM Driver either based on its own logic, or after the APM BIOS indicates that the state transition should be made, and subsequently the APM Driver has checked with all APM-aware applications to make sure that it is an appropriate time to change system states. However, there are three cases where the APM BIOS may make the system state transition itself. The first case is if the APM Driver does not pick up a posted Standby Request, Suspend Request or Critical Suspend Notification event within the 2 second (one second plus one second grace period) time limit. The second is when the APM Driver, after picking up the event, does not respond to a Standby Request, Suspend Request or Critical Suspend Notification event with an appropriate Set Power State call within 5 seconds of receiving the event. The last situation is when the APM Driver has responded to an event with a Request Processing Set Power call and does not reply again within another 5 seconds. The CPU is power managed according to *CPU Idle* and *CPU Busy* calls made by the APM Driver to the APM BIOS.

APM Driver Requirements

APM Driver power management event processing.

An APM Driver uses the *Get PM Event* APM call to see if a power management event has occurred. When a power management event is detected, the APM Driver processes that event as follows:

(User) System Standby Request Notification: The APM Driver notifies all APM-aware applications of the request. Each APM-aware application is given the opportunity to accept or reject the request. If all APM-aware applications accept the request, the APM Driver calls the *Set Power State* function to set the System Power State to Standby. Otherwise the same function is used to reject the request. If the request is rejected, the system will not transition into the Standby state.

(User) System Suspend Request Notification: The APM Driver notifies all APM-aware applications of the request. Each APM-aware application is given the opportunity to accept or reject the request. If all APM-aware applications accept the request, the APM Driver calls the *Set Power State* function to set the System Power State to Suspend. The APM Driver should call the *Get PM Event* function as soon as possible after a successful call to the *Set Power State* function returns so that applications and drivers can be notified that the system has resumed with minimal delay. If the request is rejected, the system will not transition into the Suspend state.

Normal Resume System Notification: The APM Driver notifies all APM-aware applications that the system has resumed from a Normal Suspend.

Critical Resume System Notification: The APM Driver notifies all APM-aware applications that the system has resumed from a Critical Suspend.

Battery Low Notification: The APM Driver notifies all APM-aware applications that a battery low indication has been received.

Power Status Change Notification: The APM Driver calls the *Get Power Status* function to determine the change to the power status.

Update Time Notification: The APM Driver updates the system time from the real time clock.

Critical System Suspend Notification: The APM Driver immediately tries to call the *Set Power State* function to set the System Power State to Suspend. The APM Driver should call *Get PM Event* and process the Critical Resume System Notification that will be returned by that call as soon after the system resumes operation as possible.

System Standby Resume Notification: The APM Driver notifies all APM-aware applications that the system has resumed from Standby.

APM Driver Recommendations

APM Drivers can use a 2-phase protocol for notifying APM-aware applications of (User) System Suspend Request or (User) System Standby Request Notifications if the driver allows APM-aware applications to reject either of these requests.

When the APM Driver detects a (User) System Standby Request Notification or a (User) System Suspend Request Notification, it should allow each application to indicate whether or not it is acceptable to make the requested state transition. If no rejections are received, the APM Driver should notify each application that the transition is about to occur before calling the *Set Power State* function to set the System Power State to Standby or Suspend.

If a single phase protocol is used, an application or driver may not be able to track the system power state correctly. There is no way for the application to determine whether the APM Driver actually suspended the system after the application accepts a request since the application can not determine whether another application rejected the request. As a result, the application can never be sure whether it will receive a resume notification state or whether the proposed standby/suspend has been aborted.

With the System Standby and System Suspend Notifications, it is possible for an APM Driver to use a single staged protocol with APM aware applications and drivers.

It is possible to design systems where the APM Driver allows APM applications and drivers to accept or reject System Standby and System Suspend requests. In such systems, a situation may occur where one or more APM aware applications or drivers have already accepted the request and have completed their save state procedures. Later down the notification chain, if an APM aware application or driver should reject the request, the applications that had previously accepted the request are now awaiting notification to resume. An APM 1.1 Driver can handle this situation by sending the

proper resume notification to the applications/drivers that require them. Note that this scheme will not work in an APM 1.0 environment because there is no Standby Resume defined for APM 1.0.

Appendix E - OEM Developer Notes for Hibernation

Previously OEM developers had a difficult time implementing hibernation in an APM 1.0 environment. They were forced to save the system state using hidden SCSI partitions or by using an OS specific file system. Now, a reserved fixed disk partition with an ID of 84H has been defined to give developers a better option. This enables the creation of a hibernation fixed disk partition which can be used to avoid file system and OS dependence. This hibernation partition ID can be used as either a Primary or Extended partition.

Under normal conditions, a system utility would create a disk partition which the APM BIOS would use as storage to save and restore system states. When the system is powered on, POST would look at the NVRAM bit to determine if the system had previously been in the hibernation state. If this were the case, then the APM BIOS would read the saved system state back into memory.

Because no specified format for this disk partition yet exists, the format should contain a special ID sequence to prove to the APM BIOS that it is in fact APM BIOS data.

Note that hibernation is not a requirement of APM. Actual hibernation implementation is up the individual OEM developers.