Advanced Power Management

The Next Generation

Version 1.0

Intel Corporation
Microsoft Corporation

Advanced Power Management

1 Introduction

This document presents a layered approach to Power Management which is termed Advanced Power Management(APM). APM is a specification that defines a layered cooperative environment which allows applications, operating systems, and the system BIOS to work together towards the goal of reducing power consumption in Personal Computers. The primary purpose of an APM implementation is to provide portable computer users increased productivity and greater system availability by extending the useful life of system batteries without degrading performance. The goal of prolonging battery life for portable computers does not preclude the usefulness of APM in other environments and applications.

This document describes APM using a model that partitions power management functionality into a hierarchy of cooperating layers, and standardizes the flow of information and control across the layers.

2 The Advanced Power Management System Model

3 Power States

APM defines four power states: Ready, Stand-by, Suspended, and Off. Three of these states apply both to individual system components and to the system as a whole. The suspended state is a special low power condition that applies to the system as a whole, and not the individual components.

4 Ready

In the ready state, the system or device is fully powered up and ready for use. The APM definition of Ready only indicates that the system or device is fully powered on, it does not differentiate between active and idle conditions.

5 Stand-by

Stand-by is an intermediate system dependent state which attempts to conserve power. Stand-by is entered when the CPU is idle and no device activity is known to have occurred within a machine defined period of time. The machine will not return to ready until one of the following events occur:

A device raises a hardware interrupt
Any controlled device is accessed

All data and operational parameters are preserved when the machine is in the Stand-

by state.

6 Suspended

The Suspended state is a system state which is defined to be the lowest level of power consumption available that preserves operational data and parameters. The suspend state can be initiated by either the system BIOS or the software above the BIOS. The system BIOS may place the system into the suspended state without notification if it detects a situation which requires an immediate response such as the battery entering a critically low power state. When the system is in the Suspended state, computation will not be performed until normal activity is resumed. Resumption of activity will not occur until signaled by an external event such as a button press, timer alarm, etc..

7 Off

When in the Off state, the system or device is powered down and inactive. Data and operational parameters may or may not be preserved in the Off state.

8 State Transitions

The system and devices can change from one power state to another either by explicit command or automatically based on APM parameters and system activity. The power capabilities of devices differ, and some devices may not be capable of achieving all states. Additionally, some devices may have built-in automatic power management functions which are invisible to the system, and therefor outside the scope of the APM specification.

9 Power Management Software Layers

10 The Power Management BIOS

The System BIOS is the lowest level of power management software in the system. It will normally be supplied by the OEM and is specific to the hardware platform.

The System BIOS will capable of providing power management functionality without any support from the operating system or applications. This support is analogous the ad hoc power management methods implemented on current generation laptop systems. This functionality will be enhanced once an APM aware operating system or environment establishes a cooperative connection with the BIOS. Once made, this connection establishes a protocol that allows the firmware to communicate power management events to the operating system and to wait for operating system concurrence if necessary. Details of the System BIOS operational changes are documented in the description of the interface.

11 The Operating System Layer

The operating system layer has three primary power management functions;

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

Different operating systems will require different PM application-to-OS interfaces. On some systems this interface may best be implemented through software interrupts, while on other systems, a CALL and RET interface may be a more appropriate.

Operating systems that can be enhanced by optional extensions may require specialized interfaces to allow the extensions to assist in the power management function.

12 The Application Layer

The application layer assists the power management function by providing information that only the application is in a position to know or easily ascertain. Applications are not required to support the power management function, but they can greatly increase its effectiveness, particularly on less sophisticated operating systems. Under MS-DOS in particular, the application is often in the best position to know when it is idle and awaiting user input.

13 Power Management Software Interfaces

14 The Power Management BIOS Interface

The System BIOS interface must support a variety of CPU modes so that the System BIOS implementation can provide power management to a mixed mode environment like MS-Dos.

A real mode interface will be required on all APM implementations and is the primary APM interface. This interface will be implemented with an extension to the existing PC/AT Int 15h BIOS interface. The System BIOS Int 15h interface must operate in either real mode, or virtual-86 mode on 80386 and later processors. Since the INT 15h processor instruction normally disables interrupts, the System BIOS can typically expect to be entered with interrupts disabled. However, the BIOS routines should not depend on any particular setting, and should explicitly enable and disable interrupts as necessary. To avoid re-entering Int 15h, an Interrupt Service Routine (ISR) should not try to use any of the APM functions.

To better support the protected modes of 80286 and later processors, the System BIOS will optionally support a 16-bit and 32-bit protect mode interface directly callable from protected mode. The protected mode interfaces must first be initialized using the real mode Int 15h AH=53h. Systems not supporting APM will return from this call with Carry Set and AH=86h

The 16-bit and 32-bit protected mode interfaces must be initialized using the *Protected Mode 16-bit Interface Connect* and *Protected Mode 32-bit Interface Connect* functions before these interfaces can be used.

If an error condition is detected during the processing of a System BIOS function, upon return to the caller the carry (CY) flag will be set and the AH register will contain an error code. The carry flag will be clear upon return from any successful call, and the contents of the AH register will be dependent on the particular call.

To allow for direct control of devices, this interface will adopt a convention for identifying a device class and specific subclasses within that class. For example, A class would be all Disk devices and the subclasses would be the physical unit numbers. APM calls will take this parameter, in a word length register where the most significant byte will be the device class and the least significant byte will be the device subclass. The impact of this parameter will be described in the individual interface descriptions. The APM Class/Subclass IDs will be defined as follows:

BX = Device ID

00XXh	System	
	00	System BIOS
	01	All devices power managed by the System Bios
01XXh	Display	
02XXh	Secondary Storage	
03XXh	Parallel Ports	
04XXh	Serial Ports	

0500 - FFFFH Reserved

where:

XX h = Physical Device Number (0 based)

FFh = All devices in this class

15 Interface Connect

An APM compliant System BIOS will likely provide some degree of power management functionality without any support from system or application software. The OEM or BIOS provider determines the degree to which power management functionality is implemented by the BIOS in a stand-alone configuration.

APM provides a cooperative interface between the System BIOS and other system or application software that makes the other software a partner in the power management process. Once this cooperative interface is established the System BIOS will inform its partner of significant power related events, and rely on its partner to actively participate in the power management process.
Some of the System BIOS functions can be called regardless of whether or not the cooperative PM interface has been established between the BIOS and the caller. However, some functions, particularly those related to notification of power events, will not operate until the cooperative interface is established.

16 APM Installation Check

Allows the caller to determine if the system's BIOS supports the APM functionality and if so, which version of the specification it supports.

Call with:

AH	= 53h
AL	= 00h - Do APM installation check
BX	= 0000H - System Bios

Returns:

if function successful:

Carry	= 0
AH	= major version number (in BCD format)
AL	= minor version number (in BCD format)
BH	= ASCII "P" character
BL	= ASCII "M" character
CX	= flags

bit 0 = 1 if 16-bit protected mode interface is supported.
bit 1 = 1 if 32-bit protected mode interface is supported.
bit 2 = 1 if **CPU Idle** call slows processor clock speed.
bit 3 = 1 if BIOS Power Management is disabled.
all other bits reserved.

if function unsuccessful:

Carry	= 1 Indicates error condition
AH	= 86h APM not present

The interpretation of bit 2 is documented under the CPU Busy call description.

This call is only available using the real mode Int 15h interface.

17 Interface Connect

Establishes the cooperative interface between the caller and the System BIOS. Before the interface is established, the System BIOS will provide its own power management functionality as implemented by the OEM. Once the interface is established (connected), the System BIOS and the caller will coordinate PM activities according to this specification.

Call with:

AH	= 53h
AL	= 01h
BX	= 0000H - System Bios

Returns:

If function successful:
Carry = 0
If function unsuccessful:

Carry	= 1
AH	= error code

02h interface connection already in effect.
09h Unrecognized device ID

This call is only available using the APM Int 15h interface.

18 Protected Mode 16-bit Interface Connect

Initializes the optional 16-bit protected mode interface between the caller and the System BIOS. This interface allows a protected mode caller to invoke the System BIOS functions without the need to first switch into real or virtual-86 mode. A caller that does not operate in protected mode may not need to use this call. This function establishes a 16-bit protected mode interface, but this function must be invoked in either real or virtual-86 mode using the Int 15h interface.

Call with:

AH	= 53h
AL	= 02h
BX	= 0000H - System Bios

Returns:

if function successful:

Carry	= 0
AX	= PM 16-bit code segment (real mode segment base address)
BX	= offset of entry point
CX	= PM 16-bit data segment (real mode segment base address)

if function unsuccessful:

Carry	= 1
AH	= error code 05h = 16-bit protected mode interface already established. 06h = 16-bit protected mode interface not supported 09h = Unrecognized device-id

The System BIOS 16-bit protected mode interface requires 2 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 information returned in the call from the BIOS. The segment limit fields will arbitrarily be set to 64k. These selectors may either be in the GDT or LDT, but must be valid whenever the System BIOS is to be called in protected mode. The code segment descriptor must specify protection level 0, and the BIOS routines must be invoked with CPL = 0 so they can execute privileged instructions.

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

BIOS 16-bit interface requires that it be called with a 16-bit stack.

When the System 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.

The System BIOS will reject additional 16-bit protected mode connections if a connection has already been established.

This call is only available using the APM Int 15h interface.

19 Protected Mode 32-bit Interface Connect

Initializes the optional 32-bit protected mode interface between the caller and the System BIOS. This interface allows a protected mode caller to invoke the System BIOS functions without the need to first switch into real or virtual-86 mode. A caller that does not operate in protected mode may not need to use this call. This function establishes a 32-bit protected mode interface, but this function must be invoked in either real or virtual-86 mode using the Int 15h interface.

Call with:

AH	= 53h
AL	= 03h
BX	= 0000H - System Bios

Returns:

if function successful:

Carry	= 0
AX	= PM 32-bit code segment (real mode segment base address)
EBX	= offset of entry point
CX	= PM 16-bit code segment (real mode segment base address)
DX	= PM data segment (real mode segment base address)

if function unsuccessful:

Carry	= 1
AH	= error code

07h = 32-bit protected mode interface already established.
08h = 32-bit protected mode interface not supported
09h = Unrecognized Device-ID

The System 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 System BIOS 32-bit interface can call other BIOS routines in a 16-bit code segment if it necessary. The caller must initialize these descriptors using the segment base information returned in the call from the

BIOS.   The segment limit fields will arbitrarily be set to 64k. These selectors may either be in the GDT or LDT, but must be valid whenever the System BIOS is to be called in protected mode. The code segment descriptors must specify protection level 0, and the BIOS routines must be invoked with CPL = 0 so they can execute privileged instructions.

The caller invokes the System BIOS 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. The caller must supply a stack large enough for modest use by the BIOS and by potential interrupt handlers. The caller's stack will be active if or when interrupts are enabled in the BIOS routine; the BIOS will not switch stacks when interrupts are enabled, including NMI interrupts. The BIOS 32-bit entry point requires that it be called with a 32-bit stack.

When the System 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.

The System BIOS will reject additional 32-bit protected mode connections if it a 32-bit protected mode connection has already been established.

This call is only available using the APM Int 15h interface.

20 Interface Disconnect

Breaks the cooperative interaction between the System BIOS and the caller, and in the process restores the System BIOS' default (built-in) functionality. Any protected mode connection set up by the *Protected Mode 16-bit Interface Connect* or *Protected Mode 32-bit Interface Connect* functions are also invalidated by this call.

Even though this call returns control of power management strategy to the System BIOS, the parameter values (timer values, enable/disable settings, etc.) in effect at the time of the disconnect will remain in effect.

```
Call with:
        AH      = 53h
        AL      = 04h
        BX      = 0000h - System BIOS
Returns:
        if function successful:
        Carry   = 0
        If function unsuccessful:
        Carry   = 1
        AH      = error code
                        03h = interface not connected
                        09h = Unrecognized Device-ID
```

This call is available in both the APM Int 15h and the Protect mode interfaces.

21 CPU Control

22 CPU Idle

Informs the System BIOS that the system is currently idle, and that processing should be suspended until the next system event (typically an interrupt) occurs. This function allows the System BIOS to take some implementation specific power saving action, such as a CPU HLT instruction or stopping the CPU clock.

In cases where an interrupt causes the system to leave the idle state, the interrupt may or may not have been serviced when the BIOS returns from the *CPU Idle* call. The caller should not make any assumptions concerning interrupt servicing, and should allow pending interrupts to be taken upon return from *CPU Idle.*

If interrupts are serviced from within the BIOS *CPU Idle* function, the interrupt handler must return to the BIOS when the interrupt processing is completed. The caller cannot use it's knowledge of being in the idle state to retain control from an interrupt handler. For example, some system implementations may slow the processor CPU clock rate before waiting on an interrupt, and restore the normal clock rate after the interrupt is serviced but before returning from the idle call.

When the caller regains control from the System BIOS idle routine, it should determine if there is actually any processing to be performed, and reissue the *CPU Idle* call if not. If the caller 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
AL	= 05H

Returns success:

CY = 0

This function is available in both the APM Int 15h and Protect Mode interface.

23 CPU Busy

Informs the System BIOS that the system is now busy and processing should continue at full speed. Some system implementations may only be able to slow the CPU clock rate and return in response to the *CPU Idle* call. It is expected that the System BIOS will restore the CPU clock rate to its normal rate when it recognizes increased system activity (typically interrupt driven), but it may be unable to do so when interrupts are hooked by external software and do not invoke BIOS routines.

In cases where this is possible, the caller can ensure the system is running at full speed by invoking the *CPU Busy* function. Upon return from the *APM Installation Check* call, bit 2 of the CX register indicates if the System BIOS slows the CPU clock rate during the *CPU Idle* call. The caller can use this bit to determine if it wishes to call CPU Busy before executing code that it wants run at full speed.

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 will have no unexpected side effects.

Call with:

AH	= 53h
AL	= 06H

Returns success:

CY = 0

This function is available in both the APM Int 15h and Protect Mode interface.

24 Set Power State

25 Call Description

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

Call with:

AH = 53h
AL = 07H
BX = Power Device ID
CX = System State ID

0000H Ready*
0001H Stand-by
0002H Suspend
0003H Off*
0004H-FFFFH Reserved
* Not supported for System Device ID 0001H

Returns:

If function successful:

Carry	= 0

If function unsuccessful:

Carry	= 1
AH	= error code
	01h = power management functionality disabled
	09h = Unrecognized Device ID
	0Ah= Parameter value in CX out of range
	60h = Cannot enter requested state

This call is available in both the APM Int 15h and Protect Mode interfaces.

26 System Stand-by

Explicitly commands the System BIOS to put the system into the stand-by state. The caller invokes this function in response to a *System Stand-by Request Notification* from the System BIOS after the caller has performed any processing of its own. The caller may also initiate a stand-by operation at its own discretion. For example, it the caller determines that the system is idle and is not already in the stand-by state.

The state, typically, is exited when an interrupt is raised. Interrupts can be serviced normally.

Call with:

AH	= 53H
AL	= 07H
BX	= 0001H
CX	= 0001H

This call is available in both the APM Int 15h and Protect Mode interfaces.

27 Suspend System

Explicitly commands the System BIOS to put the system into the low power suspended state. The caller invokes this function in response to a *Suspend System Request Notification* from the System BIOS after the caller (presumably an operating system or similar environment) has performed any pre-suspend processing of its own. In addition to responding to System BIOS requests, the caller may initiate a suspend operation at its own discretion. For example, it the caller determines that the system has been completely idle (for other than normal 'background' processing like servicing timer interrupts, etc..) during a given time interval it could request a system suspend using this call.

The System BIOS will not return to the caller until the system is resumed from the suspended state. This eliminates the need for most resume notifications from the System BIOS. Upon return, and when safe to do so, the caller may notify any PM aware applications of the resume.

Since the system may have been suspended for a long period of time, the caller may need to update its date and time values.

Call with:

AH	= 53H
AL	= 07H
BX	= 0001H
CX	= 0002H

This call is available in both the APM Int 15h and Protect Mode interfaces.

28 Power Management Status/Control

29 Enable/Disable Power Management Functionality

This function allows the caller to enable or disable all APM automatic power down functionality. When disabled, the System BIOS will not automatically power down devices, enter the stand-by state, enter the suspended state, or take power saving steps in response to *CPU Idle* calls. In addition, many System BIOS functions will be disabled and will return error 01h, power management functionality disabled.

Call with:

AH	= 53H
AL	= 08H
BX	= FFFFH Enable/Disable all power management
CX	= 0 to disable power management
	= 1 to enable power management

Returns:

if function successful:

Carry flag	= clear

If function unsuccessful:

Carry flag	= set
AH	= error code
	01h = power management functionality disabled
	09h = Unrecognized Device ID
	0Ah=Parameter value in CX out of range

This call is available in both the APM Int 15h and Protect Mode interfaces

30 Restore SYSTEM-BIOS Power On Defaults

This call will request the BIOS reinitialize all its power-on defaults.

Call with:

AH	= 53H
AL	= 09H
BX	= FFFFH

Returns:

if function successful:

Carry flag	= clear	

If function unsuccessful:

Carry flag	= set	
AH	= error code	
		09h = Unrecognized Device ID

This call is available in both the APM Int 15h and Protect Mode interfaces.

31 Get Power Status

Returns the systems current power status.

Call with:

AH	= 53H
AL	= 0AH
BX	= 0001H

Returns:

if function successful:

Carry = 0

BH	= AC line status		
		0	= off-line
		1	= on-line
		255	= unknown
		all other values reserved	
BL	= Battery status		
		0	= high
		1	= low
		2	= critical
		3	= charging
		255	= unknown

all other values reserved

CL	= Remaining battery life
	0 - 100 = % of full charge
	255 = unknown

If function unsuccessful:

Carry	= 1
AH	= error code
	09h = Unrecognized Device ID

This call is available in both the APM Int 15h and Protect Mode interfaces.

32 Power Management Event Notification

Power events are often asynchronous to normal system operation and can occur at times when the system software is not immediately ready to process the event. In order to synchronize power events with other system software, the System BIOS will withhold event notification until polled with the *Get PM Event* function. This allows the cooperative PM software to only receive event notification when it is ready to process the event.

The PM partner receives notification of power events by periodically calling the *Get PM Event* function. Power management events are not extremely time critical, and a polling frequency of once or twice a second should be sufficient.

33 Get PM Event

Returns the next pending PM event, or indicates if no PM events are pending.

Call with:

AH	= 53H
AL	= 0BH

Returns:

if function successful:

Carry	= 0
BX	= PM event code
	01H System Stand-by Request Notification
	02H System Suspend Request Notification
	03H Normal Resume System Notification
	04H Critical Resume System Notification
	05H Battery Low Notification

If function unsuccessful:
Carry = 1
AH = error code
03h = interface connection not established
80h = no PM events pending

The PM event notification codes and their interpretation are described in the following sections.

This call is available in both the APM Int 15h and Protect Mode interfaces.

34 System Stand-by Request Notification

Notifies the PM partner that the System BIOS wishes to put the system into a stand-by state. The System BIOS will not take any action at this time, but instead wait for the partner to call the *System Stand-by* function.

35 Suspend System Request Notification

Notifies the PM partner that the System BIOS wishes to put the system into a suspended state. The System BIOS will not actually perform a suspend at this time, but instead wait for the partner to call the *Suspend System* function.

36 Battery Low Notification

Informs the PM partner that the system's battery is running low.

37 Resume System Notification

Indicates that a critical system suspend and resume operation occurred without the cooperation of the PM partner. When a suspend operation is to occur, the System BIOS will normally notify the partner via the *Suspend System Request,* but not actually perform the suspend until the partner calls the *Suspend System* function. In some emergency siltations this may not be possible, and the System BIOS may need to suspend the system immediately. In this siltation, the System BIOS informs the partner that the system has resumed from such a suspend via the *Resume System Notification.* This notification allows the partner to recover from the suspend as best it can.

38 Impact on Existing APIs

39 Watchdog Timers / System Alarm Services

These services will continue to function in an APM compliant System BIOS in all power states except off. An application using these services can expect to be given control at the specified date and time and that the system will be in the ready state.

Appendix A - APM Function Summary

Function Description	Available Interface			Connection Required
	Int 15	16bit	32bit	
Installation Check (00h)	XXX			No
Interface Connect (01h)	XXX			No
Protect Mode Connect 16bit (02h)	XXX			No
Protect Mode Connect 32bit (03h)	XXX			No
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
System Stand-by	XXX	XXX	XXX	Yes
System Suspend	XXX	XXX	XXX	Yes
Enable Power Management Function (08h)	XXX	XXX	XXX	Yes
Disable Power Management Function	XXX	XXX	XXX	Yes
Restore Power on Defaults (09h)	XXX	XXX	XXX	No
Get Power Status (0Ah)	XXX	XXX	XXX	No

Get PM Event (0Bh)	**XXX**	**XXX**	**XXX**	**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**

Interface & General Errors

01h	**Power Management functionality disabled**	**Suspend system** **Set Power State** **Get Power Event** **Enable/Disable PM functionality**
02	**Interface connection already in effect**	**Interface connect**
03h	**Interface not connected**	**Interface disconnect** **Get PM event**
04h	**Real mode connection not established**	**Protect mode 16-bit interface connect** **Protect mode 32-bit interface connect**
05h	**16-bit protected mode interface already established**	**Protect mode 16-bit interface connect**
06	**16-bit protected mode interface not supported**	**Protect mode 16-bit interface connect**
07h	**32-bit protected mode interface already established**	**Protect mode 32-bit interface connect**
08h	**32-bit protected mode interface not supported**	**Protect mode 32-bit interface connect**
09h	**Unrecognized Device ID**	**Interface Connect** **Protected Mode 16-bit Interface**

		Connect Protected Mode 32-bit Interface Connect Interface Disconnect Set Power State Enable/Disable Power Management Functionality Restore SYSTEM-BIOS Power On Defaults Get Power Status
0Ah	**Parameter value in CX out of range**	**Set Power State Enable/Disable Power Management Functionality**

System Errors

60h	**Can't enter requested state**	**Set Power State**

Power Management Event Errors

80h	**No power management events pending**	**Get PM event**
86h	**Reserved - No APM present**	