



1
2
3
4

Document Number: DSP0134

Date: 2010-07-21

Version: 2.7.0

5 **System Management BIOS (SMBIOS) Reference**
6 **Specification**

7 **Document Type: Specification**

8 **Document Status: DMTF Standard**

9 **Document Language: en-US**

10

11 Copyright Notice

12 Copyright © 2000, 2002, 2004–2010 Distributed Management Task Force, Inc. (DMTF). All rights
13 reserved.

14 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
15 management and interoperability. Members and non-members may reproduce DMTF specifications and
16 documents, provided that correct attribution is given. As DMTF specifications may be revised from time to
17 time, the particular version and release date should always be noted.

18 Implementation of certain elements of this standard or proposed standard may be subject to third party
19 patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations
20 to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose,
21 or identify any or all such third party patent right, owners or claimants, nor for any incomplete or
22 inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to
23 any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize,
24 disclose, or identify any such third party patent rights, or for such party's reliance on the standard or
25 incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any
26 party implementing such standard, whether such implementation is foreseeable or not, nor to any patent
27 owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is
28 withdrawn or modified after publication, and shall be indemnified and held harmless by any party
29 implementing the standard from any and all claims of infringement by a patent owner for such
30 implementations.

31 For information about patents held by third-parties which have notified the DMTF that, in their opinion,
32 such patent may relate to or impact implementations of DMTF standards, visit
33 <http://www.dmtf.org/about/policies/disclosures.php>.

CONTENTS

35	Foreword	9
36	Introduction	10
37	Document Conventions	10
38	Typographical Conventions	10
39	Document Version Number Conventions	10
40	1 Scope	13
41	2 Normative references	13
42	3 Terms and definitions	14
43	4 Symbols and abbreviated terms	15
44	5 Accessing SMBIOS Information	21
45	5.1 General	21
46	5.2 Table Convention	21
47	5.2.1 SMBIOS Structure Table Entry Point	21
48	6 SMBIOS Structures	22
49	6.1 Structure Standards	23
50	6.1.1 Structure Evolution and Usage Guidelines	23
51	6.1.2 Structure Header Format	24
52	6.1.3 Text Strings	24
53	6.2 Required Structures and Data	25
54	6.3 SMBIOS Fields and CIM MOF Properties	26
55	7 Structure Definitions	27
56	7.1 BIOS Information (Type 0)	27
57	7.1.1 BIOS Characteristics	29
58	7.1.2 BIOS Characteristics Extension Bytes	30
59	7.2 System Information (Type 1)	31
60	7.2.1 System — UUID	32
61	7.2.2 System — Wake-up Type	32
62	7.3 Baseboard (or Module) Information (Type 2)	33
63	7.3.1 Baseboard — Feature Flags	34
64	7.3.2 Baseboard — Board Type	34
65	7.4 System Enclosure or Chassis (Type 3)	35
66	7.4.1 System Enclosure or Chassis Types	36
67	7.4.2 System Enclosure or Chassis States	38
68	7.4.3 System Enclosure or Chassis Security Status	38
69	7.4.4 System Enclosure or Chassis — Contained Elements	38
70	7.5 Processor Information (Type 4)	39
71	7.5.1 Processor Information — Processor Type	42
72	7.5.2 Processor Information — Processor Family	42
73	7.5.3 Processor ID Field Format	48
74	7.5.4 Processor Information — Voltage	48
75	7.5.5 Processor Information — Processor Upgrade	49
76	7.5.6 Processor Information — Core Count	50
77	7.5.7 Processor Information — Core Enabled	50
78	7.5.8 Processor Information — Thread Count	50
79	7.5.9 Processor Characteristics	50
80	7.6 Memory Controller Information (Type 5, Obsolete)	51
81	7.6.1 Memory Controller Error Detecting Method	52
82	7.6.2 Memory Controller Error Correcting Capability	52
83	7.6.3 Memory Controller Information — Interleave Support	53
84	7.6.4 Memory Controller Information — Memory Speeds	53
85	7.7 Memory Module Information (Type 6, Obsolete)	53

86	7.7.1	Memory Module Information — Memory Types	54
87	7.7.2	Memory Module Information — Memory Size	55
88	7.7.3	Memory Subsystem Example	55
89	7.8	Cache Information (Type 7)	57
90	7.8.1	Cache Information — Maximum Cache Size and Installed Size	59
91	7.8.2	Cache Information — SRAM Type	59
92	7.8.3	Cache Information — Error Correction Type	59
93	7.8.4	Cache Information — System Cache Type	59
94	7.8.5	Cache Information — Associativity	60
95	7.9	Port Connector Information (Type 8)	60
96	7.9.1	Port Information Example	61
97	7.9.2	Port Information — Connector Types	61
98	7.9.3	Port Types.....	62
99	7.10	System Slots (Type 9).....	63
100	7.10.1	System Slots — Slot Type	64
101	7.10.2	System Slots — Slot Data Bus Width	65
102	7.10.3	System Slots — Current Usage.....	66
103	7.10.4	System Slots — Slot Length	66
104	7.10.5	System Slots — Slot ID	66
105	7.10.6	Slot Characteristics 1	67
106	7.10.7	Slot Characteristics 2.....	67
107	7.10.8	Segment Group Number, Bus Number, Device/Function Number.....	68
108	7.11	On Board Devices Information (Type 10, Obsolete).....	68
109	7.11.1	Onboard Device Types	68
110	7.12	OEM Strings (Type 11).....	69
111	7.13	System Configuration Options (Type 12).....	69
112	7.14	BIOS Language Information (Type 13).....	69
113	7.15	Group Associations (Type 14)	71
114	7.16	System Event Log (Type 15)	72
115	7.16.1	Supported Event Log Type Descriptors.....	74
116	7.16.2	Indexed I/O Access Method.....	75
117	7.16.3	Access Method Address — DWORD Layout	75
118	7.16.4	Event Log Organization	76
119	7.16.5	Log Header Format.....	76
120	7.16.6	Log Record Format.....	78
121	7.17	Physical Memory Array (Type 16)	82
122	7.17.1	Memory Array — Location	83
123	7.17.2	Memory Array — Use	84
124	7.17.3	Memory Array — Error Correction Types	84
125	7.18	Memory Device (Type 17)	84
126	7.18.1	Memory Device — Form Factor.....	86
127	7.18.2	Memory Device — Type	87
128	7.18.3	Memory Device — Type Detail	88
129	7.18.4	Memory Device — Extended Size	88
130	7.19	32-Bit Memory Error Information (Type 18).....	89
131	7.19.1	Memory Error — Error Type	89
132	7.19.2	Memory Error — Error Granularity.....	90
133	7.19.3	Memory Error — Error Operation	90
134	7.20	Memory Array Mapped Address (Type 19).....	90
135	7.21	Memory Device Mapped Address (Type 20)	91
136	7.22	Built-in Pointing Device (Type 21)	93
137	7.22.1	Pointing Device — Type	94
138	7.22.2	Pointing Device — Interface	94
139	7.23	Portable Battery (Type 22).....	94
140	7.23.1	Portable Battery — Device Chemistry	96
141	7.24	System Reset (Type 23)	97

142 7.25 Hardware Security (Type 24)..... 98
 143 7.26 System Power Controls (Type 25)..... 98
 144 7.26.1 System Power Controls — Calculating the Next Scheduled Power-on Time 99
 145 7.27 Voltage Probe (Type 26)..... 99
 146 7.27.1 Voltage Probe — Location and Status 100
 147 7.28 Cooling Device (Type 27) 101
 148 7.28.1 Cooling Device — Device Type and Status..... 102
 149 7.29 Temperature Probe (Type 28) 102
 150 7.29.1 Temperature Probe — Location and Status 103
 151 7.30 Electrical Current Probe (Type 29) 104
 152 7.30.1 Current Probe — Location and Status..... 104
 153 7.31 Out-of-Band Remote Access (Type 30)..... 105
 154 7.32 Boot Integrity Services (BIS) Entry Point (Type 31) 106
 155 7.33 System Boot Information (Type 32) 106
 156 7.33.1 System Boot Status 106
 157 7.34 64-Bit Memory Error Information (Type 33)..... 107
 158 7.35 Management Device (Type 34) 108
 159 7.35.1 Management Device — Type 108
 160 7.35.2 Management Device — Address Type..... 109
 161 7.36 Management Device Component (Type 35)..... 109
 162 7.37 Management Device Threshold Data (Type 36)..... 110
 163 7.38 Memory Channel (Type 37) 110
 164 7.38.1 Memory Channel — Channel Type 111
 165 7.39 IPMI Device Information (Type 38)..... 111
 166 7.39.1 IPMI Device Information — BMC Interface Type..... 113
 167 7.40 System Power Supply (Type 39) 113
 168 7.40.1 Power Supply Characteristics..... 115
 169 7.41 Additional Information (Type 40)..... 115
 170 7.41.1 Additional Information Entry Format 116
 171 7.42 Onboard Devices Extended Information (Type 41) 117
 172 7.42.1 Reference Designation 118
 173 7.42.2 Onboard Device Types 118
 174 7.42.3 Device Type Instance 118
 175 7.42.4 Segment Group Number, Bus Number, Device/Function Number..... 118
 176 7.43 Management Controller Host Interface (Type 42) 118
 177 7.44 Inactive (Type 126) 120
 178 7.45 End-of-Table (Type 127)..... 120
 179 Annex A (informative) Conformance Guidelines..... 121
 180 Annex B (informative) Using the Table Convention 124
 181 Annex C (informative) Change History 125
 182 Bibliography 135
 183

184 **Tables**

185 Table 1 – SMBIOS Entry Point Structure..... 21
 186 Table 2 – Structure Header Format Description 24
 187 Table 3 – Required Structures and Data 25
 188 Table 4 – Relationship Between SMBIOS Fields and CIM MOF Properties 26
 189 Table 5 – BIOS Information (Type 0) Structure 27
 190 Table 6 – BIOS Characteristics..... 29
 191 Table 7 – BIOS Characteristics Extension Byte 1 30
 192 Table 8 – BIOS Characteristics Extension Byte 2 30

193	Table 9 – System Information (Type 1) Structure	31
194	Table 10 – UUID Byte Order and RFC4122 Field Names	32
195	Table 11 – System: Wake-up Type Field	33
196	Table 12 – Baseboard (or Module) Information (Type 2) Structure	33
197	Table 13 – Baseboard: Feature Flags	34
198	Table 14 – Baseboard: Board Type	35
199	Table 15 – System Enclosure or Chassis (Type 3) Structure	35
200	Table 16 – System Enclosure or Chassis Types	36
201	Table 17 – System Enclosure or Chassis States	38
202	Table 18 – System Enclosure or Chassis Security Status Field	38
203	Table 19 – System Enclosure or Chassis: Contained Elements	39
204	Table 20 – Processor Information (Type 4) Structure	40
205	Table 21 – Processor Information: Processor Type Field	42
206	Table 22 – Processor Information: Processor Family Field	42
207	Table 23 – Processor Information: Voltage Field	48
208	Table 24 – Processor Information: Processor Upgrade Field	49
209	Table 25 – Processor Characteristics	51
210	Table 26 – Memory Controller Information (Type 5, Obsolete) Structure	51
211	Table 27 – Memory Controller Error Detecting Method Field	52
212	Table 28 – Memory Controller Error Correcting Capability Field	52
213	Table 29 – Memory Controller Information: Interleave Support Field	53
214	Table 30 – Memory Controller Information: Memory Speeds Bit Field	53
215	Table 31 – Memory Module Information (Type 6, Obsolete) Structure	54
216	Table 32 – Memory Module Information: Memory Types	54
217	Table 33 – Memory Module Information: Memory Size Field	55
218	Table 34 – Cache Information (Type 7) Structure	57
219	Table 35 – Cache Information: SRAM Type Field	59
220	Table 36 – Cache Information: Error Correction Type Field	59
221	Table 37 – Cache Information: System Cache Type Field	59
222	Table 38 – Cache Information: Associativity Field	60
223	Table 39 – Port Connector Information (Type 8) Structure	60
224	Table 40 – Port Information: Connector Types Field	61
225	Table 41 – Port Types Field	62
226	Table 42 – System Slots (Type 9) Structure	63
227	Table 43 – System Slots: Slot Type Field	64
228	Table 44 – System Slots: Slot Data Bus Width Field	65
229	Table 45 – System Slots: Current Usage Field	66
230	Table 46 – System Slots: Slot Length Field	66
231	Table 47 – System Slots: Slot ID	67
232	Table 48 – Slot Characteristics 1 Field	67
233	Table 49 – Slot Characteristics 2	67
234	Table 50 – On Board Devices Information (Type 10, Obsolete) Structure	68
235	Table 51 – Onboard Device Types	68
236	Table 52 – OEM Strings (Type 11) Structure	69
237	Table 53 – System Configuration Options (Type 12) Structure	69
238	Table 54 – BIOS Language Information (Type 13) Structure	69
239	Table 55 – Group Associations (Type 14) Structure	71
240	Table 56 – System Event Log (Type 15) Structure	72

241	Table 57 – Supported Event Log Type Descriptors	74
242	Table 58 – Access Method Address: DWORD Layout	76
243	Table 59 – Event Log Organization	76
244	Table 60 – Log Header Format	76
245	Table 61 – Log Header Type 1 Format	76
246	Table 62 – Log Record Format	78
247	Table 63 – Event Log Types	78
248	Table 64 – Event Log Variable Data Format Type	79
249	Table 65 – POST Results Bitmap	80
250	Table 66 – System Management Types	81
251	Table 67 – Physical Memory Array (Type 16) Structure	82
252	Table 68 – Memory Array: Location Field	83
253	Table 69 – Memory Array: Use Field	84
254	Table 70 – Memory Array: Error Correction Types Field	84
255	Table 71 – Memory Device (Type 17) Structure	84
256	Table 72 – Memory Device: Form Factor Field	86
257	Table 73 – Memory Device: Type	87
258	Table 74 – Memory Device: Type Detail Field	88
259	Table 75 – 32-Bit Memory Error Information (Type 18) Structure	89
260	Table 76 – Memory Error: Error Type Field	89
261	Table 77 – Memory Error: Error Granularity Field	90
262	Table 78 – Memory Error: Error Operation Field	90
263	Table 79 – Memory Array Mapped Address (Type 19) Structure	91
264	Table 80 – Memory Device Mapped Address (Type 20) Structure	92
265	Table 81 – Built-in Pointing Device (Type 21) Structure	93
266	Table 82 – Pointing Device: Type Field	94
267	Table 83 – Pointing Device: Interface Field	94
268	Table 84 – Portable Battery (Type 22) Structure	95
269	Table 85 – Portable Battery: Device Chemistry Field	96
270	Table 86 – System Reset (Type 23) Structure	97
271	Table 87 – Hardware Security (Type 24) Structure	98
272	Table 88 – System Power Controls (Type 25) Structure	99
273	Table 89 – Voltage Probe (Type 26) Structure	99
274	Table 90 – Voltage Probe: Location and Status Fields	100
275	Table 91 – Cooling Device (Type 27) Structure	101
276	Table 92 – Cooling Device: Device Type and Status Fields	102
277	Table 93 – Temperature Probe (Type 28) Structure	102
278	Table 94 – Temperature Probe: Location and Status Field	103
279	Table 95 – Electrical Current Probe (Type 29) Structure	104
280	Table 96 – Current Probe: Location and Status Field	105
281	Table 97 – Out-of-Band Remote Access (Type 30) Structure	105
282	Table 98 – System Boot Information (Type 32) Structure	106
283	Table 99 – System Boot Status	107
284	Table 100 – 64-Bit Memory Error Information (Type 33) Structure	107
285	Table 101 – Management Device (Type 34) Structure	108
286	Table 102 – Management Device: Type Field	108
287	Table 103 – Management Device: Address Type Field	109
288	Table 104 – Management Device Component (Type 35) Structure	109

289	Table 105 – Management Device Threshold Data (Type 36) Structure	110
290	Table 106 – Memory Channel (Type 37) Structure	111
291	Table 107 – Memory Channel: Channel Type Field	111
292	Table 108 – IPMI Device Information (Type 38) Structure	112
293	Table 109 – IPMI Device Information: BMC Interface Type Field.....	113
294	Table 110 – System Power Supply (Type 39) Structure.....	114
295	Table 111 – Power Supply Characteristics	115
296	Table 112 – Additional Information (Type 40) Structure	116
297	Table 113 – Additional Information Entry Format	116
298	Table 114 – Onboard Devices Extended Information (Type 41) Structure.....	117
299	Table 115 – Onboard Device Types Field	118
300	Table 116 – Management Controller Host Interface (Type 42) Structure.....	119
301	Table 117 – Management Controller Host Interface (Type 42) Structure General Layout.....	119
302	Table 118 – Inactive (Type 126) Structure.....	120
303	Table 119 – End-of-Table (Type 127) Structure	120
304		

305

Foreword

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

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

310 **Acknowledgments**

311 The following persons contributed to the development of version 2.7 of this specification:

- 312 • Kimon Berlin – HP
- 313 • Richard Chan – Dell
- 314 • Philip Chidester – AMD
- 315 • Tom Slaight – Intel
- 316 • Jonathan Stern – Centaur
- 317 • Gail Woodland – IBM

318

319

Introduction

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

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

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

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

339 Document Conventions

340 Typographical Conventions

341 The following typographical conventions are used in this document:

- 342 • All numbers specified in this document are in decimal format unless otherwise indicated. A number
343 followed by the letter 'h' indicates hexadecimal format; a number followed by the letter 'b' indicates
344 binary format.
345 EXAMPLE: The values 10, 0Ah, and 1010b are equivalent.
- 346 • Any value not listed in an enumerated list is reserved for future assignment by the DMTF; see clause
347 6 for more information.
- 348 • Most of the enumerated values defined in this specification simply track the like values specified by
349 the DMTF within CIM classes. Enumerated values that are controlled by the DMTF are identified
350 within their respective subsection; additional values for these fields are assigned by the DMTF; see
351 6.3 for more information.
- 352 • Code samples use a `fixed font highlighted in gray`.

353 Document Version Number Conventions

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

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

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

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

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

371 If these conventions were in place when version 2.0 of the specification was released, they would have
372 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.

373

374 System Management BIOS (SMBIOS) Reference Specification

375 1 Scope

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

382 2 Normative references

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

387 ACPI, *Advanced Configuration and Power Interface Specification*, Version 3.0, September 2, 2004,
388 <http://www.acpi.info>

389 *Boot Integrity Services API*, Version 1.0+bis37, 31 August 1999,
390 <http://sourceforge.net/projects/bis>

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

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

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

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

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

401 DMTF, *CIM Schema*, Version 2.25, 31 March 2010,
402 <http://www.dmtf.org/standards/cim/>

403 IETF RFC 4122, *A Universally Unique Identifier (UUID) URN Namespace*, The Internet Society, July
404 2005, <http://www.ietf.org/rfc/rfc4122.txt>

405 Intel, *Intelligent Platform Management Interface (IPMI) Interface Specification*, Version 2.0, February 12
406 2004, <http://developer.intel.com/design/servers/ipmi/spec.htm>

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

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

- 411 ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country*
412 *codes*
413 A list of codes is available at http://www.iso.org/iso/english_country_names_and_code_elements
- 414 Microsoft, *PCI IRQ Routing Table Specification*, Version 1.0, February 27, 1996,
415 <http://www.microsoft.com/whdc/archive/pciirq.mspix>
- 416 Microsoft, *Plug and Play BIOS Specification*, Version 1.0A, May 5, 1994,
417 <http://www.microsoft.com/whdc/system/pnppwr/pnp/default.mspix>
- 418 Microsoft, *Simple Boot Flag Specification*, Version 2.1, 28 January 2005,
419 http://www.microsoft.com/whdc/resources/respec/specs/simp_boot.mspix
- 420 PCI SIG, *PCI Firmware Specification*, version 3.0, June 20, 2005,
421 http://www.pcisig.com/specifications/conventional/pci_firmware
- 422 Phoenix Technologies, Ltd., *BIOS Boot Specification*, Version 1.01, 11 January 1996,
423 A copy is available at <http://www.scs.stanford.edu/nyu/04fa/lab/specsbbs101.pdf>
- 424 Phoenix Technologies, Ltd., *“El Torito” Bootable CD-ROM Format Specification*, Version 1.0, January 25
425 1995, <http://download.intel.com/support/motherboards/desktop/sb/specscdrom.pdf>
- 426 SBS, *Smart Battery Data Specification*, Version 1.1, 15 December 1998,
427 <http://www.sbs-forum.org/specs/>
- 428 UEFI, *Unified Extensible Firmware Interface (UEFI) Specification*, Version 2.3, May 2009,
429 <http://www.uefi.org/specs/>.
- 430 UEFI, *UEFI Platform Initialization (PI) Specification*, Version 1.2, May 2009,
431 <http://www.uefi.org/specs/>

432 **3 Terms and definitions**

- 433 In this document, some terms have a specific meaning beyond the normal English meaning. Those terms
434 are defined in this clause.
- 435 The terms "shall" ("required"), "shall not," "should" ("recommended"), "should not" ("not recommended"),
436 "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
437 in [ISO/IEC Directives, Part 2](#), Annex H. The terms in parenthesis are alternatives for the preceding term,
438 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
439 [ISO/IEC Directives, Part 2](#), Annex H specifies additional alternatives. Occurrences of such additional
440 alternatives shall be interpreted in their normal English meaning.
- 441 The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as
442 described in [ISO/IEC Directives, Part 2](#), Clause 5.
- 443 The terms "normative" and "informative" in this document are to be interpreted as described in [ISO/IEC](#)
444 [Directives, Part 2](#), Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do
445 not contain normative content. Notes and examples are always informative elements.
- 446 The terms defined in [DSP0004](#), [DSP0200](#), and [DSP1001](#) apply to this document. The following additional
447 terms are used in this document.

448 **4 Symbols and abbreviated terms**

449 The abbreviations defined in [DSP0004](#), [DSP0200](#), and [DSP1001](#) apply to this document. The following
450 additional abbreviations are used in this document.

451 **4.1**

452 **AC**

453 Alternating Current

454 **4.2**

455 **ACPI**

456 Advanced Configuration and Power Interface

457 **4.3**

458 **AGP**

459 Accelerated Graphics Port

460 **4.4**

461 **APM**

462 Advanced Power Management

463 **4.5**

464 **ASCII**

465 American Standard Code for Information Interchange

466 **4.6**

467 **ATA**

468 Advanced Technology Attachment

469 **4.7**

470 **ATAPI**

471 ATA Packet Interface

472 **4.8**

473 **BCD**

474 Binary-Coded Decimal

475 **4.9**

476 **BIOS**

477 Basic Input/Output System

478 **4.10**

479 **BMC**

480 Baseboard Management Controller

481 **4.11**

482 **CGA**

483 Color Graphics Array

484 **4.12**

485 **CIM**

486 Common Information Model

487	4.13
488	CMOS
489	Complementary Metal-Oxide Semiconductor. “CMOS” is commonly used as a shorthand for “CMOS
490	RAM”, the non-volatile RAM used on industry-standard PCs.
491	4.14
492	CPU
493	Central Processing Unit
494	4.15
495	CRC
496	Cyclic Redundancy Check
497	4.16
498	DDC
499	Display Data Channel
500	4.17
501	DDR
502	Double Data Rate SDRAM
503	4.18
504	DIMM
505	Dual In-line Memory Module
506	4.19
507	DMA
508	Direct Memory Access
509	4.20
510	DMI
511	Desktop Management Interface
512	4.21
513	DRAM
514	Dynamic RAM
515	4.22
516	DSP
517	Digital Signal Processing
518	4.23
519	ECC
520	Error Checking and Correction
521	4.24
522	EDD
523	Enhanced Disk Drive
524	4.25
525	EDO
526	Extended Data Out

527	4.26
528	EEPROM
529	Electrically Erasable PROM
530	4.27
531	EISA
532	Extended Industry-Standard Architecture
533	4.28
534	EPS
535	Entry Point Structure
536	4.29
537	ESCD
538	Extended System Configuration Data
539	4.30
540	FDC
541	Floppy Drive Controller
542	4.31
543	FIFO
544	First In, First Out
545	4.32
546	GPNV
547	General-Purpose NVRAM
548	4.33
549	I2O
550	Intelligent Input/Output
551	4.34
552	IEPS
553	Intermediate Entry Point Structure
554	4.35
555	IO
556	Input/Output
557	4.36
558	IRQ
559	Interrupt Request
560	4.37
561	ISA
562	Industry Standard Architecture
563	4.38
564	LIF
565	Low Insertion Force

566	4.39
567	LSB
568	Least-Significant Bit
569	4.40
570	MCA
571	Micro Channel Architecture
572	4.41
573	MOF
574	Managed Object Format
575	4.42
576	MSB
577	Most Significant Bit
578	4.43
579	NMI
580	Non-Maskable Interrupt
581	4.44
582	NV
583	Non-Volatile
584	4.45
585	NVRAM
586	Non-Volatile RAM
587	4.46
588	OEM
589	Original Equipment Manufacturer
590	4.47
591	OS
592	Operating System
593	4.48
594	PATA
595	Parallel ATA
596	4.49
597	PCI
598	Peripheral Component Interconnect
599	4.50
600	PCMCIA
601	Personal Computer Memory Card International Association
602	4.51
603	PME
604	Power Management Event

605	4.52
606	PNP
607	Plug-And-Play
608	4.53
609	POST
610	Power-On Self-Test
611	4.54
612	PROM
613	Programmable ROM
614	4.55
615	PXE
616	Pre-boot Execution Environment
617	4.56
618	RAID
619	Redundant Array of Inexpensive Disks
620	4.57
621	RAM
622	Random-Access Memory
623	4.58
624	ROM
625	Read-Only Memory
626	4.59
627	RPM
628	Revolutions per Minute
629	4.60
630	RTC
631	Real-Time Clock
632	4.61
633	SAS
634	Serial-Attached SCSI
635	4.62
636	SATA
637	Serial ATA
638	4.63
639	SCSI
640	Small Computer System Interface
641	4.64
642	SDRAM
643	Synchronous DRAM

644	4.65
645	SIMM
646	Single In-line Memory Module
647	4.66
648	SKU
649	Stock-Keeping Unit
650	4.67
651	SMBIOS
652	System Management BIOS
653	4.68
654	SMBus
655	System Management Bus
656	4.69
657	SRAM
658	Static RAM
659	4.70
660	UEFI
661	Unified Extensible Firmware Interface
662	4.71
663	UPS
664	Uninterruptible Power Supply
665	4.72
666	USB
667	Universal Serial Bus
668	4.73
669	UUID
670	Universally Unique Identifier
671	4.74
672	VESA
673	Video Electronics Standards Association
674	4.75
675	VL-VESA
676	VESA Video Local Bus
677	4.76
678	ZIF
679	Zero Insertion Force

680 5 Accessing SMBIOS Information

681 5.1 General

682 The only access method defined for the SMBIOS structures is a table-based method, defined in version
683 2.1 of this specification. It provides the SMBIOS structures as a packed list of data referenced by a table
684 entry point.

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

687 5.2 Table Convention

688 The table convention allows the SMBIOS structures to be accessed under 32-bit and 64-bit protected-
689 mode operating systems such as Microsoft Windows XP®, Microsoft Windows Server®, or Linux®. This
690 convention provides a searchable entry-point structure (which can be queried on EFI-based systems) that
691 contains a pointer to the packed SMBIOS structures residing somewhere in 32-bit physical address space
692 (that is, below 4 GB).

693 See Annex B for pseudo-code using this convention.

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

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

697 5.2.1 SMBIOS Structure Table Entry Point

698 The SMBIOS Entry Point Structure is described in Table 1.

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

703 On EFI-based systems, the SMBIOS Entry Point structure can be located by looking in the EFI
704 Configuration Table for the SMBIOS GUID (SMBIOS_TABLE_GUID) and using the associated pointer.
705 See section 4.6 of the [UEFI Specification](#) for details. See section 2.3 and table 5-6 of the [UEFI](#)
706 [Specification](#) for how to report the underlying memory type.

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

710 **Table 1 – SMBIOS 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 1Fh value, but version 2.2 or later implementations must use the 1Fh value.

Offset	Name	Length	Description
06h	SMBIOS Major Version	BYTE	Identifies the 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	Identifies the 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. This is the value returned as StructureSize from the Plug-and-Play Get SMBIOS Information function.
0Ah	Entry Point Revision	BYTE	Identifies the 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	The 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	The 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.
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.

711 6 SMBIOS Structures

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

715 6.1 Structure Standards

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

719 6.1.1 Structure Evolution and Usage Guidelines

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

- 722 • If a new field is added to an existing structure, that field is added at the end of the formatted
723 area of that structure and the structure's *Length* field is increased by the new field's size.
- 724 • Any software that interprets a structure shall use the structure's *Length* field to determine the
725 formatted area size for the structure rather than hard-coding or deriving the *Length* from a
726 structure field.
- 727 • Each structure shall be terminated by a double-null (0000h), either directly following the
728 formatted area (if no strings are present) or directly following the last string. This includes
729 system- and OEM-specific structures and allows upper-level software to easily traverse the
730 structure table. (See structure-termination examples later in this clause.)
- 731 • The unformed section of the structure is used for passing variable data such as text strings; see
732 6.1.3 for more information.
- 733 • When an enumerated field's values are controlled by the DMTF, new values can be used as
734 soon as they are defined by the DMTF without requiring an update to this specification.
- 735 • Starting with version 2.3, each SMBIOS structure type has a *minimum* length — enabling the
736 addition of new, but optional, fields to SMBIOS structures. In no case shall a structure's length
737 result in a field being less than fully populated. For example, a Voltage Probe structure with
738 *Length* of 15h is invalid because the *Nominal Value* field would not be fully specified.
- 739 • Software that interprets a structure field must verify that the structure's length is sufficient to
740 encompass the optional field; if the length is insufficient, the optional field's value is *Unknown*.
741 For example, if a Voltage Probe structure has a *Length* field of 14h, the probe's *Nominal Value*
742 is *Unknown*. A Voltage Probe structure with *Length* greater than 14h always includes a *Nominal*
743 *Value* field.

744 EXAMPLE 1: BIOS Information with strings:

```

745 BIOS_Info LABEL BYTE
746 db 0 ; Indicates BIOS Structure Type
747 db 13h ; Length of information in bytes
748 dw ? ; Reserved for handle
749 db 01h ; String 1 is the Vendor Name
750 db 02h ; String 2 is the BIOS version
751 dw 0E800h ; BIOS Starting Address
752 db 03h ; String 3 is the BIOS Build Date
753 db 1 ; Size of BIOS ROM is 128K (64K * (1 + 1))
754 dq BIOS_Char ; BIOS Characteristics
755 db 0 ; BIOS Characteristics Extension Byte 1
756 db 'System BIOS Vendor Name',0 ;
757 db '4.04',0 ;
758 db '00/00/0000',0 ;
759 db 0 ; End of strings

```

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

```

761 BIOS_Info LABEL BYTE
762 db 0          ; Indicates BIOS Structure Type
763 db 13h       ; Length of information in bytes
764 dw ?        ; Reserved for handle
765 db 00h      ; No Vendor Name provided
766 db 00h      ; No BIOS version provided
767 dw 0E800h   ; BIOS Starting Address
768 db 00h      ; No BIOS Build Date provided
769 db 1        ; Size of BIOS ROM is 128K (64K * (1 + 1))
770 dq BIOS_Char ; BIOS Characteristics
771 db 0        ; BIOS Characteristics Extension Byte 1
772 dw 0000h    ; Structure terminator

```

773 6.1.2 Structure Header Format

774 Each SMBIOS structure begins with a 4-byte header as shown in Table 2.

775 **Table 2 – 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.
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.
<p>^[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.</p>			

776 6.1.3 Text Strings

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

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

790 **6.2 Required Structures and Data**

791 Beginning with SMBIOS version 2.3, compliant SMBIOS implementations include the base set of required
 792 structures and data within those structures shown in Table 3. For a detailed list of conformance
 793 guidelines, refer to Annex A.

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

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

797 **Table 3 – 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. <i>BIOS Version</i> and <i>BIOS Release Date</i> 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)	<i>Manufacturer</i> and <i>Product Name</i> strings are non-null. <i>UUID</i> field identifies the system’s non-zero UUID value. <i>Wake-up Type</i> 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)	<i>Manufacturer</i> string is non-null; the <i>Type</i> field identifies the type of enclosure. (Unknown is disallowed.)
Processor Information (Type 4)	One structure is required for each system processor. The presence of two structures with the <i>Processor Type</i> field set to <i>Central Processor</i> , for instance, identifies that the system is capable of dual-processor operations. <i>Socket Designation</i> string is non-null. <i>Processor Type</i> , <i>Max Speed</i> , and <i>Processor Upgrade</i> 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 <i>CPU Socket Populated</i> sub-field of the <i>Status</i> field indicates that the socket is populated), the <i>Processor Manufacturer</i> string is non-null and the <i>Processor Family</i> , <i>Current Speed</i> , and <i>CPU Status</i> sub-field of the <i>Status</i> field are all set to “known” values. Each of the <i>Lx Cache Handle</i> fields is either set to 0xFFFF (no further cache description) or references a valid <i>Cache Information</i> structure.
Cache Information (Type 7)	One structure is required for each cache that is external to the processor. <i>Socket Designation</i> string is non-null if the cache is external to the processor. If the cache is present (that is, the <i>Installed Size</i> is non-zero), the <i>Cache Configuration</i> 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). <i>Slot Designation</i> string is non-null. <i>Slot Type</i> , <i>Slot Data Bus Width</i> , <i>Slot ID</i> , and <i>Slot Characteristics 1 & 2</i> are all set to “known” values. If device presence is detectable within the slot (for example, PCI), the <i>Current Usage</i> field must be set to either <i>Available</i> or <i>In-use</i> . Otherwise (for example, ISA), the Unknown value for the field is also allowed.

Structure Name and Type	Data Requirements
Physical Memory Array (Type 16)	One structure is required for the system memory. <i>Location</i> , <i>Use</i> , and <i>Memory Error Correction</i> are all set to “known” values. Either <i>Maximum Capacity</i> or <i>Extended Maximum Capacity</i> must be set to a known, non-zero value. <i>Number of Memory Devices</i> is non-zero and identifies the number of <i>Memory Device</i> structures that are associated with this <i>Physical Memory Array</i> .
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. <i>Device Locator</i> string is set to a non-null value. <i>Memory Array Handle</i> contains the handle associated with the <i>Physical Memory Array</i> structure to which this device belongs. <i>Data Width</i> , <i>Size</i> , <i>Form Factor</i> , and <i>Device Set</i> are all set to “known” values. If the device is present (for instance, <i>Size</i> is non-zero), the <i>Total Width</i> 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 <i>Physical Memory Array</i> . Either the pair of <i>Starting Address</i> and <i>Ending Address</i> is set to a valid address range or the pair of <i>Extended Starting Address</i> and <i>Extended Ending Address</i> is set to a valid address range. If the pair of <i>Starting Address</i> and <i>Ending Address</i> is used, then <i>Ending Address</i> must be larger than <i>Starting Address</i> . If the pair of <i>Extended Starting Address</i> and <i>Extended Ending Address</i> is used, <i>Extended Ending Address</i> must be larger than <i>Extended Starting Address</i> . Each structure’s address range is unique and non-overlapping. <i>Memory Array Handle</i> references a <i>Physical Memory Array</i> structure. <i>Partition Width</i> is non-zero.
System Boot Information (Type 32)	The structure’s length is at least 0x0B (for instance, at least one byte of <i>System Boot Status</i> is provided).

798 6.3 SMBIOS Fields and CIM MOF Properties

799 Many of the enumerated values are shared between SMBIOS fields and Common Information Model
800 (CIM) MOF properties. Table 4 identifies the relationships; any additions to these enumerated lists should
801 be reflected in both documents by submitting change requests to <mailto:schema-sc@dmf.org> and
802 <mailto:smbios@dmf.org> for the CIM-related and SMBIOS-related updates, respectively. Any other
803 enumerated value identified in this specification is controlled by this specification; change requests should
804 be sent to <mailto:smbios@dmf.org>.

805 **Table 4 – Relationship Between SMBIOS Fields and CIM MOF Properties**

Name	Section	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.

Name	Section	MOF Class.Property
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
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

806 **7 Structure Definitions**

807 **7.1 BIOS Information (Type 0)**

808 Table 5 shows the BIOS Information structure.

809 **Table 5 – BIOS Information (Type 0) Structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	0	BIOS Information Indicator

Offset	Name	Length	Value	Description
01h	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	Handle	WORD	Varies	
04h	Vendor	BYTE	STRING	String number of the BIOS Vendor's Name.
05h	BIOS Version	BYTE	STRING	String number of the BIOS Version. This is a free-form string that may contain Core and OEM version information.
06h	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	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	BIOS ROM Size	BYTE	Varies (n)	Size (n) where $64K * (n+1)$ is the size of the physical device containing the BIOS, in bytes
0Ah	BIOS Characteristics	QWORD	Bit Field	Defines which functions the BIOS supports: PCI, PCMCIA, Flash, etc. (see 7.1.1).
12h	BIOS Characteristics Extension Bytes	Zero or more BYTES	Bit Field	Optional space reserved for future supported functions. The number of Extension Bytes that are 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	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 and/or the System BIOS Minor Release field is 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	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.

Offset	Name	Length	Value	Description
16h	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 and/or the Embedded Controller Firmware Minor Release field is 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	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.

810 **7.1.1 BIOS Characteristics**

811 Table 6 shows the BIOS Characteristics layout.

812 **Table 6 – 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.

QWORD Bit Position	Meaning If Set
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
Bits 32:47	Reserved for BIOS Vendor
Bits 48:63	Reserved for System Vendor

813 7.1.2 BIOS Characteristics Extension Bytes

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

815 7.1.2.1 BIOS Characteristics Extension Byte 1

816 Table 7 shows the BIOS Characteristics Extension Byte 1 layout. This information, available for SMBIOS
817 version 2.1 and later, appears at offset 12h within the BIOS Information structure.

818 **Table 7 – 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.

819 7.1.2.2 BIOS Characteristics Extension Byte 2

820 Table 8 shows the BIOS Characteristics for Extension Byte 2 layout. This information, available for
821 SMBIOS version 2.3 and later, appears at offset 13h within the BIOS Information structure.

822 **Table 8 – 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	The 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

823 **7.2 System Information (Type 1)**

824 The information in this structure defines attributes of the overall system and is intended to be associated
 825 with the *Component ID* group of the system's MIF. An SMBIOS implementation is associated with a single
 826 system instance and contains one and only one System Information (Type 1) structure. Table 9 shows
 827 the contents of this structure.

828 **Table 9 – 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: <ul style="list-style-type: none"> • 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 is used to identify 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.

Offset	Spec. Version	Name	Length	Value	Description
1Ah	2.4+	Family	BYTE	STRING	Number of Null terminated string This text string is used to identify the family a particular computer belongs to. 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.

829 7.2.1 System — UUID

830 A UUID is an identifier that is designed to be unique across both time and space. It requires no central
831 registration process. The UUID is 128 bits long. Its format is described in [RFC 4122](#), but the actual field
832 contents are opaque and not significant to the SMBIOS specification, which is only concerned with the
833 byte order. Table 10 shows the field names; these field names, particularly for multiplexed fields, follow
834 historical practice.

835 **Table 10 – UUID Byte Order and RFC4122 Field Names**

Offset	RFC 4122 Name	Length	Value	Description
00h	time_low	DWORD	Varies	The low field of the timestamp
04h	time_mid	WORD	Varies	The middle field of the timestamp
06h	time_hi_and_version	WORD	Varies	The high field of the timestamp multiplexed with the version number
08h	clock_seq_hi_and_reserved	BYTE	Varies	The high field of the clock sequence multiplexed with the variant
09h	clock_seq_low	BYTE	Varies	The low field of the clock sequence
0Ah	Node	6 BYTES	Varies	The spatially unique node identifier

836 Although [RFC 4122](#) recommends network byte order for all fields, the PC industry (including the [ACPI](#),
837 [UEFI](#), and [Microsoft](#) specifications) has consistently used little-endian byte encoding for the first three
838 fields: *time_low*, *time_mid*, *time_hi_and_version*. The same encoding, also known as *wire format*, should
839 also be used for the SMBIOS representation of the UUID.

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

841 33 22 11 00 55 44 77 66 88 99 AA BB CC DD EE FF.

842 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,
843 the ID is not present in the system.

844 7.2.2 System — Wake-up Type

845 Table 11 shows what the byte values mean for the System — Wake-up Type field.

846

Table 11 – 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

847 **7.3 Baseboard (or Module) Information (Type 2)**

848 As shown in Table 12, the information in this structure defines attributes of a system baseboard (for
849 example, a motherboard, planar, server blade, or other standard system module).

850 NOTE: If more than one Type 2 structure is provided by an SMBIOS implementation, each structure shall include the
851 *Number of Contained Object Handles* and *Contained Object Handles* fields to specify which system elements are
852 contained on which boards. If a single Type 2 structure is provided and the contained object information is not
853 present¹, or if no Type 2 structure is provided, then all system elements identified by the SMBIOS implementation are
854 associated with a single motherboard.

855

Table 12 – 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	A collection of flags that identify features of this baseboard. See 7.3.1.

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

Offset	Name	Length	Value	Description
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: <ul style="list-style-type: none"> LocationWithinContainer is this field. GroupComponent is the chassis referenced by Chassis Handle. PartComponent is this baseboard.
0Bh	Chassis Handle	WORD	Varies	The handle, or instance number, associated with the chassis in which this board resides (see 7.4)
0Dh	Board Type	BYTE	ENUM	Identifies the type of board (see 7.3.2)
0Eh	Number of Contained Object Handles (n)	BYTE	Varies	Identifies the number (0 to 255) of Contained Object Handles that follow
0Fh	Contained Object Handles	n WORDs	Varies	A list of handles of other structures (for example, Baseboard, Processor, Port, System Slots, Memory Device) that are contained by this baseboard

856 7.3.1 Baseboard — Feature Flags

857 Table 13 shows the baseboard feature flags.

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

859 **Table 13 – 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 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).

860 7.3.2 Baseboard — Board Type

861 Table 14 shows what the byte values mean for the Baseboard — Board Type field.

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

864

Table 14 – 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

865 **7.4 System Enclosure or Chassis (Type 3)**

866 The information in this structure (see Table 15) defines attributes of the system’s mechanical
 867 enclosure(s). For example, if a system included a separate enclosure for its peripheral devices, two
 868 structures would be returned: one for the main system enclosure and the second for the peripheral device
 869 enclosure. The additions to this structure in version 2.1 of this specification support the population of the
 870 CIM_Chassis class.

871

Table 15 – 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

Offset	Specification Version	Name	Length	Value	Description
09h	2.1+	Boot-up State	BYTE	ENUM	Identifies the state of the enclosure when it was last booted. See 7.4.2 for definitions.
0Ah	2.1+	Power Supply State	BYTE	ENUM	Identifies the 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	Identifies the enclosure's thermal state when last booted. See 7.4.2 for definitions.
0Ch	2.1+	Security Status	BYTE	ENUM	Identifies the enclosure's physical security status when last booted. See 7.4.3 for definitions.
0Dh	2.3+	OEM-defined	DWORD	Varies	Contains OEM- or BIOS vendor-specific information.
11h	2.3+	Height	BYTE	Varies	The 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	Identifies the 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	Identifies the number of Contained Element records that follow, in the range 0 to 255. Each Contained Element group comprises <i>m</i> bytes, as specified by the Contained Element Record Length field that follows. If no Contained Elements are included, this field is set to 0.
14h	2.3+	Contained Element Record Length (m)	BYTE	Varies	Identifies the 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	Identifies the 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

872 7.4.1 System Enclosure or Chassis Types

873 Table 16 shows what the byte values mean for the System Enclosure or Chassis Types field.

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

875

Table 16 – System Enclosure or Chassis Types

Byte Value	Meaning
01h	Other

Byte Value	Meaning
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
16h	RAID Chassis
17h	Rack Mount Chassis
18h	Sealed-case PC
19h	<p>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.</p> <p>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.</p>
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.

Byte Value	Meaning
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.

876 7.4.2 System Enclosure or Chassis States

877 Table 17 shows what the byte values mean for the System Enclosure or Chassis States field.

878 **Table 17 – System Enclosure or Chassis States**

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

879 7.4.3 System Enclosure or Chassis Security Status

880 Table 18 shows what the byte values mean for the System Enclosure or Chassis Security Status field.

881 **Table 18 – 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

882 7.4.4 System Enclosure or Chassis — Contained Elements

883 Each *Contained Element* record consists of sub-fields that further describe elements contained by the
 884 chassis, as shown in Table 19. Relative offset and size of fields within each record shall remain the same
 885 in future revisions to this specification, but new fields might be added to the end of the current definitions.

886

Table 19 – System Enclosure or Chassis: Contained Elements

Offset	Spec. Version	Name	Length	Value	Description						
00h	2.3+	Contained Element Type	BYTE	Bit Field	<p>Specifies the type of element associated with this record:</p> <table border="1"> <thead> <tr> <th>Bit(s)</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>7</td> <td><u>Type Select</u>. Identifies whether the Type contains an SMBIOS structure type enumeration (1) or an SMBIOS Baseboard Type enumeration (0).</td> </tr> <tr> <td>6:0</td> <td><u>Type</u>. The value 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.</td> </tr> </tbody> </table> <p>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.</p>	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> . The value 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.
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> . The value 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.										
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.						

887 **7.5 Processor Information (Type 4)**

888 The information in this structure (see Table 20) defines the attributes of a single processor; a separate
 889 structure instance is provided for each system processor socket/slot. For example, a system with an
 890 IntelDX2™ processor would have a single structure instance while a system with an IntelSX2™ processor
 891 would have a structure to describe the main CPU and a second structure to describe the 80487 co-
 892 processor.

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

897

Table 20 – 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	The Length is 1Ah for version 2.0 implementations or a minimum of 20h for version 2.1 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 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 <i>Processor ID</i> 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	<p>Bit 7 Reserved, must be zero</p> <p>Bit 6 CPU Socket Populated 1 – CPU Socket Populated 0 – CPU Socket Unpopulated</p> <p>Bits 5:3 Reserved, must be zero</p> <p>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</p>
19h	2.0+	Processor Upgrade	BYTE	ENUM	See 7.5.5.
1Ah	2.1+	L1 Cache Handle	WORD	Varies	The 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	The 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	The 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]
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.

Offset	Spec. Version	Name	Length	Value	Description
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.
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.
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.
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.

^[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.

898 7.5.1 Processor Information — Processor Type

899 Table 21 shows what the byte values mean for the Processor Information — Processor Type field.

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

901 **Table 21 – Processor Information: Processor Type Field**

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

902 7.5.2 Processor Information — Processor Family

903 Table 22 details the values for the Processor Information — Processor Family field.

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

905 **Table 22 – Processor Information: Processor Family Field**

Hex Value	Decimal Value	Meaning
01h	1	Other
02h	2	Unknown

Hex Value	Decimal Value	Meaning
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
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-2Fh	44-47	Available for assignment

Hex Value	Decimal Value	Meaning
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-3Fh	61-63	Available for assignment
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-4Fh	70-79	Available for assignment
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 Ili
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-6Fh	102-111	Available for assignment
70h	112	Hobbit Family
71h-77h	113-119	Available for assignment
78h	120	Crusoe™ TM5000 Family

Hex Value	Decimal Value	Meaning
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
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

Hex Value	Decimal Value	Meaning
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
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/Architectur 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

Hex Value	Decimal Value	Meaning
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-E5h	225-229	Available for assignment
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
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
<p>^[1] Note that the meaning associated with this value is different from the meaning defined in CIM_Processor.Family for the same value.</p>		
<p>^[2] Some version 2.0 specification implementations used <i>Processor Family</i> type value 30h to represent a Pentium® Pro processor.</p>		

Hex Value	Decimal Value	Meaning
^[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.		

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

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

909 The following values are reserved:

- 910 • FFh Not used. FFh is the un-initialized value of Flash memory.
- 911 • FFFFh Not used. FFFFh is the un-initialized value of Flash memory.
- 912 • FFFEh For special use in the future, such as FEh as the extension indicator.

913 7.5.3 Processor ID Field Format

914 The Processor ID field contains processor-specific information that describes the processor’s features.

915 7.5.3.1 x86-Class CPUs

916 For x86 class CPUs, the field’s format depends on the processor’s support of the CPUID instruction. If the
917 instruction is supported, the *Processor ID* field contains two DWORD-formatted values. The first (offsets
918 08h-0Bh) is the EAX value returned by a CPUID instruction with input EAX set to 1; the second (offsets
919 0Ch-0Fh) is the EDX value returned by that instruction.

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

922 7.5.4 Processor Information — Voltage

923 Two forms of information can be specified by the SMBIOS in this field, dependent on the value present in
924 bit 7 (the most-significant bit). If bit 7 is 0 (legacy mode), the remaining bits of the field represent the
925 specific voltages that the processor socket can accept, as shown in Table 23.

926 **Table 23 – 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	<p><i>Voltage Capability</i>. A Set bit indicates that the voltage is supported.</p> <p>Bit 0 – 5V</p> <p>Bit 1 – 3.3V</p> <p>Bit 2 – 2.9V</p> <p>Bit 3 – Reserved, must be zero.</p> <p>NOTE: Setting of multiple bits indicates the socket is configurable.</p>

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

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

930 $92h = 80h + (1.8 * 10) = 80h + 18 = 80h + 12h$

931 7.5.5 Processor Information — Processor Upgrade

932 Table 24 shows what the byte values mean for the Processor Information — Processor Upgrade field.

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

934

Table 24 – Processor Information: Processor Upgrade Field

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

Byte Value	Meaning
1Dh	Socket LGA1156
1Eh	Socket LGA1567
1Fh	Socket PGA988A
20h	Socket BGA1288

935 7.5.6 Processor Information — Core Count

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

942 7.5.7 Processor Information — Core Enabled

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

946 7.5.8 Processor Information — Thread Count

947 *Thread Count* is the total number of threads detected by the BIOS for this processor socket. It is a
 948 processor-wide count, not a thread-per-core count. It does not necessarily indicate the full capability of
 949 the processor. For example, platform hardware may have the capability to limit the number of threads
 950 reported by the processor without BIOS intervention or knowledge. For a dual-thread processor installed
 951 in a platform where the hardware is set to limit it to one thread, the BIOS reports a value of 1 in *Thread*
 952 *Count*. For a dual-thread processor with multi-threading disabled by BIOS, the BIOS reports a value of 2
 953 in *Thread Count*. For a dual-core, dual-thread-per-core processor, the BIOS reports a value of 4 in
 954 *Thread Count*.

955 7.5.9 Processor Characteristics

956 Table 25 describes the Processor Characteristics field.

957 *64-bit Capable* indicates the maximum data width capability of the processor. For example, this bit is set
 958 for Intel Itanium, AMD Opteron, and Intel Xeon (with EM64T) processors; this bit is cleared for Intel Xeon
 959 processors that do not have EM64T. This bit indicates the maximum capability of the processor and does
 960 not indicate the current enabled state.

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

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

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

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

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

972 **Table 25 – 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

973 **7.6 Memory Controller Information (Type 5, Obsolete)**

974 The information in this structure defines the attributes of the system’s memory controller(s) and the
 975 supported attributes of any memory-modules present in the sockets controlled by this controller. See
 976 Table 26 for the details of this structure.

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

981 **Table 26 – 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)	The 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.

Offset	Spec. Version	Name	Length	Value	Description
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	This 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
0Fh to 0Fh + (2*x) - 1	2.0+	Memory Module Configuration Handles	x WORDs	Varies	A list of 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.

982 7.6.1 Memory Controller Error Detecting Method

983 Table 27 shows the byte values for the Memory Controller Error Detecting Method field.

984 **Table 27 – 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

985 7.6.2 Memory Controller Error Correcting Capability

986 Table 28 shows the values for the Memory Controller Error Correcting Capability field.

987 **Table 28 – 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

988 **7.6.3 Memory Controller Information — Interleave Support**

989 Table 29 shows the byte values for the Memory Controller Information — Interleave Support field.

990 **Table 29 – Memory Controller Information: Interleave Support Field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	One-Way Interleave
04h	Two-Way Interleave
05h	Four-Way Interleave
06h	Eight-Way Interleave
07h	Sixteen-Way Interleave

991 **7.6.4 Memory Controller Information — Memory Speeds**

992 The bit-field shown in Table 30 describes the speed of the memory modules supported by the system.

993 **Table 30 – 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

994 **7.7 Memory Module Information (Type 6, Obsolete)**

995 One *Memory Module Information* structure is included for each memory-module socket in the system. As
 996 shown in Table 31, the structure describes the speed, type, size, and error status of each system memory
 997 module. The supported attributes of each module are described by the “owning” *Memory Controller*
 998 *Information* structure.

999 NOTE: This structure and its companion Memory Controller Information (Type 5, Obsolete) are **obsolete** starting
 1000 with version 2.1 of this specification; the Physical Memory Array (Type 16) and Memory Device (Type 17) structures
 1001 should be used instead. BIOS providers might choose to implement both memory description types to allow existing
 1002 DMI browsers to properly display the system’s memory attributes.

1003

Table 31 – 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	The 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.
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 1 and 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.

1004 **7.7.1 Memory Module Information — Memory Types**

1005 The bit-field shown in Table 32 describes the physical characteristics of the memory modules that are
1006 supported by (and currently installed in) the system.

1007

Table 32 – 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

1008 **7.7.2 Memory Module Information — Memory Size**

1009 The Size fields of the Memory Module Configuration Information structure define the amount of memory
 1010 currently installed (and enabled) in a memory-module connector. Table 33 shows the meaning of the
 1011 bytes and bits in the Memory Size field.

1012 The *Installed Size* fields identify the size of the memory module that is installed in the socket, as
 1013 determined by reading and correlating the module’s presence-detect information. If the system does not
 1014 support presence-detect mechanisms, the *Installed Size* field is set to 7Dh to indicate that the installed
 1015 size is not determinable. The *Enabled Size* field identifies the amount of memory currently enabled for the
 1016 system’s use from the module. If a module is known to be installed in a connector, but all memory in the
 1017 module has been disabled due to error, the *Enabled Size* field is set to 7Eh.

1018 **Table 33 – Memory Module Information: Memory Size Field**

Byte Bit Range	Meaning
Bits 0:6	Size (n), where 2**n is the size in MB, with the following special-case values: <ul style="list-style-type: none"> • 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.

1019 **7.7.3 Memory Subsystem Example**

1020 A system utilizes a memory controller that supports up to 4-32 MB 5 V 70 ns parity SIMMs. The memory
 1021 module sockets are used in pairs A1/A2 and B1/B2 to provide a 64-bit data path to the CPU. No
 1022 mechanism is provided by the system to read the SIMM IDs. RAS-0 and -1 are connected to the front-
 1023 and back-size banks of the SIMMs in the A1/A2 sockets and RAS-2 and -3 are similarly connected to the
 1024 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 jjjjMB (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 0000010b  ; 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

```

1025 **7.8 Cache Information (Type 7)**

1026 As shown in Table 34, the information in this structure defines the attributes of CPU cache device in the
 1027 system. One structure is specified for each such device, whether the device is internal to or external to
 1028 the CPU module. Cache modules can be associated with a processor structure in one or two ways
 1029 depending on the SMBIOS version; see 7.5 and 7.15 for more information.

1030 **Table 34 – 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	The 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	The cache module speed, in nanoseconds. The value is 0 if the speed is unknown.
10h	2.1+	Error Correction Type	BYTE	ENUM	The error-correction scheme supported by this cache component; see 7.8.3.
11h	2.1+	System Cache Type	BYTE	ENUM	The logical type of cache; see 7.8.4.
12h	2.1+	Associativity	BYTE	ENUM	The associativity of the cache; see 7.8.5.

1031 **7.8.1 Cache Information — Maximum Cache Size and Installed Size**

1032 For multi-core processors, the cache size for the different levels of the cache (L1, L2, L3) is the total
 1033 amount of cache per level per processor socket. The cache size is independent of the core count. For
 1034 example, the cache size is 2 MB for both a dual core processor with a 2 MB L3 cache shared between
 1035 the cores and a dual core processor with 1 MB L3 cache (non-shared) per core.

1036 **7.8.2 Cache Information — SRAM Type**

1037 Table 35 shows the values for the Cache Information — SRAM Type field.

1038 **Table 35 – 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

1039 **7.8.3 Cache Information — Error Correction Type**

1040 Table 36 shows the values for the Cache Information — Error Correction Type field.

1041 **Table 36 – 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

1042 **7.8.4 Cache Information — System Cache Type**

1043 Table 37 shows the values for the Cache Information — System Cache Type field.

1044 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1045 **Table 37 – Cache Information: System Cache Type Field**

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

05h	Unified
-----	---------

1046 7.8.5 Cache Information — Associativity

1047 Table 38 shows the values for the Cache Information — Associativity field.

1048 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1049 **Table 38 – 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

1050 7.9 Port Connector Information (Type 8)

1051 As shown in Table 39, the information in this structure defines the attributes of a system port connector
 1052 (for example, parallel, serial, keyboard, or mouse ports). The port's type and connector information are
 1053 provided. One structure is present for each port provided by the system.

1054 **Table 39 – 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	
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

Offset	Name	Length	Value	Description
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.

1055 **7.9.1 Port Information Example**

1056 The following structure shows an example where a DB-9 Pin Male connector on the System Backpanel
 1057 (COM A) is connected to the System Board through a 9-Pin Dual Inline connector (J101).

```

1058 db 8          ; Indicates Connector Type
1059 db 9h        ; Length
1060 dw ?         ; Reserved for handle
1061 db 01h       ; String 1 - Internal Reference Designation
1062 db 18h       ; 9 Pin Dual Inline
1063 db 02h       ; String 2 - External Reference Designation
1064 db 08h       ; DB-9 Pin Male
1065 db 09h       ; 16550A Compatible
1066 db 'J101',0  ; Internal reference
1067 db 'COM A',0 ; External reference
1068 db 0
    
```

1069 If an External Connector is not used (as in the case of a CD-ROM Sound connector), then the External
 1070 Reference Designator and the External Connector type should be set to zero. If an Internal Connector is
 1071 not used (as in the case of a soldered-on Parallel Port connector that extends outside of the chassis),
 1072 then the Internal Reference Designation and Connector Type should be set to zero.

1073 **7.9.2 Port Information — Connector Types**

1074 Table 40 shows the values of the bytes in the Port Information — Connector Types field.

1075 **Table 40 – 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
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

Byte Value	Meaning
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.

1076 7.9.3 Port Types

1077 Table 41 shows the values for the Port Types field.

1078

Table 41 – Port Types Field

Byte Value	Meaning
00h	None
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

Byte Value	Meaning
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 II
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

1079 **7.10 System Slots (Type 9)**

1080 As shown in Table 42, the information in this structure defines the attributes of a system slot. One
 1081 structure is provided for each slot in the system.

1082 **Table 42 – 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	

Offset	Spec. Version	Name	Length	Value	Description
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.

1083 **7.10.1 System Slots — Slot Type**

1084 Table 43 shows the values of the System Slots — Slot Type field.

1085

Table 43 – System Slots: Slot Type Field

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

Byte Value	Meaning
11h	AGP 4X
12h	PCI-X
13h	AGP 8X
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
ADh	PCI Express Gen 2 x2
A Eh	PCI Express Gen 2 x4
AFh	PCI Express Gen 2 x8
B0h	PCI Express Gen 2 x16

1086 NOTE: Slot type A5h and ABh should be used only for PCI Express slots where the physical width is identical to
 1087 the electrical width; in that case the System Slots – Slot Data Bus Width field specifies the width. Other PCI Express
 1088 slot types (A6h-AAh, ACh-B0h) should be used to describe slots where the physical width is different from the
 1089 maximum electrical width; in these cases the width indicated in this field refers to the physical width of the slot, while
 1090 electrical width is described in the System Slots – Slot Data Bus Width field.

1091 **7.10.2 System Slots — Slot Data Bus Width**

1092 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1093 Table 44 shows the values for the System Slots – Slot Data Bus Width field. Slot Data Bus Width
 1094 meanings of type “n bit” are for parallel buses such as PCI. Slot Data Bus Width meanings of type “nx or
 1095 xn” are for serial buses such as PCI Express.

1096 NOTE: For PCI Express, width refers to the maximum supported electrical width of the “data bus”; physical slot width
 1097 is described in System Slots – Slot Type, and the actual link width resulting from PCI Express link training can be
 1098 read from configuration space.

1099 **Table 44 – System Slots: Slot Data Bus Width Field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	8 bit
04h	16 bit
05h	32 bit

Byte Value	Meaning
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

1100 7.10.3 System Slots — Current Usage

1101 Table 45 shows the values of the System Slots — Current Usage field.

1102 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1103 **Table 45 – System Slots: Current Usage Field**

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

1104 7.10.4 System Slots — Slot Length

1105 Table 46 shows the values of the System Slots — Slot Length field.

1106 **Table 46 – System Slots: Slot Length Field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Short Length
04h	Long Length

1107 7.10.5 System Slots — Slot ID

1108 The *Slot ID* field of the System Slot structure provides a mechanism to correlate the physical attributes of
 1109 the slot to its logical access method (which varies based on the *Slot Type* field). The *Slot ID* field has
 1110 meaning only for the slot types described in Table 47.

1111

Table 47 – 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.
PCMCIA	Identifies the Adapter Number (offset 09h) and Socket Number (offset 0Ah) to be passed to PCMCIA Socket Services to identify this slot

1112 **7.10.6 Slot Characteristics 1**

1113 Table 48 shows the values for the Slot Characteristics 1 field.

1114

Table 48 – 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.

1115 **7.10.7 Slot Characteristics 2**

1116 Table 49 shows the values for the Slot Characteristics 2 field.

1117

Table 49 – 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

1118 7.10.8 Segment Group Number, Bus Number, Device/Function Number

1119 For slots that are not of types PCI, AGP, PCI-X, or PCI-Express that do not have bus/device/function
1120 information, 0FFh should be populated in the fields of *Segment Group Number*, *Bus Number*,
1121 *Device/Function Number*.

1122 *Segment Group Number* is defined in the [PCI Firmware Specification](#). The value is 0 for a single-segment
1123 topology.

1124 7.11 On Board Devices Information (Type 10, Obsolete)

1125 NOTE: This structure is obsolete starting with version 2.6 of this specification; the *Onboard Devices Extended*
1126 *Information* (Type 41) structure should be used instead (see 7.42). BIOS providers can choose to implement both
1127 types to allow existing SMBIOS browsers to properly display the system's onboard devices information.

1128 The information in this structure (shown in Table 50) defines the attributes of devices that are onboard
1129 (soldered onto) a system element, usually the baseboard. In general, an entry in this table implies that the
1130 BIOS has some level of control over the enabling of the associated device for use by the system.

1131 NOTE: Because this structure was originally defined with the *Length* implicitly defining the number of devices
1132 present, no further fields can be added to this structure without adversely affecting existing software's ability to
1133 properly parse the data. Thus, if additional fields are required for this structure type a brand new structure must be
1134 defined to add a device count field, carry over the existing fields, and add the new information.

1135 **Table 50 – 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 $4 + 2 * (\text{Number of Devices})$. The user of this structure determines the number of devices as $(\text{Length} - 4) / 2$.
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

1136 NOTE: There may be a single structure instance containing the information for all onboard devices, or there may
1137 be a unique structure instance for each onboard device.

1138 7.11.1 Onboard Device Types

1139 Table 51 shows what the bytes mean for the Onboard Device Types field.

1140 **Table 51 – Onboard Device Types**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Video
04h	SCSI Controller
05h	Ethernet
06h	Token Ring

Byte Value	Meaning
07h	Sound
08h	PATA Controller
09h	SATA Controller
0Ah	SAS Controller

1141 **7.12 OEM Strings (Type 11)**

1142 This structure (shown in Table 52) contains free-form strings defined by the OEM. Examples of this are
 1143 part numbers for system reference documents, contact information for the manufacturer, etc.

1144 **Table 52 – OEM Strings (Type 11) Structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	11	OEM Strings Indicator
01h	Length	BYTE	5h	
02h	Handle	WORD	Varies	
04h	Count	BYTE	Varies	Number of strings

1145 **7.13 System Configuration Options (Type 12)**

1146 This structure (shown in Table 53) contains information required to configure the baseboard's Jumpers
 1147 and Switches.

1148 EXAMPLES:

1149 "JP2: 1-2 Cache Size is 256K, 2-3 Cache Size is 512K"
 1150 "SW1-1: Close to Disable On Board Video"

1151 **Table 53 – 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

1152 **7.14 BIOS Language Information (Type 13)**

1153 The information in this structure (shown in Table 54) defines the installable language attributes of the
 1154 BIOS.

1155 **Table 54 – 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	

Offset	Spec. Version	Name	Length	Value	Description
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

1156 The strings describing the languages follow the *Current Language* byte. The format of the strings
1157 depends on the value present in bit 0 of the byte at offset 05h in the structure.

1158 • If the bit is 0, each language string is in the form "ISO 639-1 Language Name | ISO 3166-1-alpha-
1159 2 Territory Name | Encoding Method". See Example 1 below.

1160 • If the bit is 1, each language string consists of the two-character ISO 639-1 Language Name
1161 directly followed by the two-character ISO 3166-1-alpha-2 Territory Name. See Example 2 below.

1162 NOTE: Refer to ISO 639-1 and ISO 3166-1 for additional information.

1163 EXAMPLE 1: BIOS Language Information (Long Format)

```

1164 db 13          ; language information
1165 db 16h        ; length
1166 dw ??        ; handle
1167 db 3          ; three languages available
1168 db 0          ; use long-format for language strings
1169 db 15 dup (0) ; reserved
1170 db 2          ; current language is French Canadian
1171 db 'en|US|iso8859-1',0 ; language 1 is US English
1172 db 'fr|CA|iso8859-1',0 ; language 2 is French Canadian
1173 db 'ja|JP|unicode',0 ; language 3 is Japanese
1174 db 0          ; Structure termination

```

1175 EXAMPLE 2: BIOS Language Information (Abbreviated Format)

```

1176 db 13          ; language information
1177 db 16h        ; length
1178 dw ??        ; handle
1179 db 3          ; three languages available
1180 db 01h       ; use abbreviated format for language strings
1181 db 15 dup (0) ; reserved
1182 db 2          ; current language is French Canadian
1183 db 'enUS',0   ; language 1 is US English
1184 db 'frCA',0   ; language 2 is French Canadian

```

```
1185 db 'jaJP',0 ; language 3 is Japanese
1186 db 0 ; Structure termination
```

1187 **7.15 Group Associations (Type 14)**

1188 Table 55 shows the values for the Group Associations (Type 14) structure.

1189 NOTE: Since this structure was originally defined with the Length implicitly defining the number of items present,
 1190 no further fields can be added to this structure without adversely affecting existing software's ability to properly parse
 1191 the data. Thus, if additional fields are required for this structure type a brand new structure must be defined to add an
 1192 item count field, carry over the existing fields, and add the new information.

1193 **Table 55 – Group Associations (Type 14) Structure**

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

1194 The Group Associations structure is provided for OEMs who want to specify the arrangement or hierarchy
 1195 of certain components (including other Group Associations) within the system. For example, you can use
 1196 the Group Associations structure to indicate that two CPUs share a common external cache system.
 1197 These structures might look similar to the examples shown in Example 1 and Example 2.

1198 **EXAMPLE 1: First Group Association Structure**

```
1199 db 14 ; Group Association structure
1200 db 11 ; Length
1201 dw 28h ; Handle
1202 db 01h ; String Number (First String)
1203 db 04 ; CPU Structure
1204 dw 08h ; CPU Structure's Handle
1205 db 07 ; Cache Structure
1206 dw 09h ; Cache Structure's Handle
1207 db 'Primary CPU Module', 0
1208 db 0
```

1209 **EXAMPLE 2: Second Group Association Structure**

```
1210 db 14 ; Group Association structure
1211 db 11 ; Length
1212 dw 29h ; Handle
1213 db 01h ; String Number (First String)
1214 db 04 ; CPU Structure
1215 dw 0Ah ; CPU Structure's Handle
1216 db 07 ; Cache Structure
1217 dw 09h ; Cache Structure's Handle
```

```
1218 db `Secondary CPU Module`, 0
1219 db 0
```

1220 In the examples above, CPU structures 08h and 0Ah are associated with the same cache, 09h. This
1221 relationship could also be specified as a single group, as shown in Example 3.

1222 EXAMPLE 3:

```
1223 db 14 ; Group Association structure
1224 db 14 ; Length (5 + 3 * 3)
1225 dw 28h; Structure handle for Group Association
1226 db 1 ; String Number (First string)
1227 db 4 ; 1st CPU
1228 dw 08h; CPU structure handle
1229 db 4 ; 2nd CPU
1230 dw 0Ah; CPU structure handle
1231 db 7 ; Shared cache
1232 dw 09h; Cache structure handle
1233 db `Dual-Processor CPU Complex`, 0
1234 db 0
```

1235 7.16 System Event Log (Type 15)

1236 The presence of this structure within the SMBIOS data returned for a system indicates that the system
1237 supports an event log. See Table 56 for details. An event log is a fixed-length area within a non-volatile
1238 storage element, starting with a fixed-length (and vendor-specific) header record, followed by one or more
1239 variable-length log records. See 7.16.4 for more information.

1240 An application can implement event-log change notification by periodically reading the System Event Log
1241 structure (by its assigned handle) and looking for a change in the *Log Change Token*. This token uniquely
1242 identifies the last time the event log was updated. When it sees the token changed, the application can
1243 retrieve the entire event log and determine the changes since the last time it read the event log.

1244 **Table 56 – 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	The handle, or instance number, associated with the structure
04h	2.0+	Log Area Length	WORD	Varies	The 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.

Offset	Spec. Version	Name	Length	Value	Description
08h	2.0+	Log Data Start Offset	WORD	Varies	<p>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.</p> <p>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.</p>
0Ah	2.0+	Access Method	BYTE	Varies	<p>Defines the Location and Method used by higher-level software to access the log area, one of:</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>03h Memory-mapped physical 32-bit address. The Access Method Address field contains the 4-byte (Intel DWORD format) starting physical address.</p> <p>04h Available through General-Purpose NonVolatile Data functions.</p> <p>The Access Method Address field contains the 2-byte (Intel WORD format) GPNV handle.</p> <p>05h-7Fh Available for future assignment by this specification</p> <p>80h-FFh BIOS Vendor/OEM-specific</p>
0Bh	2.0+	Log Status ^[1]	BYTE	Varies	<p>This bit-field describes the current status of the system event-log:</p> <p>Bits 7:2 Reserved, set to 0s</p> <p>Bit 1 Log area full, if 1</p> <p>Bit 0 Log area valid, if 1</p>
0Ch	2.0+	Log Change Token	DWORD	Varies	<p>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.</p>
10h	2.0+	Access Method Address	DWORD	Varies	<p>The 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-</p>

Offset	Spec. Version	Name	Length	Value	Description
					packed 'C' union: union { struct { short IndexAddr; short DataAddr; } IO; long PhysicalAddr32; short GPNVHandle; } AccessMethodAddress;
14h	2.1+	Log Header Format	BYTE	ENUM	Identifies the 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	Identifies the 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	Contains a list of Event Log Type Descriptors (see 7.16.1), as long as the value specified in offset 15h is non-zero.
^[1] The <i>Log Status</i> and <i>Log Change Token</i> fields might not be up-to-date (dynamic) when the structure is accessed using the table interface.					

1245 7.16.1 Supported Event Log Type Descriptors

1246 Each entry consists of a 1-byte type field and a 1-byte data-format descriptor, as shown in Table 57. The
 1247 presence of an entry identifies that the Log Type is supported by the system and the format of any
 1248 variable data that accompanies the first bytes of the log's variable data — a specific log record might
 1249 have more variable data than specified by its Variable Data Format Type.

1250 **Table 57 – 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.

1251 7.16.2 Indexed I/O Access Method

1252 This clause contains examples (in x86 assembly language) that detail the code required to access the
1253 “indexed I/O” event-log information.

1254 7.16.2.1 One 8-bit Index, One 8-bit Data (00h)

1255 To access the event-log, the caller selects 1 of 256 unique data bytes by

- 1256 1) Writing the byte data-selection value (index) to the *IndexAddr* I/O address
- 1257 2) Reading or writing the byte data value to (or from) the *DataAddr* I/O address

```
1258 mov dx, IndexAddr ;Value from event-log structure
1259 mov al, WhichLoc ;Identify offset to be accessed
1260 out dx, al
1261 mov dx, DataAddr ;Value from event-log structure
1262 in al, dx ;Read current value
```

1263 7.16.2.2 Two 8-bit Index, One 8-bit Data (01h)

1264 To access the event-log, the caller selects 1 of 65536 unique data bytes by

- 1265 1) Writing the least-significant byte data-selection value (index) to the *IndexAddr* I/O address
- 1266 2) Writing the most-significant byte data-selection value (index) to the (*IndexAddr+1*) I/O address
- 1267 3) Reading or writing the byte data value to (or from) the *DataAddr* I/O address

```
1268 mov dx, IndexAddr ;Value from event-log structure
1269 mov ax, WhichLoc ;Identify offset to be accessed
1270 out dx, al ;Select LSB offset
1271 inc dx
1272 xchg ah, al
1273 out dx, al ;Select MSB offset
1274 mov dx, DataAddr ;Value from event-log structure
1275 in al, dx ;Read current value
```

1276 7.16.2.3 One 16-bit Index, One 8-bit Data (02h)

1277 To access the event-log, the caller selects 1 of 65536 unique data bytes by

- 1278 1) Writing the word data-selection value (index) to the *IndexAddr* I/O address
- 1279 2) Reading or writing the byte data value to (or from) the *DataAddr* I/O address

```
1280 mov dx, IndexAddr ;Value from event-log structure
1281 mov ax, WhichLoc ;Identify offset to be accessed
1282 out dx, ax
1283 mov dx, DataAddr ;Value from event-log structure
1284 in al, dx ;Read current value
```

1285 7.16.3 Access Method Address — DWORD Layout

1286 Table 58 shows the DWORD layout of the Access Method Address.

1287

Table 58 – 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

1288 **7.16.4 Event Log Organization**

1289 The event log is organized as an optional (and implementation-specific) fixed-length header, followed by
 1290 one or more variable-length event records, as illustrated in Table 59. From one implementation to the
 1291 next, the format of the log header and the size of the overall log area might change; all other required
 1292 fields of the event log area are consistent across all systems.

1293

Table 59 – 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

1294 **7.16.5 Log Header Format**

1295 Table 60 contains the byte enumeration values (available for SMBIOS 2.1 and later) that identify the
 1296 standard formats of the event log headers.

1297

Table 60 – 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

1298 **7.16.5.1 Log Header Type 1 Format**

1299 The type 1 event log header consists of the following fields shown in Table 61.

1300

Table 61 – 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	The 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	The 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	Identifies the 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.
08h	Pre-boot Event Log Reset — CMOS Bit Index	BYTE	Varies	Identifies the 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	Identifies the 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	Identifies the 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	Identifies the 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	Identifies the version of Type 1 header implemented

1301 The Type 1 Log Header also provides pre-boot event log reset support. Application software can set a
 1302 system-specific location of CMOS RAM memory (accessible through I/O ports 70h and 71h) to cause the
 1303 event log to be cleared by the BIOS on the next reboot of the system.

1304 To perform the field setting, application software follows these steps, as long as the *Pre-boot Event Log*
 1305 *Reset – CMOS Address* field of the header is non-zero:

- 1306 • Read the address specified by *Pre-boot Event Log Reset — CMOS Address* from CMOS RAM.
 1307 Set the bit specified by the *CMOS Bit Index* field to 1. Rewrite the CMOS RAM address with the
 1308 updated data.
- 1309 • If the *CMOS Checksum – Starting Offset* field is non-zero, recalculate the CMOS RAM
 1310 checksum value for the range starting at the *Starting Offset* field for *Byte Count* bytes into a 2-
 1311 byte value. Subtract that value from 0 to create the checksum value for the range and store that
 1312 2-byte value into the CMOS RAM; the least-significant byte of the value is stored at the CMOS
 1313 RAM *Checksum Offset* and the most-significant byte of the value is stored at (*Checksum*
 1314 *Offset*)+1.

1315 **7.16.6 Log Record Format**

1316 Each log record consists of a *required* fixed-length record header, followed by (optional) additional data
 1317 that is defined by the event type. The fixed-length log record header is present as the first eight bytes of
 1318 each log record, regardless of event type. Details are shown in Table 62.

1319

Table 62 – 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
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	These fields contain 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	This field contains the (optional) event-specific additional status information.

1320 **7.16.6.1 Event Log Types**

1321 Table 63 shows the values for Event Log Types.

1322

Table 63 – 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

Value	Description
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
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.

1323 **7.16.6.2 Event Log Variable Data Format Type**

1324 The Variable Data Format Type, specified in the Event Log structure's Supported Event Type fields,
 1325 identifies the standard format that application software can apply to the first *n* bytes of the associated Log
 1326 Type's variable data. Additional OEM-specific data might follow in the log's variable data field. Table 64
 1327 shows the values for this field.

1328 **Table 64 – 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

1329 7.16.6.3 Multiple-Event Counter

1330 Some system events can be persistent; after they occur, it is possible to quickly fill the log with redundant
1331 multiple logs. The Multiple Event Count Increment (*MECI*) and Multiple Event Time Window (*METW*)
1332 values can be used to reduce the occurrence of these multiple logs while providing multiple event counts.

1333 NOTE: These values are normally specified within the event log header; see 7.16.5.1 for an example. If the values
1334 are not specified in the header, the application software can assume that the *MECI* value is 1 and the *METW* value is
1335 60 (minutes).

1336 The multiple-event counter is a DWORD (32-bit) value that tracks the number of logs of the same type
1337 that have occurred within *METW* minutes. The counter value is initialized (in the log entry) to FFFFFFFFh,
1338 implying that only a single event of that type has been detected, and the internal BIOS counter² specific
1339 to that log type is reset to 0. The counter is incremented by setting its next non-zero bit to zero; this allows
1340 counting up to 33 events. When the counter reaches 00000000h, it is full.

1341 EXAMPLE: if the current counter value is FFFFFFFCh (meaning a count of 3 events), it is incremented to
1342 FFFFFFF8h (meaning a count of 4).

1343 When the BIOS receives the next event of that type, it increments its internal counter and checks to see
1344 what recording of the error is to be performed:

- 1345 1) if the date/time of the original log entry is outside of *METW* minutes: a new log entry is written,
1346 and the internal BIOS counter is reset to 0;
- 1347 2) if the log's current multiple-event counter is 00000000h or if the internal BIOS counter is less
1348 than the *MECI* value: no recording happens (other than the internal counter increment);
- 1349 3) Otherwise: The next non-zero bit of the multiple-event counter is set to 0.

1350 7.16.6.4 POST Results Bitmap

1351 This variable data type, when present, is expected to be associated with the POST Error (08h) event log
1352 type and identifies that one or more error types have occurred. The bitmap consists of two DWORD
1353 values, described in Table 65. Any bit within the DWORD pair that is specified as Reserved is set to 0
1354 within the log data and is available for assignment by this specification. A set bit ('1'b) at a DWORD bit
1355 position implies that the error associated with that position has occurred.

1356

Table 65 – POST Results Bitmap

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

² All BIOS counters that support the Multiple-Event Counters are reset to zero each time the system boots.

Bit Position	First DWORD	Second DWORD
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
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

1357 **7.16.6.5 System Management Types**

1358 Table 66 defines the system management types present in an event log record's variable data. In
 1359 general, each type is associated with a management event that occurred within the system.

1360 **Table 66 – System Management Types**

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

Value	Name
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
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

1361 7.17 Physical Memory Array (Type 16)

1362 This structure describes a collection of memory devices that operate together to form a memory address
1363 space. Table 67 provides the details.

1364 **Table 67 – 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	0Fh	Length of the structure
02h	2.1+	Handle	WORD	Varies	The handle, or instance number, associated with the structure
04h	2.1+	Location	BYTE	ENUM	The 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	The function for which the array is used. See 7.17.2 for definitions.
06h	2.1+	Memory Error Correction	BYTE	ENUM	The 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	The 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	The 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	The number of slots or sockets available for Memory Devices in this array. This value represents the number of Memory Device structures that comprise this Memory Array. Each Memory Device has a reference to the “owning” Memory Array.
0Fh	2.7+	Extended Maximum Capacity	QWORD	Varies	The 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.

1365 **7.17.1 Memory Array — Location**

1366 Table 68 shows what the byte values mean for the Memory Array — Location field.

1367 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1368

Table 68 – 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

1369 **7.17.2 Memory Array — Use**

1370 Table 69 shows what the byte values mean for the Memory Array — Use field.

1371 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1372 **Table 69 – 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

1373 **7.17.3 Memory Array — Error Correction Types**

1374 Table 70 shows what the byte values mean for the Memory Array — Error Correction Types field.

1375 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1376 **Table 70 – 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

1377 **7.18 Memory Device (Type 17)**1378 This structure describes a single memory device that is part of a larger Physical Memory Array (Type 16)
1379 structure. See 7.17 for more details.

1380 Table 71 provides information about the Memory Device (Type 17) Structure.

1381 NOTE: If a system includes memory-device sockets, the SMBIOS implementation includes a *Memory Device*
1382 structure instance for each slot whether or not the socket is currently populated.1383 **Table 71 – 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, a minimum of 15h
02h	2.1+	Handle	WORD	Varies	The handle, or instance number, associated with the structure
04h	2.1+	Physical Memory Array Handle	WORD	Varies	The handle, or instance number, associated with the Physical Memory Array to which this device belongs
06h	2.1+	Memory Error Information Handle	WORD	Varies	The 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	The 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	The 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	The 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	The 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.
10h	2.1+	Device Locator	BYTE	STRING	The string number of the string that identifies the physically-labeled socket or board position where the memory device is located EXAMPLE: "SIMM 3"

Offset	Spec. Version	Name	Length	Value	Description
11h	2.1+	Bank Locator	BYTE	STRING	The 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	The 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	The 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)

1384 **7.18.1 Memory Device — Form Factor**

1385 Table 72 shows what the byte values mean for the Memory Device — Form Factor field.

1386 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1387

Table 72 – Memory Device: Form Factor Field

Byte Value	Meaning
01h	Other
02h	Unknown
03h	SIMM

Byte Value	Meaning
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

1388 **7.18.2 Memory Device — Type**

1389 Table 73 shows what the byte values mean for the Memory Device — Type field.

1390 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1391

Table 73 – 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	FEPRM
0Ch	EPRM
0Dh	CDRAM
0Eh	3DRAM
0Fh	SDRAM
10h	SGRAM
11h	RDRAM
12h	DDR
13h	DDR2
14h	DDR2 FB-DIMM

Byte Value	Meaning
15h-17h	Reserved
18h	DDR3
19h	FBD2

1392 7.18.3 Memory Device — Type Detail

1393 Table 74 shows what the word bit positions mean for the Memory Device — Type Detail field.

1394 NOTE: Multiple bits are set if more than one attribute applies.

1395 **Table 74 – 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)
Bit 14	Unbuffered (Unregistered)
Bit 15	Reserved, set to 0.

1396 7.18.4 Memory Device — Extended Size

1397 The *Extended Size* field is intended to represent memory devices larger than 32,767 MB (32 GB - 1 MB),
 1398 which cannot be described using the *Size* field. This field is only meaningful if the value in the *Size* field is
 1399 7FFFh. For compatibility with older SMBIOS parsers, memory devices smaller than (32 GB - 1 MB)
 1400 should be represented using their size in the *Size* field, leaving the *Extended Size* field set to 0.

1401 Bit 31 is reserved for future use and must be set to 0.

1402 Bits 30:0 represent the size of the memory device in megabytes.

1403 EXAMPLE: 0000_8000h indicates a 32 GB memory device (32,768 MB), 0002_0000h represents a 128 GB memory
 1404 device (131,072 MB), and 0000_7FFFh represents a 32,767 MB (32 GB - 1 MB) device.

1405 **7.19 32-Bit Memory Error Information (Type 18)**

1406 This structure identifies the specifics of an error that might be detected within a Physical Memory Array.
 1407 Table 75 shows the details for this structure.

1408 **Table 75 – 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	The handle, or instance number, associated with the structure
04h	2.1+	Error Type	BYTE	ENUM	The 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	Identifies the 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	The memory access operation that caused the error. See 7.19.3 for definitions.
07h	2.1+	Vendor Syndrome	DWORD	Varies	The 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	The 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.
0Fh	2.1+	Device Error Address	DWORD	Varies	The 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	The 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.

1409 **7.19.1 Memory Error — Error Type**

1410 Table 76 shows what the byte values mean for the Memory Error — Error Type field.

1411 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1412 **Table 76 – Memory Error: Error Type Field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	OK
04h	Bad read

Byte Value	Meaning
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

1413 7.19.2 Memory Error — Error Granularity

1414 Table 77 shows what the byte values mean for the Memory Error — Error Granularity field.

1415 **Table 77 – Memory Error: Error Granularity Field**

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

1416 7.19.3 Memory Error — Error Operation

1417 Table 78 shows what the byte values mean for the Memory Error — Error Operation field.

1418 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1419 **Table 78 – Memory Error: Error Operation Field**

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

1420 7.20 Memory Array Mapped Address (Type 19)

1421 This structure provides the address mapping for a Physical Memory Array. Details are provided in Table
1422 79.

1423 One structure is present for each contiguous address range described.

1424 See 7.17, 7.18, and 7.21 for more information.

1425

Table 79 – 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	0Fh	Length of the structure
02h	2.1+	Handle	WORD	Varies	The handle, or instance number, associated with the structure
04h	2.1+	Starting Address	DWORD	Varies	The 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	The 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.
0Ch	2.1+	Memory Array Handle	WORD	Varies	The 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	Identifies the 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	The 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	The 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.

1426 **7.21 Memory Device Mapped Address (Type 20)**

1427 This structure maps memory address space usually to a device-level granularity. Details are provided in
 1428 Table 80.

1429 One structure is present for each contiguous address range described.

1430 NOTE: A Memory Device Mapped Address structure is provided only if a Memory Device has a mapped address;
1431 there is no provision within this structure to map a zero-length address space.

1432 See 7.17, 7.18, and 7.21 for more information.

1433 **Table 80 – 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	13h	Length of the structure
02h	2.1+	Handle	WORD	Varies	The handle, or instance number, associated with the structure
04h	2.1+	Starting Address	DWORD	Varies	The 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	The 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	The 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	The 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	Identifies the 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	The 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.

Offset	Spec. Version	Name	Length	Value	Description
12h	2.1+	Interleaved Data Depth	BYTE	Varies	The 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.
13h	2.7+	Extended Starting Address	QWORD	Varies	The 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	The 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.

1434 **7.22 Built-in Pointing Device (Type 21)**

1435 This structure describes the attributes of the built-in pointing device for the system. Details are provided in
1436 Table 81.

1437 The presence of this structure does not imply that the built-in pointing device is active for the system's
1438 use.

1439 **Table 81 – 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	The handle, or instance number, associated with the structure
04h	2.1+	Type	BYTE	ENUM	The type of pointing device; see 7.22.1.
05h	2.1+	Interface	BYTE	ENUM	The interface type for the pointing device; see 7.22.2.
06h	2.1+	Number of Buttons	BYTE	Varies	The number of buttons on the pointing device. If the device has three buttons, the field value is 03h.

1440 **7.22.1 Pointing Device — Type**

1441 Table 82 shows what the byte values mean for the Pointing Device — Type field.

1442 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1443 **Table 82 – Pointing Device: Type Field**

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Mouse
04h	Track Ball
05h	Track Point
06h	Glide Point
07h	Touch Pad
08h	Touch Screen
09h	Optical Sensor

1444 **7.22.2 Pointing Device — Interface**

1445 Table 83 shows what the byte values mean for the Pointing Device — Interface field.

1446 **Table 83 – 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

1447 **7.23 Portable Battery (Type 22)**

1448 This structure describes the attributes of the portable battery or batteries for the system. The structure
 1449 contains the static attributes for the group. Each structure describes a single battery pack's attributes.
 1450 Details are provided in Table 84.

1451

Table 84 – 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	The handle, or instance number, associated with the structure
04h	2.1+	Location	BYTE	STRING	The 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	The number of the string that names the company that manufactured the battery
06h	2.1+	Manufacture Date	BYTE	STRING	The 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	The 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	The 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	The 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	The design voltage of the battery in mVolts. If the value is unknown, the field contains 0.
0Eh	2.1+	SBDS Version Number	BYTE	STRING	The 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	The 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.

Offset	Spec. Version	Name	Length	Value	Description
10h	2.2+	SBDS Serial Number	WORD	Varies	The 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.
12h	2.2+	SBDS Manufacture Date	WORD	Varies	The 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	The 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	The 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

1452 7.23.1 Portable Battery — Device Chemistry

1453 Table 85 shows what the byte values mean for the Portable Battery — Device Chemistry field.

1454 NOTE: Refer to 6.3 for the CIM properties associated with this enumerated value.

1455 **Table 85 – 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

1456 **7.24 System Reset (Type 23)**

1457 This structure describes whether Automatic System Reset functions enabled (*Status*). Details are
 1458 provided in Table 86.

1459 If the system has a watchdog timer and the timer is not reset (*Timer Reset*) before the *Interval* elapses,
 1460 an automatic system reset occurs. The system re-boots according to the *Boot Option*. This function may
 1461 repeat until the *Limit* is reached, at which time the system re-boots according to the *Boot Option at Limit*.

1462 NOTE: This structure type was added for version 2.2 of this specification.

1463 **Table 86 – 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	The 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	The number of automatic system resets since the last intentional reset. A value of 0FFFFh indicates unknown.
07h	Reset Limit	WORD	Varies	The number of consecutive times the system reset is attempted. A value of 0FFFFh indicates unknown.
09h	Timer Interval	WORD	Varies	The 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	Identifies the 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.

1464 **7.25 Hardware Security (Type 24)**

1465 This structure describes the system-wide hardware security settings. Details are provided in Table 87.

1466 NOTE: This structure type was added in version 2.2 of this specification.

1467 **Table 87 – 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	The 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

1468 **7.26 System Power Controls (Type 25)**1469 This structure describes the attributes for controlling the main power supply to the system. Details are
1470 provided in Table 88.1471 Software that interprets this structure uses the month, day, hour, minute, and second values to determine
1472 the number of seconds until the next power-on of the system. The presence of this structure implies that a
1473 timed power-on facility is available for the system.

1474 NOTE: This structure type was added in version 2.2 of the specification.

1475

Table 88 – System Power Controls (Type 25) Structure

Offset	Name	Length	Value	Description
00h	Type	BYTE	25	System Power Controls indicator
01h	Length	BYTE	09h	Length of the structure
02h	Handle	WORD	Varies	The handle, or instance number, associated with the structure
04h	Next Scheduled Power-on Month	BYTE	Varies	Contains the 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	Contains the 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	Contains the 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	Contains the 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	Contains the 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.

1476 **7.26.1 System Power Controls — Calculating the Next Scheduled Power-on Time**

1477 The DMTF *System Power Controls* group contains a *Next Scheduled Power-on Time*, specified as the
 1478 number of seconds until the next scheduled power-on of the system. Management software uses the date
 1479 and time information specified in the associated SMBIOS structure to calculate the total number of
 1480 seconds.

1481 Any date or time field in the structure whose value is outside of the field’s specified range does not
 1482 contribute to the total-seconds count. For example, if the Month field contains the value 0xFF the next
 1483 power-on is scheduled to fall within the next month, perhaps on a specific day-of-month and time.

1484 **7.27 Voltage Probe (Type 26)**

1485 This describes the attributes for a voltage probe in the system. Each structure describes a single voltage
 1486 probe. Details are provided in Table 89.

1487 NOTE: This structure type was added in version 2.2 of this specification.

1488 **Table 89 – 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	The handle, or instance number, associated with the structure
04h	Description	BYTE	STRING	The number of the string that contains additional descriptive information about the probe or its location

Offset	Name	Length	Value	Description
05h	Location and Status	BYTE	Bit-field	Defines the probe's physical location and status of the voltage monitored by this voltage probe. See 7.27.1.
06h	Maximum Value	WORD	Varies	The 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	The minimum voltage level readable by this probe, in millivolts. If the value is unknown, the field is set to 0x8000.
0Ah	Resolution	WORD	Varies	The 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	The 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	The 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	Contains OEM- or BIOS vendor-specific information.
14h	Nominal Value	WORD	Varies	The 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.

1489 **7.27.1 Voltage Probe — Location and Status**

1490 Table 90 provides details about the Location and Status fields.

1491 **Table 90 – 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

Bit Range	Field Name	Value	Meaning
		...01001	Processor Module
		...01010	Power Unit
		...01011	Add-in Card

1492 **7.28 Cooling Device (Type 27)**

1493 This structure describes the attributes for a cooling device in the system. Each structure describes a
 1494 single cooling device. Details are provided in Table 91.

1495 NOTE: This structure type was added in version 2.2 of this specification.

1496 **Table 91 – 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	The handle, or instance number, associated with the structure
04h	2.2+	Temperature Probe Handle	WORD	Varies	The 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	Identifies the cooling device type and the status of this cooling device; see 7.28.1.
07h	2.2+	Cooling Unit Group	BYTE	Varies	Identifies the 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	Contains OEM- or BIOS vendor-specific information.
0Ch	2.2+	Nominal Speed	WORD	Varies	The 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	The 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.

1497 **7.28.1 Cooling Device — Device Type and Status**

1498 Table 92 provides details about the Device Type and Status fields.

1499 **Table 92 – 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

1500 **7.29 Temperature Probe (Type 28)**1501 This structure describes the attributes for a temperature probe in the system. Each structure describes a
1502 single temperature probe. Table 93 provides details.

1503 NOTE: This structure type was added in version 2.2 of this specification.

1504 **Table 93 – 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	The handle, or instance number, associated with the structure
04h	Description	BYTE	STRING	The number of the string that contains additional descriptive information about the probe or its location
05h	Location and Status	BYTE	Bit-field	Defines the probe's physical location and the status of the temperature monitored by this temperature probe. See 7.29.1.
06h	Maximum Value	WORD	Varies	The maximum 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
08h	Minimum Value	WORD	Varies	The minimum temperature readable by this probe, in 1/10 th degrees C. If the value is unknown, the field is set to 0x8000.
0Ah	Resolution	WORD	Varies	The resolution for the probe's reading, in 1/1000 th degrees C. If the value is unknown, the field is set to 0x8000.
0Ch	Tolerance	WORD	Varies	The tolerance for reading from this probe, in plus/minus 1/10 th degrees C. If the value is unknown, the field is set to 0x8000.
0Eh	Accuracy	WORD	Varies	The 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	Contains OEM- or BIOS vendor-specific information
14h	Nominal Value	WORD	Varies	The nominal value for the probe's reading in 1/10 th 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.

1505 **7.29.1 Temperature Probe — Location and Status**

1506 Table 94 provides details about the Location and Status fields.

1507 NOTE: Refer to 6.3 for the CIM properties associated with these enumerated values.

1508 **Table 94 – 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

1509 **7.30 Electrical Current Probe (Type 29)**

1510 This structure describes the attributes for an electrical current probe in the system. Each structure
 1511 describes a single electrical current probe. Table 95 provides details.

1512 NOTE: This structure type was added in version 2.2 of this specification.

1513 **Table 95 – 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	The handle, or instance number, associated with the structure
04h	Description	BYTE	STRING	The 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	The maximum current readable by this probe, in milliamps. If the value is unknown, the field is set to 0x8000.
08h	Minimum Value	WORD	Varies	The minimum current readable by this probe, in milliamps. If the value is unknown, the field is set to 0x8000.
0Ah	Resolution	WORD	Varies	The 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	The 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	The 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	Contains OEM- or BIOS vendor-specific information
14h	Nominal Value	WORD	Varies	The 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.

1514 **7.30.1 Current Probe — Location and Status**

1515 Table 96 provides details about the Location and Status fields.

1516 NOTE: Refer to 6.3 for the CIM properties associated with these enumerated values.

1517

Table 96 – 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

1518 **7.31 Out-of-Band Remote Access (Type 30)**

1519 This structure describes the attributes and policy settings of a hardware facility that may be used to gain
 1520 remote access to a hardware system when the operating system is not available due to power-down
 1521 status, hardware failures, or boot failures. Details are provided in Table 97.

1522 NOTE: This structure type was added in version 2.2 of this specification.

1523 **Table 97 – 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	The handle, or instance number, associated with the structure
04h	Manufacturer Name	BYTE	STRING	The 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	Identifies the 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

1524 7.32 Boot Integrity Services (BIS) Entry Point (Type 31)

1525 Structure type 31 (decimal) is reserved for use by the Boot Integrity Services (BIS). Refer to the [Boot Integrity Services API Specification](#) for details.

1527 NOTE: This structure type was added in version 2.3 of this specification.

1528 7.33 System Boot Information (Type 32)

1529 The client system firmware (for example, BIOS) communicates the *System Boot Status* to the client's Pre-
 1530 boot Execution Environment (PXE) boot image or OS-present management application through this
 1531 structure. Table 98 provides details on this structure.

1532 When used in the PXE environment, for example, this code identifies the reason the PXE was initiated
 1533 and can be used by boot-image software to further automate an enterprise's PXE sessions. For example,
 1534 an enterprise could choose to automatically download a hardware-diagnostic image to a client whose
 1535 reason code indicated either a firmware- or operating system-detected hardware failure.

1536 NOTE: This structure type was added in version 2.3 of this specification.

1537 **Table 98 – 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	The Status and Additional Data fields that identify the boot status. See 7.33.1 for additional information.

1538 7.33.1 System Boot Status

1539 Table 99 provides information about system boot status.

1540

Table 99 – System Boot Status

Description	Status	Additional Data
No errors detected	0	None
No bootable media	1	None
The “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
A 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

1541 **7.34 64-Bit Memory Error Information (Type 33)**

1542 This structure describes an error within a Physical Memory Array, when the error address is above 4G
 1543 (0xFFFFFFFF). Table 100 provides details.

1544 NOTE: This structure type was added in version 2.3 of this specification.

1545

Table 100 – 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	The handle, or instance number, associated with the structure
04h	Error Type	BYTE	ENUM	The 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	Identifies the granularity (for example, device versus Partition) to which the error can be resolved. See 7.19.2 for definitions.
06h	Error Operation	BYTE	ENUM	The memory access operation that caused the error. See 7.19.3 for definitions.

Offset	Name	Length	Value	Description
07h	Vendor Syndrome	DWORD	Varies	The 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	The 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	The 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	The 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.

1546 7.35 Management Device (Type 34)

1547 The information in this structure defines the attributes of a *Management Device*. Table 101 provides
1548 details.

1549 A *Management Device* might control one or more fans or voltage, current, or temperature probes as
1550 defined by one or more *Management Device Component* structures. See 7.36 for more information.

1551 NOTE: This structure type was added in version 2.3 of this specification.

1552 **Table 101 – 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	The handle, or instance number, associated with the structure
04h	Description	BYTE	STRING	The number of the string that contains additional descriptive information about the device or its location
05h	Type	BYTE	Varies	Defines the device's type; see 7.35.1.
06h	Address	DWORD	Varies	Defines the device's address
0Ah	Address Type	BYTE	Varies	Defines the type of addressing used to access the device; see 7.35.2.

1553 7.35.1 Management Device — Type

1554 Table 102 shows what the byte values mean for the Management Device — Type field.

1555 **Table 102 – 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

1556 **7.35.2 Management Device — Address Type**

1557 Table 103 shows what the byte values mean for the Management Device — Address Type field.

1558 **Table 103 – Management Device: Address Type Field**

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

1559 **7.36 Management Device Component (Type 35)**

1560 This structure associates a cooling device or environmental probe with structures that define the
 1561 controlling hardware device and (optionally) the component’s thresholds. Table 104 provides details.

1562 NOTE: This structure type was added in version 2.3 of this specification.

1563 **Table 104 – 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	The handle, or instance number, associated with the structure
04h	Description	BYTE	STRING	The number of the string that contains additional descriptive information about the component
05h	Management Device Handle	WORD	Varies	The 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	The 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	The 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.

1564 7.37 Management Device Threshold Data (Type 36)

1565 The information in this structure defines threshold information for a component (probe or cooling-unit)
1566 contained within a *Management Device*. Table 105 provides details.

1567 For each threshold field present in the structure:

- 1568 • The threshold units (millivolts, milliamps, 1/10th degrees C, or RPMs) are as defined by the
1569 associated probe or cooling-unit component structure.
- 1570 • If the value is unavailable, the field is set to 0x8000.

1571 NOTE: This structure type was added in version 2.3 of this specification.

1572 **Table 105 – 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	The handle, or instance number, associated with the structure
04h	Lower Threshold – Non-critical	WORD	Varies	The lower non-critical threshold for this component
06h	Upper Threshold – Non-critical	WORD	Varies	The upper non-critical threshold for this component
08h	Lower Threshold – Critical	WORD	Varies	The lower critical threshold for this component
0Ah	Upper Threshold – Critical	WORD	Varies	The upper critical threshold for this component
0ch	Lower Threshold – Non-recoverable	WORD	Varies	The lower non-recoverable threshold for this component
0eh	Upper Threshold – Non-recoverable	WORD	Varies	The upper non-recoverable threshold for this component

1573 7.38 Memory Channel (Type 37)

1574 The information in this structure provides the correlation between a Memory Channel and its associated
1575 Memory Devices. Table 106 provides details.

1576 Each device presents one or more loads to the channel; the sum of all device loads cannot exceed the
1577 channel's defined maximum.

1578 NOTE: This structure type was added in version 2.3 of this specification.

1579 **Table 106 – 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	The handle, or instance number, associated with the structure.
04h	Channel Type	BYTE	Varies	Identifies the type of memory associated with the channel; see 7.38.1.
05h	Maximum Channel Load	BYTE	Varies	The maximum load supported by the channel; the sum of all device loads cannot exceed this value.
06h	Memory Device Count (n)	BYTE	Varies	Identifies the 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	The channel load provided by the first Memory Device associated with this channel.
08h	Memory Device1 Handle	WORD	Varies	The structure handle that identifies the first Memory Device associated with this channel.
7 + 3*(n-1)	Memory Device _n Load	BYTE	Varies	The channel load provided by the nth Memory Device associated with this channel.
8 + 3*(n-1)	Memory Device _n Handle	WORD	Varies	The structure handle that identifies the nth Memory Device associated with this channel.

1580 **7.38.1 Memory Channel — Channel Type**

1581 Table 107 shows what the byte values mean for the Memory Channel — Channel Type field.

1582 NOTE: Enumerated values are controlled by the DMTF, not by this specification.

1583 **Table 107 – Memory Channel: Channel Type Field**

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

1584 **7.39 IPMI Device Information (Type 38)**

1585 The information in this structure defines the attributes of an Intelligent Platform Management Interface
1586 (IPMI) Baseboard Management Controller (BMC). Table 108 provides the details about this structure.

1587 Refer to the [Intelligent Platform Management Interface \(IPMI\) Interface Specification](#) for full
1588 documentation of IPMI and additional information on the use of this structure.

1589 The Type 42 structure can also be used to describe a physical management controller host interface and
1590 one or more protocols that share that interface. If IPMI is not shared with other protocols, either the Type
1591 38 or Type 42 structures can be used. Providing Type 38 is recommended for backward compatibility.
1592 See 7.43 for additional information on Type 42.

1593 **Table 108 – IPMI Device Information (Type 38) Structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	38	IPMI Device Information structure indicator
01h	Length	BYTE		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	Identifies the 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	The 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	Identifies the 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	<p>Base Address Modifier (This field is unused and set to 00h for SSIF.)</p> <p>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.</p> <p>bit 5 – Reserved. Return as 0b. bit 4 – LS-bit for addresses: 0b = Address bit 0 = 0b 1b = Address bit 0 = 1b</p> <p>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</p>
11h	Interrupt Number	BYTE	Varies	<p>Interrupt number for IPMI System Interface</p> <p>00h = unspecified/unsupported</p>

1594 **7.39.1 IPMI Device Information — BMC Interface Type**

1595 Table 109 shows what the byte values mean for the IPMI Device Information — BMC Interface Type field.

1596 **Table 109 – 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

1597 **7.40 System Power Supply (Type 39)**

1598 This structure identifies attributes of a system power supply. Table 110 provides details. One instance of
 1599 this structure is present for each possible power supply in a system.

1600 NOTE: This structure type was added in version 2.3.1 of this specification.

1601

Table 110 – 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	The handle, or instance number, associated with the power supply structure
04h	Power Unit Group	BYTE	Varies	Identifies the 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	The 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	The number of the string that names the power supply device EXAMPLE: "DR-36"
07h	Manufacturer	BYTE	STRING	The number of the string that names the company that manufactured the supply
08h	Serial Number	BYTE	STRING	The number of the string that contains the serial number for the power supply
09h	Asset Tag Number	BYTE	STRING	The number of the string that contains the Asset Tag Number
0Ah	Model Part Number	BYTE	STRING	The 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	The 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	The 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	The 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.

1602 **7.40.1 Power Supply Characteristics**

1603 Table 111 provides information about power supply characteristics.

1604 NOTE: Refer to 6.3 for the CIM properties associated with these enumerated values.

1605 **Table 111 – 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	Power supply is unplugged from the wall, if 1.
1	Power supply is present, if 1.
0	Power supply is hot replaceable, if 1.

1606 **7.41 Additional Information (Type 40)**

1607 This structure is intended to provide additional information for handling unspecified enumerated values
 1608 and interim field updates in another structure. Table 112 provides details.

1609 NOTE: This structure type was added in version 2.6 of this specification.

1610

Table 112 – 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	The handle, or instance number, associated with the structure
04h	Number of Additional Information entries (n)	BYTE	Varies	The number of Additional Information Entries that follow
05h	Additional Information entries	Varies	Varies	Additional Information entries; see 7.41.1

1611 **7.41.1 Additional Information Entry Format**

1612 Table 113 describes an Additional Information Entry format.

1613

Table 113 – 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	The handle, or instance number, associated with the structure for which additional information is provided
03h	Referenced Offset	BYTE	Varies	The offset of the field within the structure referenced by the <i>Referenced Handle</i> for which additional information is provided
04h	String	BYTE	STRING	The number of the optional string to be associated with the field referenced by the <i>Referenced Offset</i>
05h	Value	Varies	Varies	The 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 <i>Referenced Offset</i> NOTE: This field is the same type and size as the field being referenced by this Additional Information Entry.

1614 The following guidance is given for using this structure to provide additional information for an
1615 enumerated value field, such as processor type:

- 1616 • If a value has been proposed:
 - 1617 – Set the field in the original structure to “Other”.
 - 1618 – Use the proposed value in the value field of the Additional Information Entry that
 - 1619 references the enumerated field in the original structure.
 - 1620 – The Additional Information Entry string field may also be used to uniquely describe this
 - 1621 new item (for example the CPU ID string).
- 1622 • If a value has not been proposed:
 - 1623 – The field in the original structure and the Additional Information Entry value field that
 - 1624 references it should both be set to “Other”.

- 1625 – The Additional Information Entry string field should be filled so as to uniquely describe this
- 1626 new item (for example the CPU ID string).

1627 The following guidance is given for using this structure to provide additional information for a field update:

- 1628 • If a change has been proposed:
 - 1629 – Set the field in the original structure as best as possible using only fully approved settings.
 - 1630 – Place the modified value in the value field of the Additional Information Entry that
 - 1631 references the field in the original structure.
 - 1632 – The Additional Information Entry string field may also be used to uniquely describe this
 - 1633 modification.
- 1634 • If a change has not been proposed:
 - 1635 – The field in the original structure and Additional Information Entry value field that
 - 1636 references it should both be set to the same value (the best possible value using only fully
 - 1637 approved settings).
 - 1638 – The Additional Information Entry string field should be filled so as to uniquely describe what
 - 1639 needs to be modified (for example, “XYZ capability needs to be defined”).

1640 **7.42 Onboard Devices Extended Information (Type 41)**

1641 The information in this structure defines the attributes of devices that are onboard (soldered onto) a

1642 system element, usually the baseboard. Table 114 provides details.

1643 In general, an entry in this table implies that the BIOS has some level of control over the enabling of the

1644 associated device for use by the system.

1645 NOTE: This structure replaces Onboard Device Information (Type 10) starting with version 2.6 of this specification.

1646 BIOS providers can choose to implement both types to allow existing SMBIOS browsers to properly display the

1647 system’s onboard devices information.

1648 **Table 114 – Onboard Devices Extended Information (Type 41) Structure**

Offset	Name	Length	Value	Description
00h	Type	BYTE	41	Onboard Devices Extended Information
01h	Length	BYTE	Varies	0Bh for version 2.6 and later
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.

Offset	Name	Length	Value	Description
0Ah	Device/Function Number	BYTE	Bit Field	Bits 7:3 – Device number Bits 2:0 – Function number See 7.42.4.

1649 7.42.1 Reference Designation

1650 The *Reference Designation* string is typically the silkscreen label.

1651 7.42.2 Onboard Device Types

1652 Table 115 shows what the byte values mean for the Onboard Device Types field.

1653 **Table 115 – 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

1654 7.42.3 Device Type Instance

1655 *Device Type Instance* is a unique value (within a given *onboard device type*) used to indicate the order
1656 the device is designated by the system. For example, a system with two identical Ethernet NICs may
1657 designate one NIC (with higher Bus/Device/Function=15/0/0) as the first onboard NIC (instance 1) and
1658 the other NIC (with lower Bus/Device/Function =3/0/0) as the second onboard NIC (instance 2).

1659 7.42.4 Segment Group Number, Bus Number, Device/Function Number

1660 For devices that are not of types PCI, AGP, PCI-X, or PCI-Express and that do not have
1661 bus/device/function information, 0FFh should be populated in the fields of *Segment Group Number*, *Bus*
1662 *Number*, *Device/Function Number*.

1663 *Segment Group Number* is defined in the [PCI Firmware Specification](#). The value is 0 for a single-segment
1664 topology.

1665 7.43 Management Controller Host Interface (Type 42)

1666 The information in this structure defines the attributes of a Management Controller Host Interface that is
1667 not discoverable by “Plug and Play” mechanisms. Table 116 provides details. The Type 42 structure can

1668 be used to describe a physical management controller host interface and one or more protocols that
 1669 share that interface.

1670 Type 42 should be used for management controller host interfaces that use protocols other than IPMI or
 1671 that use multiple protocols on a single host interface type.

1672 This structure should also be provided if IPMI is shared with other protocols over the same interface
 1673 hardware. If IPMI is not shared with other protocols, either the Type 38 or Type 42 structures can be
 1674 used. Providing Type 38 is recommended for backward compatibility. The structures are not required to
 1675 be mutually exclusive. Type 38 and Type 42 structures may be implemented simultaneously to provide
 1676 backward compatibility with IPMI applications or drivers that do not yet recognize the Type 42 structure.
 1677 Refer to the [Intelligent Platform Management Interface \(IPMI\) Interface Specification](#) for full
 1678 documentation of IPMI and additional information on the use of this structure with IPMI.

1679 **Table 116 – 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 then 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.

1680 After the generic structure header, this structure has the following general layout shown in Table 117.

1681 **Table 117 – 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	

1682 **7.44 Inactive (Type 126)**

1683 This structure definition supports a system implementation where the SMBIOS structure-table is a
 1684 superset of all supported system attributes and provides a standard mechanism for the system BIOS to
 1685 signal that a structure is currently inactive and should not be interpreted by the upper-level software.
 1686 Table 118 provides details.

1687 For example, a portable system might include *System Slot* structures that are reported only when the
 1688 portable is docked. An undocked system would report those structures as *Inactive*. When the system is
 1689 docked, the system-specific software would change the Type structure from *Inactive* to the *System Slot*
 1690 equivalent.

1691 Upper-level software that interprets the SMBIOS structure-table should bypass an *Inactive* structure just
 1692 as it would for a structure type that the software does not recognize.

1693 NOTE: This structure type was added in version 2.2 of this specification.

1694 **Table 118 – 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	The handle, or instance number, associated with the structure

1695 **7.45 End-of-Table (Type 127)**

1696 This structure type identifies the end of the structure table that might be earlier than the last byte within
 1697 the buffer specified by the structure. Table 119 provides details.

1698 To ensure backward compatibility with management software written to previous versions of this
 1699 specification, a system implementation should use the end-of-table indicator in a manner similar to the
 1700 *Inactive (Type 126)* structure type; the structure table is still reported as a fixed-length, and the entire
 1701 length of the table is still indexable. If the end-of-table indicator is used in the last physical structure in a
 1702 table, the field's length is encoded as 4.

1703 NOTE: This structure type was added in version 2.2 of this specification.

1704 **Table 119 – 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	The handle, or instance number, associated with the structure

1705

Annex A (informative)

Conformance Guidelines

- 1706
1707
1708
1709
- 1710 The following conformance requirements apply for SMBIOS 2.5 or later implementations.
- 1711 1. The table anchor string "_SM_" is present in the address range 0xF0000 to 0xFFFFF on a 16-byte boundary.
- 1712 2. Table entry-point verification:
- 1713 2.1 The Entry Point Length field value is at least 0x1F.
- 1714 2.2 The entry-point checksum evaluates to 0.
- 1715 2.3 The SMBIOS Version (Major.Minor) is at least 2.4.
- 1716 2.4 The Intermediate Anchor String is "_DMI_".
- 1717 2.5 The intermediate checksum evaluates to 0.
- 1718 3. The structure-table is traversable and conforms to the entry-point specifications:
- 1719 3.1 The structure-table's linked-list is traversable within the length and structure-count bounds specified by the
1720 entry-point structure.
- 1721 3.2 The overall size of the structure table is less than or equal to the Structure Table Length specified by the
1722 entry-point structure.
- 1723 3.3 Each structure's length must be at least 4 (the size of a structure header).
- 1724 3.4 No structure handle number is repeated.
- 1725 3.5 The last structure is the end-of-table (0x7F).
- 1726 3.6 The number of structures found within the table equals the Number of SMBIOS Structures field present in
1727 the entry-point.
- 1728 3.7 The maximum structure size (formatted area plus its string-pool) is less than or equal to the Maximum
1729 Structure Size specified by the entry-point.
- 1730 4. Required structures and corresponding data are present (see 6.2):
- 1731 4.1 BIOS Information (Type 0)
- 1732 4.1.1 One and only one structure of this type is present.
- 1733 4.1.2 The structure Length field is at least 18h.
- 1734 4.1.3 BIOS Version string is present and non-null.
- 1735 4.1.4 BIOS Release Date string is present, non-null, and includes a 4-digit year.
- 1736 4.1.5 BIOS Characteristics: bits 3:0 are all 0, and at least one of bits 31:4 is set to 1.
- 1737 4.2 System Information (Type 1)
- 1738 4.2.1 One and only one structure of this type is present.
- 1739 4.2.2 The structure Length field is at least 1Bh.
- 1740 4.2.3 Manufacturer string is present and non-null.
- 1741 4.2.4 Product Name string is present and non-null.
- 1742 4.2.5 UUID field is neither 00000000 00000000 nor FFFFFFFF FFFFFFFF.
- 1743 4.2.6 Wake-up Type field is neither 00h (Reserved) nor 02h (Unknown).
- 1744 4.3 System Enclosure (Type 3)

- 1745 4.3.1 One or more structures of this type is present.
- 1746 4.3.2 The structure length is at least 0Dh.
- 1747 4.3.3 The Manufacturer string is present and non-null in each structure.
- 1748 4.3.4 Type field is neither 00h (Reserved) nor 02h (Unknown).
- 1749 4.4 Processor Information (Type 4)
- 1750 4.4.1 The number of structures defines the maximum number of processors supported by the system;
1751 at least one structure with a Processor Type field of "Central Processor" must be present.
- 1752 4.4.2 Each structure's length is at least 20h.
- 1753 4.4.3 Socket Designation string is present and non-null.
- 1754 4.4.4 Processor Type field is neither 00h (Reserved) nor 02h (Unknown).
- 1755 4.4.5 (*)Processor Family field is neither 00h (Reserved) nor 02h (Unknown).
- 1756 4.4.6 (*)Processor Manufacturer string is present and non-null.
- 1757 4.4.7 Max Speed field is non-0.
- 1758 4.4.8 (*)CPU Status sub-field of the Status field is not 0 (Unknown).
- 1759 4.4.9 Processor Upgrade field is neither 00h (Reserved) nor 02h (Unknown).
- 1760 4.4.10 Lx (x=1,2,3) Cache Handle fields, if not set to 0xFFFF, reference Cache Information (Type 7)
1761 structures.
- 1762 NOTE: Fields preceded by (*) are checked only if the CPU Socket Populated sub-field of the Status field is set to
1763 "CPU Populated".
- 1764 4.5 Cache Information (Type 7)
- 1765 4.5.1 One structure is present for each external-to-the-processor cache.
- 1766 4.5.2 Each structure's Length is at least 13h.
- 1767 4.5.3 Socket Designation string is present and non-null if the cache is external to the processor
1768 (Location sub-field of Cache Configuration field is 01b).
- 1769 4.5.4 Operational Mode and Location sub-fields of the Cache Configuration field are not 11b
1770 (Unknown).
- 1771 4.6 System Slots (Type 9)
- 1772 4.6.1 One structure is present for each upgradeable system slot.
- 1773 4.6.2 Each structure's Length is at least 0Dh.
- 1774 4.6.3 Slot Designation string is present and non-null.
- 1775 4.6.4 Slot Type is neither 00h (Reserved) nor 02h (Unknown).
- 1776 4.6.5 Slot Data Bus Width is neither 00h (Reserved) or 02h (Unknown).
- 1777 4.6.6 Current Usage is not set to 00h (Reserved). If the "Slot Type" provides device presence-detect
1778 capabilities (for example, PCI or AGP), Current Usage is not set to 02h (Unknown).
- 1779 4.6.7 Slot ID is set to a meaningful value.
- 1780 4.6.8 Slot Characteristics 1, bit 0, is not set to 1.
- 1781 4.7 Physical Memory Array (Type 16)
- 1782 4.7.1 At least one structure is present with "Use" set to 03h (System memory).
- 1783 4.7.2 Each structure's length is at least 0Fh.
- 1784 4.7.3 Location is neither 00h (Reserved) nor 02h (Unknown).
- 1785 4.7.4 Use is neither 00h (Reserved) nor 02h (Unknown).
- 1786 4.7.5 Memory Error Correction is neither 00h (Reserved) nor 02h (Unknown).

1787	4.7.6	Either Maximum Capacity or Extended Maximum Capacity must be set to a known, non-zero value.
1788		
1789	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.
1790		
1791	4.8	Memory Device (Type 17)
1792	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.
1793		
1794		
1795		
1796		
1797	4.8.2	Each structure's length is at least 15h.
1798	4.8.3	Memory Array Handle references a Physical Memory Array (Type 16) structure.
1799	4.8.4	Total Width is not 0FFFFh (Unknown) if the memory device is installed. (Size is not 0.)
1800	4.8.5	Data Width is not 0FFFFh (Unknown).
1801	4.8.6	Size is not 0FFFFh (Unknown).
1802	4.8.7	Form Factor is not 00h (Reserved) or 02h (Unknown).
1803	4.8.8	Device Set is not 0FFh (Unknown).
1804	4.8.9	Device Locator string is present and non-null.
1805	4.9	Memory Array Mapped Address (Type 19)
1806	4.9.1	One structure is provided for each contiguous block of memory addresses mapped to a Physical Memory Array.
1807		
1808	4.9.2	Each structure's length is at least 0Fh.
1809	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.
1810		
1811	4.9.4	Memory Array Handle references a Physical Memory Array (Type 16).
1812	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.
1813		
1814	4.9.6	Partition Width is not 0.
1815	4.10	Boot Integrity Services (BIS) Entry Point (Type 31). This structure is optional, but if it is present the following checks are performed:
1816		
1817	4.10.1	The structure's length is at least 1Ch.
1818	4.10.2	The structure-level checksum evaluates to 00h.
1819	4.10.3	16-bit Entry Point is not 0.
1820	4.10.4	32-bit Entry Point is not 0.
1821	4.11	System Boot Information (Type 32)
1822	4.11.1	One and only one structure of this type is present.
1823	4.11.2	The structure's length is at least 0Bh.
1824		

Annex B (informative)

Using the Table Convention

1825
1826
1827
1828

1829 This clause contains pseudo-code that describes the method that application software can use to parse
1830 the table-based SMBIOS structures. The example searches for the first structure of the type specified,
1831 returning the handle of the structure found or 0xFFFF if no structure of the type was found in the list.
1832 *TableAddress* and *StructureCount* values are those previously found by locating the Table Entry Point
1833 structure in low memory.

```

1834 typedef unsigned short ushort;
1835 typedef unsigned char uchar;
1836 typedef struct
1837 {
1838     uchar Type;
1839     uchar Length;
1840     ushort Handle;
1841 } HEADER;
1842 ushort FindStructure( char *TableAddress, ushort StructureCount, uchar Type )
1843 {
1844     ushort i, handle;
1845     uchar lasttype;
1846     i = 0;
1847     handle = 0xFFFF;
1848     while( i < StructureCount && handle == 0xFFFF )
1849     {
1850         i++;
1851         lasttype = ((HEADER *)TableAddress)->Type;
1852         if( lasttype == Type )
1853         {
1854             handle = ((HEADER *)TableAddress)->Handle;
1855         } /* Found first structure of the requested type */
1856         else
1857         {
1858             TableAddress += ((HEADER *)TableAddress)->Length;
1859             while( *((int *)TableAddress) != 0 )
1860             {
1861                 TableAddress++;
1862             } /* Get past trailing string-list */
1863             TableAddress += 2;
1864         } /* Increment address to start of next structure */
1865     } /* END while-loop looking for structure type */
1866     return handle;
1867 } /* END FindStructure */

```

1868

Annex C (informative)

Change History

1869
1870
1871
1872

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: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> Added definitions for Pentium® Pro, Burst EDO, and SDRAM. Added clarifications to the Memory Controller Error Status.
2.1	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: <ul style="list-style-type: none"> 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
		<ul style="list-style-type: none"> • 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	1998-03-16	<p>The following changes were made to version 2.1 of the document to produce this version:</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> – 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 <ul style="list-style-type: none"> – 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
		<p>Hardware Security group.</p> <ul style="list-style-type: none"> – 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	1998-08-12	<p>The following changes were made to version 2.2 of the document to produce this version:</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – Added enumeration for Wake-up Type • System Enclosure or Chassis: <ul style="list-style-type: none"> – Added OEM-defined field. • Processor Information: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – Corrected example, adding double-null to terminate the structure. • System Slots: <ul style="list-style-type: none"> – Added hot-plug characteristic definition and clarified usage of the PCI “Slot ID” field. • Memory Device: <ul style="list-style-type: none"> – Added enumerations for Form Factor and Device Type – Added new field for memory Speed • System Event Log: <ul style="list-style-type: none"> – Added note describing how century portion of the 2-digit year within a log record is to be interpreted.

Version	Release Date	Description
		<ul style="list-style-type: none"> • Voltage Probe, Temperature Probe, Electrical Current Probe, Cooling Device: <ul style="list-style-type: none"> – 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	<p>The following changes were made to version 2.3 of the document to produce this version:</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> – Added BIOS Characteristic Extension Byte 2, bit 1, to identify that the BIOS supports F12=Network Boot functionality • Processor Information: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – Added slot type enumeration for PCI-X – Added slot characteristic to identify support for (to-be) standard SMBus interface for PCI slots • Memory Device: <ul style="list-style-type: none"> – 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: <ul style="list-style-type: none"> – 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	<p>The following changes were made to version 2.3.1 of the document to produce this version:</p> <ul style="list-style-type: none"> • 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
		<p>deprecated at this specification version.</p> <ul style="list-style-type: none"> • 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) <ul style="list-style-type: none"> – 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) <ul style="list-style-type: none"> – 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) <ul style="list-style-type: none"> – 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) <ul style="list-style-type: none"> – Added new enumerations to Associativity • Memory Device (Type 17) <ul style="list-style-type: none"> – Added new enumerations to Memory Type • Built-in Pointing Device (Type 21) <ul style="list-style-type: none"> – 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	<p>The following changes were made to version 2.3.2 of the document to produce this version:</p> <ul style="list-style-type: none"> • Accepted all changes introduced at version 2.3.2 • Updated the Abstract to contain the updated DMTF copyright statement. • Processor Information (Type 4) <ul style="list-style-type: none"> – Added new enumerations to Processor Family and Processor Upgrade
2.3.4	2002-12-06	<p>The following changes were made to version 2.3.3 of the document to produce this version:</p> <ul style="list-style-type: none"> • System Enclosure Information (Type 3) <ul style="list-style-type: none"> – Provided clarification regarding contained element types • Processor Information (Type 4) <ul style="list-style-type: none"> – 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) <ul style="list-style-type: none"> – Added AGP8X enumeration to Slot Type
2.4	2004-07-21	<p>The following changes were made to version 2.3.4 of the document to produce this version:</p>

Version	Release Date	Description
		<ul style="list-style-type: none"> • Processor Information (Type 4) <ul style="list-style-type: none"> – Added new enumerations to Processor Family (CR00951, CR01152) • System Slots (Type 9) <ul style="list-style-type: none"> – Added PCI Express enumeration to Slot Type (CR01259) – Added new enumerations to Slot Data Bus Width (CR01324) • Memory Device (Type 17) <ul style="list-style-type: none"> – Added DDR2 enumeration to Type (CR01263) • BIOS Information (Type 0) <ul style="list-style-type: none"> – 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) <ul style="list-style-type: none"> – Added fields: SKU Number and Family (CR01270) – Updated Conformance Guidelines and added corrections
2.5	2006-09-05	<p>The following changes were made to version 2.4 of the document to produce this version:</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> – Updated specification revisions and URLs (PreOSCR00019.001) • Table Convention: <ul style="list-style-type: none"> – Added EFI-specific information (PreOSCR00011.005) • SMBIOS Structure Table Entry Point: <ul style="list-style-type: none"> – Corrected typo, the SMBIOS BCD Revision is at offset 1Eh, not 1Dh (PreOSCR00020.000) • Required Structures and Data: <ul style="list-style-type: none"> – Added DIG64 information (PreOSCR00013.000) • System Enclosure or Chassis (Type 3) <ul style="list-style-type: none"> – Added new types for CompactPCI and AdvancedTCA (PreOSCR00012.001) • Processor Information (Type 4) <ul style="list-style-type: none"> – 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) <ul style="list-style-type: none"> – Added note on cache size for multi-core processors (PreOSCR00002)

Version	Release Date	Description
		<ul style="list-style-type: none"> • Port connector Information (Type 8) <ul style="list-style-type: none"> – Added SATA and SAS (PreOSCR00021.002) • System Slots (Type 9) <ul style="list-style-type: none"> – Updated Slot ID description with ACPI and PCI Express (PreOSCR00018.000) • Onboard Devices Information (Type 10) <ul style="list-style-type: none"> – Added SATA and SAS (PreOSCR00021.002) • Memory Device (Type 17) <ul style="list-style-type: none"> – Added values for FB-DIMM (PreOSCR00010.004) • Memory Device Mapped Address (Type 20) <ul style="list-style-type: none"> – Moved structure from 'required' to 'optional' (PreOSCR00009.002) – Moved 'Plug-and-Play Calling Convention' to Appendix C (PreOSCR00022.001)
2.6	2008-06-30	<p>The following changes were made to version 2.5 of the document to produce this version:</p> <ul style="list-style-type: none"> • 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): <ul style="list-style-type: none"> – 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 Ili” (was “UltraSPARC Iii”) (SMBIOSCR00054) • System Slots (Type 9): <ul style="list-style-type: none"> – 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	<p>The following changes were made to version 2.6 of the document to produce this version:</p>

Version	Release Date	Description
		<ul style="list-style-type: none"> • System Information (Type 1): <ul style="list-style-type: none"> – 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): <ul style="list-style-type: none"> – 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): <ul style="list-style-type: none"> – SMBIOSCR00062: Add values to cache associativity enum to cover new processors • System Slots (Type 9): <ul style="list-style-type: none"> – SMBIOSCR00064: Add PCIe Gen 2 slot types to Type 9 • Memory Device (Type 17): <ul style="list-style-type: none"> – SMBIOSCR00052: Add new memory device types: DDR3 and FBD2
2.7	2010-07-21	<p>The following changes were made to version 2.6.1 of the document to produce this version:</p> <ul style="list-style-type: none"> • Document layout: <ul style="list-style-type: none"> – SMBIOSCR00073: Move SMBIOS structure definitions to a new top-level section – SMBIOSCR00074: Remove Appendix C, "Plug-and-Play Calling Convention" • Various sections: <ul style="list-style-type: none"> – SMBIOSCR00096: Miscellaneous clerical changes • Section 1.1, Document Version Number Conventions: <ul style="list-style-type: none"> – SMBIOSCR00085: Add more description to the document version number convention • Section 3.1.2, Structure Header Format: <ul style="list-style-type: none"> – SMBIOSCR00048: Reserve handle number for consistency with UEFI PI specification • Section 3.1.3, Text Strings: <ul style="list-style-type: none"> – SMBIOSCR00086: Remove maximum string size limitation • Section 3.2, Required Structures and Data: <ul style="list-style-type: none"> – SMBIOSCR00095: Increase the capacity to represent system memory of 4 terabytes or greater.

Version	Release Date	Description
		<ul style="list-style-type: none"> • Bios Information (Type 0): <ul style="list-style-type: none"> – SMBIOSCR00056: Add UEFI support to BIOS characteristics – SMBIOSCR00071: Add support to describe virtualized platforms (bit 4) • System Enclosure or Chassis (Type 3): <ul style="list-style-type: none"> – 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): <ul style="list-style-type: none"> – 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): <ul style="list-style-type: none"> – SMBIOSCR00095: Increase the capacity to represent system memory of 4 terabytes or greater. • Memory Device (Type 17): <ul style="list-style-type: none"> – 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): <ul style="list-style-type: none"> – SMBIOSCR00095: Increase the capacity to represent system memory of 4 terabytes or greater. • Cooling Device (Type 27): <ul style="list-style-type: none"> – SMBIOSCR00075: Add new "description" field in structure type 27 • IPMI Device Information (Type 38): <ul style="list-style-type: none"> – 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
		<ul style="list-style-type: none"><li data-bbox="548 268 1344 300">– SMBIOSCR00079: Add Type 42 Management Controller Host Interface<li data-bbox="548 306 1138 338">– SMBIOSCR00096: Replace “record” with “structure”<li data-bbox="509 344 954 375">• Appendix A, Conformance Guidelines:<ul style="list-style-type: none"><li data-bbox="548 382 1378 438">– SMBIOSCR00095: Increase the capacity to represent system memory of 4 terabytes or greater.

1873

1874

Bibliography

1875 DMTF DSP4004, *DMTF Release Process 2.2*,
1876 http://www.dmtf.org/standards/published_documents/DSP4004_2.2.pdf

1877