- IPMI -

Platform Event Trap Format Specification

v1.0

Document Revision 1.0

December 7, 1998

Intel Hewlett-Packard NEC Dell

Revision History

Date	Revision	Modifications
12/7/98	1.0	Initial release.

Contents

Tables

Introduction

This document provides the specification of the *Platform Event Trap* format. A platform event is defined as an event that is originated directly from platform firmware (BIOS) or platform hardware (ASIC, chip set, or microcontroller) independently of the state of the operating system or system management hardware. The Platform Event Trap format is used for sending a platform event in an SNMP Trap. The trap may be directly issued from the platform or may be indirectly issued via a proxy (local or remote) that acts on events or alternatively formatted traps from the platform.

The Platform Event Trap allows traps to be generated from various sources including:

- BIOS
- OS Bootstrap Loader
- NIC
- System Alert ASIC
- System Management Micro-controller
- System Management Software
- Alert Proxy Software

Goals

The following goals were taken into consideration in defining the Platform Event Trap:

- Provide common fields for traps regardless of trap source (BIOS, software, Alert ASIC, IPMI, etc.)
- Merge 'sensor' related events with 'BIOS' related events.
- Extract fields from existing specifications, where possible.
- Create a trap format that is suited to ASIC or microcontroller implementation
- Include provisions for 'value added' content from system integrators, OSV, ISV, etc.
- Provide versioning fields to cleanly identify potential future revisions to the trap format.
- Support information that enables a management application that has received the trap to access the system and query the status of the subsystem that originated the event.
- Provide for codes indicating that a field is 'unspecified' (unused) to provide for devices that return a subset of the trap information. Attempt to have all 'unspecified' codes be either all 0's or all 1's to make it easy for a state machine to 'jam' the 'unspecified' code when transmitting the trap.

References

The following references provide supplementary and background information associated with this specification:

[1] *Intelligent Platform Management Interface v1.0 Specification*, ©1998 Intel Corporation, Hewlett-Packard Company, NEC Corporation, Dell Computer Corporation. Softcopy and information available from http://developer.intel.com/design/servers/ipmi.

[2] *System Management BIOS Reference Specification, v2.2*, © 1997, 1998 American Megatrends Inc., Award Software International, Compaq Computer Corporation, Dell Computer Corporation, Hewlett-Packard Company, Intel Corporation, International Business Machines Corporation, Phoenix Technologies Limited, and SystemSoft Corporation. Softcopy available from http://www.phoenix.com/techs, http://developer.intel.com/ial/wfm, and http://www.ibm.com/products/surepath.

SNMP Trap Format

The following table is a summary of the SNMP Trap PDU (protocol data unit) format. In addition the SNMP header shall carry the following fields:

Version SNMP rev-1

Community String Default = 'public'. This string may optionally be used to hold a vendor-specific string that is used to identify or provide SNMP access to the system that generated the event.

Table 1 - Trap PDU format per RFC 1157

enterprise	OID = iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).wired_for_management(3183).PET(1).version(1)
agent-addr	NetworkAddress
generic-trap	EnterpriseSpecific(6)
specific-trap	See below.
time-stamp	Time elapsed between last (re) initialization of the network entity and the generation of the trap
variable-bindings	Other information, defined below

Specific Trap and Variable Bindings Fields

The specific-trap and variable-bindings fields carry the heart of the Platform Event Trap information. The content and definition of these fields is specified in the following tables.

Table 2 -"Specific Trap" field

Field #	Name	size/ type	Description
1	Event	integer	31:24 reserved. 0000_0000b 23:16 Event Sensor Type An *Event Sensor* is a logical entity that is responsible for detecting events. The *Event Sensor Type* field indicates what types of events the sensor is monitoring. E.g. *temperature, voltage, current, BIOS, POST, processor, fan, etc.* (This field corresponds to the IPMI 'Sensor Type' field, and conceptually maps to the 'cause of trap' field in the Phoenix proposal.) 15:8 Event Type Code indicating what type of transition / state change triggered the trap. (Corresponds to IPMI 'Event Type' field) The code is split into the following ranges: 00-0Bh = generic - can be used with any type of sensor 6Fh = sensor specific 70h-7Fh = OEM all other = reserved See Table 4, below, for generic event type codes 7:0 Event Offset Indicates which particular event occurred for a given Event Type. This field allows events to be extended on a per Event Type basis—making it easier to manage the Event Type 'name space'. 7 0 = Assertion Event. (Event occurred when state became asserted) 1 = Deassertion Event. 6:4 reserved. 000b. 3:0 Offset Value. Per IPMI, up to 15 different discrete states are allowed per each Event Type. 0Fh = unspecified.

Variable Bindings Fields

The Platform Event Trap uses a single octet-string varbind (variable binding) for holding the bulk of the event information. This is done to keep the overall trap size compact to facilitate implementation in devices that have limited storage capabilities, such as ASICs or microcontrollers. The octet string can vary from 47 to 110 octets, depending on whether the OEM Custom Fields have data or not.

The OID for this varbind is:
iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).wired_for_management(3183).PET(1).version(1).1

The following table lists the different fields and their position in the octet-string. Data is in network order (ms-byte first) unless otherwise specified.

Table 3 - Variable Bindings Fields

octet #	Name	size/ type	Description
1:16	GUID	16 bytes	GUID for the platform, per SMBIOS 2.2 / PXE specifications. All 0's = unspecified (need to use agent-addr to identify the platform that generated the trap). The following specifies the octet ordering of sub-fields within the GUID field: time_low 0:3 msbyte in offset 0 time_mid 4:5 msbyte in offset 4 time_hi_and_version 6:7 msbyte in offset 6 clock_seq_hi_and_reserved 8 clock_seq_low 9 node 10-15 msbyte in offset 10
17:18	Sequence # / Cookie	word	0000h = unspecified. The function of this field is specific to the trap source type. The intent of the field is to provide a 'sequence #' that can be used to differentiate a re-transmitted (re-tried) trap from a new trap instance. It may also be used for applications that know how to respond to the trap source to give a positive acknowledge. There are restrictions on the use of this field: • An application must not be required to interpret this field in order to accept the trap or decode the fields in the trap. • The trap source must not rely on getting a response or other action from an application that interprets this field. [I.e. must be designed with the assumption that it may not get a response.] • All trap source types must support a 0000h=unspecified content for this field.
19:22	Local Timestamp	dword	Differs from SNMP trap timestamp in that this is platform local time based. Encoded as number of seconds from 0:00 1/1/98. 0000 0000 = unspecified.
23:24	UTC Offset	word	UTC Offset in minutes (two's complement, signed. -720 to +720, 0xFFFF=unspecified).

octet #	Name	size/ type	Description
25	Trap Source Type	byte	Class of the device or software that originated the trap on the network. This information provides information that an application can use for responding to, or controlling generation of, the trap. Platform Firmware (e.g. BIOS) 00h-07h SMI Handler 08h-0Fh ISV System Management Software 10h-17h Alert ASIC 18h-1Fh IPMI 20h-27h BIOS Vendor 28h-2Fh System Board Set Vendor 30h-37h System Integrator 38h-3Fh Third Party Add-in 40h-47h OSV 48h-4Fh NIC 50h-57h System Management Card 58h-5Fh unspecified FFh
26	Event Source Type	byte	Class of device or type of software that originated the *event*: This can be different than the device or type of software that sends the *trap.* This field is used for interpreting the event. Platform Firmware (e.g. BIOS) 00h-07h SMI Handler 08h-0Fh ISV System Management Software 10h-17h Alert ASIC 18h-1Fh IPMI 20h-27h BIOS Vendor 28h-2Fh System Board Set Vendor 30h-37h System Integrator 38h-3Fh Third Party Add-in 40h-47h OSV 48h-4Fh NIC 50h-57h System Management Card 58h-5Fh unspecified FFh
27	Event Severity	byte	Severity (based on DMI Event Severity). 0x00 = unspecified 0x01 = Monitor 00 0001 0x02 = Information 00 0010 0x04 = OK (return to OK condition) 00 0100 0x08 = Non-critical condition 00 1000 a.k.a. 'warning' 0x10 = Critical condition 01 0000 0x20 = Non-recoverable condition 10 0000
28	Sensor Device	byte	Identifies the instance of the device that holds the sensor that generated the event. E.g. in IPMI this holds an ID (I^2C address) of the controller that generated the event. FFh = unspecified.
29	Sensor Number	byte	The Sensor Number field is used to identify a given instance of a sensor relative to the Sensor Device. The ID is a 'handle'—that is, there is no linkage between the sensor type and the sensor number. E.g. sensor number 1 could be a temperature sensor on one device, and a voltage sensor on another. FFh = unspecified. 00h = unspecified.
30	Entity	byte	Entity ID from IPMI v1.0 specification (see Table 6, below). Indicates the platform entity the event is associated with - e.g. processor, system board, power supply, etc. Under IPMI, the Entity information can be used to subsequently access the Entities Field Replaceable Unit data, if any. 00h = unspecified.
31	Entity Instance	byte	Indicates which instance of the Entity the event is for. E.g. processor 1 or processor 2. 00h = unspecified.
32:39	Event Data	octet string (8)	Additional parametric data byte—formatted as specified by Event Type in combination with Event Source. Interpreted as individual octet fields. Event Data 1 Event Data 2

octet #	Name	size/ type	Description
			Event Data 3 Event Data 4 Event Data 5 Event Data 6 Event Data 7 Event Data 8
40	Language Code	byte	Per IPMI v1.0 FRU Information Format. FFh = 'unspecified'. This field can be used in conjunction with the OEM fields, below, to indicate the language that any strings are in. Note that language is different than character set. Character sets are specified as ASCII or UNICODE, per type/length bytes.
41:44	Manufacturer ID	dword	Manufacturer ID using Private Enterprise IDs per IANA.
45:46	System ID	word	Specified by manufacturer given by Manufacturer ID field, this number can be used to identify the particular system/product model or type.
47:(110)	OEM Custom Fields	octet string (max. 64)	One or more fields given in IPMI v1.0 FRU Information field format: Type/length code byte followed by N data bytes for each field. Fields end when type/length byte indicates 'no more records' (C1h). A C1h in octet 47 indicates no OEM Custom Fields.

Supporting Tables

Generic Event Types and Offsets

The following table is taken from the IPMI v1.0 specification. If differences exist, the IPMI v1.0 specification takes precedence.

Table 4 - IPMI v1.0 Generic Event Types and Offsets

Generic Event/Reading Type Code	Event/Reading Class	Generic Offset	Description
			THRESHOLD BASED STATES
01h	Threshold	00h	Lower Non-critical - going low
		01h	Lower Non-critical - going high
		02h	Lower Critical - going low
		03h	Lower Critical - going high
		04h	Lower Non-recoverable - going low
		05h	Lower Non-recoverable - going high
		06h	Upper Non-critical - going low
		07h	Upper Non-critical - going high
		08h	Upper Critical - going low
		09h	Upper Critical - going high
		0Ah	Upper Non-recoverable - going low
		0Bh	Upper Non-recoverable - going high
			DMI-based “Usage State” STATES
02h	Discrete	00h	Transition to Idle
		01h	Transition to Active
		02h	Transition to Busy
			DIGITAL/DISCRETE EVENT STATES
03h	‘digital’ Discrete	00h	State Deasserted
		01h	State Asserted
04h	‘digital’ Discrete	00h	Predictive Failure Deasserted
		01h	Predictive Failure Asserted
05h	‘digital’ Discrete	00h	Limit Not Exceeded
		01h	Limit Exceeded
06h	‘digital’ Discrete	00h	Performance Met
		01h	Performance Lags
			DMI-based SEVERITY EVENT STATES
07h	Discrete	00h	transition to OK
		01h	transition to Non-Critical from OK
		02h	transition to Critical from Less Severe
		03h	transition to Non-recoverable from Less Severe
		04h	transition to Non-Critical from More Severe
		05h	transition to Critical from Non-recoverable
		06h	transition to Non-recoverable
		07h	Monitor
		08h	Informational
			DMI-based AVAILABILITY STATUS STATES
08h	‘digital’ Discrete	00h	Device Removed / Device Absent
		01h	Device Inserted / Device Present
09h	‘digital’ Discrete	00h	Device Disabled
		01h	Device Enabled

Generic Event/Reading Type Code	Event/Reading Class	Generic Offset	Description
0Ah	Discrete	00h	transition to Running
		01h	transition to In Test
		02h	transition to Power Off
		03h	transition to On Line
		04h	transition to Off Line
		05h	transition to Off Duty
		06h	transition to Degraded
		07h	transition to Power Save
		08h	Install Error
			Other AVAILABILITY STATUS STATES
0Bh	Discrete	00h	Redundancy Regained
		01h	Redundancy Lost
		02h	Redundancy Degraded
			ACPI DEVICE POWER STATES
0Ch	Discrete	00h	D0
		01h	D1
		02h	D2
		03h	D3

Sensor Types and Sensor-specific Event Offsets

The following table is taken from the IPMI v1.0 specification. If differences exist, the IPMI v1.0 specification takes precedence.

Table 5 - IPMI v1.0 Sensor Types and Sensor-specific Event Offsets

Sensor Type	Sensor Type Code	Sensor-specific Offset	Event
reserved	00h	-	Reserved
Temperature	01h	-	Temperature
Voltage	02h	-	Voltage
Current	03h	-	Current
Fan	04h	-	Fan
Physical Security (Chassis Intrusion)	05h	00h 01h 02h 03h 04h 05h	General Chassis Intrusion Drive Bay Intrusion I/O Card area Intrusion Processor area Intrusion LAN Leash Lost (system has been unplugged from LAN) Unauthorized Dock/Undock
Platform Security Violation Attempt	06h	00h 01h 02h 03h 04h 05h	Secure Mode Violation Attempt Pre-boot Password Violation - user password Pre-boot Password Violation Attempt - setup password Pre-boot Password Violation - network boot password Other pre-boot Password Violation Out-of-band Access Password Violation
Processor	07h	00h 01h 02h 03h 04h 05h 06h 07h 08h 09h	IERR Thermal Trip FRB1/BIST Failure FRB2/Hang in POST Failure FRB3/Processor Startup/Initialization failure (CPU didn't start) Configuration Error (for DMI) SM BIOS 'Uncorrectable CPU-complex Error' Processor Presence Detected Processor Disabled Terminator Presence Detected
Power Supply	08h	00h 01h 02h	Presence Detected Power Supply Failure Detected Predictive Failure Asserted
Power Unit	09h	00h 01h 02h 03h 04h 05h 06h	Power Off / Power Down Power Cycle 240VA Power Down Interlock Power Down A/C Lost Soft Power Control Failure (unit did not respond to request to turn on) Power Unit Failure Detected
Cooling Device	0Ah	-	-
Other Units-based Sensor (per units given in SDR)	0Bh	-	-
Memory	0Ch	00h 01h 02h 03h	Correctable ECC Uncorrectable ECC Parity Memory Scrub Failed (stuck bit) *The Event Data 3 field for this command can be used to provide an event extension code, with the following definition:* 7:0 DIMM/SIMM/RIMM identification, relative to the entity that the sensor is associated with (if SDR provided for this sensor)
Drive Slot (Bay)	0Dh	-	-
POST Memory Resize	0Eh	-	-
POST Error	0Fh	-	-

Sensor Type	Sensor Type Code	Sensor-specific Offset	Event
Event Logging Disabled	10h	00h	Correctable Memory Error Logging Disabled
		01h	Event 'Type' Logging Disabled
		02h	Log Area Reset/Cleared
		03h	All Event Logging Disabled
Watchdog 1	11h		This sensor is provided to support IPMI v0.9 to v1.0 transition. See sensor 23h for recommended definition of Watchdog sensor for new v1.0 implementations.
		00h	BIOS Watchdog Reset
		01h	OS Watchdog Reset
		02h	OS Watchdog Shut Down
		03h	OS Watchdog Power Down
		04h	OS Watchdog Power Cycle
		05h	OS Watchdog NMI
		06h	OS Watchdog Expired, status only
		07h	OS Watchdog Pre-timeout Interrupt, non-NMI
System Event	12h	00h	System Reconfigured
		01h	OEM System Boot Event
		02h	Undetermined system hardware failure (this event would typically require system-specific diagnostics to determine FRU / failure type)
Critical Interrupt	13h	00h	Front Panel NMI
		01h	Bus Timeout
		02h	I/O Channel Check NMI
		03h	Software NMI
		04h	PCI PERR
		05h	PCI SERR
		06h	EISA Fail Safe Timeout
		07h	Bus Correctable Error
		08h	Bus Uncorrectable Error
		09h	Fatal NMI (port 61h, bit 7)
Button	14h	-	Button Event
Module / Board	15h	-	-
Microcontroller / Coprocessor	16h	-	-
Add-in Card	17h	-	-
Chassis	18h	-	-
Chip Set	19h	-	-
Other FRU	1Ah	-	-
Cable / Interconnect	1Bh	-	-
Terminator	1Ch	-	-
System Boot Initiated	1Dh	00h	Initiated by power up
		01h	Initiated by hard reset
		02h	Initiated by warm reset
		03h	User requested PXE boot
		04h	Automatic boot to diagnostic
Boot Error	1Eh	00h	No bootable media
		01h	Non-bootable diskette left in drive
		02h	PXE Server not found
		03h	Invalid boot sector
		04h	Timeout waiting for user selection of boot source
OS Boot	1Fh	00h	A: boot completed
		01h	C: boot completed
		02h	PXE boot completed
		03h	Diagnostic boot completed
		04h	CD-ROM boot completed
		05h	ROM boot completed
		06h	Boot completed - boot device not specified
OS Critical Stop	20h	00h	Stop during OS load / initialization
		01h	Run-time Stop

Sensor Type	Sensor Type Code	Sensor-specific Offset	Event
Slot / Connector	21h	00h	Fault Status asserted
		01h	Identify Status asserted
		02h	Slot / Connector Device installed/attached [This can include dock events]
		03h	Slot / Connector Ready for Device Installation - Typically, this means that the slot power is off. The Ready for Installation, Ready for Removal, and Slot Power states can transition together, depending on the slot implementation.
		04h	Slot/Connector Ready for Device Removal - Typically, this means that the slot power is off.
		05h	Slot Power is Off
		06h	Slot / Connector Device Removal Request - This is typically connected to a switch that becomes asserted to request removal of the device)
		07h	Interlock asserted - This is typically connected to a switch that mechanically enables/disables power to the slot, or locks the slot in the 'Ready for Installation / Ready for Removal states' - depending on the slot implementation. The asserted state indicates that the lockout is active. *The Event Data 2 & 3 fields for this command can be used to provide an event extension code, with the following definition:* <u>Event Data 2</u> 7 Reserved 6:0 Slot/Connector Type 0 PCI 1 Drive Array 2 External Peripheral Connector 3 Docking 4 Other standard internal expansion slot 5 Slot associated with entity specified by Entity ID for sensor all other = Reserved <u>Event Data 3</u> 7:0 Slot/Connector Number
System ACPI Power State	22h	00h	S0 / G0 "working"
		01h	S1 "sleeping with system h/w & processor context maintained"
		02h	S2 "sleeping, processor context lost"
		03h	S3 "sleeping, processor & h/w context lost, memory retained."
		04h	S4 "non-volatile sleep / suspend-to disk"
		05h	S5 / G2 "soft-off"
		06h	S4 / S5 soft-off, particular S4 / S5 state cannot be determined
		07h	G3 / Mechanical Off
		08h	Sleeping in an S1, S2, or S3 states (used when particular S1, S2, S3 state cannot be determined)
		09h	G1 sleeping (S1-S4 state cannot be determined)

Sensor Type	Sensor Type Code	Sensor-specific Offset	Event
Watchdog 2	23h		This sensor is recommended for new IPMI v1.0 implementations.
		00h	Timer expired, status only (no action, no interrupt)
		01h	Hard Reset
		02h	Power Down
		03h	Power Cycle
		04h-07h	reserved
		08h	Timer interrupt
			The Event Data 2 field for this command can be used to provide an event extension code, with the following definition: 7:0 interrupt type 000 = none 001 = SMI 010 = NMI 011 = Interrupt Fh = Unspecified all other = Reserved 3:0 timer use at expiration: 0h = Reserved 1h = BIOS FRB2 2h = BIOS/POST 3h = OS Load 4h = SMS/OS 5h = OEM Fh = Unspecified all other = Reserved
Platform Alert	24h		This sensor can be used for returning the state and generating events associated with alerts that have been generated by the Platform mgmt. subsystem.
		00h	Platform generated page.
		01h	Platform generated LAN alert.
		02h	Platform Event Trap generated, formatted per IPMI PET specification.
		03h	Platform generated SNMP trap, OEM format.
Entity Presence	25h		This sensor type provides a mechanism that allows a management controller to direct system management software to ignore a set of sensors based on detecting that presence of an entity. This sensor type is not typically used for event generation, but to just provide a present reading.
		00h	Entity Present. This indicates that the Entity identified by the Entity ID for the sensor is present.
		01h	Entity Absent. This indicates that the Entity identified by the Entity ID for the sensor is absent. If the entity is absent, system management software should consider all sensors associated with that Entity to be absent as well—and ignore those sensors.
Monitor ASIC / IC	26h	-	-
LAN	27h	00h	LAN Heartbeat Lost
Reserved	Remaining	-	-
OEM RESERVED	C0h-FFh	-	-

Entity ID Codes

The following table is taken from the IPMI v1.0 specification. If differences exist, the IPMI v1.0 specification takes precedence.

Table 6 - Entity ID Codes

code	Entity
0	Unspecified
1*	Other
2*	Unknown (unspecified)
3*	Processor
4*	Disk or disk bay
5*	Peripheral bay
6*	System management module
7*	System board (main system board, may also be a processor board and/or internal expansion board)
8*	Memory module (board holding memory devices)
9*	Processor module (holds processors, use this designation when processors are not mounted on system board)
10*	Power supply (DMI refers to this as a “power unit”, but it’s used to represent a power supply)
11*	Add-in card
12	Front panel board (control panel)
13	Back panel board
14	Power system board
15	Drive backplane
16	System internal expansion board (contains expansion slots)
17	Other system board (part of board set)
18	Processor board (holds 1 or more processors - includes boards that hold SECC modules)
19	Reserved
20	Power unit / power domain (typically used as a pre-defined logical entity for grouping power supplies)
21	Power module / converter
22	Power management / power distribution board
23	Chassis back panel board
24	System chassis
25	Sub-chassis
26	Other chassis board
27	Disk Drive Bay
28	Peripheral Bay
29	Device Bay
30	Fan / cooling device
31	Cooling unit (can be used as a pre-defined logical entity for grouping fans or other cooling devices)
32	Cable / interconnect
33	Memory device (Should be used for replaceable memory devices, e.g. DIMM/SIMM. It is recommended that Entity IDs not be used for individual non-replaceable memory devices. Rather, monitoring and error reporting should be associated with the FRU [e.g. memory card] holding the memory.)
34	System Management Software
35	BIOS
36	Operating System
37	System Bus
38	Group - this is a logical entity for use with Entity Association records. It is provided to allow a sensor data record to point to an Entity-association record when there is no appropriate pre-defined logical entity for the entity grouping.
90h-AFh	Chassis-specific Entities. These IDs are system specific and can be assigned by the chassis provider.
B0h-CFh	Board-set specific Entities. These IDs are system specific and can be assigned by the Board-set provider.
D0h-FFh	OEM System Integrator defined. These IDs are system specific and can be assigned by the system integrator, or OEM.
-	All other values reserved

* = DMI standard groups compatible.

Language Codes

The following table should be used when interpreting the Language Code fields. This table is taken from the IPMI v1.0 specification. If differences exist, the IPMI v1.0 specification takes precedence. The number (1-136) for the desired language is stored in the Language Code field. These codes are used to specify the language that the information is in. Since the standard FRU information consists of numeric fields, alpha-numeric strings, and names, there is little linguistic content to actually be concerned with. The language code can, however, serve to guide utilities in the presenting the FRU information by offering user interface elements, such as user prompts, menus, etc., in the specified language. This encoding is based on ISO 639:1988 (E/F) "Code for the representation of names of languages."

Table 7- Language Codes

1.	aa	Afar	51.	it	Italian	101.	si	Singhalese
2.	ab	Abkhazian	52.	iw	Hebrew	102.	sk	Slovak
3.	af	Afrikaans	53.	ja	Japanese	103.	sl	Slovenian
4.	am	Amharic	54.	ji	Yiddish	104.	sm	Samoan
5.	ar	Arabic	55.	jw	Javanese	105.	sn	Shona
6.	as	Assamese	56.	ka	Georgian	106.	so	Somali
7.	ay	Aymara	57.	kk	Kazakh	107.	sq	Albanian
8.	az	Azerbaijani	58.	kl	Greenlandic	108.	sr	Serbian
9.	ba	Bashkir	59.	km	Cambodian	109.	ss	Siswati
10.	be	Byelorussian	60.	kn	Kannada	110.	st	Sesotho
11.	bg	Bulgarian	61.	ko	Korean	111.	su	Sudanese
12.	bh	Bihari	62.	ks	Kashmiri	112.	sv	Swedish
13.	bi	Bislama	63.	ku	Kurdish	113.	sw	Swahili
14.	bn	Bengali; Bangla	64.	ky	Kirghiz	114.	ta	Tamil
15.	bo	Tibetan	65.	la	Latin	115.	te	Tegulu
16.	br	Breton	66.	ln	Lingala	116.	tg	Tajik
17.	ca	Catalan	67.	lo	Laothian	117.	th	Thai
18.	co	Corsican	68.	lt	Lithuanian	118.	ti	Tigrinya
19.	cs	Czech	69.	lv	Latvian, Lettish	119.	tk	Turkmen
20.	cy	Welsh	70.	mg	Malagasy	120.	tl	Tagalog
21.	da	Danish	71.	mi	Maori	121.	tn	Setswana
22.	de	German	72.	mk	Macedonian	122.	to	Tonga
23.	dz	Bhutani	73.	ml	Malayalam	123.	tr	Turkish
24.	el	Greek	74.	mn	Mongolian	124.	ts	Tsonga
25.	en	English	75.	mo	Moldavian	125.	tt	Tatar
26.	eo	Esperanto	76.	mr	Marathi	126.	tw	Twi
27.	es	Spanish	77.	ms	Malay	127.	uk	Ukrainian
28.	et	Estonian	78.	mt	Maltese	128.	ur	Urdu
29.	eu	Basque	79.	my	Burmese	129.	uz	Uzbek
30.	fa	Persian	80.	na	Nauru	130.	vi	Vietnamese
31.	fi	Finnish	81.	ne	Nepali	131.	vo	Volapuk
32.	fj	Fiji	82.	nl	Dutch	132.	wo	Wolof
33.	fo	Faeroese	83.	no	Norwegian	133.	xh	Xhosa
34.	fr	French	84.	oc	Occitan	134.	yo	Yoruba
35.	fy	Frisian	85.	om	(Afan) Oromo	135.	zh	Chinese
36.	ga	Irish	86.	or	Oriya	136.	zu	Zulu
37.	gd	Scots Gaelic	87.	pa	Punjabi			
38.	gl	Galician	88.	pl	Polish			
39.	gn	Guarani	89.	ps	Pashto, Pushto			
40.	gu	Gujarati	90.	pt	Portuguese			
41.	ha	Hausa	91.	qu	Quechua			
42.	hi	Hindi	92.	rm	Rhaeto-Romance			
43.	hr	Croatian	93.	rn	Kirundi			
44.	hu	Hungarian	94.	ro	Romanian			
45.	hy	Armenian	95.	ru	Russian			
46.	ia	Interlingua	96.	rw	Kinyarwanda			
47.	ie	Interlingue	97.	sa	Sanskrit			
48.	ik	Inupiak	98.	sd	Sindhi			
49.	in	Indonesian	99.	sg	Sangro			
50.	is	Icelandic	100.	sh	Serbo-Croatian			

Type/Length Byte Format

7:6 - type code
- 00 - binary or unspecified
- 01 - BCD plus (see below)
- 10 - 6-bit ASCII, packed
- 11 - 8-bit ASCII + Latin 1. At least two bytes of data must be present when this type is used. Therefore, the length (number of data bytes) will be >1 if data is present, 0 if data is not present.

5:0 - number of data bytes

000000 indicates that the field is empty.

When the type code is 11b, a length of 000001 indicates 'end of fields'. I.e. Type/Length = C1h indicates 'end of fields'.

BCD PLUS definition

- 0h - 9h = digits 0 through 9
- Ah = space
- Bh = dash '-'
- Ch = period '.'
- Dh = reserved
- Eh = reserved
- Fh = reserved

Mapping to System Boot Status Codes

The Platform Event Trap Sensor and Event offset codes map to the same underlying events that are reported in the SMBIOS table record as system boot status codes. Note that reason codes #4 and #7 do not have a mapping per this proposal. The rationale for this is given in the corresponding Event Offset / Description column.

Table 8 - SMBIOS System Boot Status Code to Platform Event Mapping

SYSTEM BOOT STATUS CODES			PLATFORM EVENT TRAP CODES	
System Boot Status Code Name	Reason Type	Offset	Sensor Type / Name	Event Offset / Description
System booted with no detectable error.	0	none	20h = OS Boot	00h = A: boot completed 01h = C: boot completed 02h = PXE boot completed 03h = Diagnostic boot completed 04h = CD-ROM boot completed 05h = ROM boot completed 06h = Boot completed - boot device not specified
No bootable media.	1	none	1Fh = Boot Error	00h = No bootable media
The “normal” operating system failed to load.	2	none	21h = OS Critical Stop	00h = Stop during OS load / initialization
Firmware-detected hardware failure, diagnostics requested.	3	none	1Dh = System	00h = Undetermined system hardware failure
Operating system-detected hardware failure, diagnostics requested. For ACPI OS’s, the system firmware might set this reason code when the OS reports a boot failure via interfaces defined in the *Simple Boot Flag Specification*.	4	none	NOT COVERED	Run time hardware failure traps left to DMI / CIM definition.
User-requested boot, usually via a keystroke.	5	none	System Boot Initiated	03h = User requested boot.
System security violation.	6		06h = Security Violation Attempt	See Sensor Types and Sensor-specific Event Offsets
Previously requested image. This reason code allows coordination between OS-present software and the OS-absent environment. For example, an OS-present application might enable (via a platform-specific interface) the system to boot to the PXE and request a specific boot-image.	7	varies	NOT COVERED	Appears valid for PXE, but not as a trap.
A system watchdog timer expired, causing the system to reboot.	8	none	Watchdog	See Sensor Types and Sensor-specific Event Offsets
Vendor/OEM-specific implementations.	C0h-FFh	OEM	C0h-FFh	OEM defined, per OEM ID.

* NOTE: There is no linkage between ‘action’ and ‘event’ in the trap. Linking ‘diagnostics requested’ to a particular event is left up to the receiving application.

LAST PAGE