System Management BIOS (SMBIOS) Reference Specification

13

CONTENTS

187 **Tables**

307 # Foreword

308 The *System Management BIOS (SMBIOS) Reference Specification* (DSP0134) was prepared by the
309 SMBIOS Working Group.

310 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
311 management and interoperability. For information about the DMTF, see http://www.dmtf.org.

312 ## Acknowledgments

313 The DMTF acknowledges the following individuals for their contributions to this document:

314 • Kimon Berlin – Hewlett-Packard Company

315 • Richard Chan – Dell

316 • Garrett Kirkendall – AMD

317 • Tom Slaight – Intel

318 • Gail Woodland – IBM

319

320 # Introduction

321 Continuing the DMTF's mission of leading the development of management standards for distributed
322 desktop, network, enterprise, and Internet environments, the *System Management BIOS Reference*
323 *Specification* addresses how motherboard and system vendors present management information about
324 their products in a standard format by extending the BIOS interface on Intel architecture systems. The
325 information is intended to allow generic instrumentation to deliver this data to management applications
326 that use CIM (the WBEM data model) or direct access and eliminates the need for error prone operations
327 such as probing system hardware for presence detection.

328 This specification is intended to provide enough information for BIOS developers to implement the
329 necessary extensions to allow their product's hardware and other system-related information to be
330 accurately determined by users of the defined interfaces.

331 This specification is also intended to provide enough information for developers of management
332 instrumentation to develop generic routines for translating from SMBIOS format to the format used by
333 their chosen management technology, whether it is a DMTF technology such as CIM, or another
334 technology, such as SNMP. To support this translation for DMTF technologies, sections of this
335 specification describe the CIM classes intended to convey the information retrieved from an SMBIOS-
336 compatible system through the interfaces described in this document.

337 NOTE   The DMTF's SMBIOS Working Group controls changes to this document; change requests should be
338 submitted to mailto:smbios@dmtf.org. Refer to http://www.dmtf.org/standards/smbios for the most recent version of
339 this document.

340 ## Document conventions

341 ### Typographical conventions

342 The following typographical conventions are used in this document:

343 • All numbers specified in this document are in decimal format unless otherwise indicated. A number
344 followed by the letter 'h' indicates hexadecimal format; a number followed by the letter 'b' indicates
345 binary format.

346 EXAMPLE: The values 10, 0Ah, and 1010b are equivalent.

347 • Any value not listed in an enumerated list is reserved for future assignment by the DMTF; see clause
348 5.2.2 for more information.

349 • Most of the enumerated values defined in this specification simply track the like values specified by
350 the DMTF within CIM classes. Enumerated values that are controlled by the DMTF are identified
351 within their respective subsection; additional values for these fields are assigned by the DMTF; see
352 6.3 for more information.

353 • Code samples use a `fixed font highlighted in gray`.

354   **Document version number conventions**

355   Beginning with version 2.3.1 of this document, the document's version number is specified in a
356   ***major.minor***[.***docrev***] format. The addition of ***docrev*** enables document updates to keep current with
357   hardware technology without causing implementations to continually "chase" a specification version.

358   • The ***major*** value of the document version increments by one whenever a major interface
359   change is introduced. Looking back, the value should have been incremented in the
360   transition from version 2.0 to version 2.1 because the table-based method was a major
361   interface change.

362   • The ***minor*** value of the document version either resets to zero if the ***major*** value
363   increments, or increments by one if a change in implementation requirements is introduced
364   *within* the same major version (for example, the addition of a new *required* structure or
365   structure field, or the new definition of a previously reserved bit).

366   • The ***docrev*** value of the document version either resets to zero if either the ***major*** or
367   ***minor*** value increments, or increments by one each time this document is updated.
368   Extending an existing enumeration with a new value is an example of when only updating
369   the ***docrev*** is required. This value does not factor into the specification version; an
370   implementation based on document version 2.3 complies with specification version 2.3, as
371   does an implementation based on document version 2.3.11.

372   • A ***docrev*** value of 0 displays as blank (that is, 2.4 instead of 2.4.0).

373   If these conventions were in place when version 2.0 of the specification was released, they would have
374   been applied to specification versions 2.1 through 2.3 as follows:

Specification Version	Would Have Been …	Rationale
2.1	3.0	The addition of the table-based method constitutes a major interface change.
2.2	3.1	The table-based method was made a requirement for compliance.
2.3	3.2	A minimum set of structures was made a requirement for compliance.

375

376 # System Management BIOS (SMBIOS) Reference Specification

377 ## 1 Scope

378 The *System Management BIOS (SMBIOS) Reference Specification* addresses how motherboard and
379 system vendors present management information about their products in a standard format by extending
380 the BIOS interface on Intel architecture systems. The information is intended to allow generic
381 instrumentation to deliver this data to management applications that use CIM (the WBEM data model) or
382 direct access and eliminates the need for error prone operations like probing system hardware for
383 presence detection.

384 ### 1.1 Supported processor architectures

385 This specification was originally designed for Intel® processor architecture systems. The following
386 processor architectures are now supported:

387 • IA-32 (x86),

388 • x64 (x86-64, Intel64, AMD64, EM64T),

389 • Intel® Itanium® architecture,

390 • 32-bit ARM (Aarch32),

391 • 64-bit ARM (Aarch64).

392 This specification may be compatible with other processor architectures, but support has not been
393 explicitly targeted.

394 ## 2 Normative references

395 The following referenced documents are indispensable for the application of this document. For dated or
396 versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies.
397 For references without a date or version, the latest published edition of the referenced document
398 (including any corrigenda or DMTF update versions) applies.

399 ACPI, *Advanced Configuration and Power Interface Specification*, Version 3.0, September 2, 2004,
400 http://www.acpi.info

401 *Boot Integrity Services API*, Version 1.0+bis37, 31 August 1999,
402 http://sourceforge.net/projects/bis

403 DIG64, *Developer's Interface Guide for Intel* Itanium* Architecture-based Server* 2.2, December 2004,
404 http://www.dig64.org/specifications/

405 DMTF DSP0004, *CIM Infrastructure Specification 2.6,*
406 http://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf

407 DMTF DSP0200, *CIM Operations over HTTP 1.3,*
408 http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

409 DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes 1.1*,
410 http://www.dmtf.org/standards/published_documents/DSP0239_1.1.pdf

411 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0,*
412 http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf

413 DMTF, *CIM Schema*, Version 2.34.0, 6 September 2012,
414 http://www.dmtf.org/standards/cim/

415 IETF RFC4122, *A Universally Unique IDentifier (UUID) URN Namespace,* The Internet Society, July
416 2005, http://www.ietf.org/rfc/rfc4122.txt

417 Intel, *Intelligent Platform Management Interface (IPMI) Interface Specification*, Version 2.0, February 12
418 2004, http://developer.intel.com/design/servers/ipmi/spec.htm

419 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards,*
420 http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype

421 ISO 639-1:2002, *Codes for the representation of names of languages — Part 1: Alpha-2 code*
422 A list of codes is available at http://www.loc.gov/standards/iso639-2/php/code_list.php

423 ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country*
424 *codes*
425 A list of codes is available at http://www.iso.org/iso/country_names_and_code_elements

426 Microsoft, *PCI IRQ Routing Table Specification*, Version 1.0, February 27, 1996

427 Microsoft, *Plug and Play BIOS Specification*, Version 1.0A, May 5, 1994

428 Microsoft, *Simple Boot Flag Specification*, Version 2.1, 28 January 2005,
429 http://www.microsoft.com/whdc/resources/respec/specs/simp_boot.mspx

430 PCI SIG, *PCI Firmware Specification*, version 3.0, June 20, 2005,
431 http://www.pcisig.com/specifications/conventional/pci_firmware

432 Phoenix Technologies, Ltd., *BIOS Boot Specification*, Version 1.01, 11 January 1996,
433 A copy is available at http://www.scs.stanford.edu/nyu/04fa/lab/specsbbs101.pdf

434 Phoenix Technologies, Ltd., *"El Torito" Bootable CD-ROM Format Specification*, Version 1.0, January 25
435 1995, http://download.intel.com/support/motherboards/desktop/sb/specscdrom.pdf

436 SBS, *Smart Battery Data Specification*, Version 1.1, 15 December 1998,
437 http://www.sbs-forum.org/specs/

438 UEFI, *Unified Extensible Firmware Interface (UEFI) Specification*, Version 2.3.1, Errata C, June 2012,
439 http://www.uefi.org/specifications.

440 UEFI, *UEFI Platform Initialization (PI) Specification*, Version 1.2.1, May 2012,
441 http://www.uefi.org/specifications

442 3   Terms and definitions

443 In this document, some terms have a specific meaning beyond the normal English meaning. Those terms
444 are defined in this clause.

445 The terms "shall" ("required"), "shall not," "should" ("recommended"), "should not" ("not recommended"),
446 "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
447 in ISO/IEC Directives, Part 2, Annex H. The terms in parenthesis are alternatives for the preceding term,
448 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
449 ISO/IEC Directives, Part 2, Annex H specifies additional alternatives. Occurrences of such additional
450 alternatives shall be interpreted in their normal English meaning.

451 The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as
452 described in ISO/IEC Directives, Part 2, Clause 5.

453 The terms "normative" and "informative" in this document are to be interpreted as described in ISO/IEC
454 Directives, Part 2, Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do
455 not contain normative content. Notes and examples are always informative elements.

456 The terms defined in DSP0004, DSP0200, and DSP1001 apply to this document. The following additional
457 terms are used in this document.

458 4   Symbols and abbreviated terms

459 The abbreviations defined in DSP0004, DSP0200, and DSP1001 apply to this document. The following
460 additional abbreviations are used in this document.

461 **4.1**
462 **AC**
463 Alternating Current

464 **4.2**
465 **ACPI**
466 Advanced Configuration and Power Interface

467 **4.3**
468 **AGP**
469 Accelerated Graphics Port

470 **4.4**
471 **APM**
472 Advanced Power Management

473 **4.5**
474 **ASCII**
475 American Standard Code for Information Interchange

476 **4.6**
477 **ATA**
478 Advanced Technology Attachment

479 **4.7**
480 **ATAPI**
481 ATA Packet Interface

482 **4.8**
483 **BCD**
484 Binary-Coded Decimal

485 **4.9**
486 **BIOS**
487 Basic Input/Output System

488 **4.10**
489 **BMC**
490 Baseboard Management Controller

491 **4.11**
492 **CGA**
493 Color Graphics Array

494 **4.12**
495 **CIM**
496 Common Information Model

497 **4.13**
498 **CMOS**
499 Complementary Metal-Oxide Semiconductor. "CMOS" is commonly used as a shorthand for "CMOS
500 RAM", the non-volatile RAM used on industry-standard PCs.

501 **4.14**
502 **CPU**
503 Central Processing Unit

504 **4.15**
505 **CRC**
506 Cyclic Redundancy Check

507 **4.16**
508 **DDC**
509 Display Data Channel

510 **4.17**
511 **DDR**
512 Double Data Rate SDRAM

513 **4.18**
514 **DIMM**
515 Dual In-line Memory Module

516 **4.19**
517 **DMA**
518 Direct Memory Access

519 **4.20**
520 **DMI**
521 Desktop Management Interface

522 **4.21**
523 **DRAM**
524 Dynamic RAM

525 **4.22**
526 **DSP**
527 Digital Signal Processing

528 **4.23**
529 **ECC**
530 Error Checking and Correction

531 **4.24**
532 **EDD**
533 Enhanced Disk Drive

534 **4.25**
535 **EDO**
536 Extended Data Out

537 **4.26**
538 **EEPROM**
539 Electrically Erasable PROM

540 **4.27**
541 **EISA**
542 Extended Industry-Standard Architecture

543 **4.28**
544 **EPS**
545 Entry Point Structure

546 **4.29**
547 **ESCD**
548 Extended System Configuration Data

549 **4.30**
550 **FDC**
551 Floppy Drive Controller

552 **4.31**
553 **FIFO**
554 First In, First Out

555 **4.32**
556 **GPNV**
557 General-Purpose NVRAM

558 **4.33**
559 **I2O**
560 Intelligent Input/Output

561 **4.34**
562 **IEPS**
563 Intermediate Entry Point Structure

564 **4.35**
565 **IO**
566 Input/Output

567 **4.36**
568 **IRQ**
569 Interrupt Request

570 **4.37**
571 **ISA**
572 Industry Standard Architecture

573 **4.38**
574 **LIF**
575 Low Insertion Force

576 **4.39**
577 **LSB**
578 Least-Significant Bit

579 **4.40**
580 **MCA**
581 Micro Channel Architecture

582 **4.41**
583 **MOF**
584 Managed Object Format

585 **4.42**
586 **MSB**
587 Most Significant Bit

588 **4.43**
589 **NMI**
590 Non-Maskable Interrupt

591 **4.44**
592 **NV**
593 Non-Volatile

594 **4.45**
595 **NVRAM**
596 Non-Volatile RAM

597 **4.46**
598 **OEM**
599 Original Equipment Manufacturer

600 **4.47**
601 **OS**
602 Operating System

603 **4.48**
604 **PATA**
605 Parallel ATA

606 **4.49**
607 **PCI**
608 Peripheral Component Interconnect

609 **4.50**
610 **PCMCIA**
611 Personal Computer Memory Card International Association

612 **4.51**
613 **PME**
614 Power Management Event

615 **4.52**
616 **PNP**
617 Plug-And-Play

618 **4.53**
619 **POST**
620 Power-On Self-Test

621 **4.54**
622 **PROM**
623 Programmable ROM

624 **4.55**
625 **PXE**
626 Pre-boot Execution Environment

627 **4.56**
628 **RAID**
629 Redundant Array of Inexpensive Disks

630 **4.57**
631 **RAM**
632 Random-Access Memory

633 **4.58**
634 **ROM**
635 Read-Only Memory

636 **4.59**
637 **RPM**
638 Revolutions per Minute

639 **4.60**
640 **RTC**
641 Real-Time Clock

642 **4.61**
643 **SAS**
644 Serial-Attached SCSI

645 **4.62**
646 **SATA**
647 Serial ATA

648   **4.63**
649   **SCSI**
650   Small Computer System Interface

651   **4.64**
652   **SDRAM**
653   Synchronous DRAM

654   **4.65**
655   **SIMM**
656   Single In-line Memory Module

657   **4.66**
658   **SKU**
659   Stock-Keeping Unit

660   **4.67**
661   **SMBIOS**
662   System Management BIOS

663   **4.68**
664   **SMBus**
665   System Management Bus

666   **4.69**
667   **SRAM**
668   Static RAM

669   **4.70**
670   **UEFI**
671   Unified Extensible Firmware Interface

672   **4.71**
673   **UPS**
674   Uninterruptible Power Supply

675   **4.72**
676   **USB**
677   Universal Serial Bus

678   **4.73**
679   **UUID**
680   Universally Unique Identifier

681   **4.74**
682   **VESA**
683   Video Electronics Standards Association

684   **4.75**
685   **VL-VESA**
686   VESA Video Local Bus

687   **4.76**

688   **ZIF**

689   Zero Insertion Force

5   Accessing SMBIOS information

5.1   General

692 The only access method defined for the SMBIOS structures is a table-based method, defined in version
693 2.1 of this specification. It provides the SMBIOS structures as a packed list of data referenced by a table
694 entry point.

695 NOTE   The Plug-and-Play function interface was deprecated in version 2.3.2 of this specification. It was
696 completely removed in version 2.7.

697 NOTE   The Entry Point Structure and all SMBIOS structures assume a little-endian ordering convention, unless
698 explicitly specified otherwise, i.e., multi-byte numbers (WORD, DWORD, etc.) are stored with the low-order byte at
699 the lowest address and the high-order byte at the highest address.

5.2   Table convention

701 The table convention allows the SMBIOS structures to be accessed under 32-bit and 64-bit protected-
702 mode operating systems, such as Microsoft® Windows XP®, Microsoft® Windows Server®, or Linux®.
703 This convention provides a searchable entry-point structure (which can be queried on EFI-based
704 systems) that contains a pointer to the packed SMBIOS structures.

705 The original SMBIOS 2.1 (32-bit) entry point, described in section 5.2.1, allows the SMBIOS structure
706 table to reside anywhere in 32-bit physical address space (that is, below 4 GB).

707 The SMBIOS 3.0 (64-bit) entry point, described in section 5.2.2, allows the SMBIOS structure table to
708 reside anywhere in 64-bit memory.

709 An implementation may provide either the 32-bit entry point or the 64-bit entry point, or both. For
710 compatibility with existing SMBIOS parsers, an implementation should provide the 32-bit entry point.

711 If an implementation provides both a 32-bit and a 64-bit entry point and they point to distinct SMBIOS
712 structure tables, the 32-bit table must be a consistent subset of the 64-bit table: for any structure type
713 (between 0 and 125) that exists in the 32-bit table, there must be a corresponding structure in the 64-bit
714 table. The 64-bit table may contain structure types not found in the 32-bit table.

715 See ANNEX B for pseudo-code using this convention.

716 NOTE 1   The table convention is required for SMBIOS version 2.2 and later implementations.

717 NOTE 2   The information that is present in the table-based structures is boot-time static, and SMBIOS consumers
718 should not expect the information to be updated during normal system operations.

5.2.1   SMBIOS 2.1 (32-bit) Entry Point

720 The 32-bit SMBIOS Entry Point Structure is described in Table 1.

721 On non-UEFI systems, the 32-bit SMBIOS Entry Point structure, can be located by application software
722 by searching for the anchor-string on paragraph (16-byte) boundaries within the physical memory address
723 range 000F0000h to 000FFFFFh. This entry point encapsulates an intermediate anchor string that is used
724 by some existing DMI browsers.

725 On UEFI-based systems, the SMBIOS Entry Point structure can be located by looking in the EFI
726 Configuration Table for the SMBIOS GUID (SMBIOS_TABLE_GUID, {EB9D2D31-2D88-11D3-9A16-

727 0090273FC14D}) and using the associated pointer. See section 4.6 of the *UEFI Specification* for details.
728 See section 2.3 of the *UEFI Specification* for how to report the containing memory type.

729 NOTE   While the SMBIOS Major and Minor Versions (offsets 06h and 07h) currently duplicate the information that
730 is present in the SMBIOS BCD Revision (offset 1Eh), they provide a path for future growth in this specification. The
731 BCD Revision, for example, provides only a single digit for each of the major and minor version numbers.

732   **Table 1 – SMBIOS 2.1 (32-bit) Entry Point structure**

Offset	Name	Length	Description
00h	Anchor String	4 BYTEs	_SM_, specified as four ASCII characters (5F 53 4D 5F).
04h	Entry Point Structure Checksum	BYTE	Checksum of the Entry Point Structure (EPS) This value, when added to all other bytes in the EPS, results in the value 00h (using 8-bit addition calculations). Values in the EPS are summed starting at offset 00h, for Entry Point Length bytes.
05h	Entry Point Length	BYTE	Length of the Entry Point Structure, starting with the Anchor String field, in bytes, currently 1Fh NOTE: This value was incorrectly stated in version 2.1 of this specification as 1Eh. Because of this, there might be version 2.1 implementations that use either the 1Eh or the 1Fh value, but version 2.2 or later implementations must use the 1Fh value.
06h	SMBIOS Major Version	BYTE	Major version of this specification implemented in the table structures (for example, the value is 0Ah for revision 10.22 and 02h for revision 2.1)
07h	SMBIOS Minor Version	BYTE	Minor version of this specification implemented in the table structures (for example, the value is 16h for revision 10.22 and 01h for revision 2.1)
08h	Maximum Structure Size	WORD	Size of the largest SMBIOS structure, in bytes, and encompasses the structure's formatted area and text strings
0Ah	Entry Point Revision	BYTE	EPS revision implemented in this structure and identifies the formatting of offsets 0Bh to 0Fh as follows: 00h Entry Point is based on SMBIOS 2.1 definition; formatted area is reserved and set to all 00h. 01h-FFh Reserved for assignment by this specification
0Bh - 0Fh	Formatted Area	5 BYTEs	Value present in the Entry Point Revision field defines the interpretation to be placed upon these 5 bytes
10h	Intermediate Anchor String	5 BYTEs	_DMI_, specified as five ASCII characters (5F 44 4D 49 5F). NOTE: This field is paragraph-aligned, to allow legacy DMI browsers to find this entry point within the SMBIOS Entry Point Structure.
15h	Intermediate Checksum	BYTE	Checksum of Intermediate Entry Point Structure (IEPS). This value, when added to all other bytes in the IEPS, results in the value 00h (using 8-bit addition calculations). Values in the IEPS are summed starting at offset 10h, for 0Fh bytes.
16h	Structure Table Length	WORD	Total length of SMBIOS Structure Table, pointed to by the Structure Table Address, in bytes
18h	Structure Table Address	DWORD	32-bit physical starting address of the read-only SMBIOS Structure Table, which can start at any 32-bit address This area contains all of the SMBIOS structures fully packed together. These structures can then be parsed to produce exactly the same format as that returned from a Get SMBIOS Structure function call.

Offset	Name	Length	Description
1Ch	Number of SMBIOS Structures	WORD	Total number of structures present in the SMBIOS Structure Table This is the value returned as NumStructures from the Get SMBIOS Information function.
1Eh	SMBIOS BCD Revision	BYTE	Indicates compliance with a revision of this specification It is a BCD value where the upper nibble indicates the major version and the lower nibble the minor version. For revision 2.1, the returned value is 21h. If the value is 00h, only the Major and Minor Versions in offsets 6 and 7 of the Entry Point Structure provide the version information.

733 ### 5.2.2   SMBIOS 3.0 (64-bit) Entry Point

734 The 64-bit SMBIOS Entry Point Structure is described in Table 2.

735 On non-UEFI systems, the 64-bit SMBIOS Entry Point structure can be located by application software by
736 searching for the anchor-string on paragraph (16-byte) boundaries within the physical memory address
737 range 000F0000h to 000FFFFFh.

738 On UEFI-based systems, the SMBIOS Entry Point structure can be located by looking in the EFI
739 Configuration Table for the SMBIOS 3.x GUID (SMBIOS3_TABLE_GUID, {F2FD1544-9794-4A2C-992E-
740 E5BBCF20E394}) and using the associated pointer. See section 4.6 of the UEFI Specification for details.
741 See section 2.3 of the UEFI Specification for how to report the containing memory type.

742   **Table 2- SMBIOS 3.0 (64-bit) Entry Point Structure**

Offset	Name	Length	Description
00h	Anchor String	5 BYTEs	_SM3_, specified as five ASCII characters (5F 53 4D 33 5F).
05h	Entry Point Structure Checksum	BYTE	Checksum of the Entry Point Structure (EPS) This value, when added to all other bytes in the EPS, results in the value 00h (using 8-bit addition calculations). Values in the EPS are summed starting at offset 00h, for Entry Point Length bytes.
06h	Entry Point Length	BYTE	Length of the Entry Point Structure, starting with the Anchor String field, in bytes, currently 18h
07h	SMBIOS Major Version	BYTE	Major version of this specification implemented in the table structures (for example, the value is 0Ah for revision 10.22 and 02h for revision 2.1)
08h	SMBIOS Minor Version	BYTE	Minor version of this specification implemented in the table structures (for example, the value is 16h for revision 10.22 and 01h for revision 2.1)
09h	SMBIOS Docrev	BYTE	Identifies the docrev of this specification implemented in the table structures (for example, the value is 00h for revision 10.22.0 and 01h for revision 2.7.1).
0Ah	Entry Point Revision	BYTE	EPS revision implemented in this structure and identifies the formatting of offsets 0Bh and beyond as follows: 00h Reserved for assignment by this specification 01h Entry Point is based on SMBIOS 3.0 definition; 02h-FFh Reserved for assignment by this specification; offsets 0Ch-17h are defined per revision 01h
0Bh	Reserved	BYTE	Reserved for assignment by this specification, set to 0

Offset	Name	Length	Description
0Ch	Structure table maximum size	DWORD	Maximum size of SMBIOS Structure Table, pointed to by the Structure Table Address, in bytes. The actual size is guaranteed to be less or equal to the maximum size.
10h	Structure table address	QWORD	The 64-bit physical starting address of the read-only SMBIOS Structure Table, which can start at any 64-bit address. This area contains all of the SMBIOS structures fully packed together.

6   SMBIOS structures

744 The total number of structures can be obtained from the SMBIOS Entry Point Structure (see 5.2). The
745 System Information is presented to an application as a set of structures that are obtained by traversing
746 the SMBIOS structure table referenced by the SMBIOS Entry Point Structure (see 5.2).

6.1   Structure standards

748 Each SMBIOS structure has a formatted section and an optional unformed section. The formatted section
749 of each structure begins with a 4-byte header. Remaining data in the formatted section is determined by
750 the structure type, as is the overall length of the formatted section.

6.1.1   Structure evolution and usage guidelines

752 As the industry evolves, the structures defined in this specification will evolve. To ensure that the
753 evolution occurs in a nondestructive fashion, the following guidelines must be followed:

754 • If a new field is added to an existing structure, that field is added at the end of the formatted
755   area of that structure and the structure's *Length* field is increased by the new field's size.

756 • Any software that interprets a structure shall use the structure's *Length* field to determine the
757   formatted area size for the structure rather than hard-coding or deriving the *Length* from a
758   structure field.

759 • Each structure shall be terminated by a double-null (0000h), either directly following the
760   formatted area (if no strings are present) or directly following the last string. This includes
761   system- and OEM-specific structures and allows upper-level software to easily traverse the
762   structure table. (See structure-termination examples later in this clause.)

763 • The unformed section of the structure is used for passing variable data such as text strings; see
764   6.1.3 for more information.

765 • When an enumerated field's values are controlled by the DMTF, new values can be used as
766   soon as they are defined by the DMTF without requiring an update to this specification.

767 • Starting with version 2.3, each SMBIOS structure type has a *minimum* length — enabling the
768   addition of new, but optional, fields to SMBIOS structures. In no case shall a structure's length
769   result in a field being less than fully populated. For example, a Voltage Probe structure with
770   *Length* of 15h is invalid because the *Nominal Value* field would not be fully specified.

771 • Software that interprets a structure field must verify that the structure's length is sufficient to
772   encompass the optional field; if the length is insufficient, the optional field's value is *Unknown*.
773   For example, if a Voltage Probe structure has a *Length* field of 14h, the probe's *Nominal Value*
774   is *Unknown*. A Voltage Probe structure with *Length* greater than 14h always includes a *Nominal*
775   *Value* field.

776 EXAMPLE 1: BIOS Information with strings:

```
777  BIOS_Info LABEL BYTE
778  db  0                               ; Indicates BIOS Structure Type
779  db  13h                             ; Length of information in bytes
780  dw  ?                               ; Reserved for handle
781  db  01h                             ; String 1 is the Vendor Name
782  db  02h                             ; String 2 is the BIOS version
783  dw  0E800h                          ; BIOS Starting Address
784  db  03h                             ; String 3 is the BIOS Build Date
785  db  1                               ; Size of BIOS ROM is 128K (64K * (1 + 1))
786  dq  BIOS_Char                       ; BIOS Characteristics
787  db  0                               ; BIOS Characteristics Extension Byte 1
788  db  'System BIOS Vendor Name',0   ;
789  db  '4.04',0                       ;
790  db  '00/00/0000',0                 ;
791  db  0                               ; End of strings
```

792 EXAMPLE 2: BIOS Information without strings (example-only):

```
793  BIOS_Info LABEL BYTE
794  db  0            ; Indicates BIOS Structure Type
795  db  13h          ; Length of information in bytes
796  dw  ?            ; Reserved for handle
797  db  00h          ; No Vendor Name provided
798  db  00h          ; No BIOS version provided
799  dw  0E800h       ; BIOS Starting Address
800  db  00h          ; No BIOS Build Date provided
801  db  1            ; Size of BIOS ROM is 128K (64K * (1 + 1))
802  dq  BIOS_Char    ; BIOS Characteristics
803  db  0            ; BIOS Characteristics Extension Byte 1
804  dw  0000h        ; Structure terminator
```

805 ## 6.1.2 Structure header format

806 Each SMBIOS structure begins with a four-byte header as shown in Table 3.

807 **Table 3 – Structure header format description**

Offset	Name	Length	Description
00h	Type	BYTE	Specifies the type of structure. Types 0 through 127 (7Fh) are reserved for and defined by this specification. Types 128 through 256 (80h to FFh) are available for system- and OEM-specific information.
01h	Length	BYTE	Specifies the length of the formatted area of the structure, starting at the Type field. The length of the structure's string-set is not included.

Offset	Name	Length	Description
02h	Handle	WORD	Specifies the structure's handle, a unique 16-bit number in the range 0 to 0FFFEh (for version 2.0) or 0 to 0FEFFh (for version 2.1 and later). The handle can be used with the Get SMBIOS Structure function to retrieve a specific structure; the handle numbers are not required to be contiguous. For version 2.1 and later, handle values in the range 0FF00h to 0FFFFh are reserved for use by this specification.[1] If the system configuration changes, a previously assigned handle might no longer exist. However, after a handle has been assigned by the BIOS, the BIOS cannot re-assign that handle number to another structure.

[1] The *UEFI Platform Initialization Specification* reserves handle number FFFEh for its EFI_SMBIOS_PROTOCOL.Add() function to mean "assign an unused handle number automatically." This number is not used for any other purpose by the SMBIOS specification.

6.1.3 Text strings

Text strings associated with a given SMBIOS structure are appended directly after the formatted portion of the structure. This method of returning string information eliminates the need for application software to deal with pointers embedded in the SMBIOS structure. Each string is terminated with a null (00h) BYTE and the set of strings is terminated with an additional null (00h) BYTE. When the formatted portion of an SMBIOS structure references a string, it does so by specifying a non-zero string number within the structure's string-set. For example, if a string field contains 02h, it references the second string following the formatted portion of the SMBIOS structure. If a string field references no string, a null (0) is placed in that string field. If the formatted portion of the structure contains string-reference fields and all the string fields are set to 0 (no string references), the formatted section of the structure is followed by two null (00h) BYTES. See 6.1.1 for a string-containing example.

NOTE   There is no limit on the length of each individual text string. However, the length of the entire structure table (including all strings) must be reported in the Structure Table Length field of the SMBIOS Structure Table Entry Point (see 5.2.1), which is a WORD field limited to 65,535 bytes.

6.2 Required structures and data

Beginning with SMBIOS version 2.3, compliant SMBIOS implementations include the base set of required structures and data within those structures shown in Table 4. For a detailed list of conformance guidelines, refer to ANNEX A.

NOTE 1   DIG64-compliant systems are only required to provide a type 1 structure (which includes the UUID); see section 4.6.2 of DIG64 for details.

NOTE 2   As of version 2.5 of this specification, structure type 20 is optional.

Table 4 – Required structures and data

Structure Name and Type	Data Requirements
BIOS Information (Type 0)	One and only one structure is present in the structure-table. *BIOS Version* and *BIOS Release Date* strings are non-null; the date field uses a 4-digit year (for example, 1999). All other fields reflect full BIOS support information.
System Information (Type 1)	*Manufacturer* and *Product Name* strings are non-null. *UUID* field identifies the system's non-zero UUID value. *Wake-up Type* field identifies the wake-up source and cannot be Unknown. One and only one structure is present in the structure-table.
System Enclosure (Type 3)	*Manufacturer* string is non-null; the *Type* field identifies the type of enclosure. (Unknown is disallowed.)

Structure Name and Type	Data Requirements
Processor Information (Type 4)	One structure is required for each system processor. The presence of two structures with the *Processor Type* field set to *Central Processor*, for instance, identifies that the system is capable of dual-processor operations.
	Socket Designation string is non-null. *Processor Type*, *Max Speed*, and *Processor Upgrade* fields are all set to "known" values (that is, the Unknown value is disallowed for each field).
	If the associated processor is present (that is, the *CPU Socket Populated* sub-field of the *Status* field indicates that the socket is populated), the *Processor Manufacturer* string is non-null and the *Processor Family*, *Current Speed*, and *CPU Status* sub-field of the *Status* field are all set to "known" values.
	Each of the *Lx Cache Handle* fields is either set to 0xFFFF (no further cache description) or references a valid *Cache Information* structure.
Cache Information (Type 7)	One structure is required for each cache that is external to the processor.
	Socket Designation string is non-null if the cache is external to the processor. If the cache is present (that is, the *Installed Size* is non-zero), the *Cache Configuration* field is set to a "known" value (that is, the Unknown value is disallowed).
System Slots (Type 9)	One structure is required for each upgradeable system slot. A structure is not required if the slot must be populated for proper system operation (for example, the system contains a single memory-card slot).
	Slot Designation string is non-null. *Slot Type*, *Slot Data Bus Width*, *Slot ID*, and *Slot Characteristics 1 & 2* are all set to "known" values.
	If device presence is detectable within the slot (for example, PCI), the *Current Usage* field must be set to either *Available* or *In-use*. Otherwise (for example, ISA), the Unknown value for the field is also allowed.
Physical Memory Array (Type 16)	One structure is required for the system memory.
	Location, *Use*, and *Memory Error Correction* are all set to "known" values. Either *Maximum Capacity* or *Extended Maximum Capacity* must be set to a known, non-zero value. *Number of Memory Devices* is non-zero and identifies the number of *Memory Device* structures that are associated with this *Physical Memory Array*.
Memory Device (Type 17)	One structure is required for each socketed system-memory device, whether or not the socket is currently populated; if the system includes soldered system-memory, one additional structure is required to identify that memory device.
	Device Locator string is set to a non-null value. *Memory Array Handle* contains the handle associated with the *Physical Memory Array* structure to which this device belongs. *Data Width*, *Size*, *Form Factor*, and *Device Set* are all set to "known" values. If the device is present (for instance, *Size* is non-zero), the *Total Width* field is not set to 0xFFFF (Unknown).
Memory Array Mapped Address (Type 19)	One structure is required for each contiguous block of memory addresses mapped to a *Physical Memory Array*.
	Either the pair of *Starting Address* and *Ending Address* is set to a valid address range or the pair of *Extended Starting Address* and *Extended Ending Address* is set to a valid address range. If the pair of *Starting Address* and *Ending Address* is used, *Ending Address* must be larger than *Starting Address*. If the pair of *Extended Starting Address* and *Extended Ending Address* is used, *Extended Ending Address* must be larger than *Extended Starting Address*. Each structure's address range is unique and non-overlapping. *Memory Array Handle* references a *Physical Memory Array* structure. *Partition Width* is non-zero.
System Boot Information (Type 32)	Structure's length is at least 0x0B (for instance, at least one byte of *System Boot Status* is provided).

830 ## 6.3 SMBIOS fields and CIM MOF properties

831 Many of the enumerated values are shared between SMBIOS fields and Common Information Model
832 (CIM) MOF properties. Table 5 identifies the relationships; any additions to these enumerated lists should
833 be reflected in both documents by submitting change requests to mailto:schema-sc@dmtf.org and
834 mailto:smbios@dmtf.org for the CIM-related and SMBIOS-related updates, respectively. Any other
835 enumerated value identified in this specification is controlled by this specification; change requests should
836 be sent to mailto:smbios@dmtf.org.

837 **Table 5 – Relationship between SMBIOS fields and CIM MOF properties**

Name	Clause	MOF Class.Property
Baseboard	7.3.1	Originally, the baseboard feature flags mapped to CIM properties CIM_PhysicalPackage.HotSwappable, CIM_PhysicalPackage.Replaceable, and CIM_PhysicalPackage.Removable. These properties are deprecated and replaced with CIM_PhysicalPackage.RemovalConditions. CIM_Card. RequiresDaughterCard CIM_Card.HostingBoard
Enclosure or Chassis Type	7.4.1	CIM_Chassis.ChassisPackageType
Processor Type	7.5.1	CIM defines a CIM_Processor.Role property, which is a free-form string.
Processor Family	7.5.2	CIM_Processor.Family CIM_ArchitectureCheck.ArchitectureType
Processor Upgrade	7.5.5	CIM_Processor.UpgradeMethod
System Cache Type	7.8.4	CIM_AssociatedCacheMemory.CacheType
Cache Associativity	7.8.5	CIM_AssociatedCacheMemory.Associativity
Slot Data Bus Width	7.10.2	CIM_Slot.MaxDataWidth
Slot Current Usage	7.10.3	CIM handles slot population more explicitly than SMBIOS or DMI, by using a CIM_CardInSlot class to associate the card (CIM_Card) with the slot (CIM_Slot) into which it is inserted.
Memory Array Location	7.17.1	CIM handles memory location more specifically than SMBIOS or DMI, by using a CIM_AssociatedMemory class to associate the memory (CIM_Memory) with the device on which it is installed.
Memory Array Use	7.17.2	CIM handles memory array use more specifically than SMBIOS or DMI, by defining classes that inherit from CIM_Memory to define the specific use (for example, CIM_CacheMemory or CIM_NonVolatileStorage).
Memory Array Error Correction Types	7.17.3	CIM_Memory.ErrorMethodology CIM maps memory error correction types into string values rather than enumerations.
Memory Device Form Factor	7.18.1	CIM_PhysicalMemory.FormFactor is inherited from CIM_Chip.FormFactor and uses a different enumeration than SMBIOS.
Memory Device Type	7.18.2	CIM_PhysicalMemory.MemoryType uses a different enumeration than SMBIOS.
Memory Error Type	7.19.1	CIM_MemoryError.ErrorInfo values 0Ch-0Eh have no match in the CIM_MemoryError.ErrorInfo property; instead, they are reported through CIM_MemoryError.CorrectableError (Boolean).
Memory Error Operation	7.19.3	CIM_MemoryError.Access
Pointing Device Type	7.22.1	CIM_PointingDevice.PointingType

Name	Clause	MOF Class.Property
Portable Battery Device Chemistry	7.23.1	CIM_Battery.Chemistry
Power Supply Type	7.40.1	Linear/switching is reported through CIM_PowerSupply.IsSwitchingSupply (Boolean).
Power Supply Input Voltage Range Switching	7.40.1	CIM_PowerSupply.TypeOfRangeSwitching

838 7   Structure definitions

839 7.1   BIOS Information (Type 0)

840   Table 6 shows the BIOS Information structure.

841   **Table 6 – BIOS Information (Type 0) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.0+	Type	BYTE	0	BIOS Information indicator
01h	2.0+	Length	BYTE	Varies	12h + number of BIOS Characteristics Extension Bytes. If no Extension Bytes are used the Length is 12h. For version 2.1 and 2.2 implementations, the length is 13h because one extension byte is defined. For version 2.3 and later implementations, the length is at least 14h because two extension bytes are defined. For version 2.4 and later implementations, the length is at least 18h because bytes 14-17h are defined.
02h	2.0+	Handle	WORD	Varies	
04h	2.0+	Vendor	BYTE	STRING	String number of the BIOS Vendor's Name.
05h	2.0+	BIOS Version	BYTE	STRING	String number of the BIOS Version. This value is a free-form string that may contain Core and OEM version information.
06h	2.0+	BIOS Starting Address Segment	WORD	Varies	Segment location of BIOS starting address (for example, 0E800h). NOTE: The size of the runtime BIOS image can be computed by subtracting the Starting Address Segment from 10000h and multiplying the result by 16.
08h	2.0+	BIOS Release Date	BYTE	STRING	String number of the BIOS release date. The date string, if supplied, is in either mm/dd/yy or mm/dd/yyyy format. If the year portion of the string is two digits, the year is assumed to be 19yy. NOTE: The mm/dd/yyyy format is required for SMBIOS version 2.3 and later.
09h	2.0+	BIOS ROM Size	BYTE	Varies (n)	Size (n) where 64K * (n+1) is the size of the physical device containing the BIOS, in bytes

Offset	Spec. Version	Name	Length	Value	Description
0Ah	2.0+	BIOS Characteristics	QWORD	Bit Field	Defines which functions the BIOS supports: PCI, PCMCIA, Flash, etc. (see 7.1.1).
12h	2.4+	BIOS Characteristics Extension Bytes	Zero or more BYTEs	Bit Field	Optional space reserved for future supported functions. The number of Extension Bytes that is present is indicated by the Length in offset 1 minus 12h. See 7.1.2 for extensions defined for version 2.1 and later implementations. For version 2.4 and later implementations, two BIOS Characteristics Extension Bytes are defined (12-13h) and bytes 14-17h are also defined.
14h	2.4+	System BIOS Major Release	BYTE	Varies	Identifies the major release of the System BIOS; for example, the value is 0Ah for revision 10.22 and 02h for revision 2.1. This field or the System BIOS Minor Release field or both are updated each time a System BIOS update for a given system is released. If the system does not support the use of this field, the value is 0FFh for both this field and the System BIOS Minor Release field.
15h	2.4+	System BIOS Minor Release	BYTE	Varies	Identifies the minor release of the System BIOS; for example, the value is 16h for revision 10.22 and 01h for revision 2.1.
16h	2.4+	Embedded Controller Firmware Major Release	BYTE	Varies	Identifies the major release of the embedded controller firmware; for example, the value would be 0Ah for revision 10.22 and 02h for revision 2.1. This field or the Embedded Controller Firmware Minor Release field or both are updated each time an embedded controller firmware update for a given system is released. If the system does not have field upgradeable embedded controller firmware, the value is 0FFh.
17h	2.4+	Embedded Controller Firmware Minor Release	BYTE	Varies	Identifies the minor release of the embedded controller firmware; for example, the value is 16h for revision 10.22 and 01h for revision 2.1. If the system does not have field upgradeable embedded controller firmware, the value is 0FFh.

842   **7.1.1   BIOS Characteristics**

843   Table 7 shows the BIOS Characteristics layout.

844   **Table 7 – BIOS Characteristics**

QWORD Bit Position	Meaning If Set
Bit 0	Reserved.
Bit 1	Reserved.
Bit 2	Unknown.
Bit 3	BIOS Characteristics are not supported.
Bit 4	ISA is supported.
Bit 5	MCA is supported.
Bit 6	EISA is supported.
Bit 7	PCI is supported.
Bit 8	PC card (PCMCIA) is supported.
Bit 9	Plug and Play is supported.
Bit 10	APM is supported.
Bit 11	BIOS is upgradeable (Flash).
Bit 12	BIOS shadowing is allowed.
Bit 13	VL-VESA is supported.
Bit 14	ESCD support is available.
Bit 15	Boot from CD is supported.
Bit 16	Selectable boot is supported.
Bit 17	BIOS ROM is socketed.
Bit 18	Boot from PC card (PCMCIA) is supported.
Bit 19	EDD specification is supported.
Bit 20	Int 13h — Japanese floppy for NEC 9800 1.2 MB (3.5", 1K bytes/sector, 360 RPM) is supported.
Bit 21	Int 13h — Japanese floppy for Toshiba 1.2 MB (3.5", 360 RPM) is supported.
Bit 22	Int 13h — 5.25" / 360 KB floppy services are supported.
Bit 23	Int 13h — 5.25" /1.2 MB floppy services are supported.
Bit 24	Int 13h — 3.5" / 720 KB floppy services are supported.
Bit 25	Int 13h — 3.5" / 2.88 MB floppy services are supported.
Bit 26	Int 5h, print screen Service is supported.
Bit 27	Int 9h, 8042 keyboard services are supported.
Bit 28	Int 14h, serial services are supported.
Bit 29	Int 17h, printer services are supported.
Bit 30	Int 10h, CGA/Mono Video Services are supported.
Bit 31	NEC PC-98.
Bits32:47	Reserved for BIOS vendor.
Bits 48:63	Reserved for system vendor.

845   **7.1.2 BIOS Characteristics Extension Bytes**

846   NOTE   All Characteristics Extension Bytes are reserved for assignment through this specification.

847 **7.1.2.1   BIOS Characteristics Extension Byte 1**

848   Table 8 shows the BIOS Characteristics Extension Byte 1 layout. This information, available for SMBIOS
849   version 2.1 and later, appears at offset 12h within the BIOS Information structure.

850   **Table 8 – BIOS Characteristics Extension Byte 1**

Byte Bit Position	Meaning If Set
Bit 0	ACPI is supported.
Bit 1	USB Legacy is supported.
Bit 2	AGP is supported.
Bit 3	I2O boot is supported.
Bit 4	LS-120 SuperDisk boot is supported.
Bit 5	ATAPI ZIP drive boot is supported.
Bit 6	1394 boot is supported.
Bit 7	Smart battery is supported.

851 **7.1.2.2   BIOS Characteristics Extension Byte 2**

852   Table 9 shows the BIOS Characteristics for Extension Byte 2 layout. This information, available for
853   SMBIOS version 2.3 and later, appears at offset 13h within the BIOS Information structure.

854   **Table 9 – BIOS Characteristics Extension Byte 2**

Byte Bit Position	Meaning If Set
Bit 0	BIOS Boot Specification is supported.
Bit 1	Function key-initiated network service boot is supported. When function key-uninitiated network service boot is not supported, a network adapter option ROM may choose to offer this functionality on its own, thus offering this capability to legacy systems. When the function is supported, the network adapter option ROM shall not offer this capability.
Bit 2	Enable targeted content distribution. The manufacturer has ensured that the SMBIOS data is useful in identifying the computer for targeted delivery of model-specific software and firmware content through third-party content distribution services.
Bit 3	UEFI Specification is supported.
Bit 4	SMBIOS table describes a virtual machine. (If this bit is not set, no inference can be made about the virtuality of the system.)
Bits 5:7	Reserved for future assignment by this specification.

855 ## 7.2   System Information (Type 1)

856   The information in this structure defines attributes of the overall system and is intended to be associated
857   with the *Component ID* group of the system's MIF. An SMBIOS implementation is associated with a single
858   system instance and contains one and only one System Information (Type 1) structure. Table 10 shows
859   the contents of this structure.

860   **Table 10 – System Information (Type 1) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.0+	Type	BYTE	1	System Information indicator
01h	2.0+	Length	BYTE	08h or 19h	Length dependent on version supported: • 08h for 2.0 • 19h for 2.1 – 2.3.4 • 1Bh for 2.4 and later
02h	2.0+	Handle	WORD	Varies	
04h	2.0+	Manufacturer	BYTE	STRING	Number of null-terminated string
05h	2.0+	Product Name	BYTE	STRING	Number of null-terminated string
06h	2.0+	Version	BYTE	STRING	Number of null-terminated string
07h	2.0+	Serial Number	BYTE	STRING	Number of null-terminated string
08h	2.1+	UUID	16 BYTEs	Varies	Universal unique ID number; see 7.2.1.
18h	2.1+	Wake-up Type	BYTE	ENUM	Identifies the event that caused the system to power up. See 7.2.2.
19h	2.4+	SKU Number	BYTE	STRING	Number of null-terminated string This text string identifies a particular computer configuration for sale. It is sometimes also called a product ID or purchase order number. This number is frequently found in existing fields, but there is no standard format. Typically for a given system board from a given OEM, there are tens of unique processor, memory, hard drive, and optical drive configurations.
1Ah	2.4+	Family	BYTE	STRING	Number of null-terminated string This text string identifies the family to which a particular computer belongs. A family refers to a set of computers that are similar but not identical from a hardware or software point of view. Typically, a family is composed of different computer models, which have different configurations and pricing points. Computers in the same family often have similar branding and cosmetic features.

861   **7.2.1   System — UUID**

862   A UUID is an identifier that is designed to be unique across both time and space. It requires no central
863   registration process. The UUID is 128 bits long. Its format is described in RFC4122, but the actual field
864   contents are opaque and not significant to the SMBIOS specification, which is only concerned with the
865   byte order. Table 11 shows the field names; these field names, particularly for multiplexed fields, follow
866   historical practice.

867 **Table 11 – UUID byte order and RFC4122 field names**

Offset	RFC 4122 Name	Length	Value	Description
00h	time_low	DWORD	Varies	Low field of the timestamp
04h	time_mid	WORD	Varies	Middle field of the timestamp
06h	time_hi_and_version	WORD	Varies	High field of the timestamp multiplexed with the version number
08h	clock_seq_hi_and_reserved	BYTE	Varies	High field of the clock sequence multiplexed with the variant
09h	clock_seq_low	BYTE	Varies	Low field of the clock sequence
0Ah	Node	6 BYTEs	Varies	Spatially unique node identifier

868 Although RFC4122 recommends network byte order for all fields, the PC industry (including the ACPI,
869 UEFI, and Microsoft specifications) has consistently used little-endian byte encoding for the first three
870 fields: *time_low*, *time_mid*, *time_hi_and_version*. The same encoding, also known as *wire format*, should
871 also be used for the SMBIOS representation of the UUID.

872 The UUID {00112233-4455-6677-8899-AABBCCDDEEFF} would thus be represented as:

873 33 22 11 00 55 44 77 66 88 99 AA BB CC DD EE FF.

874 If the value is all FFh, the ID is not currently present in the system, but it can be set. If the value is all 00h,
875 the ID is not present in the system.

876 **7.2.2 System — Wake-up Type**

877 Table 12 shows what the byte values mean for the System — Wake-up Type field.

878 **Table 12 – System: Wake-up Type field**

Byte Value	Meaning
00h	Reserved
01h	Other
02h	Unknown
03h	APM Timer
04h	Modem Ring
05h	LAN Remote
06h	Power Switch
07h	PCI PME#
08h	AC Power Restored

879 **7.3 Baseboard (or Module) Information (Type 2)**

880 As shown in Table 13, the information in this structure defines attributes of a system baseboard (for
881 example, a motherboard, planar, server blade, or other standard system module).

882 NOTE   If more than one Type 2 structure is provided by an SMBIOS implementation, each structure shall include
883 the *Number of Contained Object Handles* and *Contained Object Handles* fields to specify which system elements are
884 contained on which boards. If a single Type 2 structure is provided and the contained object information is not

885   present[1], or if no Type 2 structure is provided, all system elements identified by the SMBIOS implementation are
886   associated with a single motherboard.

887   **Table 13 – Baseboard (or Module) Information (Type 2) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	2	Baseboard Information indicator
01h	Length	BYTE	Varies	Length of the structure, at least 08h
02h	Handle	WORD	Varies	
04h	Manufacturer	BYTE	STRING	Number of null-terminated string
05h	Product	BYTE	STRING	Number of null-terminated string
06h	Version	BYTE	STRING	Number of null-terminated string
07h	Serial Number	BYTE	STRING	Number of null-terminated string
08h	Asset Tag	BYTE	STRING	Number of a null-terminated string
09h	Feature Flags	BYTE	Bit Field	Collection of flags that identify features of this baseboard; see 7.3.1
0Ah	Location in Chassis	BYTE	STRING	Number of a null-terminated string that describes this board's location within the chassis referenced by the Chassis Handle (described below in this table) NOTE: This field supports a CIM_Container class mapping where: • LocationWithinContainer is this field. • GroupComponent is the chassis referenced by Chassis Handle. • PartComponent is this baseboard.
0Bh	Chassis Handle	WORD	Varies	Handle, or instance number, associated with the chassis in which this board resides (see 7.4)
0Dh	Board Type	BYTE	ENUM	Type of board (see 7.3.2)
0Eh	Number of Contained Object Handles (n)	BYTE	Varies	Number (0 to 255) of Contained Object Handles that follow
0Fh	Contained Object Handles	n WORDs	Varies	List of handles of other structures (for example, Baseboard, Processor, Port, System Slots, Memory Device) that are contained by this baseboard

888   **7.3.1   Baseboard — feature flags**

889   Table 14 shows the baseboard feature flags.

890   NOTE   Refer to 6.3 for the CIM properties associated with these bit fields.

---

[1] This information is "not present" if either the *Length* of the Type 2 structure is less than 14 (0Eh) or the *Number of Contained Object Handles* field at offset 0Dh is set to 0.

891 **Table 14 – Baseboard: feature flags**

Bit Position(s)	Description
7:5	Reserved for future definition by this specification; set to 000b.
4	Set to 1 if the board is hot swappable; it is possible to replace the board with a physically different but equivalent board while power is applied to the board. The board is inherently replaceable and removable.
3	Set to 1 if the board is replaceable; it is possible to replace (either as a field repair or as an upgrade) the board with a physically different board. The board is inherently removable.
2	Set to 1 if the board is removable; it is designed to be taken in and out of the chassis without impairing the function of the chassis.
1	Set to 1 if the board requires at least one daughter board or auxiliary card to function properly.
0	Set to 1 if the board is a hosting board (for example, a motherboard).

892 ### 7.3.2 Baseboard — Board Type

893 Table 15 shows the byte values for the Baseboard — Board Type field.

894 NOTE These enumerations are also used within the System Enclosure or Chassis (Type 3) structure's *Contained*
895 *Element* record (see 7.4).

896 **Table 15 – Baseboard: Board Type**

Byte Value	Meaning
01h	Unknown
02h	Other
03h	Server Blade
04h	Connectivity Switch
05h	System Management Module
06h	Processor Module
07h	I/O Module
08h	Memory Module
09h	Daughter board
0Ah	Motherboard (includes processor, memory, and I/O)
0Bh	Processor/Memory Module
0Ch	Processor/IO Module
0Dh	Interconnect board

897 ## 7.4 System Enclosure or Chassis (Type 3)

898 The information in this structure (see Table 16) defines attributes of the system's mechanical
899 enclosure(s). For example, if a system included a separate enclosure for its peripheral devices, two
900 structures would be returned: one for the main system enclosure and the second for the peripheral device
901 enclosure. The additions to this structure in version 2.1 of this specification support the population of the
902 CIM_Chassis class.

903   **Table 16 – System Enclosure or Chassis (Type 3) structure**

Offset	Specification Version	Name	Length	Value	Description
00h	2.0+	Type	BYTE	3	System Enclosure indicator
01h	2.0+	Length	BYTE	Varies	09h for version 2.0 implementations or a minimum of 0Dh for version 2.1 and later implementations
02h	2.0+	Handle	WORD	Varies	
04h	2.0+	Manufacturer	BYTE	STRING	Number of null-terminated string
05h	2.0+	Type	BYTE	Varies	Bit 7 Chassis lock is present if 1. Otherwise, either a lock is not present or it is unknown if the enclosure has a lock. Bits 6:0 Enumeration value; see below.
06h	2.0+	Version	BYTE	STRING	Number of null-terminated string
07h	2.0+	Serial Number	BYTE	STRING	Number of null-terminated string
08h	2.0+	Asset Tag Number	BYTE	STRING	Number of null-terminated string
09h	2.1+	Boot-up State	BYTE	ENUM	State of the enclosure when it was last booted; see 7.4.2 for definitions
0Ah	2.1+	Power Supply State	BYTE	ENUM	State of the enclosure's power supply (or supplies) when last booted; see 7.4.2 for definitions
0Bh	2.1+	Thermal State	BYTE	ENUM	Thermal state of the enclosure when last booted; see 7.4.2 for definitions
0Ch	2.1+	Security Status	BYTE	ENUM	Physical security status of the enclosure when last booted; see 7.4.3 for definitions
0Dh	2.3+	OEM-defined	DWORD	Varies	OEM- or BIOS vendor-specific information
11h	2.3+	Height	BYTE	Varies	Height of the enclosure, in 'U's A U is a standard unit of measure for the height of a rack or rack-mountable component and is equal to 1.75 inches or 4.445 cm. A value of 00h indicates that the enclosure height is unspecified.
12h	2.3+	Number of Power Cords	BYTE	Varies	Number of power cords associated with the enclosure or chassis A value of 00h indicates that the number is unspecified.
13h	2.3+	Contained Element Count (n)	BYTE	Varies	Number of Contained Element records that follow, in the range 0 to 255 Each Contained Element group comprises *m* bytes, as specified by the Contained Element Record Length field that follows. If no Contained Elements are included, this field is set to 0.

Offset	Specification Version	Name	Length	Value	Description
14h	2.3+	Contained Element Record Length (m)	BYTE	Varies	Byte length of each Contained Element record that follows, in the range 0 to 255 If no Contained Elements are included, this field is set to 0. For version 2.3.2 and later of this specification, this field is set to at least 03h when Contained Elements are specified.
15h	2.3+	Contained Elements	n * m BYTEs	Varies	Elements, possibly defined by other SMBIOS structures, present in this chassis; see 7.4.4 for definitions
15h + n*m	2.7+	SKU Number	BYTE	STRING	Number of null-terminated string describing the chassis or enclosure SKU number

904   ### 7.4.1   System Enclosure or Chassis Types

905   Table 17 shows the byte values for the System Enclosure or Chassis Types field.

906   NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

907   **Table 17 – System Enclosure or Chassis Types**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Desktop
04h	Low Profile Desktop
05h	Pizza Box
06h	Mini Tower
07h	Tower
08h	Portable
09h	Laptop
0Ah	Notebook
0Bh	Hand Held
0Ch	Docking Station
0Dh	All in One
0Eh	Sub Notebook
0Fh	Space-saving
10h	Lunch Box
11h	Main Server Chassis
12h	Expansion Chassis
13h	SubChassis
14h	Bus Expansion Chassis
15h	Peripheral Chassis

Byte Value	Meaning
16h	RAID Chassis
17h	Rack Mount Chassis
18h	Sealed-case PC
19h	Multi-system chassis When this value is specified by an SMBIOS implementation, the physical chassis associated with this structure supports multiple, independently reporting physical systems—regardless of the chassis' current configuration. Systems in the same physical chassis are required to report the same value in this structure's Serial Number field. For a chassis that may also be configured as either a single system or multiple physical systems, the Multi-system chassis value is reported even if the chassis is currently configured as a single system. This allows management applications to recognize the multi-system potential of the chassis.
1Ah	Compact PCI
1Bh	Advanced TCA
1Ch	Blade An SMBIOS implementation for a Blade would contain a Type 3 Chassis structure for the individual Blade system as well as one for the Blade Enclosure that completes the Blade system.
1Dh	Blade Enclosure A Blade Enclosure is a specialized chassis that contains a set of Blades. It provides much of the non-core computing infrastructure for a set of Blades (power, cooling, networking, etc.). A Blade Enclosure may itself reside inside a Rack or be a standalone chassis.
1Eh	Tablet
1Fh	Convertible
20h	Detachable

7.4.2 System Enclosure or Chassis States

Table 18 shows the byte values for the System Enclosure or Chassis States field.

Table 18 – System Enclosure or Chassis States

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Safe
04h	Warning
05h	Critical
06h	Non-recoverable

911 ### 7.4.3 System Enclosure or Chassis Security Status

912 Table 19 shows the byte values for the System Enclosure or Chassis Security Status field.

913 **Table 19 – System Enclosure or Chassis Security Status Field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	None
04h	External interface locked out
05h	External interface enabled

914 ### 7.4.4 System Enclosure or Chassis — Contained Elements

915 Each *Contained Element* record consists of sub-fields that further describe elements contained by the
916 chassis, as shown in Table 20. Relative offset and size of fields within each record shall remain the same
917 in future revisions to this specification, but new fields might be added to the end of the current definitions.

918 **Table 20 – System Enclosure or Chassis: Contained Elements**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.3+	Contained Element Type	BYTE	Bit Field	Specifies the type of element associated with this record: Bit(s) — Meaning 7 — <u>Type Select</u>. Identifies whether the Type contains an SMBIOS structure type enumeration (1) or an SMBIOS Baseboard Type enumeration (0). 6:0 — <u>Type</u>. Specifies either an SMBIOS Board Type enumeration (see 7.3.2 for definitions) or an SMBIOS structure type, dependent on the setting of the Type Select. For example, a contained Power Supply is specified as A7h (1 0100111b) — the MSB is 1, so the remaining seven bits (27h = 39) represent an SMBIOS structure type; structure type 39 represents a System Power Supply. A contained Server Blade is specified as 03h — the MSB is 0, so the remaining seven bits represent an SMBIOS board type; board type 03h represents a Server Blade.
01h	2.3+	Contained Element Minimum	BYTE	Varies	Specifies the minimum number of the element type that can be installed in the chassis for the chassis to properly operate, in the range 0 to 254. The value 255 (0FFh) is reserved for future definition by this specification.
02h	2.3+	Contained Element Maximum	BYTE	Varies	Specifies the maximum number of the element type that can be installed in the chassis, in the range 1 to 255. The value 0 is reserved for future definition by this specification.

919 ## 7.5   Processor Information (Type 4)

920 The information in this structure (see Table 21) defines the attributes of a single processor; a separate
921 structure instance is provided for each system processor socket/slot. For example, a system with an
922 IntelDX2™ processor would have a single structure instance while a system with an IntelSX2™ processor
923 would have a structure to describe the main CPU and a second structure to describe the 80487 co-
924 processor.

925 NOTE   One structure is provided for each processor instance in a system. For example, a system that supports up
926 to two processors includes two *Processor Information* structures — even if only one processor is currently installed.
927 Software that interprets the SMBIOS information can count the *Processor Information* structures to determine the
928 maximum possible configuration of the system.

929 **Table 21 – Processor Information (Type 4) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.0+	Type	BYTE	4	Processor Information indicator
01h	2.0+	Length	BYTE	Varies	Length is 1Ah for version 2.0 implementations; 23h for 2.3; 28h for 2.5; 2Ah for 2.6, and 30h for version 3.0 and later implementations.
02h	2.0+	Handle	WORD	Varies	
04h	2.0+	Socket Designation	BYTE	STRING	String number for Reference Designation EXAMPLE: 'J202',0
05h	2.0+	Processor Type	BYTE	ENUM	See 7.5.1.
06h	2.0+	Processor Family	BYTE	ENUM	See 7.5.2.
07h	2.0+	Processor Manufacturer	BYTE	STRING	String number of Processor Manufacturer
08h	2.0+	Processor ID	QWORD	Varies	Raw processor identification data See 7.5.3 for details.
10h	2.0+	Processor Version	BYTE	STRING	String number describing the Processor
11h	2.0+	Voltage	BYTE	Varies	See 7.5.4.
12h	2.0+	External Clock	WORD	Varies	External Clock Frequency, in MHz If the value is unknown, the field is set to 0.
14h	2.0+	Max Speed	WORD	Varies	Maximum processor speed (in MHz) supported by the system for this processor socket 0E9h is for a 233 MHz processor. If the value is unknown, the field is set to 0. NOTE: This field identifies a capability for the system, not the processor itself.
16h	2.0+	Current Speed	WORD	Varies	Same format as Max Speed NOTE: This field identifies the processor's speed at system boot, and the *Processor ID* field implies the processor's additional speed characteristics (that is, single speed or multiple speed).

Offset	Spec. Version	Name	Length	Value	Description
18h	2.0+	Status	BYTE	Varies	Bit 7 Reserved, must be zero Bit 6 CPU Socket Populated 1 – CPU Socket Populated 0 – CPU Socket Unpopulated Bits 5:3 Reserved, must be zero Bits 2:0 CPU Status 0h – Unknown 1h – CPU Enabled 2h – CPU Disabled by User through BIOS Setup 3h – CPU Disabled By BIOS (POST Error) 4h – CPU is Idle, waiting to be enabled. 5-6h – Reserved 7h – Other
19h	2.0+	Processor Upgrade	BYTE	ENUM	See 7.5.5.
1Ah	2.1+	L1 Cache Handle	WORD	Varies	Handle of a Cache Information structure that defines the attributes of the primary (Level 1) cache for this processor For version 2.1 and version 2.2 implementations, the value is 0FFFFh if the processor has no L1 cache. For version 2.3 and later implementations, the value is 0FFFFh if the Cache Information structure is not provided.[1]
1Ch	2.1+	L2 Cache Handle	WORD	Varies	Handle of a Cache Information structure that defines the attributes of the secondary (Level 2) cache for this processor For version 2.1 and version 2.2 implementations, the value is 0FFFFh if the processor has no L2 cache. For version 2.3 and later implementations, the value is 0FFFFh if the Cache Information structure is not provided.[1]
1Eh	2.1+	L3 Cache Handle	WORD	Varies	Handle of a Cache Information structure that defines the attributes of the tertiary (Level 3) cache for this processor For version 2.1 and version 2.2 implementations, the value is 0FFFFh if the processor has no L3 cache. For version 2.3 and later implementations, the value is 0FFFFh if the Cache Information structure is not provided.[1]

Offset	Spec. Version	Name	Length	Value	Description
20h	2.3+	Serial Number	BYTE	STRING	String number for the serial number of this processor This value is set by the manufacturer and normally not changeable.
21h	2.3+	Asset Tag	BYTE	STRING	String number for the asset tag of this processor
22h	2.3+	Part Number	BYTE	STRING	String number for the part number of this processor This value is set by the manufacturer and normally not changeable.
23h	2.5+	Core Count	BYTE	Varies	Number of cores per processor socket See 7.5.6. If the value is unknown, the field is set to 0. For core counts of 256 or greater, the *Core Count* field is set to FFh and the *Core Count 2* field is set to the number of cores.
24h	2.5+	Core Enabled	BYTE	Varies	Number of enabled cores per processor socket See 7.5.7. If the value is unknown, the field is set 0. For core counts of 256 or greater, the *Core Enabled* field is set to FFh and the *Core Enabled 2* field is set to the number of enabled cores.
25h	2.5+	Thread Count	BYTE	Varies	Number of threads per processor socket See 7.5.8. If the value is unknown, the field is set to 0. For thread counts of 256 or greater, the *Thread Count* field is set to FFh and the *Thread Count 2* field is set to the number of threads.
26h	2.5+	Processor Characteristics	WORD	Bit Field	Defines which functions the processor supports See 7.5.9.
28h	2.6+	Processor Family 2	WORD	Enum	See 7.5.2.
2Ah	3.0+	Core Count 2	WORD	Varies	Number of Cores per processor socket. Supports core counts >255. If this field is present, it holds the core count for the processor socket. *Core Count* will also hold the core count, except for core counts that are 256 or greater. In that case, *Core Count* shall be set to FFh and *Core Count 2* will hold the count.. See 7.5.6. Legal values: 0000h = unknown 0001h-00FFh = core counts 1 to 255. Matches *Core Count* value. 0100h-FFFEh = Core counts 256 to 65534, respectively. FFFFh = reserved.

Offset	Spec. Version	Name	Length	Value	Description
2Ch	3.0+	Core Enabled 2	WORD	Varies	Number of enabled cores per processor socket. Supports core enabled counts >255. If this field is present, it holds the core enabled count for the processor socket. *Core Enabled* will also hold the core enabled count, except for core counts that are 256 or greater. In that case, *Core Enabled* shall be set to FFh and *Core Enabled 2* will hold the count. See 7.5.7. Legal values: 0000h = unknown 0001h-00FFh = core enabled counts 1 to 255. Matches *Core Enabled* value. 0100h-FFFEh = core enabled counts 256 to 65534, respectively. FFFFh = reserved.
2Eh	3.0+	Thread Count 2	WORD	Varies	Number of threads per processor socket. Supports thread counts >255. If this field is present, it holds the thread count for the processor socket. *Thread Count* will also hold the thread count, except for thread counts that are 256 or greater. In that case, *Thread Count* shall be set to FFh and *Thread Count 2* will hold the count. See 7.5.8. Legal values:0000h = unknown 0001h-00FFh = thread counts 1 to 255. Matches *Thread Count* value. 0100h-FFFEh = thread counts 256 to 65534, respectively. FFFFh = reserved.

[1] Beginning with version 2.3 implementations, if the *Cache Handle* is 0FFFFh, management software must make no assumptions about the cache's attributes and should report all cache-related attributes as unknown. The definitive absence of a specific cache is identified by referencing a *Cache Information* structure and setting that structure's *Installed Size* field to 0.

7.5.1 Processor Information — Processor Type

931 Table 22 shows what the byte values mean for the Processor Information — Processor Type field.

932 NOTE Refer to 6.3 for the CIM properties associated with this enumerated value.

933 **Table 22 – Processor Information: Processor Type field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Central Processor
04h	Math Processor
05h	DSP Processor
06h	Video Processor

934   **7.5.2   Processor Information — Processor Family**

935   Table 23 details the values for the Processor Information — Processor Family field.

936   NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value. ® and ™ in this table are
937   equivalent to (R) and (TM) in the MOF file.

938   **Table 23 – Processor Information: Processor Family field**

Hex Value	Decimal Value	Meaning
01h	1	Other
02h	2	Unknown
03h	3	8086
04h	4	80286
05h	5	Intel386™ processor
06h	6	Intel486™ processor
07h	7	8087
08h	8	80287
09h	9	80387
0Ah	10	80487
0Bh	11	Intel® Pentium® processor
0Ch	12	Pentium® Pro processor
0Dh	13	Pentium® II processor
0Eh	14	Pentium® processor with MMX™ technology
0Fh	15	Intel® Celeron® processor
10h	16	Pentium® II Xeon™ processor
11h	17	Pentium® III processor
12h	18	M1 Family
13h	19	M2 Family
14h	20	Intel® Celeron® M processor
15h	21	Intel® Pentium® 4 HT processor
16h-17h	22-23	Available for assignment
18h	24	AMD Duron™ Processor Family [1]
19h	25	K5 Family [1]
1Ah	26	K6 Family [1]
1Bh	27	K6-2 [1]
1Ch	28	K6-3 [1]
1Dh	29	AMD Athlon™ Processor Family [1]
1Eh	30	AMD29000 Family
1Fh	31	K6-2+
20h	32	Power PC Family
21h	33	Power PC 601
22h	34	Power PC 603
23h	35	Power PC 603+
24h	36	Power PC 604

Hex Value	Decimal Value	Meaning
25h	37	Power PC 620
26h	38	Power PC x704
27h	39	Power PC 750
28h	40	Intel® Core™ Duo processor
29h	41	Intel® Core™ Duo mobile processor
2Ah	42	Intel® Core™ Solo mobile processor
2Bh	43	Intel® Atom™ processor
2Ch	44	Intel® Core™ M processor
2Dh-2Fh	45-47	Available for assignment
30h	48	Alpha Family [2]
31h	49	Alpha 21064
32h	50	Alpha 21066
33h	51	Alpha 21164
34h	52	Alpha 21164PC
35h	53	Alpha 21164a
36h	54	Alpha 21264
37h	55	Alpha 21364
38h	56	AMD Turion™ II Ultra Dual-Core Mobile M Processor Family
39h	57	AMD Turion™ II Dual-Core Mobile M Processor Family
3Ah	58	AMD Athlon™ II Dual-Core M Processor Family
3Bh	59	AMD Opteron™ 6100 Series Processor
3Ch	60	AMD Opteron™ 4100 Series Processor
3Dh	61	AMD Opteron™ 6200 Series Processor
3Eh	62	AMD Opteron™ 4200 Series Processor
3Fh	63	AMD FX™ Series Processor
40h	64	MIPS Family
41h	65	MIPS R4000
42h	66	MIPS R4200
43h	67	MIPS R4400
44h	68	MIPS R4600
45h	69	MIPS R10000
46h	70	AMD C-Series Processor
47h	71	AMD E-Series Processor
48h	72	AMD A-Series Processor
49h	73	AMD G-Series Processor
4Ah	74	AMD Z-Series Processor
4Bh	75	AMD R-Series Processor
4Ch	76	AMD Opteron™ 4300 Series Processor
4Dh	77	AMD Opteron™ 6300 Series Processor
4Eh	78	AMD Opteron™ 3300 Series Processor
4Fh	79	AMD FirePro™ Series Processor

Hex Value	Decimal Value	Meaning
50h	80	SPARC Family
51h	81	SuperSPARC
52h	82	microSPARC II
53h	83	microSPARC IIep
54h	84	UltraSPARC
55h	85	UltraSPARC II
56h	86	UltraSPARC Iii
57h	87	UltraSPARC III
58h	88	UltraSPARC IIIi
59h-5Fh	89-95	Available for assignment
60h	96	68040 Family
61h	97	68xxx
62h	98	68000
63h	99	68010
64h	100	68020
65h	101	68030
66h	102	AMD Athlon(TM) X4 Quad-Core Processor Family
67h	103	AMD Opteron(TM) X1000 Series Processor
68h	104	AMD Opteron(TM) X2000 Series APU
69h-6Fh	105-111	Available for assignment
70h	112	Hobbit Family
71h-77h	113-119	Available for assignment
78h	120	Crusoe™ TM5000 Family
79h	121	Crusoe™ TM3000 Family
7Ah	122	Efficeon™ TM8000 Family
7Bh-7Fh	123-127	Available for assignment
80h	128	Weitek
81h	129	Available for assignment
82h	130	Itanium™ processor
83h	131	AMD Athlon™ 64 Processor Family
84h	132	AMD Opteron™ Processor Family
85h	133	AMD Sempron™ Processor Family
86h	134	AMD Turion™ 64 Mobile Technology
87h	135	Dual-Core AMD Opteron™ Processor Family
88h	136	AMD Athlon™ 64 X2 Dual-Core Processor Family
89h	137	AMD Turion™ 64 X2 Mobile Technology
8Ah	138	Quad-Core AMD Opteron™ Processor Family
8Bh	139	Third-Generation AMD Opteron™ Processor Family
8Ch	140	AMD Phenom™ FX Quad-Core Processor Family
8Dh	141	AMD Phenom™ X4 Quad-Core Processor Family
8Eh	142	AMD Phenom™ X2 Dual-Core Processor Family

Hex Value	Decimal Value	Meaning
8Fh	143	AMD Athlon™ X2 Dual-Core Processor Family
90h	144	PA-RISC Family
91h	145	PA-RISC 8500
92h	146	PA-RISC 8000
93h	147	PA-RISC 7300LC
94h	148	PA-RISC 7200
95h	149	PA-RISC 7100LC
96h	150	PA-RISC 7100
97h-9Fh	151-159	Available for assignment
A0h	160	V30 Family
A1h	161	Quad-Core Intel® Xeon® processor 3200 Series
A2h	162	Dual-Core Intel® Xeon® processor 3000 Series
A3h	163	Quad-Core Intel® Xeon® processor 5300 Series
A4h	164	Dual-Core Intel® Xeon® processor 5100 Series
A5h	165	Dual-Core Intel® Xeon® processor 5000 Series
A6h	166	Dual-Core Intel® Xeon® processor LV
A7h	167	Dual-Core Intel® Xeon® processor ULV
A8h	168	Dual-Core Intel® Xeon® processor 7100 Series
A9h	169	Quad-Core Intel® Xeon® processor 5400 Series
AAh	170	Quad-Core Intel® Xeon® processor
ABh	171	Dual-Core Intel® Xeon® processor 5200 Series
ACh	172	Dual-Core Intel® Xeon® processor 7200 Series
ADh	173	Quad-Core Intel® Xeon® processor 7300 Series
AEh	174	Quad-Core Intel® Xeon® processor 7400 Series
AFh	175	Multi-Core Intel® Xeon® processor 7400 Series
B0h	176	Pentium® III Xeon™ processor
B1h	177	Pentium® III Processor with Intel® SpeedStep™ Technology
B2h	178	Pentium® 4 Processor
B3h	179	Intel® Xeon® processor
B4h	180	AS400 Family
B5h	181	Intel® Xeon™ processor MP
B6h	182	AMD Athlon™ XP Processor Family
B7h	183	AMD Athlon™ MP Processor Family
B8h	184	Intel® Itanium® 2 processor
B9h	185	Intel® Pentium® M processor
BAh	186	Intel® Celeron® D processor
BBh	187	Intel® Pentium® D processor
BCh	188	Intel® Pentium® Processor Extreme Edition
BDh	189	Intel® Core™ Solo Processor
BEh	190	Reserved [3]
BFh	191	Intel® Core™ 2 Duo Processor

Hex Value	Decimal Value	Meaning
C0h	192	Intel® Core™ 2 Solo processor
C1h	193	Intel® Core™ 2 Extreme processor
C2h	194	Intel® Core™ 2 Quad processor
C3h	195	Intel® Core™ 2 Extreme mobile processor
C4h	196	Intel® Core™ 2 Duo mobile processor
C5h	197	Intel® Core™ 2 Solo mobile processor
C6h	198	Intel® Core™ i7 processor
C7h	199	Dual-Core Intel® Celeron® processor
C8h	200	IBM390 Family
C9h	201	G4
CAh	202	G5
CBh	203	ESA/390 G6
CCh	204	z/Architecture base
CDh	205	Intel® Core™ i5 processor
CEh	206	Intel® Core™ i3 processor
CFh-D1h	207-209	Available for assignment
D2h	210	VIA C7™-M Processor Family
D3h	211	VIA C7™-D Processor Family
D4h	212	VIA C7™ Processor Family
D5h	213	VIA Eden™ Processor Family
D6h	214	Multi-Core Intel® Xeon® processor
D7h	215	Dual-Core Intel® Xeon® processor 3xxx Series
D8h	216	Quad-Core Intel® Xeon® processor 3xxx Series
D9h	217	VIA Nano™ Processor Family
DAh	218	Dual-Core Intel® Xeon® processor 5xxx Series
DBh	219	Quad-Core Intel® Xeon® processor 5xxx Series
DCh	220	Available for assignment
DDh	221	Dual-Core Intel® Xeon® processor 7xxx Series
DEh	222	Quad-Core Intel® Xeon® processor 7xxx Series
DFh	223	Multi-Core Intel® Xeon® processor 7xxx Series
E0h	224	Multi-Core Intel® Xeon® processor 3400 Series
E1h-E3h	225-227	Available for assignment
E4h	228	AMD Opteron™ 3000 Series Processor
E5h	229	AMD Sempron™ II Processor
E6h	230	Embedded AMD Opteron™ Quad-Core Processor Family
E7h	231	AMD Phenom™ Triple-Core Processor Family
E8h	232	AMD Turion™ Ultra Dual-Core Mobile Processor Family
E9h	233	AMD Turion™ Dual-Core Mobile Processor Family
EAh	234	AMD Athlon™ Dual-Core Processor Family
EBh	235	AMD Sempron™ SI Processor Family
ECh	236	AMD Phenom™ II Processor Family

Hex Value	Decimal Value	Meaning
EDh	237	AMD Athlon™ II Processor Family
EEh	238	Six-Core AMD Opteron™ Processor Family
EFh	239	AMD Sempron™ M Processor Family
F0h-F9h	240-249	Available for assignment
FAh	250	i860
FBh	251	i960
FCh-FDh	252-253	Available for assignment
FEh	254	Indicator to obtain the processor family from the Processor Family 2 field
FFh	255	Reserved
100h-1FFh	256-511	These values are available for assignment, except for the following:
104h	260	SH-3
105h	261	SH-4
118h	280	ARM
119h	281	StrongARM
12Ch	300	6x86
12Dh	301	MediaGX
12Eh	302	MII
140h	320	WinChip
15Eh	350	DSP
1F4h	500	Video Processor
200h-FFFDh	512- 65533	Available for assignment
FFFEh-FFFFh	65534-65535	Reserved

[1] Note that the meaning associated with this value is different from the meaning defined in CIM_Processor.Family for the same value.

[2] Some version 2.0 specification implementations used *Processor Family* type value 30h to represent a Pentium® Pro processor.

[3] Version 2.5 of this specification listed this value as "available for assignment". CIM_Processor.mof files assigned this value to AMD K7 processors in the CIM_Processor.Family property, and an SMBIOS change request assigned it to Intel Core 2 processors. Some implementations of the SMBIOS version 2.5 specification are known to use BEh to indicate Intel Core 2 processors. Some implementations of SMBIOS and some implementations of CIM-based software may also have used BEh to indicate AMD K7 processors.

939   For processor family enumerations from 0 to FDh, Processor Family is identical to Processor Family 2.

940   For processor family enumerations from 100h to FFFDh, Processor Family has a value of FEh and
941   Processor Family 2 has the enumerated value.

942   The following values are reserved:

943   • FFh   Not used. FFh is the un-initialized value of Flash memory.

944   • FFFFh   Not used. FFFFh is the un-initialized value of Flash memory.

945   • FFFEh   For special use in the future, such as FEh as the extension indicator.

946   ### 7.5.3   Processor ID field format

947   The Processor ID field contains processor-specific information that describes the processor's features.

948 **7.5.3.1   x86-class CPUs**

949 For x86 class CPUs, the field's format depends on the processor's support of the CPUID instruction. If the
950 instruction is supported, the *Processor ID* field contains two DWORD-formatted values. The first (offsets
951 08h-0Bh) is the EAX value returned by a CPUID instruction with input EAX set to 1; the second (offsets
952 0Ch-0Fh) is the EDX value returned by that instruction.

953 Otherwise, only the first two bytes of the *Processor ID* field are significant (all others are set to 0) and
954 contain (in WORD-format) the contents of the DX register at CPU reset.

955 **7.5.4   Processor Information — Voltage**

956 Two forms of information can be specified by the SMBIOS in this field, dependent on the value present in
957 bit 7 (the most-significant bit). If bit 7 is 0 (legacy mode), the remaining bits of the field represent the
958 specific voltages that the processor socket can accept, as shown in Table 24.

959 **Table 24 – Processor Information: Voltage field**

Byte Bit Range	Meaning
Bit 7	Set to 0, indicating 'legacy' mode for processor voltage
Bits 6:4	Reserved, must be zero
Bits 3:0	*Voltage Capability* A set bit indicates that the voltage is supported. Bit 0 – 5V Bit 1 – 3.3V Bit 2 – 2.9V Bit 3 – Reserved, must be zero. NOTE: Setting of multiple bits indicates the socket is configurable.

960 If bit 7 is set to 1, the remaining seven bits of the field are set to contain the processor's current *voltage*
961 *times 10*.

962 EXAMPLE: The field value for a processor voltage of 1.8 volts would be:

963   92h = 80h + (1.8 * 10) = 80h + 18 = 80h +12h

964 **7.5.5   Processor Information — Processor Upgrade**

965 Table 25 shows what the byte values mean for the Processor Information — Processor Upgrade field.

966 NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

967 **Table 25 – Processor Information: Processor Upgrade field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Daughter Board
04h	ZIF Socket
05h	Replaceable Piggy Back

Byte Value	Meaning
06h	None
07h	LIF Socket
08h	Slot 1
09h	Slot 2
0Ah	370-pin socket
0Bh	Slot A
0Ch	Slot M
0Dh	Socket 423
0Eh	Socket A (Socket 462)
0Fh	Socket 478
10h	Socket 754
11h	Socket 940
12h	Socket 939
13h	Socket mPGA604
14h	Socket LGA771
15h	Socket LGA775
16h	Socket S1
17h	Socket AM2
18h	Socket F (1207)
19h	Socket LGA1366
1Ah	Socket G34
1Bh	Socket AM3
1Ch	Socket C32
1Dh	Socket LGA1156
1Eh	Socket LGA1567
1Fh	Socket PGA988A
20h	Socket BGA1288
21h	Socket rPGA988B
22h	Socket BGA1023
23h	Socket BGA1224
24h	Socket LGA1155
25h	Socket LGA1356
26h	Socket LGA2011
27h	Socket FS1
28h	Socket FS2
29h	Socket FM1

Byte Value	Meaning
2Ah	Socket FM2
2Bh	Socket LGA2011-3
2Ch	Socket LGA1356-3
2Dh	Socket LGA1150
2Eh	Socket BGA1168
2Fh	Socket BGA1234
30h	Socket BGA1364

968   **7.5.6   Processor Information — Core Count**

969   *Core Count* is the number of cores detected by the BIOS for this processor socket. It does not necessarily
970   indicate the full capability of the processor. For example, platform hardware may have the capability to
971   limit the number of cores reported by the processor without BIOS intervention or knowledge. For a dual-
972   core processor installed in a platform where the hardware is set to limit it to one core, the BIOS reports a
973   value of 1 in *Core Count*. For a dual-core processor with multi-core support disabled by BIOS, the BIOS
974   reports a value of 2 in *Core Count*.

975   The *Core Count 2* field supports core counts > 255. For core counts of 256 or greater, the *Core Count*
976   field is set to FFh and the *Core Count 2* field is set to the number of cores. For core counts of 255 or
977   fewer, if *Core Count 2* is present it shall be set the same value as *Core Count*. Table 26 presents
978   examples of the use and interpretation of the *Core Count* and *Core Count 2* fields.

979   **Table 26 - Examples of *Core Count* and *Core Count 2* use**

Core Count Field	*Core Count 2* Field	Core Count
00h	absent	Unknown
20h	absent	32
FFh	absent	255
00h	0000h	Unknown
20h	0020h	32
FFh	00FFh	255
FFh	0100h	256
FFh	0200h	512
FFh	FFFFh	Reserved

980   **7.5.7   Processor Information — Core Enabled**

981   *Core Enabled* is the number of cores that are enabled by the BIOS and available for Operating System
982   use. For example, if the BIOS detects a dual-core processor, it would report a value of 2 if it leaves both
983   cores enabled, and it would report a value of 1 if it disables multi-core support.

984   The *Core Enabled 2* field supports core enabled counts > 255. For core enabled counts of 256 or greater,
985   the *Core Enabled* field is set to FFh and the *Core Enabled 2* field is set to the number of enabled cores.
986   For core enabled counts of 255 or fewer, if *Core Enabled 2* is present it shall be set to the same value as
987   *Core Enabled*. This follows the approach used for the *Core Count* and *Core Count 2* fields. See Table 26
988   for examples.

989   7.5.8   Processor Information — Thread Count

990   *Thread Count* is the total number of threads detected by the BIOS for this processor socket. It is a
991   processor-wide count, not a thread-per-core count. It does not necessarily indicate the full capability of
992   the processor. For example, platform hardware may have the capability to limit the number of threads
993   reported by the processor without BIOS intervention or knowledge. For a dual-thread processor installed
994   in a platform where the hardware is set to limit it to one thread, the BIOS reports a value of 1 in *Thread*
995   *Count*. For a dual-thread processor with multi-threading disabled by BIOS, the BIOS reports a value of 2
996   in *Thread Count*. For a dual-core, dual-thread-per-core processor, the BIOS reports a value of 4 in
997   *Thread Count*.

998   The *Thread Count 2* field supports thread counts > 255. For thread counts of 256 or greater, the *Thread*
999   *Count* field is set to FFh and the *Thread Count 2* field is set to the number of threads. For thread counts
1000   of 255 or fewer, if *Thread Count 2* is present it shall be set to the same value as *Thread Count*. This
1001   follows the approach used for the *Core Count* and *Core Count 2* fields. See Table 26 for examples.

1002   7.5.9   Processor Characteristics

1003   Table 27 describes the Processor Characteristics field.

1004   *64-bit Capable* indicates the maximum data width capability of the processor. For example, this bit is set
1005   for Intel Itanium, AMD Opteron, and Intel Xeon (with EM64T) processors; this bit is cleared for Intel Xeon
1006   processors that do not have EM64T. This bit indicates the maximum capability of the processor and does
1007   not indicate the current enabled state.

1008   *Multi-Core* indicates the processor has more than one core. This bit does not indicate the number of
1009   cores (*Core Count*) enabled by hardware or the number of cores (*Core Enabled*) enabled by BIOS.

1010   *Hardware Thread* indicates that the processor supports multiple hardware threads per core. This bit does
1011   not indicate the state or number of threads.

1012   *Execute Protection* indicates that the processor supports marking specific memory regions as non-
1013   executable. For example, this is the NX (No eXecute) feature of AMD processors and the XD (eXecute
1014   Disable) feature of Intel processors. This bit does not indicate the present state of this feature.

1015   *Enhanced Virtualization* indicates that the processor is capable of executing enhanced virtualization
1016   instructions. This bit does not indicate the present state of this feature.

1017   *Power/Performance Control* indicates that the processor is capable of load-based power savings. This bit
1018   does not indicate the present state of this feature.

1019   **Table 27 – Processor Characteristics**

WORD Bit Position	Meaning if Set
Bit 0	Reserved
Bit 1	Unknown
Bit 2	64-bit Capable
Bit 3	Multi-Core
Bit 4	Hardware Thread
Bit 5	Execute Protection
Bit 6	Enhanced Virtualization
Bit 7	Power/Performance Control
Bits 8:15	Reserved

1020   ## 7.6   Memory Controller Information (Type 5, Obsolete)

1021   The information in this structure defines the attributes of the system's memory controller(s) and the
1022   supported attributes of any memory-modules present in the sockets controlled by this controller. See
1023   Table 28 for the details of this structure.

1024   NOTE   This structure, and its companion, Memory Module Information (Type 6, Obsolete), are **obsolete** starting
1025   with version 2.1 of this specification; the Physical Memory Array (Type 16) and Memory Device (Type 17) structures
1026   should be used instead. BIOS providers might choose to implement both memory description types to allow existing
1027   DMI browsers to properly display the system's memory attributes.

1028   **Table 28 – Memory Controller Information (Type 5, Obsolete) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.0+	Type	BYTE	5	Memory Controller indicator
01h	2.0+	Length	BYTE	Varies	Computed by the BIOS as either 15 + (2 * x) for version 2.0 implementations or 16 + (2 * x) for version 2.1 and later implementations, where x is the value present in offset 0Eh.
02h	2.0+	Handle	WORD	Varies	
04h	2.0+	Error Detecting Method	BYTE	ENUM	See 7.6.1.
05h	2.0+	Error Correcting Capability	BYTE	Bit Field	See 7.6.2.
06h	2.0+	Supported Interleave	BYTE	ENUM	See 7.6.3.
07h	2.0+	Current Interleave	BYTE	ENUM	See 7.6.3.
08h	2.0+	Maximum Memory Module Size	BYTE	Varies (n)	Size of the largest memory module supported (per slot), specified as n, where 2**n is the maximum size in MB The maximum amount of memory supported by this controller is that value times the number of slots, as specified in offset 0Eh of this structure.
09h	2.0+	Supported Speeds	WORD	Bit Field	See 7.6.4 for bit-wise descriptions.
0Bh	2.0+	Supported Memory Types	WORD	Bit Field	See 7.7.1 for bit-wise descriptions.
0Dh	2.0+	Memory Module Voltage	BYTE	Bit Field	Describes the required voltages for each of the memory module sockets controlled by this controller: Bits 7:3 Reserved, must be zero Bit 2 2.9V Bit 1 3.3V Bit 0 5V NOTE: Setting of multiple bits indicates that the sockets are configurable.
0Eh	2.0+	Number of Associated Memory Slots (x)	BYTE	Varies	Defines how many of the Memory Module Information blocks are controlled by this controller

Offset	Spec. Version	Name	Length	Value	Description
0Fh to 0Fh + (2*x) - 1	2.0+	Memory Module Configuration Handles	x WORDs	Varies	Lists memory information structure handles controlled by this controller Value in offset 0Eh (x) defines the count.
0Fh + (2*x)	2.1+	Enabled Error Correcting Capabilities	BYTE	Bit Field	Identifies the error-correcting capabilities that were enabled when the structure was built See 7.6.2 for bit-wise definitions.

1029   **7.6.1   Memory Controller Error Detecting Method**

1030   Table 29 shows the byte values for the Memory Controller Error Detecting Method field.

1031   **Table 29 – Memory Controller Error Detecting Method field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	None
04h	8-bit Parity
05h	32-bit ECC
06h	64-bit ECC
07h	128-bit ECC
08h	CRC

1032   **7.6.2   Memory Controller Error Correcting Capability**

1033   Table 30 shows the values for the Memory Controller Error Correcting Capability field.

1034   **Table 30 – Memory Controller Error Correcting Capability field**

Byte Bit Position	Meaning
Bit 0	Other
Bit 1	Unknown
Bit 2	None
Bit 3	Single-Bit Error Correcting
Bit 4	Double-Bit Error Correcting
Bit 5	Error Scrubbing

1035   **7.6.3   Memory Controller Information — Interleave Support**

1036   Table 31 shows the byte values for the Memory Controller Information — Interleave Support field.

1037   **Table 31 – Memory Controller Information: Interleave Support field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	One-Way Interleave

Byte Value	Meaning
04h	Two-Way Interleave
05h	Four-Way Interleave
06h	Eight-Way Interleave
07h	Sixteen-Way Interleave

1038   **7.6.4   Memory Controller Information — Memory Speeds**

1039   The bit-field shown in Table 32 describes the speed of the memory modules supported by the system.

1040   **Table 32 – Memory Controller Information: Memory Speeds Bit field**

Word Bit Position	Meaning
Bit 0	Other
Bit 1	Unknown
Bit 2	70ns
Bit 3	60ns
Bit 4	50ns
Bits 5:15	Reserved, must be zero

1041   **7.7   Memory Module Information (Type 6, Obsolete)**

1042   One *Memory Module Information* structure is included for each memory-module socket in the system. As
1043   shown in Table 33, the structure describes the speed, type, size, and error status of each system memory
1044   module. The supported attributes of each module are described by the "owning" *Memory Controller*
1045   *Information* structure.

1046   NOTE   This structure and its companion Memory Controller Information (Type 5, Obsolete) are **obsolete** starting
1047   with version 2.1 of this specification; the Physical Memory Array (Type 16) and Memory Device (Type 17) structures
1048   should be used instead. BIOS providers might choose to implement both memory description types to allow existing
1049   DMI browsers to properly display the system's memory attributes.

1050   **Table 33 – Memory Module Information (Type 6, Obsolete) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	6	Memory Module Configuration indicator
01h	Length	BYTE	0Ch	
02h	Handle	WORD	Varies	
04h	Socket Designation	BYTE	STRING	String number for reference designation EXAMPLE: 'J202',0
05h	Bank Connections	BYTE	Varies	Each nibble indicates a bank (RAS#) connection; 0xF means no connection. EXAMPLE: If banks 1 & 3 (RAS# 1 & 3) were connected to a SIMM socket the byte for that socket would be 13h. If only bank 2 (RAS 2) were connected, the byte for that socket would be 2Fh.
06h	Current Speed	BYTE	Varies	Speed of the memory module, in ns (for example, 70d for a 70ns module) If the speed is unknown, the field is set to 0.

Offset	Name	Length	Value	Description
07h	Current Memory Type	WORD	Bit Field	See 7.7.1.
09h	Installed Size	BYTE	Varies	See 7.7.2.
0Ah	Enabled Size	BYTE	Varies	See 7.7.2.
0Bh	Error Status	BYTE	Varies	Bits 7:3 Reserved, set to 0s Bit 2 If set, the Error Status information should be obtained from the event log; bits 1and 0 are reserved. Bit 1 Correctable errors received for the module, if set. This bit is reset only during a system reset. Bit 0 Uncorrectable errors received for the module, if set. All or a portion of the module has been disabled. This bit is only reset on power-on.

1051 **7.7.1 Memory Module Information — Memory Types**

1052 The bit-field shown in Table 34 describes the physical characteristics of the memory modules that are
1053 supported by (and currently installed in) the system.

1054   **Table 34 – Memory Module Information: Memory Types**

Word Bit Position	Meaning
Bit 0	Other
Bit 1	Unknown
Bit 2	Standard
Bit 3	Fast Page Mode
Bit 4	EDO
Bit 5	Parity
Bit 6	ECC
Bit 7	SIMM
Bit 8	DIMM
Bit 9	Burst EDO
Bit 10	SDRAM
Bits 11:15	Reserved, must be zero

1055 **7.7.2 Memory Module Information — Memory Size**

1056 The Size fields of the Memory Module Configuration Information structure define the amount of memory
1057 currently installed (and enabled) in a memory-module connector. Table 35 shows the meaning of the
1058 bytes and bits in the Memory Size field.

1059 The *Installed Size* fields identify the size of the memory module that is installed in the socket, as
1060 determined by reading and correlating the module's presence-detect information. If the system does not
1061 support presence-detect mechanisms, the *Installed Size* field is set to 7Dh to indicate that the installed
1062 size is not determinable. The *Enabled Size* field identifies the amount of memory currently enabled for the

1063   system's use from the module. If a module is known to be installed in a connector, but all memory in the
1064   module has been disabled due to error, the *Enabled Size* field is set to 7Eh.

1065   **Table 35 – Memory Module Information: Memory Size field**

Byte Bit Range	Meaning
Bits 0:6	Indicates size (n), where 2**n is the size in MB, with the following special-case values: • 7Dh Not determinable (Installed Size only) • 7Eh Module is installed, but no memory has been enabled • 7Fh Not installed
Bit 7	Defines whether the memory module has a single- (0) or double-bank (1) connection

7.7.3   Memory subsystem example

1066

1067   A system utilizes a memory controller that supports up to four 32 MB 5 V 70 ns parity SIMMs. The
1068   memory module sockets are used in pairs A1/A2 and B1/B2 to provide a 64-bit data path to the CPU. No
1069   mechanism is provided by the system to read the SIMM IDs. RAS-0 and -1 are connected to the front-
1070   and back-size banks of the SIMMs in the A1/A2 sockets and RAS-2 and -3 are similarly connected to the
1071   B1/B2 sockets. The current installation is an 8 MB SIMM in sockets A1 and A2, 16 MB total.

```
db  5          ; Memory Controller Information
db  23         ; Length = 15 + 2*4
dw  14         ; Memory Controller Handle
db  4          ; 8-bit parity error detection
db  00000100b  ; No error correction provided
db  03h        ; 1-way interleave supported
db  03h        ; 1-way interleave currently used
db  5          ; Maximum memory-module size supported is 32 MB (2**5)
dw  00000100b  ; Only 70ns SIMMs supported
dw  00A4h      ; Standard, parity SIMMs supported
db  00000001b  ; 5V provided to each socket
db  4          ; 4 memory-module sockets supported
dw  15         ; 1st Memory Module Handle
dw  16
dw  17
dw  18         ; 4th ...
dw  0000h      ; End-of-structure termination
```

```
db  6          ; Memory Module Information
db  0Ch
dw  15         ; Handle
db  1          ; Reference Designation string #1
db  01h        ; Socket connected to RAS-0 and RAS-1
db  00000010b  ; Current speed is Unknown, since can't read SIMM IDs
db  00000100b  ; Upgrade speed is 70ns, since that's all that's
               ; supported
dw  00A4h      ; Current SIMM must be standard parity
db  7Dh        ; Installed size indeterminable (no SIMM IDs)
db  83h        ; Enabled size is double-bank 8MB (2**3)
db  0          ; No errors
db  "A1",0     ; String#1: Reference Designator
db  0          ; End-of-strings
```

```
db  6          ; Memory Module Information
db  0Ch
dw  16         ; Handle
db  1          ; Reference Designation string #1
db  01h        ; Socket connected to RAS-0 and RAS-1
db  0          ; Current speed is Unknown, since can't read SIMM IDs
dw  00A4h      ; Current SIMM must be standard parity
db  7Dh        ; Installed size indeterminable (no SIMM IDs)
db  83h        ; Enabled size is double-bank 8MB (2**3)
db  0          ; No errors
db  "A2",0     ; String#1: Reference Designator
db  0          ; End-of-strings
```

```
db  6          ; Memory Module Information
db  0Ch
dw  17         ; Handle
db  1          ; Reference Designation string #1
db  23h        ; Socket connected to RAS-2 and RAS-3
db  0          ; Current speed is Unknown, since can't read SIMM IDs
dw  0001h      ; Nothing appears to be installed (Other)
db  7Dh        ; Installed size indeterminable (no SIMM IDs)
db  7Fh        ; Enabled size is 0 (nothing installed)
db  0          ; No errors
db  "B1",0     ; String#1: Reference Designator
db  0          ; End-of-strings
```

```
db  6          ; Memory Module Information
db  0Ch
dw  18         ; Handle
db  1          ; Reference Designation string #1
db  23h        ; Socket connected to RAS-2 and RAS-3
db  0          ; Current speed is Unknown, since can't read SIMM IDs
dw  0001h      ; Nothing appears to be installed (Other)
db  7Dh        ; Installed size indeterminable (no SIMM IDs)
db  7Fh        ; Enabled size is 0 (nothing installed)
db  0          ; No errors
db  "B2",0     ; String#1: Reference Designator
db  0          ; End-of-strings
```

1072 ## 7.8 Cache Information (Type 7)

1073 As shown in Table 36, the information in this structure defines the attributes of CPU cache device in the
1074 system. One structure is specified for each such device, whether the device is internal to or external to
1075 the CPU module. Cache modules can be associated with a processor structure in one or two ways
1076 depending on the SMBIOS version; see 7.5 and 7.15 for more information.

1077 **Table 36 – Cache Information (Type 7) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.0+	Type	BYTE	7	Cache Information indicator
01h	2.0+	Length	BYTE	Varies	Value is 0Fh for version 2.0 implementations, or 13h for version 2.1 implementations.
02h	2.0+	Handle	WORD	Varies	
04h	2.0+	Socket Designation	BYTE	STRING	String number for reference designation EXAMPLE: "CACHE1", 0

Offset	Spec. Version	Name	Length	Value	Description	
05h	2.0+	Cache Configuration	WORD	Varies	Bits 15:10	Reserved, must be zero
					Bits 9:8	Operational Mode
						00b Write Through
						01b Write Back
						10b Varies with Memory Address
						11b Unknown
					Bit 7	Enabled/Disabled (at boot time)
						1b Enabled
						0b Disabled
					Bits 6:5	Location, relative to the CPU module:
						00b Internal
						01b External
						10b Reserved
						11b Unknown
					Bit 4	Reserved, must be zero
					Bit 3	Cache Socketed
						1b Socketed
						0b Not Socketed
					Bits 2:0	Cache Level – 1 through 8 (For example, an L1 cache would use value 000b and an L3 cache would use 010b.)
07h	2.0+	Maximum Cache Size	WORD	Varies	Maximum size that can be installed	
					Bit 15 Granularity	
					0 – 1K granularity	
					1 – 64K granularity	
					Bits 14:0 Max size in given granularity	
					See 7.8.1.	
09h	2.0+	Installed Size	WORD	Varies	Same format as Max Cache Size field; set to 0 if no cache is installed See 7.8.1.	
0Bh	2.0+	Supported SRAM Type	WORD	Bit Field	See 7.8.1.	
0Dh	2.0+	Current SRAM Type	WORD	Bit Field	See 7.8.1.	
0Fh	2.1+	Cache Speed	BYTE	Varies	Cache module speed, in nanoseconds The value is 0 if the speed is unknown.	
10h	2.1+	Error Correction Type	BYTE	ENUM	Error-correction scheme supported by this cache component; see 7.8.3	
11h	2.1+	System Cache Type	BYTE	ENUM	Logical type of cache; see 7.8.4	
12h	2.1+	Associativity	BYTE	ENUM	Associativity of the cache; see 7.8.5	

1078　**7.8.1　Cache Information — Maximum Cache Size and Installed Size**

1079　For multi-core processors, the cache size for the different levels of the cache (L1, L2, L3) is the total
1080　amount of cache per level per processor socket. The cache size is independent of the core count. For
1081　example, the cache size is 2 MB for both a dual core processor with a 2 MB L3 cache shared between
1082　the cores and a dual core processor with 1 MB L3 cache (non-shared) per core.

1083　**7.8.2　Cache Information — SRAM Type**

1084　Table 37 shows the values for the Cache Information — SRAM Type field.

1085　　　　　　　　　　**Table 37 – Cache Information: SRAM Type field**

Word Bit Position	Meaning
Bit 0	Other
Bit 1	Unknown
Bit 2	Non-Burst
Bit 3	Burst
Bit 4	Pipeline Burst
Bit 5	Synchronous
Bit 6	Asynchronous
Bits 7:15	Reserved, must be zero

1086　**7.8.3　Cache Information — Error Correction Type**

1087　Table 38 shows the values for the Cache Information — Error Correction Type field.

1088　　　　　　　　　　**Table 38 – Cache Information: Error Correction Type field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	None
04h	Parity
05h	Single-bit ECC
06h	Multi-bit ECC

1089　**7.8.4　Cache Information — System Cache Type**

1090　Table 39 shows the values for the Cache Information — System Cache Type field.

1091　The cache type for a cache level (L1, L2, L3, ...) is type 03h (Instruction) when all the caches at that level
1092　are Instruction caches. The cache type for a specific cache level (L1, L2, L3, ...) is type 04h (Data) when
1093　all the caches at that level are Data caches. The cache type for a cache level (L1, L2, L3, ...) is type 05h
1094　(Unified) when the caches at that level are a mix of Instruction and Data caches.

1095　NOTE　　Refer to 6.3 for the CIM properties associated with this enumerated value.

1096   **Table 39 – Cache Information: System Cache Type field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Instruction
04h	Data
05h	Unified

1097   **7.8.5   Cache Information — Associativity**

1098   Table 40 shows the values for the Cache Information — Associativity field.

1099   NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

1100   **Table 40 – Cache Information: Associativity field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Direct Mapped
04h	2-way Set-Associative
05h	4-way Set-Associative
06h	Fully Associative
07h	8-way Set-Associative
08h	16-way Set-Associative
09h	12-way Set-Associative
0Ah	24-way Set-Associative
0Bh	32-way Set-Associative
0Ch	48-way Set-Associative
0Dh	64-way Set-Associative
0Eh	20-way Set-Associative

1101   **7.9   Port Connector Information (Type 8)**

1102   As shown in Table 41, the information in this structure defines the attributes of a system port connector
1103   (for example, parallel, serial, keyboard, or mouse ports). The port's type and connector information are
1104   provided. One structure is present for each port provided by the system.

1105   **Table 41 – Port Connector Information (Type 8) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	8	Connector Information indicator
01h	Length	BYTE	9h	
02h	Handle	WORD	Varies	

Offset	Name	Length	Value	Description
04h	Internal Reference Designator	BYTE	STRING	String number for Internal Reference Designator, that is, internal to the system enclosure EXAMPLE: 'J101', 0
05h	Internal Connector Type	BYTE	ENUM	Internal Connector type See 7.9.2.
06h	External Reference Designator	BYTE	STRING	String number for the External Reference Designation external to the system enclosure EXAMPLE: 'COM A', 0
07h	External Connector Type	BYTE	ENUM	External Connector type See 7.9.2.
08h	Port Type	BYTE	ENUM	Describes the function of the port See 7.9.3.

7.9.1   Port Information example

The following structure shows an example where a DB-9 Pin Male connector on the System Backpanel (COM A) is connected to the System Board through a 9-Pin Dual Inline connector (J101).

```
db 8         ; Indicates Connector Type
db 9h        ; Length
dw ?         ; Reserved for handle
db 01h       ; String 1 - Internal Reference Designation
db 18h       ; 9 Pin Dual Inline
db 02h       ; String 2 - External Reference Designation
db 08h       ; DB-9 Pin Male
db 09h       ; 16550A Compatible
db 'J101',0  ; Internal reference
db 'COM A',0 ; External reference
db 0
```

If an External Connector is not used (as in the case of a CD-ROM Sound connector), the External Reference Designator and the External Connector type should be set to zero. If an Internal Connector is not used (as in the case of a soldered-on Parallel Port connector that extends outside of the chassis), the Internal Reference Designation and Connector Type should be set to zero.

7.9.2   Port Information — Connector Types

Table 42 shows the values of the bytes in the Port Information — Connector Types field.

Table 42 – Port Information: Connector Types Field

Byte Value	Meaning
00h	None
01h	Centronics
02h	Mini Centronics
03h	Proprietary
04h	DB-25 pin male
05h	DB-25 pin female

Byte Value	Meaning
06h	DB-15 pin male
07h	DB-15 pin female
08h	DB-9 pin male
09h	DB-9 pin female
0Ah	RJ-11
0Bh	RJ-45
0Ch	50-pin MiniSCSI
0Dh	Mini-DIN
0Eh	Micro-DIN
0Fh	PS/2
10h	Infrared
11h	HP-HIL
12h	Access Bus (USB)
13h	SSA SCSI
14h	Circular DIN-8 male
15h	Circular DIN-8 female
16h	On Board IDE
17h	On Board Floppy
18h	9-pin Dual Inline (pin 10 cut)
19h	25-pin Dual Inline (pin 26 cut)
1Ah	50-pin Dual Inline
1Bh	68-pin Dual Inline
1Ch	On Board Sound Input from CD-ROM
1Dh	Mini-Centronics Type-14
1Eh	Mini-Centronics Type-26
1Fh	Mini-jack (headphones)
20h	BNC
21h	1394
22h	SAS/SATA Plug Receptacle
A0h	PC-98
A1h	PC-98Hireso
A2h	PC-H98
A3h	PC-98Note
A4h	PC-98Full
FFh	Other – Use Reference Designator Strings to supply information.

1127   ### 7.9.3 Port Types

1128   Table 43 shows the values for the Port Types field.

1129   **Table 43 – Port Types field**

Byte Value	Meaning
00h	None

Byte Value	Meaning
01h	Parallel Port XT/AT Compatible
02h	Parallel Port PS/2
03h	Parallel Port ECP
04h	Parallel Port EPP
05h	Parallel Port ECP/EPP
06h	Serial Port XT/AT Compatible
07h	Serial Port 16450 Compatible
08h	Serial Port 16550 Compatible
09h	Serial Port 16550A Compatible
0Ah	SCSI Port
0Bh	MIDI Port
0Ch	Joy Stick Port
0Dh	Keyboard Port
0Eh	Mouse Port
0Fh	SSA SCSI
10h	USB
11h	FireWire (IEEE P1394)
12h	PCMCIA Type I[2]
13h	PCMCIA Type II
14h	PCMCIA Type III
15h	Cardbus
16h	Access Bus Port
17h	SCSI II
18h	SCSI Wide
19h	PC-98
1Ah	PC-98-Hireso
1Bh	PC-H98
1Ch	Video Port
1Dh	Audio Port
1Eh	Modem Port
1Fh	Network Port
20h	SATA
21h	SAS
A0h	8251 Compatible
A1h	8251 FIFO Compatible
0FFh	Other

---

[2] Prior to version 2.7.1, this specification incorrectly described this value as "PCMCIA Type II".

1130 **7.10 System Slots (Type 9)**

1131 As shown in Table 44, the information in this structure defines the attributes of a system slot. One
1132 structure is provided for each slot in the system.

1133 **Table 44 – System Slots (Type 9) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.0+	Type	BYTE	9	System Slot Structure indicator
01h	2.0+	Length	BYTE	Varies	0Ch for version 2.0 implementations 0Dh for versions 2.1 to 2.5 11h for version 2.6 and later
02h	2.0+	Handle	WORD	Varies	
04h	2.0+	Slot Designation	BYTE	STRING	String number for reference designation EXAMPLE: 'PCI-1',0
05h	2.0+	Slot Type	BYTE	ENUM	See 7.10.1.
06h	2.0+	Slot Data Bus Width	BYTE	ENUM	See 7.10.2.
07h	2.0+	Current Usage	BYTE	ENUM	See 7.10.3.
08h	2.0+	Slot Length	BYTE	ENUM	See 7.10.4.
09h	2.0+	Slot ID	WORD	Varies	See 7.10.5.
0Bh	2.0+	Slot Characteristics 1	BYTE	Bit Field	See 7.10.6.
0Ch	2.1+	Slot Characteristics 2	BYTE	Bit Field	See 7.10.7.
0Dh	2.6+	Segment Group Number	WORD	Varies	See 7.10.8.
0Fh	2.6+	Bus Number	BYTE	Varies	See 7.10.8.
10h	2.6+	Device/Function Number	BYTE	Bit field	Bits 7:3 – device number Bits 2:0 – function number See 7.10.8.

1134 **7.10.1 System Slots — Slot Type**

1135 Table 45 shows the values of the System Slots — Slot Type field.

1136 **Table 45 – System Slots: Slot Type field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	ISA
04h	MCA
05h	EISA
06h	PCI

Byte Value	Meaning
07h	PC Card (PCMCIA)
08h	VL-VESA
09h	Proprietary
0Ah	Processor Card Slot
0Bh	Proprietary Memory Card Slot
0Ch	I/O Riser Card Slot
0Dh	NuBus
0Eh	PCI – 66MHz Capable
0Fh	AGP
10h	AGP 2X
11h	AGP 4X
12h	PCI-X
13h	AGP 8X
14h	M.2 Socket 1-DP (Mechanical Key A)
15h	M.2 Socket 1-SD (Mechanical Key E)
16h	M.2 Socket 2 (Mechanical Key B)
17h	M.2 Socket 3 (Mechanical Key M)
18h	MXM Type I
19h	MXM Type II
1Ah	MXM Type III (standard connector)
1Bh	MXM Type III (HE connector)
1Ch	MXM Type IV
1Dh	MXM 3.0 Type A
1Eh	MXM 3.0 Type B
1Fh	PCI Express Gen 2 SFF-8639
20h	PCI Express Gen 3 SFF-8639
A0h	PC-98/C20
A1h	PC-98/C24
A2h	PC-98/E
A3h	PC-98/Local Bus
A4h	PC-98/Card
A5h	PCI Express (see below)
A6h	PCI Express x1
A7h	PCI Express x2
A8h	PCI Express x4
A9h	PCI Express x8
AAh	PCI Express x16
ABh	PCI Express Gen 2 (see below)
ACh	PCI Express Gen 2 x1

Byte Value	Meaning
ADh	PCI Express Gen 2 x2
AEh	PCI Express Gen 2 x4
AFh	PCI Express Gen 2 x8
B0h	PCI Express Gen 2 x16
B1h	PCI Express Gen 3 (see below)
B2h	PCI Express Gen 3 x1
B3h	PCI Express Gen 3 x2
B4h	PCI Express Gen 3 x4
B5h	PCI Express Gen 3 x8
B6h	PCI Express Gen 3 x16

1137 NOTE   Slot types A5h, ABh, and B1h should be used only for PCI Express slots where the physical width is
1138 identical to the electrical width; in that case the System Slots – Slot Data Bus Width field specifies the width. Other
1139 PCI Express slot types (A6h-AAh, ACh-B0h, B2h-B6h) should be used to describe slots where the physical width is
1140 different from the maximum electrical width; in these cases the width indicated in this field refers to the physical width
1141 of the slot, while electrical width is described in the System Slots – Slot Data Bus Width field.

1142 **7.10.2 System Slots — Slot Data Bus Width**

1143 NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

1144 Table 46 shows the values for the System Slots – Slot Data Bus Width field. Slot Data Bus Width
1145 meanings of type "n bit" are for parallel buses such as PCI. Slot Data Bus Width meanings of type "nx or
1146 xn" are for serial buses such as PCI Express.

1147 NOTE   For PCI Express, width refers to the maximum supported electrical width of the "data bus"; physical slot
1148 width is described in System Slots – Slot Type, and the actual link width resulting from PCI Express link training can
1149 be read from configuration space.

1150   **Table 46 – System Slots: Slot Data Bus Width field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	8 bit
04h	16 bit
05h	32 bit
06h	64 bit
07h	128 bit
08h	1x or x1
09h	2x or x2
0Ah	4x or x4
0Bh	8x or x8
0Ch	12x or x12
0Dh	16x or x16
0Eh	32x or x32

1151 **7.10.3 System Slots — Current Usage**

1152 Table 47 shows the values of the System Slots — Current Usage field.

1153 NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

1154 **Table 47 – System Slots: Current Usage field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Available
04h	In use

1155 **7.10.4 System Slots — Slot Length**

1156 Table 48 shows the values of the System Slots — Slot Length field.

1157 **Table 48 – System Slots: Slot Length field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Short Length
04h	Long Length
05h	2.5" drive form factor
06h	3.5" drive form factor

1158 **7.10.5 System Slots — Slot ID**

1159 The *Slot ID* field of the System Slot structure provides a mechanism to correlate the physical attributes of
1160 the slot to its logical access method (which varies based on the *Slot Type* field). The *Slot ID* field has
1161 meaning only for the slot types described in Table 49.

1162 **Table 49 – System Slots: Slot ID**

Slot Type	Slot ID Field Meaning
MCA	Identifies the logical Micro Channel slot number, in the range 1 to 15, in offset 09h. Offset 0Ah is set to 0.
EISA	Identifies the logical EISA slot number, in the range 1 to 15, in offset 09h. Offset 0Ah is set to 0.
PCI, AGP, PCI-X, PCI Express	On a system that supports ACPI, identifies the value returned in the _SUN object for this slot On a system that supports the *PCI IRQ Routing Table Specification*, identifies the value present in the Slot Number field of the PCI Interrupt Routing table entry that is associated with this slot, in offset 09h—offset 0Ah is set to 0. The table is returned by the "Get PCI Interrupt Routing Options" PCI BIOS function call and provided directly in the *PCI IRQ Routing Table Specification* ($PIRQ). Software can determine the PCI bus number and device associated with the slot by matching the "Slot ID" to an entry in the routing-table and ultimately determine what device is present in that slot. NOTE: This definition also applies to the 66 MHz-capable PCI slots.

Slot Type	Slot ID Field Meaning
PCMCIA	Identifies the Adapter Number (offset 09h) and Socket Number (offset 0Ah) to be passed to PCMCIA Socket Services to identify this slot.

7.10.6 Slot Characteristics 1

Table 50 shows the values for the Slot Characteristics 1 field.

Table 50 – Slot Characteristics 1 field

BYTE Bit Position	Meaning if Set
Bit 0	Characteristics unknown.
Bit 1	Provides 5.0 volts.
Bit 2	Provides 3.3 volts.
Bit 3	Slot's opening is shared with another slot (for example, PCI/EISA shared slot).
Bit 4	PC Card slot supports PC Card-16.
Bit 5	PC Card slot supports CardBus.
Bit 6	PC Card slot supports Zoom Video.
Bit 7	PC Card slot supports Modem Ring Resume.

7.10.7 Slot Characteristics 2

Table 51 shows the values for the Slot Characteristics 2 field.

Table 51 – Slot Characteristics 2

BYTE Bit Position	Meaning if Set
Bit 0	PCI slot supports Power Management Event (PME#) signal.
Bit 1	Slot supports hot-plug devices.
Bit 2	PCI slot supports SMBus signal.
Bits 3:7	Reserved, set to 0.

7.10.8 Segment Group Number, Bus Number, Device/Function Number

For slots that are not of types PCI, AGP, PCI-X, or PCI-Express that do not have bus/device/function information, 0FFh should be populated in the fields of *Segment Group Number*, *Bus Number*, *Device/Function Number*.

Segment Group Number is defined in the *PCI Firmware Specification*. The value is 0 for a single-segment topology.

7.11 On Board Devices Information (Type 10, Obsolete)

NOTE   This structure is obsolete starting with version 2.6 of this specification; the *Onboard Devices Extended Information* (Type 41) structure should be used instead (see 7.42). BIOS providers can choose to implement both types to allow existing SMBIOS browsers to properly display the system's onboard devices information.

The information in this structure (shown in Table 52) defines the attributes of devices that are onboard (soldered onto) a system element, usually the baseboard. In general, an entry in this table implies that the BIOS has some level of control over the enabling of the associated device for use by the system.

1182 NOTE   Because this structure was originally defined with the *Length* implicitly defining the number of devices
1183 present, no further fields can be added to this structure without adversely affecting existing software's ability to
1184 properly parse the data. Thus, if additional fields are required for this structure type a brand new structure must be
1185 defined to add a device count field, carry over the existing fields, and add the new information.

1186   **Table 52 – On Board Devices Information (Type 10, Obsolete) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	10	On Board Devices Information indicator
01h	Length	BYTE	Varies	Computed by the BIOS as 4 + 2 * (Number of Devices). The user of this structure determines the number of devices as (Length - 4) / 2.
02h	Handle	WORD	Varies	
4+2*(n-1)	Device$_n$ Type, n ranges from 1 to Number of Devices	BYTE	Varies	Bit 7 Device$_n$ Status 1 – Device Enabled 0 – Device Disabled Bits 6:0 Type of Device (see 7.11.1)
5+2*(n-1)	Description String	BYTE	STRING	String number of device description

1187 NOTE   There may be a single structure instance containing the information for all onboard devices, or there may
1188 be a unique structure instance for each onboard device.

7.11.1 Onboard Device Types

1190 Table 53 shows what the bytes mean for the Onboard Device Types field.

1191   **Table 53 – Onboard Device Types**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Video
04h	SCSI Controller
05h	Ethernet
06h	Token Ring
07h	Sound
08h	PATA Controller
09h	SATA Controller
0Ah	SAS Controller

7.12 OEM Strings (Type 11)

1193 This structure (shown in Table 54) contains free-form strings defined by the OEM. Examples of this are
1194 part numbers for system reference documents, contact information for the manufacturer, etc.

1195   **Table 54 – OEM Strings (Type 11) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	11	OEM Strings indicator

Offset	Name	Length	Value	Description
01h	Length	BYTE	5h	
02h	Handle	WORD	Varies	
04h	Count	BYTE	Varies	Number of strings

7.13 System Configuration Options (Type 12)

This structure (shown in Table 55) contains information required to configure the baseboard's Jumpers and Switches.

EXAMPLES:

```
"JP2: 1-2 Cache Size is 256K, 2-3 Cache Size is 512K"
"SW1-1: Close to Disable On Board Video"
```

Table 55 – System Configuration Options (Type 12) structure

Offset	Name	Length	Value	Description
00h	Type	BYTE	12	Configuration Information indicator
01h	Length	BYTE	5h	
02h	Handle	WORD	Varies	
04h	Count	BYTE	Varies	Number of strings

7.14 BIOS Language Information (Type 13)

The information in this structure (shown in Table 56) defines the installable language attributes of the BIOS.

Table 56 – BIOS Language Information (Type 13) structure

Offset	Spec. Version	Name	Length	Value	Description
00h	2.0+	Type	BYTE	13	Language Information indicator
01h	2.0+	Length	BYTE	16h	
02h	2.0+	Handle	WORD	Varies	
04h	2.0+	Installable Languages	BYTE	Varies	Number of languages available Each available language has a description string. This field contains the number of strings that follow the formatted area of the structure.
05h	2.1+	Flags	BYTE	Bit Field	Bits 7:1 Reserved Bit 0 If set to 1, the Current Language strings use the abbreviated format. Otherwise, the strings use the long format. See below for details.
06h	2.0+	Reserved	15 BYTEs	0	Reserved for future use
015h	2.0+	Current Language	BYTE	STRING	String number (one-based) of the currently installed language

1207 The strings describing the languages follow the *Current Language* byte. The format of the strings
1208 depends on the value present in bit 0 of the byte at offset 05h in the structure.

1209 • If the bit is 0, each language string is in the form "ISO 639-1 Language Name | ISO 3166-1-alpha-
1210 2 Territory Name | Encoding Method". See Example 1 below.

1211 • If the bit is 1, each language string consists of the two-character "ISO 639-1 Language Name"
1212 directly followed by the two-character "ISO 3166-1-alpha-2 Territory Name". See Example 2
1213 below.

1214 NOTE   Refer to ISO 639-1 and ISO 3166-1 for additional information.

1215 EXAMPLE 1: BIOS Language Information (Long Format)

```
1216  db 13              ; language information
1217  db 16h             ; length
1218  dw ??              ; handle
1219  db 3               ; three languages available
1220  db 0               ; use long-format for language strings
1221  db 15 dup (0)        ; reserved
1222  db 2               ; current language is French Canadian
1223  db 'en|US|iso8859-1',0    ; language 1 is US English
1224  db 'fr|CA|iso8859-1',0    ; language 2 is French Canadian
1225  db 'ja|JP|unicode',0     ; language 3 is Japanese
1226  db 0               ; Structure termination
```

1227 EXAMPLE 2: BIOS Language Information (Abbreviated Format)

```
1228  db 13              ; language information
1229  db 16h             ; length
1230  dw ??              ; handle
1231  db 3               ; three languages available
1232  db 01h             ; use abbreviated format for language strings
1233  db 15 dup (0)        ; reserved
1234  db 2               ; current language is French Canadian
1235  db 'enUS',0         ; language 1 is US English
1236  db 'frCA',0         ; language 2 is French Canadian
1237  db 'jaJP',0         ; language 3 is Japanese
1238  db 0               ; Structure termination
```

1239 7.15 Group Associations (Type 14)

1240 Table 57 shows the values for the Group Associations (Type 14) structure.

1241 NOTE   Because this structure was originally defined with the Length implicitly defining the number of items
1242 present, no further fields can be added to this structure without adversely affecting existing software's ability to
1243 properly parse the data. Thus, if additional fields are required for this structure type, a brand new structure must be
1244 defined to add an item count field, carry over the existing fields, and add the new information.

1245   **Table 57 – Group Associations (Type 14) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	14	Group Associations indicator

Offset	Name	Length	Value	Description
01h	Length	BYTE	Varies	Computed by the BIOS as 5 + (3 bytes for each item in the group) The user of this structure determines the number of items as (Length - 5) / 3.
02h	Handle	WORD	Varies	
04h	Group Name	BYTE	STRING	String number of string describing the group
05h	Item Type	BYTE	Varies	Item (Structure) Type of this member
06h	Item Handle	WORD	Varies	Handle corresponding to this structure

1246 The Group Associations structure is provided for OEMs who want to specify the arrangement or hierarchy
1247 of certain components (including other Group Associations) within the system. For example, you can use
1248 the Group Associations structure to indicate that two CPUs share a common external cache system.
1249 These structures might look similar to the examples shown in Example 1 and Example 2.

1250 EXAMPLE 1: First Group Association Structure

```
1251  db 14 ; Group Association structure
1252  db 11 ; Length
1253  dw 28h; Handle
1254  db 01h; String Number (First String)
1255  db 04 ; CPU Structure
1256  dw 08h; CPU Structure's Handle
1257  db 07 ; Cache Structure
1258  dw 09h; Cache Structure's Handle
1259  db 'Primary CPU Module', 0
1260  db 0
```

1261 EXAMPLE 2: Second Group Association Structure

```
1262  db 14 ; Group Association structure
1263  db 11 ; Length
1264  dw 29h; Handle
1265  db 01h; String Number (First String)
1266  db 04 ; CPU Structure
1267  dw 0Ah; CPU Structure's Handle
1268  db 07 ; Cache Structure
1269  dw 09h; Cache Structure's Handle
1270  db 'Secondary CPU Module', 0
1271  db 0
```

1272 In the examples above, CPU structures 08h and 0Ah are associated with the same cache, 09h. This
1273 relationship could also be specified as a single group, as shown in Example 3.

1274 EXAMPLE 3:

```
1275  db 14 ; Group Association structure
1276  db 14 ; Length (5 + 3 * 3)
1277  dw 28h; Structure handle for Group Association
1278  db 1  ; String Number (First string)
1279  db 4  ; 1st CPU
```

```
1280    dw 08h; CPU Structure's Handle
1281    db 4  ; 2nd CPU
1282    dw 0Ah; CPU Structure's Handle
1283    db 7  ; Shared cache
1284    dw 09h; Cache Structure's Handle
1285    db 'Dual-Processor CPU Complex', 0
1286    db 0
```

1287   7.16 System Event Log (Type 15)

1288   The presence of this structure within the SMBIOS data returned for a system indicates that the system
1289   supports an event log. See Table 58 for details. An event log is a fixed-length area within a non-volatile
1290   storage element, starting with a fixed-length (and vendor-specific) header record, followed by one or more
1291   variable-length log records. See 7.16.4 for more information.

1292   An application can implement event-log change notification by periodically reading the System Event Log
1293   structure (by its assigned handle) and looking for a change in the *Log Change Token*. This token uniquely
1294   identifies the last time the event log was updated. When it sees the token changed, the application can
1295   retrieve the entire event log and determine the changes since the last time it read the event log.

1296   **Table 58 – System Event Log (Type 15) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.0+	Type	BYTE	15	Event Log Type indicator
01h	2.0+	Length	BYTE	Varies	Length of the structure, including the Type and Length fields The Length is 14h for version 2.0 implementations. For version 2.1 and higher implementations the length is computed by the BIOS as 17h+(x*y), where x is the value present at offset 15h and y is the value present at offset 16h.
02h	2.0+	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	2.0+	Log Area Length	WORD	Varies	Length, in bytes, of the overall event log area, from the first byte of header to the last byte of data
06h	2.0+	Log Header Start Offset	WORD	Varies	Defines the starting offset (or index) within the nonvolatile storage of the event-log's header, from the Access Method Address For single-byte indexed I/O accesses, the most-significant byte of the start offset is set to 00h.
08h	2.0+	Log Data Start Offset	WORD	Varies	Defines the starting offset (or index) within the nonvolatile storage of the event-log's first data byte, from the Access Method Address For single-byte indexed I/O accesses, the most-significant byte of the start offset is set to 00h. NOTE: The data directly follows any header information. Therefore, the header length can be determined by subtracting the Header Start Offset from the Data Start Offset.
0Ah	2.0+	Access	BYTE	Varies	Defines the Location and Method used by

Offset	Spec. Version	Name	Length	Value	Description
		Method			higher-level software to access the log area, one of:
					00h Indexed I/O: 1 8-bit index port, 1 8-bit data port. The Access Method Address field contains the 16-bit I/O addresses for the index and data ports. See 7.16.2.1 for usage details.
					01h Indexed I/O: 2 8-bit index ports, 1 8-bit data port. The Access Method Address field contains the 16-bit I/O address for the index and data ports. See 7.16.2.2 for usage details.
					02h Indexed I/O: 1 16-bit index port, 1 8-bit data port. The Access Method Address field contains the 16-bit I/O address for the index and data ports. See 7.16.2.3 for usage details.
					03h Memory-mapped physical 32-bit address. The Access Method Address field contains the 4-byte (Intel DWORD format) starting physical address.
					04h Available through General-Purpose NonVolatile Data functions.
					The Access Method Address field contains the 2-byte (Intel WORD format) GPNV handle.
					05h-7Fh Available for future assignment by this specification
					80h-FFh BIOS Vendor/OEM-specific
0Bh	2.0+	Log Status [1]	BYTE	Varies	Current status of the system event-log: Bits 7:2 Reserved, set to 0s Bit 1 Log area full, if 1 Bit 0 Log area valid, if 1
0Ch	2.0+	Log Change Token	DWORD	Varies	Unique token that is reassigned every time the event log changes Can be used to determine if additional events have occurred since the last time the log was read.
10h	2.0+	Access Method Address	DWORD	Varies	Address associated with the access method; the data present depends on the Access Method field value The area's format can be described by the following 1-byte-packed 'C' union: union { struct { short IndexAddr; short DataAddr; } IO;

Offset	Spec. Version	Name	Length	Value	Description
					long PhysicalAddr32;
					short GPNVHandle;
					} AccessMethodAddress;
14h	2.1+	Log Header Format	BYTE	ENUM	Format of the log header area; see 7.16.5 for details
15h	2.1+	Number of Supported Log Type Descriptors (x)	BYTE	Varies	Number of supported event log type descriptors that follow If the value is 0, the list that starts at offset 17h is not present.
16h	2.1+	Length of each Log Type Descriptor (y)	BYTE	2	Number of bytes associated with each type entry in the list below The value is currently "hard-coded" as 2, because each entry consists of two bytes. This field's presence allows future additions to the type list. Software that interprets the following list should not assume a list entry's length.
17h to 17h+(x*y))-1	2.1+	List of Supported Event Log Type Descriptors	Varies	Varies	List of Event Log Type Descriptors (see 7.16.1), as long as the value specified in offset 15h is non-zero
[1] The *Log Status* and *Log Change Token* fields might not be up-to-date (dynamic) when the structure is accessed using the table interface.					

7.16.1 Supported Event Log Type descriptors

Each entry consists of a 1-byte type field and a 1-byte data-format descriptor, as shown in Table 59. The presence of an entry identifies that the Log Type is supported by the system and the format of any variable data that accompanies the first bytes of the log's variable data — a specific log record might have more variable data than specified by its Variable Data Format Type.

Table 59 – Supported Event Log Type Descriptors

Offset	Name	Length	Value	Description
00h	Log Type	BYTE	ENUM	See 7.16.6.1 for list.
01h	Variable Data Format Type	BYTE	ENUM	See 7.16.6.2 for list.

7.16.2 Indexed I/O Access method

This clause contains examples (in x86 assembly language) that detail the code that is required to access the "indexed I/O" event-log information.

7.16.2.1 One 8-bit Index, One 8-bit Data (00h)

To access the event-log, the caller selects 1 of 256 unique data bytes by

    1)   Writing the byte data-selection value (index) to the *IndexAddr* I/O address

    2)   Reading or writing the byte data value to (or from) the *DataAddr* I/O address

```
1310    mov    dx, IndexAddr    ;Value from event-log structure
1311    mov    al, WhichLoc     ;Identify offset to be accessed
1312    out    dx, al
1313    mov    dx, DataAddr     ;Value from event-log structure
1314    in al, dx               ;Read current value
```

1315   7.16.2.2 Two 8-bit Index, One 8-bit Data (01h)

1316   To access the event-log, the caller selects 1 of 65536 unique data bytes by

1317   1)   Writing the least-significant <u>byte</u> data-selection value (index) to the *IndexAddr* I/O address

1318   2)   Writing the most-significant <u>byte</u> data-selection value (index) to the (*IndexAddr+1*) I/O address

1319   3)   Reading or writing the <u>byte</u> data value to (or from) the *DataAddr* I/O address

```
1320    mov    dx, IndexAddr    ;Value from event-log structure
1321    mov    ax, WhichLoc     ;Identify offset to be accessed
1322    out    dx, al           ;Select LSB offset
1323    inc    dx
1324    xchg   ah, al
1325    out    dx, al           ;Select MSB offset
1326    mov    dx, DataAddr     ;Value from event-log structure
1327    in al, dx               ;Read current value
```

1328   7.16.2.3 One 16-bit Index, One 8-bit Data (02h)

1329   To access the event-log, the caller selects 1 of 65536 unique data bytes by

1330   1)   Writing the <u>word</u> data-selection value (index) to the *IndexAddr* I/O address

1331   2)   Reading or writing the <u>byte</u> data value to (or from) the *DataAddr* I/O address

```
1332    mov    dx, IndexAddr    ;Value from event-log structure
1333    mov    ax, WhichLoc     ;Identify offset to be accessed
1334    out    dx, ax
1335    mov    dx, DataAddr     ;Value from event-log structure
1336    in al, dx               ;Read current value
```

1337   7.16.3 Access Method Address — DWORD layout

1338   Table 60 shows the DWORD layout of the Access Method Address.

1339   **Table 60 – Access Method Address: DWORD layout**

Access Type	BYTE 3	BYTE 2	BYTE 1	BYTE 0
00:02 – Indexed I/O	Data MSB	Data LSB	Index MSB	Index LSB
03 – Absolute Address	Byte 3	Byte 2	Byte 1	Byte 0
04 – Use GPNV	0	0	Handle MSB	Handle LSB

1340   7.16.4 Event Log organization

1341   The event log is organized as an optional (and implementation-specific) fixed-length header, followed by
1342   one or more variable-length event records, as illustrated in Table 61. From one implementation to the

1343 next, the format of the log header and the size of the overall log area might change; all other required
1344 fields of the event log area are consistent across all systems.

1345 **Table 61 – Event Log organization**

Log Header (Optional)								
Type	**Length**	**Year**	**Month**	**Day**	**Hour**	**Minute**	**Second**	**Log Variable Data**
Required	Required	Required	Required	Required	Required	Required	Required	Optional

1346 ### 7.16.5 Log Header format

1347 Table 62 contains the byte enumeration values (available for SMBIOS 2.1 and later) that identify the
1348 standard formats of the event log headers.

1349 **Table 62 – Log Header format**

Byte Value	Meaning
00h	No header (for example, the header is 0 bytes in length)
01h	Type 1 log header; see 7.16.5.1
02h-7Fh	Available for future assignment by this specification
80h-FFh	BIOS vendor or OEM-specific format

1350 ### 7.16.5.1 Log Header Type 1 format

1351 The type 1 event log header consists of the following fields shown in Table 63.

1352 **Table 63 – Log Header Type 1 format**

Offset	Name	Length	Value	Description
00h	OEM Reserved	5 BYTES	Varies	Reserved area for OEM customization, not assignable by this specification
05h	Multiple Event Time Window	BYTE	Varies	Number of minutes that must pass between duplicate log entries that utilize a multiple-event counter, specified in BCD The value ranges from 00h to 99h to represent 0 to 99 minutes. See 7.16.6.3 for usage details.
06h	Multiple Event Count Increment	BYTE	Varies	Number of occurrences of a duplicate event that must pass before the multiple-event counter associated with the log entry is updated, specified as a numeric value in the range 1 to 255 (The value 0 is reserved.) See 7.16.6.3 for usage details.
07h	Pre-boot Event Log Reset — CMOS Address	BYTE	Varies	CMOS RAM address (in the range 10h - FFh) associated with the Pre-boot Event Log Reset; the value is 00h if the feature is not supported See below for usage details.

Offset	Name	Length	Value	Description
08h	Pre-boot Event Log Reset — CMOS Bit Index	BYTE	Varies	Bit within the above CMOS RAM location that is set to indicate that the log should be cleared The value is specified in the range 0 to 7, where 0 specifies the LSB and 7 specified the MSB. See below for usage details.
09h	CMOS Checksum — Starting Offset	BYTE	Varies	CMOS RAM address associated with the start of the area that is to be checksummed, if the value is non-0 If the value is 0, the CMOS Address field lies outside of a checksummed region in CMOS RAM. See below for usage details.
0Ah	CMOS Checksum — Byte Count	BYTE	Varies	Number of consecutive CMOS RAM addresses, starting at the Starting Offset, that participate in the CMOS Checksum region associated with the pre-boot event log reset See below for usage details.
0Bh	CMOS Checksum — Checksum Offset	BYTE	Varies	CMOS RAM address associated with the start of two consecutive bytes into which the calculated checksum value is stored See below for usage details.
0Ch - 0Eh	Reserved	3 BYTEs	000000h	Available for future assignment by this specification
0Fh	Header Revision	BYTE	01h	Version of Type 1 header implemented

The Type 1 Log Header also provides pre-boot event log reset support. Application software can set a system-specific location of CMOS RAM memory (accessible through I/O ports 70h and 71h) to cause the event log to be cleared by the BIOS on the next reboot of the system.

To perform the field setting, application software follows these steps, as long as the *Pre-boot Event Log Reset – CMOS Address* field of the header is non-zero:

- Read the address specified by *Pre-boot Event Log Reset — CMOS Address* from CMOS RAM. Set the bit specified by the *CMOS Bit Index* field to 1. Rewrite the CMOS RAM address with the updated data.

- If the *CMOS Checksum – Starting Offset* field is non-zero, recalculate the CMOS RAM checksum value for the range starting at the *Starting Offset* field for *Byte Count* bytes into a 2-byte value. Subtract that value from 0 to create the checksum value for the range and store that 2-byte value into the CMOS RAM; the least-significant byte of the value is stored at the CMOS RAM *Checksum Offset* and the most-significant byte of the value is stored at (*Checksum Offset*)+1.

7.16.6 Log Record format

Each log record consists of a *required* fixed-length record header, followed by (optional) additional data that is defined by the event type. The fixed-length log record header is present as the first eight bytes of each log record, regardless of event type. Details are shown in Table 64.

Table 64 – Log Record format

Offset	Name	Format	Description
00h	Event Type	BYTE	Specifies the "Type" of event noted in an event-log entry as defined in 7.16.6.1

Offset	Name	Format	Description
01h	Length	BYTE	Specifies the byte length of the event record, including the record's Type and Length fields The most-significant bit of the field specifies whether (0) or not (1) the record has been read. The implication of the record having been read is that the information in the log record has been processed by a higher software layer.
02h-07h	Date/Time Fields	BYTE	Contains the BCD representation of the date and time (as read from CMOS RAM) of the occurrence of the event The information is present in year, month, day, hour, minute, and second order. NOTE: The century portion of the two-digit year is implied as '19' for year values in the range 80h to 99h and '20' for year values in the range 00h to 79h.
08h+	Log Variable Data	Var	Contains the (optional) event-specific additional status information.

1372   **7.16.6.1 Event Log types**

1373   Table 65 shows the values for Event Log types.

1374   **Table 65 – Event Log types**

Value	Description
00h	Reserved
01h	Single-bit ECC memory error
02h	Multi-bit ECC memory error
03h	Parity memory error
04h	Bus time-out
05h	I/O Channel Check
06h	Software NMI
07h	POST Memory Resize
08h	POST Error
09h	PCI Parity Error
0Ah	PCI System Error
0Bh	CPU Failure
0Ch	EISA FailSafe Timer time-out
0Dh	Correctable memory log disabled
0Eh	Logging disabled for a specific Event Type — too many errors of the same type received in a short amount of time
0Fh	Reserved
10h	System Limit Exceeded (for example, voltage or temperature threshold exceeded)
11h	Asynchronous hardware timer expired and issued a system reset
12h	System configuration information
13h	Hard-disk information
14h	System reconfigured
15h	Uncorrectable CPU-complex error
16h	Log Area Reset/Cleared

Value	Description
17h	System boot. If implemented, this log entry is guaranteed to be the first one written on any system boot.
18h-7Fh	Unused, available for assignment by this specification
80h-FEh	Available for system- and OEM-specific assignments
FFh	End of log When an application searches through the event-log records, the end of the log is identified when a log record with this type is found.

1375 **7.16.6.2 Event Log Variable Data Format Type**

1376 The Variable Data Format Type, specified in the Event Log structure's Supported Event Type fields,
1377 identifies the standard format that application software can apply to the first *n* bytes of the associated Log
1378 Type's variable data. Additional OEM-specific data might follow in the log's variable data field. Table 66
1379 shows the values for this field.

1380   **Table 66 – Event Log Variable Data Format Type**

Value	Name	Description
00h	None	No standard format data is available; the first byte of the variable data (if present) contains OEM-specific unformatted information.
01h	Handle	The first WORD of the variable data contains the handle of the SMBIOS structure associated with the hardware element that failed.
02h	Multiple-Event	The first DWORD of the variable data contains a multiple-event counter (see 7.16.6.3 for details).
03h	Multiple-Event Handle	The first WORD of the variable data contains the handle of the SMBIOS structure associated with the hardware element that failed; it is followed by a DWORD containing a multiple-event counter (see 7.16.6.3 for details).
04h	POST Results Bitmap	The first two DWORDs of the variable data contain the POST Results Bitmap, as described in 7.16.6.4.
05h	System Management Type	The first DWORD of the variable data contains a value that identifies a system-management condition. See 7.16.6.5 for the enumerated values.
06h	Multiple-Event System Management Type	The first DWORD of the variable data contains a value that identifies a system-management condition. (See 7.16.6.5 for the enumerated values.) This DWORD is directly followed by a DWORD that contains a multiple-event counter (see 7.16.6.3 for details).
07h-7Fh	Unused	Unused, available for assignment by this specification.
80h-FFh	OEM assigned	Available for system- and OEM-specific assignments.

1381 **7.16.6.3 Multiple-Event Counter**

1382 Some system events can be persistent; after they occur, it is possible to quickly fill the log with redundant
1383 multiple logs. The Multiple Event Count Increment (*MECI*) and Multiple Event Time Window (*METW*)
1384 values can be used to reduce the occurrence of these multiple logs while providing multiple event counts.

1385 NOTE   These values are normally specified within the event log header; see 7.16.5.1 for an example. If the values
1386 are not specified in the header, the application software can assume that the *MECI* value is 1 and the *METW* value is
1387 60 (minutes).

1388 The multiple-event counter is a DWORD (32-bit) value that tracks the number of logs of the same type
1389 that have occurred within *METW* minutes. The counter value is initialized (in the log entry) to FFFFFFFFh,
1390 implying that only a single event of that type has been detected, and the internal BIOS counter[3] specific
1391 to that log type is reset to 0. The counter is incremented by setting its next non-zero bit to zero; this allows
1392 counting up to 33 events. When the counter reaches 00000000h, it is full.

1393 EXAMPLE: If the current counter value is FFFFFFFCh (meaning a count of 3 events), it is incremented to
1394 FFFFFFF8h (meaning a count of 4).

1395 When the BIOS receives the next event of that type, it increments its internal counter and checks to see
1396 what recording of the error is to be performed:

1397   If the date/time of the original log entry is outside of *METW* minutes: a new log entry is written, and
1398   the internal BIOS counter is reset to 0;

1399   – If the log's current multiple-event counter is 00000000h or if the internal BIOS counter is
1400   less than the *MECI* value: no recording happens (other than the internal counter
1401   increment);

1402   – Otherwise: The next non-zero bit of the multiple-event counter is set to 0.

1403 **7.16.6.4 POST Results Bitmap**

1404 This variable data type, when present, is expected to be associated with the POST Error (08h) event log
1405 type and identifies that one or more error types have occurred. The bitmap consists of two DWORD
1406 values, described in Table 67. Any bit within the DWORD pair that is specified as Reserved is set to 0
1407 within the log data and is available for assignment by this specification. A set bit ('1'b) at a DWORD bit
1408 position implies that the error associated with that position has occurred.

1409 <div align="center">**Table 67 – POST Results Bitmap**</div>

Bit Position	First DWORD	Second DWORD
0	Channel 2 Timer error	Normally 0; available for OEM assignment
1	Master PIC (8259 #1) error	Normally 0; available for OEM assignment
2	Slave PIC (8259 #2) error	Normally 0; available for OEM assignment
3	CMOS RAM Battery Failure	Normally 0; available for OEM assignment
4	CMOS RAM System Options Not Set	Normally 0; available for OEM assignment
5	CMOS RAM Checksum Error	Normally 0; available for OEM assignment
6	CMOS RAM Configuration Error	Normally 0; available for OEM assignment
7	Mouse and Keyboard Swapped	PCI Memory Conflict
8	Keyboard Locked	PCI I/O Conflict
9	Keyboard Not Functional	PCI IRQ Conflict
10	Keyboard Controller Not Functional	PNP Memory Conflict
11	CMOS Memory Size Different	PNP 32 bit Memory Conflict
12	Memory Decreased in Size	PNP I/O Conflict
13	Cache Memory Error	PNP IRQ Conflict

---

[3] All BIOS counters that support the Multiple-Event Counters are reset to zero each time the system boots.

Bit Position	First DWORD	Second DWORD
14	Floppy Drive 0 Error	PNP DMA Conflict
15	Floppy Drive 1 Error	Bad PNP Serial ID Checksum
16	Floppy Controller Failure	Bad PNP Resource Data Checksum
17	Number of ATA Drives Reduced Error	Static Resource Conflict
18	RTC Time Not Set	NVRAM Checksum Error, NVRAM Cleared
19	DDC Monitor Configuration Change	System Board Device Resource Conflict
20	Reserved, set to 0	Primary Output Device Not Found
21	Reserved, set to 0	Primary Input Device Not Found
22	Reserved, set to 0	Primary Boot Device Not Found
23	Reserved, set to 0	NVRAM Cleared By Jumper
24	Second DWORD has valid data	NVRAM Data Invalid, NVRAM Cleared
25	Reserved, set to 0	FDC Resource Conflict
26	Reserved, set to 0	Primary ATA Controller Resource Conflict
27	Reserved, set to 0	Secondary ATA Controller Resource Conflict
28	Normally 0; available for OEM assignment	Parallel Port Resource Conflict
29	Normally 0; available for OEM assignment	Serial Port 1 Resource Conflict
30	Normally 0; available for OEM assignment	Serial Port 2 Resource Conflict
31	Normally 0; available for OEM assignment	Audio Resource Conflict

1410 **7.16.6.5 System management types**

1411 Table 68 defines the system management types present in an event log record's variable data. In
1412 general, each type is associated with a management event that occurred within the system.

1413 <div align="center">**Table 68 – System management types**</div>

Value	Name
00000000h	+2.5V Out of range, #1
00000001h	+2.5V Out of range, #2
00000002h	+3.3V Out of range
00000003h	+5V Out of range
00000004h	-5V Out of range
00000005h	+12V Out of range
00000006h	-12V Out of range
00000007h - 0000000Fh	Reserved for future out-of-range voltage levels, assigned by this specification
00000010h	System board temperature out of range
00000011h	Processor #1 temperature out of range
00000012h	Processor #2 temperature out of range
00000013h	Processor #3 temperature out of range

Value	Name
00000014h	Processor #4 temperature out of range
00000015h - 0000001Fh	Reserved for future out-of-range temperatures, assigned by this specification
00000020h - 00000027h	Fan n (n = 0 to 7) Out of range
00000028h - 0000002Fh	Reserved for future assignment by this specification
00000030h	Chassis secure switch activated
00000031h - 0000FFFFh	Reserved for future assignment by this specification
0001xxxxh	A system-management probe or cooling device is out of range. The xxxx portion of the value contains the handle of the SMBIOS structure associated with the errant device.
00020000h - 7FFFFFFFh	Reserved for future assignment by this specification
80000000h - FFFFFFFFh	OEM assigned

1414   ## 7.17 Physical Memory Array (Type 16)

1415   This structure describes a collection of memory devices that operate together to form a memory address
1416   space. Table 69 provides the details.

1417   **Table 69 – Physical Memory Array (Type 16) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.1+	Type	BYTE	16	Physical Memory Array type
01h	2.1+	Length	BYTE	Varies	Length of the structure, 0Fh for version 2.1, 17h for version 2.7 and later
02h	2.1+	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	2.1+	Location	BYTE	ENUM	Physical location of the Memory Array, whether on the system board or an add-in board See 7.17.1 for definitions.
05h	2.1+	Use	BYTE	ENUM	Function for which the array is used See 7.17.2 for definitions.
06h	2.1+	Memory Error Correction	BYTE	ENUM	Primary hardware error correction or detection method supported by this memory array See 7.17.3 for definitions.
07h	2.1+	Maximum Capacity	DWORD	Varies	Maximum memory capacity, in kilobytes, for this array If the capacity is not represented in this field, then this field contains 8000 0000h and the Extended Maximum Capacity field should be used. Values 2 TB (8000 0000h) or greater must be represented in the Extended Maximum Capacity field.

Offset	Spec. Version	Name	Length	Value	Description
0Bh	2.1+	Memory Error Information Handle	WORD	Varies	Handle, or instance number, associated with any error that was previously detected for the array If the system does not provide the error information structure, the field contains FFFEh; otherwise, the field contains either FFFFh (if no error was detected) or the handle of the error-information structure. See 7.18.4 and 7.34.
0Dh	2.1+	Number of Memory Devices	WORD	Varies	Number of slots or sockets available for Memory Devices in this array This value represents the number of Memory Device structures that compose this Memory Array. Each Memory Device has a reference to the "owning" Memory Array.
0Fh	2.7+	Extended Maximum Capacity	QWORD	Varies	Maximum memory capacity, in bytes, for this array This field is only valid when the Maximum Capacity field contains 8000 0000h. When Maximum Capacity contains a value that is not 8000 0000h, Extended Maximum Capacity must contain zeros.

1418 **7.17.1 Memory Array — Location**

1419 Table 70 shows what the byte values mean for the Memory Array — Location field.

1420 NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

1421 **Table 70 – Memory Array: Location field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	System board or motherboard
04h	ISA add-on card
05h	EISA add-on card
06h	PCI add-on card
07h	MCA add-on card
08h	PCMCIA add-on card
09h	Proprietary add-on card
0Ah	NuBus
A0h	PC-98/C20 add-on card
A1h	PC-98/C24 add-on card
A2h	PC-98/E add-on card
A3h	PC-98/Local bus add-on card

1422 **7.17.2 Memory Array — Use**

1423 Table 71 shows what the byte values mean for the Memory Array — Use field.

1424 NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

1425 **Table 71 – Memory Array: Use field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	System memory
04h	Video memory
05h	Flash memory
06h	Non-volatile RAM
07h	Cache memory

1426 **7.17.3 Memory Array — Error Correction Types**

1427 Table 72 shows what the byte values mean for the Memory Array — Error Correction Types field.

1428 NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

1429 **Table 72 – Memory Array: Error Correction Types field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	None
04h	Parity
05h	Single-bit ECC
06h	Multi-bit ECC
07h	CRC

1430 **7.18 Memory Device (Type 17)**

1431 This structure describes a single memory device that is part of a larger Physical Memory Array (Type 16)
1432 structure. See 7.17 for more details.

1433 Table 73 provides information about the Memory Device (Type 17) structure.

1434 NOTE   If a system includes memory-device sockets, the SMBIOS implementation includes a *Memory Device*
1435 structure instance for each slot whether or not the socket is currently populated.

1436 **Table 73 – Memory Device (Type 17) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.1+	Type	BYTE	17	Memory Device type

Offset	Spec. Version	Name	Length	Value	Description
01h	2.1+	Length	BYTE	Varies	Length of the structure, 15h for version 2.1, 1Bh for version 2.3, 1Ch for version 2.6, 22h for version 2.7, 28h for version 2.8 and later
02h	2.1+	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	2.1+	Physical Memory Array Handle	WORD	Varies	Handle, or instance number, associated with the Physical Memory Array to which this device belongs
06h	2.1+	Memory Error Information Handle	WORD	Varies	Handle, or instance number, associated with any error that was previously detected for the device If the system does not provide the error information structure, the field contains FFFEh; otherwise, the field contains either FFFFh (if no error was detected) or the handle of the error-information structure. See 7.18.4 and 7.34.
08h	2.1+	Total Width	WORD	Varies	Total width, in bits, of this memory device, including any check or error-correction bits If there are no error-correction bits, this value should be equal to Data Width. If the width is unknown, the field is set to FFFFh.
0Ah	2.1+	Data Width	WORD	Varies	Data width, in bits, of this memory device A Data Width of 0 and a Total Width of 8 indicates that the device is being used solely to provide 8 error-correction bits. If the width is unknown, the field is set to FFFFh.
0Ch	2.1+	Size	WORD	Varies	Size of the memory device If the value is 0, no memory device is installed in the socket; if the size is unknown, the field value is FFFFh. If the size is 32 GB-1 MB or greater, the field value is 7FFFh and the actual size is stored in the Extended Size field. The granularity in which the value is specified depends on the setting of the most-significant bit (bit 15). If the bit is 0, the value is specified in megabyte units; if the bit is 1, the value is specified in kilobyte units. For example, the value 8100h identifies a 256 KB memory device and 0100h identifies a 256 MB memory device.
0Eh	2.1+	Form Factor	BYTE	ENUM	Implementation form factor for this memory device See 7.18.1 for definitions.
0Fh	2.1+	Device Set	BYTE	Varies	Identifies when the Memory Device is one of a set of Memory Devices that must be populated with all devices of the same type and size, and the set to which this device belongs A value of 0 indicates that the device is not part of a set; a value of FFh indicates that the attribute is unknown. NOTE: A Device Set number must be unique within the context of the Memory Array containing this Memory Device.

Offset	Spec. Version	Name	Length	Value	Description
10h	2.1+	Device Locator	BYTE	STRING	String number of the string that identifies the physically-labeled socket or board position where the memory device is located EXAMPLE: "SIMM 3"
11h	2.1+	Bank Locator	BYTE	STRING	String number of the string that identifies the physically labeled bank where the memory device is located EXAMPLE: "Bank 0" or "A"
12h	2.1+	Memory Type	BYTE	ENUM	Type of memory used in this device; see 7.18.2 for definitions
13h	2.1+	Type Detail	WORD	Bit Field	Additional detail on the memory device type; see 7.18.3 for definitions
15h	2.3+	Speed	WORD	Varies	Identifies the maximum capable speed of the device, in megahertz (MHz) If the value is 0, the speed is unknown. NOTE: n MHz = (1000 / n) nanoseconds (ns)
17h	2.3+	Manufacturer	BYTE	STRING	String number for the manufacturer of this memory device
18h	2.3+	Serial Number	BYTE	STRING	String number for the serial number of this memory device This value is set by the manufacturer and normally is not changeable.
19h	2.3+	Asset Tag	BYTE	STRING	String number for the asset tag of this memory device
1Ah	2.3+	Part Number	BYTE	STRING	String number for the part number of this memory device This value is set by the manufacturer and normally is not changeable.
1Bh	2.6+	Attributes	BYTE	Varies	Bits 7-4: reserved Bits 3-0: rank Value=0 for unknown rank information
1Ch	2.7+	Extended Size	DWORD	Varies	Extended size of the memory device (complements the Size field at offset 0Ch) See 7.18.4 for details.
20h	2.7+	Configured Memory Clock Speed	WORD	Varies	Identifies the configured clock speed to the memory device, in megahertz (MHz) If the value is 0, the speed is unknown. NOTE: n MHz = (1000 / n) nanoseconds (ns)
22h	2.8+	Minimum voltage	WORD	Varies	Minimum operating voltage for this device, in millivolts If the value is 0, the voltage is unknown.
24h	2.8+	Maximum voltage	WORD	Varies	Maximum operating voltage for this device, in millivolts If the value is 0, the voltage is unknown.
26h	2.8+	Configured voltage	WORD	Varies	Configured voltage for this device, in millivolts If the value is 0, the voltage is unknown.

1437 **7.18.1 Memory Device — Form Factor**

1438 Table 74 shows what the byte values mean for the Memory Device — Form Factor field.

1439 NOTE Refer to 6.3 for the CIM properties associated with this enumerated value.

1440 **Table 74 – Memory Device: Form Factor field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	SIMM
04h	SIP
05h	Chip
06h	DIP
07h	ZIP
08h	Proprietary Card
09h	DIMM
0Ah	TSOP
0Bh	Row of chips
0Ch	RIMM
0Dh	SODIMM
0Eh	SRIMM
0Fh	FB-DIMM

1441 **7.18.2 Memory Device — Type**

1442 Table 75 shows what the byte values mean for the Memory Device — Type field.

1443 NOTE Refer to 6.3 for the CIM properties associated with this enumerated value.

1444 **Table 75 – Memory Device: Type**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	DRAM
04h	EDRAM
05h	VRAM
06h	SRAM
07h	RAM
08h	ROM
09h	FLASH
0Ah	EEPROM
0Bh	FEPROM

Byte Value	Meaning
0Ch	EPROM
0Dh	CDRAM
0Eh	3DRAM
0Fh	SDRAM
10h	SGRAM
11h	RDRAM
12h	DDR
13h	DDR2
14h	DDR2 FB-DIMM
15h-17h	Reserved
18h	DDR3
19h	FBD2
1Ah	DDR4
1Bh	LPDDR
1Ch	LPDDR2
1Dh	LPDDR3
1Eh	LPDDR4

1445   **7.18.3 Memory Device — Type Detail**

1446   Table 76 shows what the word bit positions mean for the Memory Device — Type Detail field.

1447   NOTE   Multiple bits are set if more than one attribute applies.

1448   **Table 76 – Memory Device: Type Detail field**

Word Bit Position	Meaning
Bit 0	Reserved, set to 0
Bit 1	Other
Bit 2	Unknown
Bit 3	Fast-paged
Bit 4	Static column
Bit 5	Pseudo-static
Bit 6	RAMBUS
Bit 7	Synchronous
Bit 8	CMOS
Bit 9	EDO
Bit 10	Window DRAM
Bit 11	Cache DRAM
Bit 12	Non-volatile
Bit 13	Registered (Buffered)

Word Bit Position	Meaning
Bit 14	Unbuffered (Unregistered)
Bit 15	LRDIMM

7.18.4 Memory Device — Extended Size

The *Extended Size* field is intended to represent memory devices larger than 32,767 MB (32 GB - 1 MB), which cannot be described using the *Size* field. This field is only meaningful if the value in the *Size* field is 7FFFh. For compatibility with older SMBIOS parsers, memory devices smaller than (32 GB - 1 MB) should be represented using their size in the *Size* field, leaving the *Extended Size* field set to 0.

Bit 31 is reserved for future use and must be set to 0.

Bits 30:0 represent the size of the memory device in megabytes.

EXAMPLE: 0000_8000h indicates a 32 GB memory device (32,768 MB), 0002_0000h represents a 128 GB memory device (131,072 MB), and 0000_7FFFh represents a 32,767 MB (32 GB – 1 MB) device.

7.19 32-Bit Memory Error Information (Type 18)

This structure identifies the specifics of an error that might be detected within a Physical Memory Array. Table 77 shows the details for this structure.

Table 77 – 32-Bit Memory Error Information (Type 18) structure

Offset	Spec. Version	Name	Length	Value	Description
00h	2.1+	Type	BYTE	18	32-bit Memory Error Information type
01h	2.1+	Length	BYTE	17h	Length of the structure
02h	2.1+	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	2.1+	Error Type	BYTE	ENUM	Type of error that is associated with the current status reported for the memory array or device See 7.19.1 for definitions.
05h	2.1+	Error Granularity	BYTE	ENUM	Granularity (for example, device versus Partition) to which the error can be resolved See 7.19.2 for definitions.
06h	2.1+	Error Operation	BYTE	ENUM	Memory access operation that caused the error See 7.19.3 for definitions.
07h	2.1+	Vendor Syndrome	DWORD	Varies	Vendor-specific ECC syndrome or CRC data associated with the erroneous access If the value is unknown, this field contains 0000 0000h.
0Bh	2.1+	Memory Array Error Address	DWORD	Varies	32-bit physical address of the error based on the addressing of the bus to which the memory array is connected If the address is unknown, this field contains 8000 0000h.

Offset	Spec. Version	Name	Length	Value	Description
0Fh	2.1+	Device Error Address	DWORD	Varies	32-bit physical address of the error relative to the start of the failing memory device, in bytes If the address is unknown, this field contains 8000 0000h.
13h	2.1+	Error Resolution	DWORD	Varies	Range, in bytes, within which the error can be determined, when an error address is given If the range is unknown, this field contains 8000 0000h.

7.19.1 Memory Error — Error Type

Table 78 shows what the byte values mean for the Memory Error — Error Type field.

NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

Table 78 – Memory Error: Error Type field

Byte Value	Meaning
01h	Other
02h	Unknown
03h	OK
04h	Bad read
05h	Parity error
06h	Single-bit error
07h	Double-bit error
08h	Multi-bit error
09h	Nibble error
0Ah	Checksum error
0Bh	CRC error
0Ch	Corrected single-bit error
0Dh	Corrected error
0Eh	Uncorrectable error

7.19.2 Memory Error — Error Granularity

Table 79 shows what the byte values mean for the Memory Error — Error Granularity field.

Table 79 – Memory Error: Error Granularity field

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Device level
04h	Memory partition level

1469   **7.19.3 Memory Error — Error Operation**

1470   Table 80 shows what the byte values mean for the Memory Error — Error Operation field.

1471   NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

1472   **Table 80 – Memory Error: Error Operation field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Read
04h	Write
05h	Partial write

1473   **7.20 Memory Array Mapped Address (Type 19)**

1474   This structure provides the address mapping for a Physical Memory Array. Details are provided in Table
1475   81.

1476   One structure is present for each contiguous address range described.

1477   See 7.17, 7.18, and 7.21 for more information.

1478   **Table 81 – Memory Array Mapped Address (Type 19) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.1+	Type	BYTE	19	Memory Array Mapped Address indicator
01h	2.1+	Length	BYTE	Varies	Length of the structure, 0Fh for version 2.1, 1Fh for version 2.7 and later.
02h	2.1+	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	2.1+	Starting Address	DWORD	Varies	Physical address, in kilobytes, of a range of memory mapped to the specified Physical Memory Array When the field value is FFFF FFFFh, the actual address is stored in the Extended Starting Address field. When this field contains a valid address, Ending Address must also contain a valid address. When this field contains FFFF FFFFh, Ending Address must also contain FFFF FFFFh.
08h	2.1+	Ending Address	DWORD	Varies	Physical ending address of the last kilobyte of a range of addresses mapped to the specified Physical Memory Array When the field value is FFFF FFFFh and the Starting Address field also contains FFFF FFFFh, the actual address is stored in the Extended Ending Address field. When this field contains a valid address, Starting Address must also contain a valid address.

Offset	Spec. Version	Name	Length	Value	Description
0Ch	2.1+	Memory Array Handle	WORD	Varies	Handle, or instance number, associated with the Physical Memory Array to which this address range is mapped Multiple address ranges can be mapped to a single Physical Memory Array.
0Eh	2.1+	Partition Width	BYTE	Varies	Number of Memory Devices that form a single row of memory for the address partition defined by this structure
0Fh	2.7+	Extended Starting Address	QWORD	Varies	Physical address, in bytes, of a range of memory mapped to the specified Physical Memory Array This field is valid when Starting Address contains the value FFFF FFFFh. If Starting Address contains a value other than FFFF FFFFh, this field contains zeros. When this field contains a valid address, Extended Ending Address must also contain a valid address.
17h	2.7+	Extended Ending Address	QWORD	Varies	Physical ending address, in bytes, of the last of a range of addresses mapped to the specified Physical Memory Array This field is valid when both Starting Address and Ending Address contain the value FFFF FFFFh. If Ending Address contains a value other than FFFF FFFFh, this field contains zeros. When this field contains a valid address, Extended Starting Address must also contain a valid address.

7.21 Memory Device Mapped Address (Type 20)

This structure maps memory address space usually to a device-level granularity. Details are provided in Table 82.

One structure is present for each contiguous address range described.

NOTE　　A Memory Device Mapped Address structure is provided only if a Memory Device has a mapped address; there is no provision within this structure to map a zero-length address space.

See 7.17, 7.18, and 7.21 for more information.

Table 82 – Memory Device Mapped Address (Type 20) structure

Offset	Spec. Version	Name	Length	Value	Description
00h	2.1+	Type	BYTE	20	Memory Device Mapped Address indicator
01h	2.1+	Length	BYTE	Varies	Length of the structure, 13h for version 2.1, 23h for version 2.7 and later.
02h	2.1+	Handle	WORD	Varies	Handle, or instance number, associated with the structure

Offset	Spec. Version	Name	Length	Value	Description
04h	2.1+	Starting Address	DWORD	Varies	Physical address, in kilobytes, of a range of memory mapped to the referenced Memory Device When the field value is FFFF FFFFh the actual address is stored in the Extended Starting Address field. When this field contains a valid address, Ending Address must also contain a valid address. When this field contains FFFF FFFFh, Ending Address must also contain FFFF FFFFh.
08h	2.1+	Ending Address	DWORD	Varies	Physical ending address of the last kilobyte of a range of addresses mapped to the referenced Memory Device When the field value is FFFF FFFFh the actual address is stored in the Extended Ending Address field. When this field contains a valid address, Starting Address must also contain a valid address.
0Ch	2.1+	Memory Device Handle	WORD	Varies	Handle, or instance number, associated with the Memory Device structure to which this address range is mapped Multiple address ranges can be mapped to a single Memory Device.
0Eh	2.1+	Memory Array Mapped Address Handle	WORD	Varies	Handle, or instance number, associated with the Memory Array Mapped Address structure to which this device address range is mapped Multiple address ranges can be mapped to a single Memory Array Mapped Address.
10h	2.1+	Partition Row Position	BYTE	Varies	Position of the referenced Memory Device in a row of the address partition For example, if two 8-bit devices form a 16-bit row, this field's value is either 1 or 2. The value 0 is reserved. If the position is unknown, the field contains FFh.
11h	2.1+	Interleave Position	BYTE	Varies	Position of the referenced Memory Device in an interleave The value 0 indicates non-interleaved, 1 indicates first interleave position, 2 the second interleave position, and so on. If the position is unknown, the field contains FFh. EXAMPLES: In a 2:1 interleave, the value 1 indicates the device in the "even" position. In a 4:1 interleave, the value 1 indicates the first of four possible positions.
12h	2.1+	Interleaved Data Depth	BYTE	Varies	Maximum number of consecutive rows from the referenced Memory Device that are accessed in a single interleaved transfer If the device is not part of an interleave, the field contains 0; if the interleave configuration is unknown, the value is FFh. EXAMPLES: If a device transfers two rows each time it is read, its Interleaved Data Depth is set to 2. If that device is 2:1 interleaved and in Interleave Position 1, the rows mapped to that device are 1, 2, 5, 6, 9, 10, etc.

Offset	Spec. Version	Name	Length	Value	Description
13h	2.7+	Extended Starting Address	QWORD	Varies	Physical address, in bytes, of a range of memory mapped to the referenced Memory Device This field is valid when Starting Address contains the value FFFF FFFFh. If Starting Address contains a value other than FFFF FFFFh, this field contains zeros. When this field contains a valid address, Extended Ending Address must also contain a valid address.
1Bh	2.7+	Extended Ending Address	QWORD	Varies	Physical ending address, in bytes, of the last of a range of addresses mapped to the referenced Memory Device This field is valid when both Starting Address and Ending Address contain the value FFFF FFFFh. If Ending Address contains a value other than FFFF FFFFh, this field contains zeros. When this field contains a valid address, Extended Starting Address must also contain a valid address.

1487 **7.22 Built-in Pointing Device (Type 21)**

1488 This structure describes the attributes of the built-in pointing device for the system. Details are provided in
1489 Table 83.

1490 The presence of this structure does not imply that the built-in pointing device is active for the system's
1491 use.

1492   **Table 83 – Built-in Pointing Device (Type 21) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.1+	Type	BYTE	21	Built-in Pointing Device indicator
01h	2.1+	Length	BYTE	07h	Length of the structure
02h	2.1+	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	2.1+	Type	BYTE	ENUM	Type of pointing device; see 7.22.1
05h	2.1+	Interface	BYTE	ENUM	Interface type for the pointing device; see 7.22.2
06h	2.1+	Number of Buttons	BYTE	Varies	Number of buttons on the pointing device If the device has three buttons, the field value is 03h.

1493 **7.22.1 Pointing Device — Type**

1494 Table 84 shows what the byte values mean for the Pointing Device — Type field.

1495 NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

1496   **Table 84 – Pointing Device: Type field**

Byte Value	Meaning
01h	Other

Byte Value	Meaning
02h	Unknown
03h	Mouse
04h	Track Ball
05h	Track Point
06h	Glide Point
07h	Touch Pad
08h	Touch Screen
09h	Optical Sensor

7.22.2 Pointing Device — Interface

Table 85 shows what the byte values mean for the Pointing Device — Interface field.

Table 85 – Pointing Device: Interface field

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Serial
04h	PS/2
05h	Infrared
06h	HP-HIL
07h	Bus mouse
08h	ADB (Apple Desktop Bus)
A0h	Bus mouse DB-9
A1h	Bus mouse micro-DIN
A2h	USB

7.23 Portable Battery (Type 22)

This structure describes the attributes of the portable battery or batteries for the system. The structure contains the static attributes for the group. Each structure describes a single battery pack's attributes. Details are provided in Table 86.

Table 86 – Portable Battery (Type 22) structure

Offset	Spec. Version	Name	Length	Value	Description
00h	2.1+	Type	BYTE	22	Portable Battery indicator
01h	2.1+	Length	BYTE	1Ah	Length of the structure
02h	2.1+	Handle	WORD	Varies	Handle, or instance number, associated with the structure

Offset	Spec. Version	Name	Length	Value	Description
04h	2.1+	Location	BYTE	STRING	Number of the string that identifies the location of the battery EXAMPLE: "in the back, on the left-hand side"
05h	2.1+	Manufacturer	BYTE	STRING	Number of the string that names the company that manufactured the battery
06h	2.1+	Manufacture Date	BYTE	STRING	Number of the string that identifies the date on which the battery was manufactured Version 2.2+ implementations that use a Smart Battery set this field to 0 (no string) to indicate that the SBDS Manufacture Date field contains the information.
07h	2.1+	Serial Number	BYTE	STRING	Number of the string that contains the serial number for the battery Version 2.2+ implementations that use a Smart Battery set this field to 0 (no string) to indicate that the SBDS Serial Number field contains the information.
08h	2.1+	Device Name	BYTE	STRING	Number of the string that names the battery device EXAMPLE: "DR-36"
09h	2.1+	Device Chemistry	BYTE	ENUM	Identifies the battery chemistry; see 7.23.1 Version 2.2+ implementations that use a Smart Battery set this field to 02h (Unknown) to indicate that the SBDS Device Chemistry field contains the information.
0Ah	2.1+	Design Capacity	WORD	Varies	Design capacity of the battery in mWatt-hours If the value is unknown, the field contains 0. For version 2.2+ implementations, this value is multiplied by the Design Capacity Multiplier to produce the actual value.
0Ch	2.1+	Design Voltage	WORD	Varies	Design voltage of the battery in mVolts If the value is unknown, the field contains 0.
0Eh	2.1+	SBDS Version Number	BYTE	STRING	Number of the string that contains the Smart Battery Data Specification version number supported by this battery If the battery does not support the function, no string is supplied.
0Fh	2.1+	Maximum Error in Battery Data	BYTE	Varies	Maximum error (as a percentage in the range 0 to 100) in the Watt-hour data reported by the battery, indicating an upper bound on how much additional energy the battery might have above the energy it reports having If the value is unknown, the field contains FFh.
10h	2.2+	SBDS Serial Number	WORD	Varies	16-bit value that identifies the battery's serial number This value, when combined with the Manufacturer, Device Name, and Manufacture Date, uniquely identifies the battery. The Serial Number field must be set to 0 (no string) for this field to be valid.

Offset	Spec. Version	Name	Length	Value	Description
12h	2.2+	SBDS Manufacture Date	WORD	Varies	Date the cell pack was manufactured, in packed format: Bits 15:9 Year, biased by 1980, in the range 0 to 127 Bits 8:5 Month, in the range 1 to 12 Bits 4:0 Date, in the range 1 to 31 EXAMPLE: 01 February 2000 would be identified as 0010 1000 0100 0001b (0x2841) The Manufacture Date field must be set to 0 (no string) for this field to be valid.
14h	2.2+	SBDS Device Chemistry	BYTE	STRING	Number of the string that identifies the battery chemistry (for example, "PbAc") The Device Chemistry field must be set to 02h (Unknown) for this field to be valid.
15h	2.2+	Design Capacity Multiplier	BYTE	Varies	Multiplication factor of the Design Capacity value, which assures that the mWatt hours value does not overflow for SBDS implementations The multiplier default is 1, SBDS implementations use the value 10 to correspond to the data as returned from the SBDS Function 18h.
16h	2.2+	OEM-specific	DWORD	Varies	Contains OEM- or BIOS vendor-specific information

1505 **7.23.1 Portable Battery — Device Chemistry**

1506 Table 87 shows what the byte values mean for the Portable Battery — Device Chemistry field.

1507 NOTE   Refer to 6.3 for the CIM properties associated with this enumerated value.

1508 **Table 87 – Portable Battery: Device Chemistry field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Lead Acid
04h	Nickel Cadmium
05h	Nickel metal hydride
06h	Lithium-ion
07h	Zinc air
08h	Lithium Polymer

1509 **7.24 System Reset (Type 23)**

1510 This structure describes whether Automatic System Reset functions are enabled (*Status*). Details are
1511 provided in Table 88.

1512   If the system has a watchdog timer and the timer is not reset (*Timer Reset*) before the *Interval* elapses,
1513   an automatic system reset occurs. The system re-boots according to the *Boot Option*. This function may
1514   repeat until the *Limit* is reached, at which time the system re-boots according to the *Boot Option at Limit*.

1515   NOTE   This structure type was added for version 2.2 of this specification.

1516   **Table 88 – System Reset (Type 23) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	23	System Reset indicator
01h	Length	BYTE	0Dh	Length of the structure
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure

04h	Capabilities	BYTE	Bit-field	Identifies the system-reset capabilities for the system
				
				Bits 7:6 Reserved for future assignment by this specification; set to 00b
				
				Bit 5 System contains a watchdog timer; either True (1) or False (0)
				
				Bits 4:3 Boot Option on Limit. Identifies one of the following system actions to be taken when the Reset Limit is reached:
				
				    00b Reserved, do not use.
				    01b Operating system
				    10b System utilities
				    11b Do not reboot
				
				Bits 2:1 Boot Option. Indicates one of the following actions to be taken after a watchdog reset:
				
				    00b Reserved, do not use.
				    01b Operating system
				    10b System utilities
				    11b Do not reboot
				
				Bit 0 Status. Identifies whether (1) or not (0) the system reset is enabled by the user.

05h	Reset Count	WORD	Varies	Number of automatic system resets since the last intentional reset A value of 0FFFFh indicates unknown.
07h	Reset Limit	WORD	Varies	Number of consecutive times the system reset is attempted A value of 0FFFFh indicates unknown.
09h	Timer Interval	WORD	Varies	Number of minutes to use for the watchdog timer If the timer is not reset within this interval, the system reset timeout begins. A value of 0FFFFh indicates unknown.
0Bh	Timeout	WORD	Varies	Number of minutes before the reboot is initiated It is used after a system power cycle, system reset (local or remote), and automatic system reset. A value of 0FFFFh indicates unknown.

1517   ## 7.25 Hardware Security (Type 24)

1518   This structure describes the system-wide hardware security settings. Details are provided in Table 89.

1519   NOTE   This structure type was added in version 2.2 of this specification.

1520   **Table 89 – Hardware Security (Type 24) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	24	Hardware Security indicator
01h	Length	BYTE	05h	Length of the structure
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Hardware Security Settings	BYTE	Bit-field	Identifies the password and reset status for the system: Bits 7:6 Power-on Password Status value: 　　00b Disabled 　　01b Enabled 　　10b Not Implemented 　　11b Unknown Bits 5:4 Keyboard Password Status value: 　　00b Disabled 　　01b Enabled 　　10b Not Implemented 　　11b Unknown Bits 3:2 Administrator Password Status value: 　　00b Disabled 　　01b Enabled 　　10b Not Implemented 　　11b Unknown Bits 1:0 Front Panel Reset Status value: 　　00b Disabled 　　01b Enabled 　　10b Not Implemented 　　11b Unknown

1521   7.26 System Power Controls (Type 25)

1522   This structure describes the attributes for controlling the main power supply to the system. Details are
1523   provided in Table 90.

1524   Software that interprets this structure uses the month, day, hour, minute, and second values to determine
1525   the number of seconds until the next power-on of the system. The presence of this structure implies that a
1526   timed power-on facility is available for the system.

1527   NOTE   This structure type was added in version 2.2 of the specification.

1528   **Table 90 – System Power Controls (Type 25) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	25	System Power Controls indicator

Offset	Name	Length	Value	Description
01h	Length	BYTE	09h	Length of the structure
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Next Scheduled Power-on Month	BYTE	Varies	BCD value of the month on which the next scheduled power-on is to occur, in the range 01h to 12h; see 7.26.1
05h	Next Scheduled Power-on Day-of-month	BYTE	Varies	BCD value of the day-of-month on which the next scheduled power-on is to occur, in the range 01h to 31h; see 7.26.1
06h	Next Scheduled Power-on Hour	BYTE	Varies	BCD value of the hour on which the next scheduled power-on is to occur, in the range 00h to 23h; see 7.26.1
07h	Next Scheduled Power-on Minute	BYTE	Varies	BCD value of the minute on which the next scheduled power-on is to occur, in the range 00h to 59h; see 7.26.1
08h	Next Scheduled Power-on Second	BYTE	Varies	BCD value of the second on which the next scheduled power-on is to occur, in the range 00h to 59h; see 7.26.1

7.26.1 System Power Controls — Calculating the Next Scheduled Power-on Time

The DMTF *System Power Controls* group contains a *Next Scheduled Power-on Time*, specified as the number of seconds until the next scheduled power-on of the system. Management software uses the date and time information specified in the associated SMBIOS structure to calculate the total number of seconds.

Any date or time field in the structure whose value is outside of the field's specified range does not contribute to the total-seconds count. For example, if the Month field contains the value 0xFF the next power-on is scheduled to fall within the next month, perhaps on a specific day-of-month and time.

7.27 Voltage Probe (Type 26)

This describes the attributes for a voltage probe in the system. Each structure describes a single voltage probe. Details are provided in Table 91.

NOTE   This structure type was added in version 2.2 of this specification.

Table 91 – Voltage Probe (Type 26) structure

Offset	Name	Length	Value	Description
00h	Type	BYTE	26	Voltage Probe indicator
01h	Length	BYTE	Varies	Length of the structure, at least 14h
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Description	BYTE	STRING	Number of the string that contains additional descriptive information about the probe or its location
05h	Location and Status	BYTE	Bit-field	Probe's physical location and status of the voltage monitored by this voltage probe; see 7.27.1
06h	Maximum Value	WORD	Varies	Maximum voltage level readable by this probe, in millivolts If the value is unknown, the field is set to 0x8000.
08h	Minimum Value	WORD	Varies	Minimum voltage level readable by this probe, in millivolts If the value is unknown, the field is set to 0x8000.

Offset	Name	Length	Value	Description
0Ah	Resolution	WORD	Varies	Resolution for the probe's reading, in tenths of millivolts If the value is unknown, the field is set to 0x8000.
0Ch	Tolerance	WORD	Varies	Tolerance for reading from this probe, in plus/minus millivolts If the value is unknown, the field is set to 0x8000.
0Eh	Accuracy	WORD	Varies	Accuracy for reading from this probe, in plus/minus $1/100^{th}$ of a percent If the value is unknown, the field is set to 0x8000.
10h	OEM-defined	DWORD	Varies	OEM- or BIOS vendor-specific information.
14h	Nominal Value	WORD	Varies	Nominal value for the probe's reading in millivolts If the value is unknown, the field is set to 0x8000. This field is present in the structure only if the structure's length is larger than 14h.

1542 **7.27.1 Voltage Probe — Location and Status**

1543 Table 92 provides details about the Location and Status fields.

1544 **Table 92 – Voltage Probe: Location and Status fields**

Bit Range	Field Name	Value	Meaning
7:5	Status	001.....	Other
		010.....	Unknown
		011.....	OK
		100.....	Non-critical
		101.....	Critical
		110.....	Non-recoverable
4:0	Location	...00001	Other
		...00010	Unknown
		...00011	Processor
		...00100	Disk
		...00101	Peripheral Bay
		...00110	System Management Module
		...00111	Motherboard
		...01000	Memory Module
		...01001	Processor Module
		...01010	Power Unit
		...01011	Add-in Card

1545   **7.28 Cooling Device (Type 27)**

1546   This structure describes the attributes for a cooling device in the system. Each structure describes a
1547   single cooling device. Details are provided in Table 93.

1548   NOTE   This structure type was added in version 2.2 of this specification.

1549   **Table 93 – Cooling Device (Type 27) structure**

Offset	Spec. Version	Name	Length	Value	Description
00h	2.2+	Type	BYTE	27	Cooling Device indicator
01h	2.2+	Length	BYTE	Varies	Length of the structure, at least 0Ch
02h	2.2+	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	2.2+	Temperature Probe Handle	WORD	Varies	Handle, or instance number, of the temperature probe (see 7.29) monitoring this cooling device A value of 0xFFFF indicates that no probe is provided.
06h	2.2+	Device Type and Status	BYTE	Bit-field	Cooling device type and status; see 7.28.1
07h	2.2+	Cooling Unit Group	BYTE	Varies	Cooling unit group to which this cooling device is associated Having multiple cooling devices in the same cooling unit implies a redundant configuration. The value is 00h if the cooling device is not a member of a redundant cooling unit. Non-zero values imply redundancy and that at least one other cooling device will be enumerated with the same value.
08h	2.2+	OEM-defined	DWORD	Varies	OEM- or BIOS vendor-specific information
0Ch	2.2+	Nominal Speed	WORD	Varies	Nominal value for the cooling device's rotational speed, in revolutions-per-minute (rpm) If the value is unknown or the cooling device is non-rotating, the field is set to 0x8000. This field is present in the structure only if the structure's length is larger than 0Ch.
0Eh	2.7+	Description	BYTE	STRING	Number of the string that contains additional descriptive information about the cooling device or its location This field is present in the structure only if the structure's length is 0Fh or larger.

1550   **7.28.1 Cooling Device — Device Type and Status**

1551   Table 94 provides details about the Device Type and Status fields.

1552 **Table 94 – Cooling Device: Device Type and Status fields**

Bit Range	Field Name	Value	Meaning
7:5	Status	001.....	Other
		010.....	Unknown
		011.....	OK
		100.....	Non-critical
		101.....	Critical
		110.....	Non-recoverable
4:0	Device Type	...00001	Other
		...00010	Unknown
		...00011	Fan
		...00100	Centrifugal Blower
		...00101	Chip Fan
		...00110	Cabinet Fan
		...00111	Power Supply Fan
		...01000	Heat Pipe
		...01001	Integrated Refrigeration
		...10000	Active Cooling
		...10001	Passive Cooling

1553 7.29 Temperature Probe (Type 28)

1554 This structure describes the attributes for a temperature probe in the system. Each structure describes a
1555 single temperature probe. Table 95 provides details.

1556 NOTE   This structure type was added in version 2.2 of this specification.

1557 **Table 95 – Temperature Probe (Type 28) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	28	Temperature Probe indicator
01h	Length	BYTE	Varies	Length of the structure, at least 14h
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Description	BYTE	STRING	Number of the string that contains additional descriptive information about the probe or its location
05h	Location and Status	BYTE	Bit-field	Probe's physical location and the status of the temperature monitored by this temperature probe; see 7.29.1
06h	Maximum Value	WORD	Varies	Maximum temperature readable by this probe, in $1/10^{th}$ degrees C If the value is unknown, the field is set to 0x8000.
08h	Minimum Value	WORD	Varies	Minimum temperature readable by this probe, in $1/10^{th}$ degrees C If the value is unknown, the field is set to 0x8000.

Offset	Name	Length	Value	Description
0Ah	Resolution	WORD	Varies	Resolution for the probe's reading, in 1/1000th degrees C If the value is unknown, the field is set to 0x8000.
0Ch	Tolerance	WORD	Varies	Tolerance for reading from this probe, in plus/minus 1/10th degrees C If the value is unknown, the field is set to 0x8000.
0Eh	Accuracy	WORD	Varies	Accuracy for reading from this probe, in plus/minus 1/100th of a percent If the value is unknown, the field is set to 0x8000.
10h	OEM-defined	DWORD	Varies	OEM- or BIOS vendor-specific information
14h	Nominal Value	WORD	Varies	Nominal value for the probe's reading in 1/10th degrees C If the value is unknown, the field is set to 0x8000. This field is present in the structure only if the structure's Length is larger than 14h.

7.29.1 Temperature Probe — Location and Status

1558

1559   Table 96 provides details about the Location and Status fields.

1560   NOTE   Refer to 6.3 for the CIM properties associated with these enumerated values.

1561   **Table 96 – Temperature Probe: Location and Status field**

Bit Range	Field Name	Value	Meaning
7:5	Status	001.....	Other
		010.....	Unknown
		011.....	OK
		100.....	Non-critical
		101.....	Critical
		110.....	Non-recoverable
4:0	Location	...00001	Other
		...00010	Unknown
		...00011	Processor
		...00100	Disk
		...00101	Peripheral Bay
		...00110	System Management Module
		...00111	Motherboard
		...01000	Memory Module
		...01001	Processor Module
		...01010	Power Unit
		...01011	Add-in Card
		...01100	Front Panel Board
		...01101	Back Panel Board
		...01110	Power System Board
		...01111	Drive Back Plane

1562 **7.30 Electrical Current Probe (Type 29)**

1563 This structure describes the attributes for an electrical current probe in the system. Each structure
1564 describes a single electrical current probe. Table 97 provides details.

1565 NOTE This structure type was added in version 2.2 of this specification.

1566 **Table 97 – Electrical Current Probe (Type 29) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	29	Electrical Current Probe indicator
01h	Length	BYTE	Varies	Length of the structure, at least 14h
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Description	BYTE	STRING	Number of the string that contains additional descriptive information about the probe or its location
05h	Location and Status	BYTE	ENUM	Defines the probe's physical location and the status of the current monitored by this current probe; see 7.30.1
06h	Maximum Value	WORD	Varies	Maximum current readable by this probe, in milliamps If the value is unknown, the field is set to 0x8000.
08h	Minimum Value	WORD	Varies	Minimum current readable by this probe, in milliamps If the value is unknown, the field is set to 0x8000.
0Ah	Resolution	WORD	Varies	Resolution for the probe's reading, in tenths of milliamps If the value is unknown, the field is set to 0x8000.
0Ch	Tolerance	WORD	Varies	Tolerance for reading from this probe, in plus/minus milliamps If the value is unknown, the field is set to 0x8000.
0Eh	Accuracy	WORD	Varies	Accuracy for reading from this probe, in plus/minus $1/100^{th}$ of a percent If the value is unknown, the field is set to 0x8000.
10h	OEM-defined	DWORD	Varies	OEM- or BIOS vendor-specific information
14h	Nominal Value	WORD	Varies	Nominal value for the probe's reading in milliamps If the value is unknown, the field is set to 0x8000. This field is present in the structure only if the structure's length is larger than 14h.

1567 **7.30.1 Current Probe — Location and Status**

1568 Table 98 provides details about the Location and Status fields.

1569 NOTE Refer to 6.3 for the CIM properties associated with these enumerated values.

1570   **Table 98 – Current Probe: Location and Status field**

Bit Range	Field Name	Value	Meaning
7:5	Status	001.....	Other
		010.....	Unknown
		011.....	OK
		100.....	Non-critical
		101.....	Critical
		110.....	Non-recoverable
4:0	Location	...00001	Other
		...00010	Unknown
		...00011	Processor
		...00100	Disk
		...00101	Peripheral Bay
		...00110	System Management Module
		...00111	Motherboard
		...01000	Memory Module
		...01001	Processor Module
		...01010	Power Unit
		...01011	Add-in Card

1571 7.31 Out-of-Band Remote Access (Type 30)

1572 This structure describes the attributes and policy settings of a hardware facility that may be used to gain
1573 remote access to a hardware system when the operating system is not available due to power-down
1574 status, hardware failures, or boot failures. Details are provided in Table 99.

1575 NOTE   This structure type was added in version 2.2 of this specification.

1576   **Table 99 – Out-of-Band Remote Access (Type 30) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	30	Out-of-Band Remote Access indicator
01h	Length	BYTE	06h	Length of the structure
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Manufacturer Name	BYTE	STRING	Number of the string that contains the manufacturer of the out-of-band access facility

Offset	Name	Length	Value	Description
05h	Connections	BYTE	Bit-field	Current remote-access connections: Bits 7:2 Reserved for future definition by this specification; set to all zeros Bit 1 Outbound Connection Enabled. Identifies whether (1) or not (0) the facility is allowed to initiate outbound connections to contact an alert management facility when critical conditions occur Bit 0 Inbound Connection Enabled. Identifies whether (1) or not (0) the facility is allowed to initiate outbound connections to receive incoming connections for the purpose of remote operations or problem management

7.32 Boot Integrity Services (BIS) Entry Point (Type 31)

Structure type 31 (decimal) is reserved for use by the Boot Integrity Services (BIS). Refer to the *Boot Integrity Services API Specification* for details.

NOTE   This structure type was added in version 2.3 of this specification.

7.33 System Boot Information (Type 32)

The client system firmware (for example, BIOS) communicates the *System Boot Status* to the client's Pre-boot Execution Environment (PXE) boot image or OS-present management application through this structure. Table 100 provides details on this structure.

When used in the PXE environment, for example, this code identifies the reason the PXE was initiated and can be used by boot-image software to further automate an enterprise's PXE sessions. For example, an enterprise could choose to automatically download a hardware-diagnostic image to a client whose reason code indicated either a firmware- or an operating system-detected hardware failure.

NOTE   This structure type was added in version 2.3 of this specification.

Table 100 – System Boot Information (Type 32) structure

Offset	Name	Length	Value	Description
00h	Type	BYTE	32	System Boot Information structure identifier
01h	Length	BYTE	Varies	Length of the structure, in bytes; at least 0Bh
02h	Handle	WORD	Varies	
04h	Reserved	6 BYTEs	00h	Reserved for future assignment by this specification; all bytes are set to 00h
0Ah	Boot Status	Length-10 Bytes	Varies	Status and Additional Data fields that identify the boot status See 7.33.1 for additional information.

7.33.1 System boot status

Table 101 provides information about system boot status.

1593   **Table 101 – System boot status**

Description	Status	Additional Data
No errors detected	0	None
No bootable media	1	None
"normal" operating system failed to load	2	None
Firmware-detected hardware failure, including "unknown" failure types	3	None
Operating system-detected hardware failure For ACPI operating systems, the system firmware might set this reason code when the OS reports a boot failure through interfaces defined in the *Simple Boot Flag Specification*.	4	None
User-requested boot, usually through a keystroke	5	None
System security violation	6	None
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 (through a platform-specific interface) the system to boot to the PXE and request a specific boot-image.	7	Varies
System watchdog timer expired, causing the system to reboot	8	None
Reserved for future assignment by this specification	9-127	Varies
Vendor/OEM-specific implementations The Vendor/OEM identifier is the "Manufacturer" string found in the System Information structure.	128-191	Varies
Product-specific implementations The product identifier is formed by the concatenation of the "Manufacturer" and "Product Name" strings found in the System Information structure.	192-255	Varies

1594   7.34 64-Bit Memory Error Information (Type 33)

1595   This structure describes an error within a Physical Memory Array, when the error address is above 4G
1596   (0xFFFFFFFF). Table 102 provides details.

1597   NOTE   This structure type was added in version 2.3 of this specification.

1598   **Table 102 – 64-Bit Memory Error Information (Type 33) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	33	64-bit Memory Error Information type
01h	Length	BYTE	1Fh	Length of the structure
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Error Type	BYTE	ENUM	Type of error that is associated with the current status reported for the memory array or device See 7.19.1 for definitions.
05h	Error Granularity	BYTE	ENUM	Granularity (for example, device versus Partition) to which the error can be resolved See 7.19.2 for definitions.
06h	Error Operation	BYTE	ENUM	Memory access operation that caused the error See 7.19.3 for definitions.

Offset	Name	Length	Value	Description
07h	Vendor Syndrome	DWORD	Varies	Vendor-specific ECC syndrome or CRC data associated with the erroneous access If the value is unknown, this field contains 0000 0000h.
0Bh	Memory Array Error Address	QWORD	Varies	64-bit physical address of the error based on the addressing of the bus to which the memory array is connected If the address is unknown, this field contains 8000 0000 0000 0000h.
13h	Device Error Address	QWORD	Varies	64-bit physical address of the error relative to the start of the failing memory device, in bytes If the address is unknown, this field contains 8000 0000 0000 0000h.
1Bh	Error Resolution	DWORD	Varies	Range, in bytes, within which the error can be determined, when an error address is given If the range is unknown, this field contains 8000 0000h.

7.35 Management Device (Type 34)

The information in this structure defines the attributes of a *Management Device*. Table 103 provides details.

A *Management Device* might control one or more fans or voltage, current, or temperature probes as defined by one or more *Management Device Component* structures. See 7.36 for more information.

NOTE   This structure type was added in version 2.3 of this specification.

Table 103 – Management Device (Type 34) structure

Offset	Name	Length	Value	Description
00h	Type	BYTE	34	Management Device indicator
01h	Length	BYTE	0Bh	Length of the structure
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Description	BYTE	STRING	Number of the string that contains additional descriptive information about the device or its location
05h	Type	BYTE	Varies	Device's type; see 7.35.1
06h	Address	DWORD	Varies	Device's address
0Ah	Address Type	BYTE	Varies	Type of addressing used to access the device; see 7.35.2

7.35.1 Management Device — Type

Table 104 shows what the byte values mean for the Management Device — Type field.

Table 104 – Management Device: Type field

Byte Value	Meaning
01h	Other
02h	Unknown
03h	National Semiconductor LM75

Byte Value	Meaning
04h	National Semiconductor LM78
05h	National Semiconductor LM79
06h	National Semiconductor LM80
07h	National Semiconductor LM81
08h	Analog Devices ADM9240
09h	Dallas Semiconductor DS1780
0Ah	Maxim 1617
0Bh	Genesys GL518SM
0Ch	Winbond W83781D
0Dh	Holtek HT82H791

1609   **7.35.2 Management Device — Address Type**

1610   Table 105 shows what the byte values mean for the Management Device — Address Type field.

1611   **Table 105 – Management Device: Address Type field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	I/O Port
04h	Memory
05h	SM Bus

1612   **7.36 Management Device Component (Type 35)**

1613   This structure associates a cooling device or environmental probe with structures that define the
1614   controlling hardware device and (optionally) the component's thresholds. Table 106 provides details.

1615   NOTE   This structure type was added in version 2.3 of this specification.

1616   **Table 106 – Management Device Component (Type 35) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	35	Management Device Component indicator
01h	Length	BYTE	0Bh	Length of the structure
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Description	BYTE	STRING	Number of the string that contains additional descriptive information about the component
05h	Management Device Handle	WORD	Varies	Handle, or instance number, of the Management Device (see 7.35) that contains this component

Offset	Name	Length	Value	Description
07h	Component Handle	WORD	Varies	Handle, or instance number, of the probe or cooling device that defines this component See 7.27, 7.28, 7.29, and 7.30.
09h	Threshold Handle	WORD	Varies	Handle, or instance number, associated with the device thresholds; see 7.37 A value of 0FFFFh indicates that no Threshold Data structure is associated with this component.

1617 7.37 Management Device Threshold Data (Type 36)

1618 The information in this structure defines threshold information for a component (probe or cooling-unit)
1619 contained within a *Management Device*. Table 107 provides details.

1620 For each threshold field present in the structure:

1621 • The threshold units (millivolts, milliamps, 1/10th degrees C, or RPMs) are as defined by the
1622 associated probe or cooling-unit component structure.

1623 • If the value is unavailable, the field is set to 0x8000.

1624 NOTE This structure type was added in version 2.3 of this specification.

1625 **Table 107 – Management Device Threshold Data (Type 36) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	36	Management Device Threshold Data structure indicator
01h	Length	BYTE	10h	Length of the structure
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Lower Threshold – Non-critical	WORD	Varies	Lower non-critical threshold for this component
06h	Upper Threshold – Non-critical	WORD	Varies	Upper non-critical threshold for this component
08h	Lower Threshold – Critical	WORD	Varies	Lower critical threshold for this component
0Ah	Upper Threshold – Critical	WORD	Varies	Upper critical threshold for this component
0ch	Lower Threshold – Non-recoverable	WORD	Varies	Lower non-recoverable threshold for this component
0eh	Upper Threshold – Non-recoverable	WORD	Varies	Upper non-recoverable threshold for this component

1626 7.38 Memory Channel (Type 37)

1627 The information in this structure provides the correlation between a Memory Channel and its associated
1628 Memory Devices. Table 108 provides details.

1629 Each device presents one or more loads to the channel; the sum of all device loads cannot exceed the
1630 channel's defined maximum.

1631 NOTE This structure type was added in version 2.3 of this specification.

1632   **Table 108 – Memory Channel (Type 37) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	37	Management Device Threshold Data structure indicator
01h	Length	BYTE		Length of the structure, computed by the BIOS as 7 + 3 * (Memory Device Count) NOTE: To allow future structure growth by appending information after the Load/Handle list, this field must not be used to determine the number of memory devices specified within the structure.
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Channel Type	BYTE	Varies	Type of memory associated with the channel; see 7.38.1
05h	Maximum Channel Load	BYTE	Varies	Maximum load supported by the channel; the sum of all device loads cannot exceed this value
06h	Memory Device Count (n)	BYTE	Varies	Number of Memory Devices (Type 11h) that are associated with this channel This value also defines the number of Load/Handle pairs that follow.
07h	Memory1 Device Load	BYTE	Varies	Channel load provided by the first Memory Device associated with this channel
08h	Memory Device1 Handle	WORD	Varies	Structure handle that identifies the first Memory Device associated with this channel
7 + 3*(n-1)	Memory Device$_n$ Load	BYTE	Varies	Channel load provided by the nth Memory Device associated with this channel
8 + 3*(n-1)	Memory Device$_n$ Handle	WORD	Varies	Structure handle that identifies the nth Memory Device associated with this channel

1633   **7.38.1 Memory Channel — Channel Type**

1634   Table 109 shows what the byte values mean for the Memory Channel — Channel Type field.

1635   NOTE: Enumerated values are controlled by the DMTF, not by this specification.

1636   **Table 109 – Memory Channel: Channel Type field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	RamBus
04h	SyncLink

1637   **7.39 IPMI Device Information (Type 38)**

1638   The information in this structure defines the attributes of an Intelligent Platform Management Interface
1639   (IPMI) Baseboard Management Controller (BMC). Table 110 provides the details about this structure.
1640   Refer to the *Intelligent Platform Management Interface (IPMI) Interface Specification* for full
1641   documentation of IPMI and additional information on the use of this structure.

1642　The Type 42 structure can also be used to describe a physical management controller host interface and
1643　one or more protocols that share that interface. If IPMI is not shared with other protocols, either the Type
1644　38 or Type 42 structures can be used. Providing Type 38 is recommended for backward compatibility.
1645　See 7.43 for additional information on Type 42.

1646　　　　　　　　　　　**Table 110 – IPMI Device Information (Type 38) Structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	38	IPMI Device Information structure indicator
01h	Length	BYTE	Varies	Length of the structure, a minimum of 10h
02h	Handle	WORD	Varies	
04h	Interface Type	BYTE	ENUM	Baseboard Management Controller (BMC) interface type; see 7.39.1
05h	IPMI Specification Revision	BYTE	Varies	IPMI specification revision, in BCD format, to which the BMC was designed Bits 7:4 hold the most significant digit of the revision, while bits 3:0 hold the least significant bits. EXAMPLE: A value of 10h indicates revision 1.0.
06h	I2C Slave Address	BYTE	Varies	Slave address on the I2C bus of this BMC
07h	NV Storage Device Address	BYTE	Varies	Bus ID of the NV storage device If no storage device exists for this BMC, the field is set to 0FFh.
08h	Base Address	QWORD	Varies	Base address (either memory-mapped or I/O) of the BMC If the least-significant bit of the field is a 1, the address is in I/O space; otherwise, the address is memory-mapped. Refer to the *IPMI Interface Specification* for usage details.

Offset	Name	Length	Value	Description

10h	Base Address Modifier / Interrupt Info	BYTE	Varies	Base Address Modifier (This field is unused and set to 00h for SSIF.)
				
				 bit 7:6 – Register spacing
				   00b = Interface registers are on successive byte boundaries.
				   01b = Interface registers are on 32-bit boundaries.
				   10b = Interface registers are on 16-byte boundaries.
				   11b = Reserved.
				 bit 5 – Reserved. Return as 0b.
				 bit 4 – LS-bit for addresses:
				   0b = Address bit 0 = 0b
				   1b = Address bit 0 = 1b
				
				 Interrupt Info
				 Identifies the type and polarity of the interrupt associated with the IPMI system interface, if any:
				 bit 3 – Interrupt Info
				   1b = Interrupt information specified
				   0b = Interrupt information not specified
				 bit 2 – Reserved. Return as 0b
				 bit 1 – Interrupt Polarity
				   1b = active high
				   0b = active low
				 bit 0 – Interrupt Trigger Mode
				   1b = level
				   0b = edge

11h	Interrupt Number	BYTE	Varies	Interrupt number for IPMI System Interface 00h = unspecified/unsupported

7.39.1 IPMI Device Information — BMC Interface Type

Table 111 shows what the byte values mean for the IPMI Device Information — BMC Interface Type field.

Table 111 – IPMI Device Information: BMC Interface Type field

Byte Value	Meaning
00h	Unknown
01h	KCS: Keyboard Controller Style
02h	SMIC: Server Management Interface Chip
03h	BT: Block Transfer
04h to 0FFh	Reserved for future assignment by this specification

7.40 System Power Supply (Type 39)

This structure identifies attributes of a system power supply. Table 112 provides details. One instance of this structure is present for each possible power supply in a system.

NOTE   This structure type was added in version 2.3.1 of this specification.

1654   **Table 112 – System Power Supply (Type 39) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	39	Power Supply Structure indicator
01h	Length	BYTE	Varies	Length of the structure, a minimum of 10h
02h	Handle	WORD	Varies	Handle, or instance number, associated with the power supply structure
04h	Power Unit Group	BYTE	Varies	Power unit group to which this power supply is associated Specifying the same Power Unit Group value for more than one System Power Supply structure indicates a redundant power supply configuration. The field's value is 00h if the power supply is not a member of a redundant power unit. Non-zero values imply redundancy and that at least one other power supply will be enumerated with the same value.
05h	Location	BYTE	STRING	Number of the string that identifies the location of the power supply EXAMPLES: "in the back, on the left-hand side" or "Left Supply Bay"
06h	Device Name	BYTE	STRING	Number of the string that names the power supply device EXAMPLE: "DR-36"
07h	Manufacturer	BYTE	STRING	Number of the string that names the company that manufactured the supply
08h	Serial Number	BYTE	STRING	Number of the string that contains the serial number for the power supply
09h	Asset Tag Number	BYTE	STRING	Number of the string that contains the Asset Tag Number
0Ah	Model Part Number	BYTE	STRING	Number of the string that contains the OEM Part Order Number
0Bh	Revision Level	BYTE	STRING	Power supply Revision String EXAMPLE: "2.30"
0Ch	Max Power Capacity	WORD	Varies	Maximum sustained power output in Watts Set to 0x8000 if unknown. Note that the units specified by the DMTF for this field are milliWatts.
0Eh	Power Supply Characteristics	WORD	Varies	See 7.40.1.
10h	Input Voltage Probe Handle	WORD	Varies	Handle, or instance number, of a voltage probe (Type 26) monitoring this power supply's input voltage A value of 0xFFFF indicates that no probe is provided.
12h	Cooling Device Handle	WORD	Varies	Handle, or instance number, of a cooling device (Type 27) associated with this power supply A value of 0xFFFF indicates that no cooling device is provided.
14h	Input Current Probe Handle	WORD	Varies	Handle, or instance number, of the electrical current probe (Type 29) monitoring this power supply's input current A value of 0xFFFF indicates that no current probe is provided.

1655 **7.40.1 Power supply characteristics**

1656 Table 113 provides information about power supply characteristics.

1657 NOTE   Refer to 6.3 for the CIM properties associated with these enumerated values.

1658 **Table 113 – Power supply characteristics**

Bit Range	Meaning
15 to 14	Reserved; set to 00b
13 to 10	DMTF Power Supply Type 0001b Other 0010b Unknown 0011b Linear 0100b Switching 0101b Battery 0110b UPS 0111b Converter 1000b Regulator 1001b to 1111b — Reserved for future assignment
9 to 7	Status 001b Other 010b Unknown 011b OK 100b Non-critical 101b Critical; power supply has failed and has been taken off-line.
6 to 3	DMTF Input Voltage Range Switching 0001b Other 0010b Unknown 0011b Manual 0100b Auto-switch 0101b Wide range 0110b Not applicable 0111b to 1111b — Reserved for future assignment
2	1b power supply is unplugged from the wall
1	1b power supply is present
0	1b power supply is hot-replaceable

1659 **7.41 Additional Information (Type 40)**

1660 This structure is intended to provide additional information for handling unspecified enumerated values
1661 and interim field updates in another structure. Table 114 provides details.

1662 NOTE   This structure type was added in version 2.6 of this specification.

1663 **Table 114 – Additional Information (Type 40) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	40	Additional Information type
01h	Length	BYTE	Varies	Length of the structure, a minimum of 0Bh
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure
04h	Number of Additional Information entries (n)	BYTE	Varies	Number of Additional Information Entries that follow
05h	Additional Information entries	Varies	Varies	Additional Information entries; see7.41.1

1664 **7.41.1 Additional Information Entry format**

1665 Table 115 describes an Additional Information Entry format.

1666 **Table 115 – Additional Information Entry format**

Offset	Name	Length	Value	Description
00h	Entry Length	BYTE	Varies	Length of this Additional Information Entry instance; a minimum of 6
01h	Referenced Handle	WORD	Varies	Handle, or instance number, associated with the structure for which additional information is provided
03h	Referenced Offset	BYTE	Varies	Offset of the field within the structure referenced by the *Referenced Handle* for which additional information is provided
04h	String	BYTE	STRING	Number of the optional string to be associated with the field referenced by the *Referenced Offset*
05h	Value	Varies	Varies	Enumerated value or updated field content that has not yet been approved for publication in this specification and therefore could not be used in the field referenced by *Referenced Offset* NOTE: This field is the same type and size as the field being referenced by this Additional Information Entry.

1667 The following guidance is given for using this structure to provide additional information for an
1668 enumerated value field, such as processor type:

1669 • If a value has been proposed:

1670 – Set the field in the original structure to "Other".

1671 – Use the proposed value in the value field of the Additional Information Entry that
1672 references the enumerated field in the original structure.

1673 – The Additional Information Entry string field may also be used to uniquely describe this
1674 new item (for example the CPU ID string).

1675 • If a value has not been proposed:

1676 – The field in the original structure and the Additional Information Entry value field that
1677 references it should both be set to "Other".

1678 – The Additional Information Entry string field should be filled so as to uniquely describe this
1679 new item (for example the CPU ID string).

1680   The following guidance is given for using this structure to provide additional information for a field update:

1681   •   If a change has been proposed:

1682   –   Set the field in the original structure as best as possible using only fully approved settings.

1683   –   Place the modified value in the value field of the Additional Information Entry that
1684   references the field in the original structure.

1685   –   The Additional Information Entry string field may also be used to uniquely describe this
1686   modification.

1687   •   If a change has not been proposed:

1688   –   The field in the original structure and Additional Information Entry value field that
1689   references it should both be set to the same value (the best possible value using only fully
1690   approved settings).

1691   –   The Additional Information Entry string field should be filled so as to uniquely describe what
1692   needs to be modified (for example, "XYZ capability needs to be defined").

7.42 Onboard Devices Extended Information (Type 41)

1694   The information in this structure defines the attributes of devices that are onboard (soldered onto) a
1695   system element, usually the baseboard. Table 116 provides details.

1696   In general, an entry in this table implies that the BIOS has some level of control over the enablement of
1697   the associated device for use by the system.

1698   NOTE   This structure replaces Onboard Device Information (Type 10) starting with version 2.6 of this specification.
1699   BIOS providers can choose to implement both types to allow existing SMBIOS browsers to properly display the
1700   system's onboard devices information.

1701   **Table 116 – Onboard Devices Extended Information (Type 41) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	41	Onboard Devices Extended Information
01h	Length	BYTE	0Bh	Length of the structure
02h	Handle	WORD	Varies	
04h	Reference Designation	BYTE	String	String number of the onboard device reference designation See 7.42.1.
05h	Device Type	BYTE	ENUM	Bit 7 – Device Status: 1 – Device Enabled 0 – Device Disabled Bits 6:0 – Type of Device (see 7.42.2)
06h	Device Type Instance	BYTE	Varies	See 7.42.3
07h	Segment Group Number	WORD	Varies	See 7.42.4
09h	Bus Number	BYTE	Varies	See 7.42.4
0Ah	Device/Function Number	BYTE	Bit Field	Bits 7:3 – Device number Bits 2:0 – Function number See 7.42.4

1702   **7.42.1 Reference Designation**

1703   The *Reference Designation* string is typically the silkscreen label.

1704   **7.42.2 Onboard Device Types**

1705   Table 117 shows what the byte values mean for the Onboard Device Types field.

1706   **Table 117 – Onboard Device Types field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Video
04h	SCSI Controller
05h	Ethernet
06h	Token Ring
07h	Sound
08h	PATA Controller
09h	SATA Controller
0Ah	SAS Controller

1707   **7.42.3 Device Type Instance**

1708   *Device Type Instance* is a unique value (within a given *onboard device type*) used to indicate the order
1709   the device is designated by the system. For example, a system with two identical Ethernet NICs may
1710   designate one NIC (with higher Bus/Device/Function=15/0/0) as the first onboard NIC (instance 1) and
1711   the other NIC (with lower Bus/Device/Function =3/0/0) as the second onboard NIC (instance 2).

1712   **7.42.4 Segment Group Number, Bus Number, Device/Function Number**

1713   For devices that are not of types PCI, AGP, PCI-X, or PCI-Express and that do not have
1714   bus/device/function information, 0FFh should be populated in the fields of *Segment Group Number*, *Bus*
1715   *Number*, *Device/Function Number*.

1716   *Segment Group Number* is defined in the *PCI Firmware Specification*. The value is 0 for a single-segment
1717   topology.

1718   **7.43 Management Controller Host Interface (Type 42)**

1719   The information in this structure defines the attributes of a Management Controller Host Interface that is
1720   not discoverable by "Plug and Play" mechanisms. Table 118 provides details. The Type 42 structure can
1721   be used to describe a physical management controller host interface and one or more protocols that
1722   share that interface.

1723   Type 42 should be used for management controller host interfaces that use protocols other than IPMI or
1724   that use multiple protocols on a single host interface type.

1725   This structure should also be provided if IPMI is shared with other protocols over the same interface
1726   hardware. If IPMI is not shared with other protocols, either the Type 38 or the Type 42 structures can be
1727   used. Providing Type 38 is recommended for backward compatibility. The structures are not required to

1728 be mutually exclusive. Type 38 and Type 42 structures may be implemented simultaneously to provide
1729 backward compatibility with IPMI applications or drivers that do not yet recognize the Type 42 structure.
1730 Refer to the *Intelligent Platform Management Interface (IPMI) Interface Specification* for full
1731 documentation of IPMI and additional information on the use of this structure with IPMI.

1732   **Table 118 – Management Controller Host Interface (Type 42) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	42	Management Controller Host Interface structure indicator
01h	Length	BYTE	Varies	Length of the structure, a minimum of 09h
02h	Handle	WORD	Varies	
04h	Interface Type	BYTE	ENUM	Management Controller Interface Type Refer to *Management Component Transport Protocol (MCTP) IDs and Codes* (DSP0239) for the definition of the Interface Type values.
05h	MC Host Interface Data	n BYTEs	Varies	Management Controller Host Interface Data as specified by the Interface Type Refer to DSP0239 to locate the specification that corresponds to the Interface Type value. This field has a minimum of four bytes. If interface type = OEM, the first four bytes are the vendor ID (MSB first), as assigned by the Internet Assigned Numbers Authority (IANA). This format uses the "Enterprise Number" that is assigned and maintained by IANA (www.iana.org) as the means of identifying a particular vendor, company, or organization.

1733 After the generic structure header, this structure has the following general layout shown in Table 119.

1734   **Table 119 – Management Controller Host Interface (Type 42) structure general layout**

Name	Length	Value	Description
Interface Type	BYTE	ENUM	Host Interface Information
Interface Specific Data	Varies	Varies	
Number of Protocols	BYTE	n	n number of Protocols for this Host Interface
Protocol 1 Type	BYTE	ENUM	Protocol 1 Information
Protocol 1 Specific Data	Varies	Varies	
...	…	…	…
Protocol n Type	BYTE	ENUM	Protocol n Information
Protocol n Specific Data	Varies	Varies	

1735 7.44 Inactive (Type 126)

1736 This structure definition supports a system implementation where the SMBIOS structure-table is a
1737 superset of all supported system attributes and provides a standard mechanism for the system BIOS to
1738 signal that a structure is currently inactive and should not be interpreted by the upper-level software.
1739 Table 120 provides details.

1740 For example, a portable system might include *System Slot* structures that are reported only when the
1741 portable is docked. An undocked system would report those structures as *Inactive*. When the system is
1742 docked, the system-specific software would change the Type structure from *Inactive* to the *System Slot*
1743 equivalent.

1744 Upper-level software that interprets the SMBIOS structure-table should bypass an *Inactive* structure just
1745 as it would for a structure type that the software does not recognize.

1746 NOTE   This structure type was added in version 2.2 of this specification.

1747   **Table 120 – Inactive (Type 126) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	126	Inactive structure indicator
01h	Length	BYTE	Varies	Length of the structure
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure

1748 ## 7.45 End-of-Table (Type 127)

1749 This structure type identifies the end of the structure table that might be earlier than the last byte within
1750 the buffer specified by the structure. Table 121 provides details.

1751 To ensure backward compatibility with management software written to previous versions of this
1752 specification, a system implementation should use the end-of-table indicator in a manner similar to the
1753 *Inactive (Type 126)* structure type; the structure table is still reported as a fixed-length, and the entire
1754 length of the table is still indexable. If the end-of-table indicator is used in the last physical structure in a
1755 table, the field's length is encoded as 4.

1756 NOTE   This structure type was added in version 2.2 of this specification.

1757   **Table 121 – End-of-Table (Type 127) structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	127	End-of-table indicator
01h	Length	BYTE	Varies	Length of the structure
02h	Handle	WORD	Varies	Handle, or instance number, associated with the structure

1758

1759
ANNEX A
1760
(informative)
1761
1762
Conformance guidelines

1763 The following conformance requirements apply for SMBIOS 2.5 or later implementations.

1764 1. The table anchor string "_SM_" is present in the address range 0xF0000 to 0xFFFFF on a 16-byte boundary.

1765 2. Table entry-point verification:

1766   2.1 The Entry Point Length field value is at least 0x1F.

1767   2.2 The entry-point checksum evaluates to 0.

1768   2.3 The SMBIOS version (Major.Minor) is at least 2.4.

1769   2.4 The Intermediate Anchor String is "_DMI_"

1770   2.5 The intermediate checksum evaluates to 0.

1771 3. The structure-table is traversable and conforms to the entry-point specifications:

1772 1773   3.1 The structure-table's linked-list is traversable within the length and structure-count bounds specified by the entry-point structure.

1774 1775   3.2 The overall size of the structure table is less than or equal to the Structure Table Length specified by the entry-point structure.

1776   3.3 Each structure's length must be at least 4 (the size of a structure header).

1777   3.4 No structure handle number is repeated.

1778   3.5 The last structure is the end-of-table (0x7F).

1779 1780   3.6 The number of structures found within the table equals the Number of SMBIOS Structures field present in the entry-point.

1781 1782   3.7 The maximum structure size (formatted area plus its string-pool) is less than or equal to the Maximum Structure Size specified by the entry-point.

1783 4. Required structures and corresponding data are present (see 6.2):

1784   4.1 BIOS Information (Type 0)

1785   4.1.1 One and only one structure of this type is present.

1786   4.1.2 The structure Length field is at least 18h.

1787   4.1.3 BIOS Version string is present and non-null.

1788   4.1.4 BIOS Release Date string is present, non-null, and includes a 4-digit year.

1789   4.1.5 BIOS Characteristics: bits 3:0 are all 0, and at least one of bits 31:4 is set to 1.

1790   4.2 System Information (Type 1)

1791   4.2.1 One and only one structure of this type is present.

1792   4.2.2 The structure Length field is at least 1Bh.

1793   4.2.3 Manufacturer string is present and non-null.

1794   4.2.4 Product Name string is present and non-null.

1795   4.2.5 UUID field is neither 00000000 00000000 nor FFFFFFFF FFFFFFFF.

1796   4.2.6 Wake-up Type field is neither 00h (Reserved) nor 02h (Unknown).

1797   4.3 System Enclosure (Type 3)

1798	4.3.1	One or more structures of this type is present.
1799	4.3.2	The structure length is at least 0Dh.
1800	4.3.3	The Manufacturer string is present and non-null in each structure.
1801	4.3.4	Type field is neither 00h (Reserved) nor 02h (Unknown).

1802   4.4   Processor Information (Type 4)

1803 1804	4.4.1	The number of structures defines the maximum number of processors supported by the system; at least one structure with a Processor Type field of "Central Processor" must be present.
1805	4.4.2	Each structure's length is at least 20h.
1806	4.4.3	Socket Designation string is present and non-null.
1807	4.4.4	Processor Type field is neither 00h (Reserved) nor 02h (Unknown).
1808	4.4.5	(*)Processor Family field is neither 00h (Reserved) nor 02h (Unknown).
1809	4.4.6	(*)Processor Manufacturer string is present and non-null.
1810	4.4.7	Max Speed field is non-0.
1811	4.4.8	(*)CPU Status sub-field of the Status field is not 0 (Unknown).
1812	4.4.9	Processor Upgrade field is neither 00h (Reserved) nor 02h (Unknown).
1813 1814	4.4.10	Lx (x=1,2,3) Cache Handle fields, if not set to 0xFFFF, reference Cache Information (Type 7) structures.

1815 NOTE   Fields preceded by (*) are checked only if the CPU Socket Populated sub-field of the Status field is set to
1816 "CPU Populated".

1817   4.5   Cache Information (Type 7)

1818	4.5.1	One structure is present for each external-to-the-processor cache.
1819	4.5.2	Each structure's Length is at least 13h.
1820 1821	4.5.3	Socket Designation string is present and non-null if the cache is external to the processor (Location sub-field of Cache Configuration field is 01b).
1822 1823	4.5.4	Operational Mode and Location sub-fields of the Cache Configuration field are not 11b (Unknown).

1824   4.6   System Slots (Type 9)

1825	4.6.1	One structure is present for each upgradeable system slot.
1826	4.6.2	Each structure's Length is at least 0Dh.
1827	4.6.3	Slot Designation string is present and non-null.
1828	4.6.4	Slot Type is neither 00h (Reserved) nor 02h (Unknown).
1829	4.6.5	Slot Data Bus Width is neither 00h (Reserved) or 02h (Unknown).
1830 1831	4.6.6	Current Usage is not set to 00h (Reserved). If the "Slot Type" provides device presence-detect capabilities (for example, PCI or AGP), Current Usage is not set to 02h (Unknown).
1832	4.6.7	Slot ID is set to a meaningful value.
1833	4.6.8	Slot Characteristics 1, bit 0, is not set to 1.

1834   4.7   Physical Memory Array (Type 16)

1835	4.7.1	At least one structure is present with "Use" set to 03h (System memory).
1836	4.7.2	Each structure's length is at least 0Fh.
1837	4.7.3	Location is neither 00h (Reserved) nor 02h (Unknown).
1838	4.7.4	Use is neither 00h (Reserved) nor 02h (Unknown).
1839	4.7.5	Memory Error Correction is neither 00h (Reserved) nor 02h (Unknown).

1840 1841	4.7.6	Either Maximum Capacity or Extended Maximum Capacity must be set to a known, non-zero value.
1842 1843	4.7.7	Number of Memory Devices is not 0 and equals the number of Memory Device (Type 17) structures that reference the handle of the Physical Memory Array structure.
1844	4.8	Memory Device (Type 17)
1845 1846 1847 1848 1849	4.8.1	For each Physical Memory Array, there must be "Number of Memory Devices" Memory Device structures that map back (through the Handle) to the referencing memory array. One structure is required for each socketed system-memory device, whether or not the socket is currently populated. If the system includes soldered-on system memory, one additional structure is required to identify that memory device.
1850	4.8.2	Each structure's length is at least 15h.
1851	4.8.3	Memory Array Handle references a Physical Memory Array (Type 16) structure.
1852	4.8.4	Total Width is not 0FFFFh (Unknown) if the memory device is installed. (Size is not 0.)
1853	4.8.5	Data Width is not 0FFFFh (Unknown).
1854	4.8.6	Size is not 0FFFFh (Unknown).
1855	4.8.7	Form Factor is not 00h (Reserved) or 02h (Unknown).
1856	4.8.8	Device Set is not 0FFh (Unknown).
1857	4.8.9	Device Locator string is present and non-null.
1858	4.9	Memory Array Mapped Address (Type 19)
1859 1860	4.9.1	One structure is provided for each contiguous block of memory addresses mapped to a Physical Memory Array.
1861	4.9.2	Each structure's length is at least 0Fh.
1862 1863	4.9.3	Ending Address value is higher in magnitude than the Starting Address value, or Extended Ending Address value is higher in magnitude than the Extended Starting Address value.
1864	4.9.4	Memory Array Handle references a Physical Memory Array (Type 16).
1865 1866	4.9.5	Each structure's address range (Starting Address to Ending Address or Extended Starting Address to Extended Ending Address) is unique and non-overlapping.
1867	4.9.6	Partition Width is not 0.
1868 1869	4.10	Boot Integrity Services (BIS) Entry Point (Type 31). This structure is optional, but if it is present the following checks are performed:
1870	4.10.1	The structure's length is at least 1Ch.
1871	4.10.2	The structure-level checksum evaluates to 00h.
1872	4.10.3	16-bit Entry Point is not 0.
1873	4.10.4	32-bit Entry Point is not 0.
1874	4.11	System Boot Information (Type 32)
1875	4.11.1	One and only one structure of this type is present.
1876	4.11.2	The structure's length is at least 0Bh.
1877		

1878 # ANNEX B
1879 # (informative)
1880
1881 # Using the table convention

1882 This clause contains pseudo-code that describes the method that application software can use to parse
1883 the table-based SMBIOS structures. The example searches for the first structure of the type specified,
1884 returning the handle of the structure found or 0xFFFF if no structure of the type was found in the list.
1885 *TableAddress* and *StructureCount* values are those previously found by locating the Table Entry Point
1886 structure in low memory.

```
1887  typedef unsigned short ushort;
1888  typedef unsigned char uchar;
1889  typedef struct
1890   {
1891    uchar Type;
1892    uchar Length;
1893    ushort Handle;
1894   } HEADER;
1895  ushort FindStructure( char *TableAddress, ushort StructureCount, uchar Type )
1896   {
1897    ushort i, handle;
1898    uchar lasttype;
1899    i   = 0;
1900    handle = 0xFFFF;
1901    while( i < StructureCount && handle == 0xFFFF )
1902     {
1903      i++;
1904      lasttype = ((HEADER *)TableAddress)->Type;
1905      if( lasttype == Type )
1906       {
1907        handle = ((HEADER *)TableAddress)->Handle;
1908       } /* Found first structure of the requested type */
1909      else
1910       {
1911        TableAddress += ((HEADER *)TableAddress)->Length;
1912        while( *((int *)TableAddress) != 0 )
1913         {
1914          TableAddress++;
1915         } /* Get past trailing string-list */
1916        TableAddress += 2;
1917       } /* Increment address to start of next structure */
1918     } /* END while-loop looking for structure type */
1919    return handle;
1920   } /* END FindStructure */
```

1921

1922
1923
1924
1925

ANNEX C
(informative)

Change log

Version	Release Date	Description
2.0D	1995-09-14	Initial Release of DRAFT COPY
2.0M	1995-12-12	Final draft released, with the following changes: • Specified that dmiStorageBase (Function 50h) and NVStorageBase (Function 55h) must be paragraph-aligned. • Added Command value to change a string to function 52h; Command enumeration values modified. • Removed redundant enumerations from Processor Family list • Corrected Memory Subsystem Example • Corrected/clarified Indexed I/O access-methods for event-log; Access Method enumeration values and Access Method Address union modified • Added clarifications to some of the event log types
2.00	1996-03-06	Final release, with the following changes: • Specified that all structures end with a terminating NULL, even if the formatted portion of the structure contains string-reference fields and all the string fields are set to 0. • Corrected the Memory Subsystem Example, handles are now correctly created with a 'dw'. • Fixed formatting of some bit definition fields and function examples.
2.00.1	1996-07-18	Minor updates for new technology and clarifications. • Added definitions for Pentium® Pro, Burst EDO, and SDRAM. • Added clarifications to the Memory Controller Error Status.
2.1.0	1997-06-16	Added definition for static table interface, to allow the information to be accessed from new operating systems (see 5.2). In addition: • Changed references to DMI BIOS to SMBIOS throughout; these changes are unmarked. • Added SubFunction DMI_CLEAR_EVENT_LOG2 to Function 54h - SMBIOS Control. • For those structure entries that are string numbers, changed the Value field definition of the field from Varies to STRING throughout; these changes are unmarked. • BIOS Information structure: Added support for 4-digit year and additional BIOS Characteristics through Characteristics Extension Byte 1. • System Information structure: Added Wakeup Type and UUID fields. • System Enclosure and Chassis structure: Added Bootup State, Power Supply State, Thermal State, and Security Status to allow the DMTF|Physical Container Global Table to be populated. • Processor Information structure: Voltage value can now be specified, rather than using bit-flags, and added enumeration values for Pentium® Pro, Pentium® II, and Slot 1. Also added notes to this section, indicating that the enumerated values for the structure are assigned by the DMTF. This structure was also updated to include the Cache Information handles identifying the L1, L2, and L3 caches associated with the processor.

Version	Release Date	Description
		• Memory Controller Information structure: Added Enabled Error Correcting field. Also added note that this structure can never be updated to add string values, to preserve backwards compatibility. • Cache Information structure: Added Speed, Error Correction Type, Type, and Associativity fields. • Port Connector Information structure: Added enumerated values to Connector Types and Port Types. • System Slots structure: Added AGP enumeration values to Slot Type field. • BIOS Language Information structure: Added abbreviated-format for language strings and corrected example. • System Event Log structure: OEM-specific Access Methods can now be defined, added standard log header definitions, and a mechanism to allow the log entry's variable data formats to be described. Added note that this structure can never be updated to include string values, to preserve backwards compatibility. • Added Physical Memory Array, Memory Device, Memory Error Information, Memory Array Mapped Address, and Memory Device Mapped Address structures to support the population of the DMTF Enhanced Physical Memory groups. • Added Built-in Pointing Device structure to support the population of the DMTF Pointing Device group. • Added Portable Battery structure to support the population of the DMTF Portable Battery group. • Added appendices that contain a structure checklist and table-convention parsing pseudo-code.

2.2.0	1998-03-16	The following changes were made to version 2.1 of the document to produce this version:
		
		• Accepted all changes introduced at Version 2.1
		
		• Added ACPI statement-of-direction for dynamic state and event notification
		
		• Table-convention is required for version 2.2 and later compliance
		
		• Corrected Structure Table entry point length value.
		
		• Added Command type 06h to the Plug-and-Play Set SMBIOS Structure function (52h).
		
		• Added new processor enumerations from the updated DMTF MASTER.MIF
		
		• System Enclosure: Added enumeration value for "Sealed-case PC", to support Net PC-type chassis'.
		
		• Memory Controller Information: Corrected description of how the BIOS computes the structure Length.
		
		• System Event Log:
		  – Added definition for end-of-log data, Event Log Type 0FFh.
		  – Added generic system-management event type; the handle of an associated probe or cooling device identifies the specific failing device.
		
		• Memory Error Information: Corrected structure size and offsets.
		
		• Portable Battery: Corrected the structure length and some of the offsets, added Smart Battery-formatted fields
		
		• Memory Device: Added RIMM form factor
		
		• Added the following new structures
		  – System Reset structure to support the population of the DMTF Automatic System Reset group.
		  – Hardware Security structure to support the population of the DMTF System

Version	Release Date	Description
		Hardware Security group.
		– System Power Control structure to support the population of the DMTF System Power Control group.
		– Added Voltage Probe structure to support the population of the DMTF Voltage Probe group.
		– Cooling Device structure to support the population of the DMTF Cooling Device group.
		– Temperature Probe structure to support the population of the DMTF Temperature Probe group.
		– Electrical Current Probe structure to support the population of the DMTF Electrical Current Probe group.
		– Out-of-Band Remote Access structure to support the population of the DMTF Out-of-Band Remote Access group.
		– Inactive structure type to support standard structure superset definitions.
		– End-of-Table structure type to facilitate easier traversing of the structure data.
2.3.0	1998-08-12	The following changes were made to version 2.2 of the document to produce this version:
		• Accepted all changes introduced at Version 2.2
		• Clarified and corrected referenced documents
		• A minimum set of structures (and their data) is now required for SMBIOS compliance.
		• Documented an additional structure usage guideline, to optional structure growth.
		• BIOS Information:
		– 4-digit year format for BIOS Release Date required for SMBIOS 2.3 and later
		– Added BIOS Characteristic Extension Byte 2 to include status that the BIOS supports the BIOS Boot Specification.
		• System Information:
		– Added enumeration for Wake-up Type
		• System Enclosure or Chassis:
		– Added OEM-defined field.
		• Processor Information:
		– Added enumerated values for new processors from the updated MASTER.MIF and identified that one structure is present for each processor instance.
		– Modified interpretation of Lx Cache Handle fields for version 2.3 and later implementations
		• Memory Module Information:
		– Corrected example, adding double-null to terminate the structure.
		• System Slots:
		– Added hot-plug characteristic definition and clarified usage of the PCI "Slot ID" field.
		• Memory Device:
		– Added enumerations for Form Factor and Device Type
		– Added new field for memory Speed
		• System Event Log:
		– Added note describing how century portion of the 2-digit year within a log record is to be interpreted.

Version	Release Date	Description
		• Voltage Probe, Temperature Probe, Electrical Current Probe, Cooling Device: – Added Nominal Value field – Added the following new structures • Boot Integrity Services (BIS) Entry Point • System Boot Information • 64-bit Memory Error Information • Management Device • Management Device Component • Management Device Threshold Data
2.3.1	1999-03-16	The following changes were made to version 2.3 of the document to produce this version: • Accepted all changes introduced at Version 2.3 • Adopted a three-tier document numbering procedure, see *Document Version Number Conventions* for more information. • BIOS Information: – Added BIOS Characteristic Extension Byte 2, bit 1, to identify that the BIOS supports F12=Network Boot functionality • Processor Information: – Added Processor Family enumeration for new Pentium processors, defined reserved values for future Pentium processors. – Added fields: Asset Tag, Serial Number, and Part Number. • System Slots: – Added slot type enumeration for PCI-X – Added slot characteristic to identify support for (to-be) standard SMBus interface for PCI slots • Memory Device: – Added enumerated values for Memory Type and Form Factor, required for RamBus implementations – Added fields: Manufacturer, Asset Tag, Serial Number, and Part Number. • Added the following new structures: – Memory Channel (to support RamBus and SyncLink memory implementations) – IPMI Device, to abstract the IPMI hardware dependencies to management software – System Power Supply
2.3.1	2000-12-14	Released as DMTF Preliminary Specification DSP0119.
2.3.2	2001-10-12	The following changes were made to version 2.3.1 of the document to produce this version: • Accepted all changes introduced at version 2.3.1 • Released as DMTF Specification DSP0130 (Preliminary) • Updated the Abstract and Overview sections to be more DMTF-general than DMI-specific. Change bars are present in the Overview section only. • Deleted section 1.1 (future direction for ACPI interface specification). Any ACPI interface to provide these structures should be provided by a future version of the ACPI specification itself. • Removed "References" that had broken links. • Modified sections 2 and 2.2 to indicate that the PnP calling interface is being

Version	Release Date	Description
		deprecated at this specification version.
		• Noted in section 2.1 that the structure table data is boot-time static.
		• For each enumerated list that indicated that the enumeration is controlled by the "DMTF, not this specification", identified which CIM class.property and DMI group.attribute are mapped to the enumerated value. Also added a note in the Overview section to indicate where change requests should be sent.
		• Baseboard Information (Type 2)
		– Added fields: Asset Tag, Feature Flags, Location in Chassis, Chassis Handle, Baseboard Type, and Contained Objects to support multi-system chassis like server blades.
		• System Enclosure or Chassis (Type 3)
		– Added fields: Height, Number of Power Cords, Contained Element Count, and Contained Elements to support multi-system chassis like server blades.
		• Processor Information (Type 4)
		– Added new enumerations to Processor Family and Processor Upgrade
		– Removed (SMBIOS-only) reserved ranges. These ranges are controlled by the DMTF, not the SMBIOS group. The DMTF Device MOF (starting with version 2.3) has commentary around the Processor Family enumeration that suggests that enumerations below 256 be used only for those processor types that are going to be reported by SMBIOS (because this specification's Processor Family field is a 1-byte entity).
		• Cache (Type 7)
		– Added new enumerations to Associativity
		• Memory Device (Type 17)
		– Added new enumerations to Memory Type
		• Built-in Pointing Device (Type 21)
		– Added new enumerations to Pointing Device Type
		– Removed out-of-date section Correlation to DMTF Groups, in favor of updated section 3.3.
2.3.3	2002-05-10	The following changes were made to version 2.3.2 of the document to produce this version: • Accepted all changes introduced at version 2.3.2 • Updated the Abstract to contain the updated DMTF copyright statement. • Processor Information (Type 4) – Added new enumerations to Processor Family and Processor Upgrade

2.3.4	2002-12-06	The following changes were made to version 2.3.3 of the document to produce this version:
		• System Enclosure Information (Type 3)
		   – Provided clarification regarding contained element types
		• Processor Information (Type 4)
		   – Added and corrected enumerations to Processor Family (CR00002)
		   – Provided clarification for Max Speed and Current Speed.
		• Additions to Processor Upgrade (CR00002)
		• System Slots (Type 9)
		   – Added AGP8X enumeration to Slot Type

2.4.0	2004-07-21	The following changes were made to version 2.3.4 of the document to produce this version:

Version	Release Date	Description
		• Processor Information (Type 4)
		– Added new enumerations to Processor Family (CR00951, CR01152)
		• System Slots (Type 9)
		– Added PCI Express enumeration to Slot Type (CR01259)
		– Added new enumerations to Slot Data Bus Width (CR01324)
		• Memory Device (Type 17)
		– Added DDR2 enumeration to Type (CR01263)
		• BIOS Information (Type 0)
		– Added fields: System BIOS Major Release, System BIOS Minor Release, Embedded Controller Firmware Major Release, and Embedded Controller Firmware Minor Release (CR01270)
		– Added BIOS Characteristic Extension Byte 2, bit 2, to identify that the BIOS supports Targeted Content Distribution (CR01270)
		• System Information (Type 1)
		– Added fields: SKU Number and Family (CR01270)
		– Updated Conformance Guidelines and added corrections
2.5.0	2006-09-05	The following changes were made to version 2.4 of the document to produce this version:
		• Shortened abstract
		• Removed obsolete references to DMI, which is no longer maintained by the DMTF. Added references to the Pre-OS and CIM Core Working Groups. (PreOSCR00017.001)
		• References:
		– Updated specification revisions and URLs (PreOSCR00019.001)
		• Table Convention:
		– Added EFI-specific information (PreOSCR00011.005)
		• SMBIOS Structure Table Entry Point:
		– Corrected typo, the SMBIOS BCD Revision is at offset 1Eh, not 1Dh (PreOSCR00020.000)
		• Required Structures and Data:
		– Added DIG64 information (PreOSCR00013.000)
		• System Enclosure or Chassis (Type 3)
		– Added new types for CompactPCI and AdvancedTCA (PreOSCR00012.001)
		• Processor Information (Type 4)
		– Added AMD Sempron to Processor Family (DMTFCR01473)
		– Added AMD Turion to Processor Family (SysdevCR00708)
		– Added multi-core, multi-thread and 64-bit extension processor characteristics (PreOSCR00002)
		– Added new processor values (Celeron D, Pentium D, Pentium Extreme Edition) (PreOSCR00005)
		– Added new processor upgrade (socket 939) (DMI CR00005)
		– Added AMD dual-core Opteron and Athlon 64 X2 (PreOSCR00015.003)
		– Added new Processor Upgrade values (PreOSCR00016.001)
		• Cache Information (Type 7)
		– Added note on cache size for multi-core processors (PreOSCR00002)

Version	Release Date	Description
		• Port connector Information (Type 8)
		– Added SATA and SAS (PreOSCR00021.002)
		• System Slots (Type 9)
		– Updated Slot ID description with ACPI and PCI Express (PreOSCR00018.000)
		• Onboard Devices Information (Type 10)
		– Added SATA and SAS (PreOSCR00021.002)
		• Memory Device (Type 17)
		– Added values for FB-DIMM (PreOSCR00010.004)
		• Memory Device Mapped Address (Type 20)
		– Moved structure from 'required' to 'optional' (PreOSCR00009.002)
		– Moved 'Plug-and-Play Calling Convention' to Appendix C (PreOSCR00022.001)
2.6.0	2008-06-30	The following changes were made to version 2.5 of the document to produce this version:
		• References: added PCI Firmware Specification (SMBIOSCR00042)
		• System Information (Type 1): clarification of UUID format (SMBIOSCR00037, SMBIOSCR00061)
		• System Enclosure or Chassis (Type 3): added new values to System Enclosure or Chassis Types (Blade, Blade Enclosure) (SMBIOSCR00034)
		• Processor Information (Type 4):
		– Added Processor Family 2 field (SMBIOSCR00043)
		– Added new values to Processor Information – Processor Family (PreOSCR00025, SMBIOSCR00035, SMBIOSCR00040, SMBIOSCR00041, SMBIOSCR00044)
		– Added footnote to Processor Information – Processor Family (SMBIOSCR00039)
		– Added new values to Processor Information – Processor Upgrade (PreOSCR00028, SMBIOSCR00029)
		– Corrected values for BDh and BFh in Processor Information – Processor Family (SMBIOSCR00057)
		– Added "decimal values" column in Processor Information – Processor Family to simplify cross-referencing with CIM_Processor.mof data
		– Corrected typos for "AMD29000" (was "AMD2900") and "UltraSPARC IIi" (was "UltraSPARC Iii") (SMBIOSCR00054)
		• System Slots (Type 9):
		– Added new fields for Segment Group Number, Bus Number, Device/Function Number (SMBIOSCR00042)
		– Added new values to System Slots – Slot Type for PCI Express (SMBIOSCR00038)
		• On Board Devices Information (Type 10): marked structure type as Obsolete, replaced with type 41 (SMBIOSCR00042)
		• Memory Device (Type 17): added new field for rank information (PreOSCR00023)
		• Additional Information (Type 40): new structure type to handle unknown enumerations and other interim field updates (SMBIOSCR00031)
		• Onboard Devices Extended Information (Type 41): new structure type to replace type 10 (SMBIOSCR00042)
2.6.1	2009-03-17	The following changes were made to version 2.6 of the document to produce this version:

Version	Release Date	Description
		• System Information (Type 1): – Fixed typo in section 3.3.2 (Type 1 structure): at offset 18h (Wake-up type), the cross-reference should be to 3.3.2.2, not 3.3.2.1. • Processor Information (Type 4): – SMBIOSCR00046: Add new Processor Family values: AMD Quad Core and Third Generation Opteron Processors – SMBIOSCR00047: Add new Processor Family values: AMD Phenom and Athlon Processors – SMBIOSCR00049: Add new Processor Family value: Embedded AMD Opteron Processor – SMBIOSCR00051: Add new processor family value: AMD Phenom Triple-Core Processor Family – SMBIOSCR00055: Add new processor values for Intel processors – SMBIOSCR00058: Add new processor family values for AMD processors – SMBIOSCR00059: Add value for Intel(R) Atom(TM) processors – SMBIOSCR00060: Add number for "Quad-Core Intel(R) Xeon(R) processor 5400 Series" and a general number for "Quad-Core Intel(R) Xeon(R) processor" – SMBIOSCR00065: Add LGA1366 to Processor Upgrade enum – SMBIOSCR00068: Add numbers for new Intel processors • Cache Information (Type 7): – SMBIOSCR00062: Add values to cache associativity enum to cover new processors • System Slots (Type 9): – SMBIOSCR00064: Add PCIe Gen 2 slot types to Type 9 • Memory Device (Type 17): – SMBIOSCR00052: Add new memory device types: DDR3 and FBD2
2.7.0	2010-07-21	The following changes were made to version 2.6.1 of the document to produce this version: • Document layout: – SMBIOSCR00073: Move SMBIOS structure definitions to a new top-level section – SMBIOSCR00074: Remove Appendix C, "Plug-and-Play Calling Convention" • Various sections: – SMBIOSCR00096: Miscellaneous clerical changes • Section 1.1, Document Version Number Conventions: – SMBIOSCR00085: Add more description to the document version number convention • Section 3.1.2, Structure Header Format: – SMBIOSCR00048: Reserve handle number for consistency with UEFI PI specification • Section 3.1.3, Text Strings: – SMBIOSCR00086: Remove maximum string size limitation • Section 3.2, Required Structures and Data: – SMBIOSCR00095: Increase the capacity to represent system memory of 4 terabytes or greater.

Version	Release Date	Description
		• Bios Information (Type 0):
		– SMBIOSCR00056: Add UEFI support to BIOS characteristics
		– SMBIOSCR00071: Add support to describe virtualized platforms (bit 4)
		• System Enclosure or Chassis (Type 3):
		– SMBIOSCR00076: Add SKU Number field to type 3 structure (System Enclosure or Chassis)
		– SMBIOSCR00096: Fix offset for SKU Number entry (to 15h+n*m instead of 16h+n*m)
		• Processor Information (Type 4):
		– SMBIOSCR00063: Add new processor characteristics to Type 4
		– SMBIOSCR00070: Add new processor family values for AMD processors
		– SMBIOSCR00072: Add new processor family values for AMD processors
		– SMBIOSCR00077: Add new processor family values for VIA processors
		– SMBIOSCR00080: Add numbers for new Intel processors
		– SMBIOSCR00082: Add number for new AMD processor family
		– SMBIOSCR00083: Add new processor upgrade type (Socket G34)
		– SMBIOSCR00087: Add new processor upgrade type (Socket AM3)
		– SMBIOSCR00088: Add number for new Intel processor family: "Intel(R) Core(TM) i3 processor"
		– SMBIOSCR00090: Add number for new AMD processor family
		– SMBIOSCR00091: Add new processor upgrade type (Socket C32)
		– SMBIOSCR00092: Add new processor upgrade type (Socket LGA1156, Socket LGA1567)
		– SMBIOSCR00093: Add new processor upgrade type (Socket PGA988A, Socket BGA1288)
		– SMBIOSCR00094: Add footnote in processor family table for types 24-29
		– SMBIOSCR00097: Update processor trademarks for Intel processors
		• Physical Memory Array (Type 16):
		– SMBIOSCR00095: Increase the capacity to represent system memory of 4 terabytes or greater.
		• Memory Device (Type 17):
		– SMBIOSCR00050: Add support for memory >= 32GB in type 17
		– SMBIOSCR00053: Add memory type details of Registered and Unbuffered
		– SMBIOSCR00081: Add configured memory clock speed
		• Memory Array Mapped Address (Type 19) and Memory Device Mapped Address (Type 20):
		– SMBIOSCR00095: Increase the capacity to represent system memory of 4 terabytes or greater.
		• Cooling Device (Type 27):
		– SMBIOSCR00075: Add new "description" field in structure type 27
		• IPMI Device Information (Type 38):
		– SMBIOSCR00078: Update Type 38 to match IPMI specification
		– SMBIOSCR00079: Add Type 42 Management Controller Host Interface
		– SMBIOSCR00096: Replace "record" with "structure"
		• Management Controller Host Interface (Type 42):

Version	Release Date	Description
		– SMBIOSCR00079: Add Type 42 Management Controller Host Interface
		– SMBIOSCR00096: Replace "record" with "structure"
		• Appendix A, Conformance Guidelines:
		– SMBIOSCR00095: Increase the capacity to represent system memory of 4 terabytes or greater.
2.7.1	2011-01-26	The following changes were made to version 2.7 of the document to produce this version:
		• Processor Information (Type 4):
		– SMBIOSCR00099: new processor upgrade types
		– SMBIOSCR00100: new processor family types
		– SMBIOSCR00101: new processor family type
		– SMBIOSCR00103: new processor upgrade types
		• Cache Information (Type 7):
		– SMBIOSCR00102: new cache associativity value
		• Port Connector Information (Type 8):
		– SMBIOSCR00104: fix typo in Port Types (table 41)
		• System Slots (Type 9):
		– SMBIOSCR00105: add PCIe Gen 3 slot types
2.8.0	2012-12-14	The following changes were made to version 2.7 of the document to produce this version:
		• Processor Information (Type 4):
		– SMBIOSCR00106: processor family name correction (48h)
		– SMBIOSCR00107: new processor family types
		– SMBIOSCR00108: new processor family type
		– SMBIOSCR00110: correct typo in table 24 (processor upgrade)
		– SMBIOSCR00118: new processor family types
		– SMBIOSCR00121: new processor family type
		– SMBIOSCR00122: new processor upgrade type
		– SMBIOSCR00125: add new Intel socket type
		• Memory Device (Type 17):
		– SMBIOSCR00109: add minimum, maximum and configured voltages
		– SMBIOSCR00114: add LRDIMM to memory device list
		• Other:
		– SMBIOSCR00116: correct/clarify structure length fields
		– SMBIOSCR00120: add new supported processor architectures
		– SMBIOSCR00123: update referenced specifications
		– Wording updates for clarity and consistency
3.0.0	2015-02-12	The following changes were made to version 2.8.0 of the document to produce this version:
		• Accessing SMBIOS Information:
		– SMBIOSCR00115: add new entry point
		– SMBIOSCR00139: add GUID values for discovering SMBIOS tables in UEFI

Version	Release Date	Description
		• System Enclosure or Chassis (Type 3):
		– SMBIOSCR00130: add new chassis types: Tablet, Convertible, and Detachable
		Processor Information (Type 4):
		– SMBIOSCR00124: extend core, core enabled and thread count ranges
		– SMBIOSCR00126: add new socket type Intel LGA1150
		– SMBIOSCR00127: add new socket type Intel BGA1168
		– SMBIOSCR00128: add processor family names
		– SMBIOSCR00136: add new Intel socket types BGA1234 and BGA1364
		– SMBIOSCR00137: add new Intel processor family type– SMBIOSCR00138: update SMBIOSCR00124. Extend core, core enabled, and thread count ranges.
		• Cache Information (Type 7):
		– SMBIOSCR00134: add additional description for Unified cache type
		• System Slots (Type 9):
		– SMBIOSCR00132: add M.2 family of form factors
		– SMBIOSCR00133: add MXM family of slots
		– SMBIOSCR00135: add SFF-8639 slot types
		• Memory Device (Type 17):
		– SMBIOSCR00129: add new memory device type: DDR4
		– SMBIOSCR00131: add new memory device types: LPDDR, LPDDR2, LPDDR3, LPDDR4

1926

Bibliography
1927

1928 DMTF DSP4014, *DMTF Process for Working Bodies 1.1.0,*
1929 http://schemas.dmtf.org/process/DSP4014_1.1.0

1930