

1



2

Document Number: DSP1012

3

Date: 2009-06-22

4

5

Version: 1.0.1

6

Boot Control Profile

7

Document Type: Specification

8

Document Status: DMTF Standard

9

Document Language: E

10 Copyright Notice

11 Copyright © 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

32

Contents

34	Foreword	7
35	Introduction	8
36	1 Scope	9
37	2 Normative References.....	9
38	2.1 Approved References	9
39	2.2 Other References.....	9
40	3 Terms and Definitions	9
41	4 Symbols and Abbreviated Terms.....	11
42	5 Synopsis.....	12
43	6 Description	12
44	6.1 Class Diagram	13
45	7 Implementation.....	14
46	7.1 CIM_BootService	14
47	7.2 CIM_ComputerSystem.....	15
48	7.3 Representing Boot Service Capabilities	15
49	7.4 Boot Configurations	16
50	7.5 Applying the Boot Configuration	17
51	7.6 Creating a Boot Configuration	18
52	7.7 Deleting a Boot Configuration.....	19
53	7.8 Identifying Boot Sources.....	19
54	7.9 Changing the Boot Order.....	22
55	7.10 Representing a Set of Aggregated Boot Sources.....	23
56	7.11 Boot Order During the Boot Process	24
57	7.12 Settings to Apply During the Boot Process.....	25
58	8 Methods.....	26
59	8.1 CIM_BootService.CreateBootConfigSetting()	26
60	8.2 CIM_BootService.ApplyBootConfigSetting()	27
61	8.3 CIM_BootConfigSetting.ChangeBootOrder()	29
62	8.4 Profile Conventions for Operations.....	29
63	8.5 CIM_BootService.....	30
64	8.6 CIM_BootConfigSetting	30
65	8.7 CIM_BootSettingData	31
66	8.8 CIM_BootSourceSetting	32
67	8.9 CIM_ConcreteComponent	32
68	8.10 CIM_ConcreteDependency	32
69	8.11 CIM_ElementCapabilities	33
70	8.12 CIM_ElementSettingData	33
71	8.13 CIM_BootServiceCapabilities	34
72	8.14 CIM_HostedService	35
73	8.15 CIM_LogicalIdentity	35
74	8.16 CIM_OrderedComponent	35
75	8.17 CIM_ServiceAffectsElement	36
76	9 Use Cases.....	36
77	9.1 Advertising the Profile Conformance	36
78	9.2 Object Diagram for a Monolithic Server.....	36
79	9.3 Object Diagram for a Monolithic Server with Service Processor.....	37
80	9.4 Object Diagram for a Modular System.....	38
81	9.5 PXE Boot Source	39
82	9.6 Disk Boot Source	40
83	9.7 Local CDROM and Floppy Boot Sources	41
84	9.8 Representing IPL and Boot Control Vector (BCV) Lists	41
85	9.9 Representing Settings and Boot Settings.....	43

86	9.10	Representing the Default Boot Configuration for a Computer System.....	43
87	9.11	Representing the Next Boot Configuration for a Computer System.....	44
88	9.12	Representing the Current Boot Configuration for a Booted Computer System.....	45
89	9.13	Create a New Boot Configuration	46
90	9.14	Apply an Existing Boot Configuration	47
91	9.15	Find the Boot Service for a Computer System	48
92	9.16	Find the Boot Configuration for a Computer System.....	48
93	9.17	Find the Boot Service Capabilities for a Computer System.....	48
94	9.18	Find the Current Boot Configuration for a Computer System.....	48
95	9.19	Find the Default Boot Configuration for a Computer System	48
96	9.20	Find the Boot Configuration that Will Be Used during the Next Reboot for a Computer System	49
97			
98	9.21	Make a Boot Configuration Applicable for Subsequent Reboots	49
99	9.22	Make a Boot Configuration Applicable for the Next Reboot Only.....	49
100	9.23	Determine If the Computer System Supports PXE Boot	49
101	9.24	Find the Boot Order for a Computer System for the Next Reboot.....	49
102	9.25	Change the Boot Order for a Computer System	50
103	9.26	Determine Whether BootService.ElementName Is Modifiable	50
104	9.27	Determine Whether a New Boot Configuration Can Be Created	51
105	9.28	Determine Whether a Boot Configuration Can Be Applied.....	51
106	10	CIM Elements	51
107	10.1	CIM_RegisteredProfile.....	52
108	10.2	CIM_BootService.....	52
109	10.3	CIM_BootServiceCapabilities	52
110	10.4	CIM_BootConfigSetting	53
111	10.5	CIM_BootSettingData	53
112	10.6	CIM_BootSourceSetting	53
113	10.7	CIM_ConcreteComponent.....	54
114	10.8	CIM_ConcreteDependency	55
115	10.9	CIM_ElementCapabilities	56
116	10.10	CIM_ElementSettingData	56
117	10.11	CIM_HostedService.....	56
118	10.12	CIM_LogicalIdentity	57
119	10.13	CIM_OrderedComponent	57
120	10.14	CIM_ServiceAffectsElement.....	58
121		ANNEX A (informative) Change Log.....	59

122

123 **List of Figures**

124	Figure 1 – Boot Control Profile: Class Diagram.....	13
125	Figure 2 – Registered Profile	36
126	Figure 3 – Monolithic Server Object Diagram.....	37
127	Figure 4 – Monolithic Server with Service Processor Object Diagram	37
128	Figure 5 – Modular System Object Diagram.....	38
129	Figure 6 – Modular System Object Diagram.....	39
130	Figure 7 – PXE Boot Sources Object Diagram.....	40
131	Figure 8 – Booting from Disk	40
132	Figure 9 – Booting from CDROM and Floppy.....	41
133	Figure 10 – Booting from IPL and BCV Devices.....	42
134	Figure 11 – Setting Data and Boot Setting Data.....	43

135 Figure 12 – Default Boot Configuration Object Diagram 44
 136 Figure 13 – Next Boot Configuration Object Diagram 45
 137 Figure 14 – Boot Configuration for a Booted System Object Diagram 46
 138 Figure 15 – System with New CIM_BootConfigSetting 47
 139

140 **List of Tables**

141 Table 1 – Related Profiles 12
 142 Table 2 – Structured Name Identifiers 22
 143 Table 3 – CreateBootConfigSetting() Method: Return Code Values 27
 144 Table 4 – CreateBootConfigSetting() Method: Parameters 27
 145 Table 5 – ApplyBootConfigSetting() Method: Return Code Values 28
 146 Table 6 – ApplyBootConfigSetting() Method: Parameters 28
 147 Table 7 – ChangeBootOrder() Method: Return Code Values 29
 148 Table 8 – ChangeBootOrder() Method: Parameters 29
 149 Table 9 – Operations: CIM_BootService 30
 150 Table 10 – Operations: CIM_BootConfigSetting 30
 151 Table 11 – Operations: CIM_BootSettingData 31
 152 Table 12 – Operations: CIM_BootSourceSetting 32
 153 Table 13 – Operations: CIM_ConcreteComponent 32
 154 Table 14 – Operations: CIM_ConcreteDependency 32
 155 Table 15 – Operations: CIM_ElementCapabilities 33
 156 Table 16 – Operations: CIM_ElementSettingData 33
 157 Table 17 – Operations: CIM_BootServiceCapabilities 35
 158 Table 18 – Operations: CIM_HostedService 35
 159 Table 19 – Operations: CIM_LogicalIdentity 35
 160 Table 20 – Operations: CIM_OrderedComponent 35
 161 Table 21 – Operations: CIM_ServiceAffectsElement 36
 162 Table 22 CIM Elements – Boot Control Profile 51
 163 Table 23 – Class: CIM_RegisteredProfile 52
 164 Table 24 – Class: CIM_BootService 52
 165 Table 25 – Class: CIM_BootServiceCapabilities 53
 166 Table 26 – Class: CIM_BootConfigSetting 53
 167 Table 27 – Class: CIM_BootSettingData 53
 168 Table 28 – Class: CIM_BootSourceSetting 54
 169 Table 29 – Class: CIM_ConcreteComponent – Use 1 54
 170 Table 30 – Class: CIM_ConcreteComponent – Use 2 54
 171 Table 31 – Class: CIM_ConcreteComponent – Use 3 55
 172 Table 32 – Class: CIM_ConcreteComponent – Use 4 55
 173 Table 33 – Class: CIM_ConcreteDependency 55
 174 Table 34 – Class: CIM_ElementCapabilities 56
 175 Table 35 – Class: CIM_ElementSettingData 56
 176 Table 36 – Class: CIM_HostedService 56
 177 Table 37 – Class: CIM_LogicalIdentity 57
 178 Table 38 – Class: CIM_OrderedComponent 57
 179 Table 39 – Class: CIM_ServiceAffectsElement 58

180

182

Foreword

183 The *Boot Control Profile* (DSP1012) was prepared by the Physical Platform Profiles Working Group.

184 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
185 management and interoperability.

186 **Acknowledgments**

187 The authors wish to acknowledge the following people.

188 **Editors:**

- 189 • John Leung – Intel
- 190 • David Simpson – IBM
- 191 • Khachatur Papanyan – Dell

192 **Contributors:**

- 193 • Aaron Merkin – IBM
- 194 • Jon Hass – Dell
- 195 • Khachatur Papanyan – Dell
- 196 • Enoch Suen – Dell
- 197 • Jeff Hilland – HP
- 198 • Hemal Shah – Broadcom
- 199 • Christina Shaw – HP
- 200 • Perry Vincent – Intel
- 201 • Arvind Kumar – Intel

202

203

Introduction

204 The information in this specification should be sufficient for a provider or consumer of this data to
205 unambiguously identify the classes, properties, methods, and values that shall be instantiated and
206 manipulated to represent and manage the boot control configurations of a computer server using the
207 DMTF CIM core and extended model definitions.

208 The target audience for this specification is implementers who are writing CIM-based providers or
209 consumers of management interfaces representing the components described in this document.

210

Boot Control Profile

211 1 Scope

212 The *Boot Control Profile* describes the classes, associations, properties, and methods used to manage
213 the boot control configurations of a physical or virtual computer system.

214 2 Normative References

215 The following referenced documents are indispensable for the application of this document. For dated
216 references, only the edition cited applies. For undated references, the latest edition of the referenced
217 document (including any amendments) applies.

218 2.1 Approved References

219 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,
220 http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf

221 DMTF DSP0200, *CIM Operations over HTTP 1.2*,
222 http://www.dmtf.org/standards/published_documents/DSP0200_1.2.pdf

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

225 DMTF DSP1033, *Profile Registration Profile 1.0*,
226 http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

227 2.2 Other References

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

230 *BIOS Boot Specification 1.01* (January 11, 1996),
231 <http://www.phoenix.com/NR/rdonlyres/56E38DE2-3E6F-4743-835F-B4A53726ABED/0/specsbbs101.pdf>

232 3 Terms and Definitions

233 3.1

234 **can**

235 used for statements of possibility and capability, whether material, physical, or causal

236 3.2

237 **cannot**

238 used for statements of possibility and capability, whether material, physical, or causal

239 3.3

240 **conditional**

241 used to indicate requirements strictly to be followed, in order to conform to the document when the
242 specified conditions are met

- 243 **3.4**
244 **mandatory**
245 used to indicate requirements strictly to be followed, in order to conform to the document and from which
246 no deviation is permitted
- 247 **3.5**
248 **may**
249 used to indicate a course of action permissible within the limits of the document
- 250 **3.6**
251 **need not**
252 used to indicate a course of action permissible within the limits of the document
- 253 **3.7**
254 **optional**
255 used to indicate a course of action permissible within the limits of the document
- 256 **3.8**
257 **referencing profile**
258 indicates a profile that owns the definition of a class used, but not defined, in this document and can be
259 included in the “Referenced Profiles” table
- 260 **3.9**
261 **shall**
262 used to indicate requirements strictly to be followed, in order to conform to the document and from which
263 no deviation is permitted
- 264 **3.10**
265 **shall not**
266 used to indicate requirements strictly to be followed, in order to conform to the document and from which
267 no deviation is permitted
- 268 **3.11**
269 **should**
270 used to indicate that among several possibilities, one is recommended as particularly suitable, without
271 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 272 **3.12**
273 **should not**
274 used to indicate that a certain possibility or course of action is deprecated but not prohibited
- 275 **3.13**
276 **unspecified**
277 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 278 **3.14**
279 **Boot Configurable System**
280 an instance of CIM_ComputerSystem whose boot configurations are being managed
- 281 **3.15**
282 **Boot Configuration**
283 a collection of settings that are applied to a boot configurable system during the boot process

284 **3.16**285 **Boot Configuration Representation**

286 the CIM representation of a boot configuration, which consists of an instance of class
287 CIM_BootConfigSetting and, optionally, all of the instances of classes CIM_BootSourceSetting,
288 CIM_BootSettingData and CIM_SettingData that it directly or indirectly aggregates

289 **3.17**290 **Current Boot Configuration**

291 the instance of CIM_BootConfigSetting that was used the last time the managed system was successfully
292 booted

293 **3.18**294 **Default Boot Configuration**

295 the instance of CIM_BootConfigSetting that the computer system manufacturer or a client has
296 informatively tagged as its default boot configuration

297 **3.19**298 **Next Boot Configuration**

299 the instance of CIM_BootConfigSetting that will be used during the next boot of the Boot Configurable
300 System

301 **3.20**302 **Next Single Use Boot Configuration**

303 the instance of CIM_BootConfigSetting that will only be used during the next boot of the Boot
304 Configurable System and then not used again

305 **3.21**306 **Not Next Boot Configuration**

307 an instance of CIM_BootConfigSetting that will not be used during the next boot

308 **3.22**309 **Template Boot Configuration**

310 an existing instance of CIM_BootConfigSetting that is to be used as the template for creating a new boot
311 configuration

312 **4 Symbols and Abbreviated Terms**313 **4.1**314 **BCV**

315 Boot Control Vector. See the [BIOS Boot Specification](#) for additional information.

316 **4.2**317 **IPL**

318 Initial Program Load. See the [BIOS Boot Specification](#) for additional information.

319 **4.3**320 **PXE**

321 Preboot Execution Environment. See the [BIOS Boot Specification](#) for additional information.

322 **5 Synopsis**

323 **Profile Name:** *Boot Control*

324 **Version:** 1.0.1

325 **Organization:** DMTF

326 **CIM Schema Version:** 2.19

327 **Central Class:** CIM_BootService

328 **Scoping Class:** CIM_ComputerSystem

329 The *Boot Control Profile* extends the management capabilities of referencing profiles by adding the
 330 capability to represent and manage boot configurations that include boot devices and settings for use
 331 during booting.

332 Table 1 identifies profiles on which this profile has a dependency.

333 CIM_BootService shall be the Central Class of this profile. The instance of CIM_BootService shall be the
 334 Central Instance of this profile.

335 CIM_ComputerSystem shall be the Scoping Class of this profile. The instance of CIM_ComputerSystem
 336 with which the Central Instance is associated through an instance of CIM_HostedService shall be the
 337 Scoping Instance of this profile.

338

Table 1 – Related Profiles

Profile Name	Organization	Version	Relationship
Profile Registration	DMTF	1.0	Mandatory

339 **6 Description**

340 The *Boot Control Profile* describes the elements needed to provide the capability to manage the boot
 341 configurations of a computer system.

342 The profile could manage the following capabilities of a typical computer system:

- 343 • A computer system can have one or more boot configurations.
- 344 • A computer system can contain a boot configuration that is used during each boot.
- 345 • A computer system can contain a single-use boot configuration that is used only during the next
 346 boot and then not used again.
- 347 • A computer system can contain a current boot configuration that represents the boot
 348 configuration successfully used in the last boot.
- 349 • A computer system can contain a default boot configuration that is set by the computer system
 350 manufacturer or a client.
- 351 • A computer system can create new boot configurations.
- 352 • A computer system can apply a boot configuration to an active or inactive computer system.

353 A typical boot configuration could have the following characteristics:

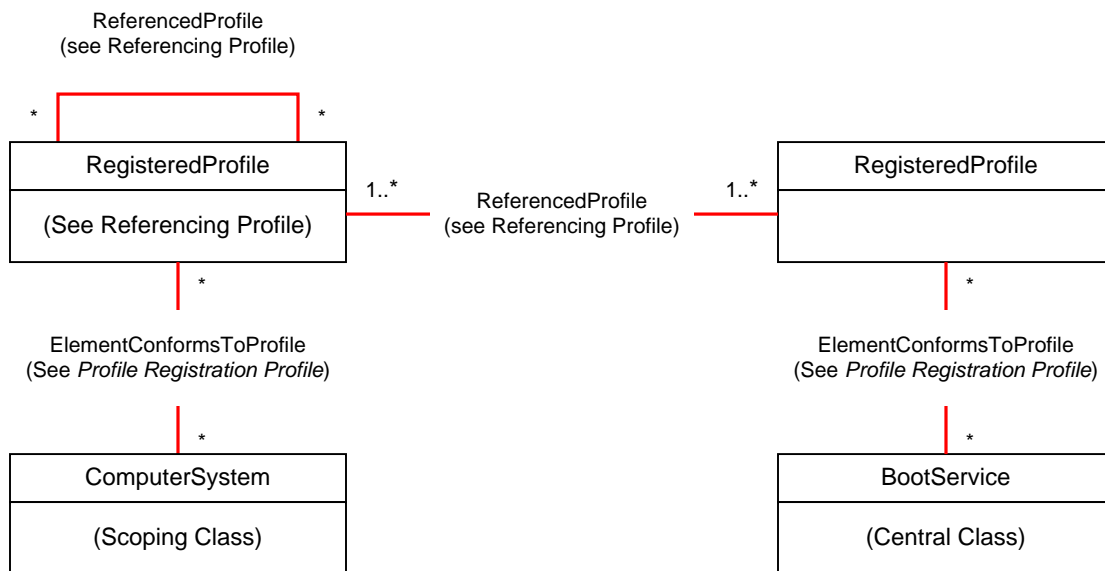
- 354 • A boot configuration can contain a boot order that specifies the order in which boot devices are
- 355 accessed. The boot devices include, but are not limited to, floppy device, CD device, hard disks,
- 356 network controllers (using the PXE protocol), and BCV devices composed of additional boot
- 357 sources.
- 358 • A boot configuration can contain data that can affect various computer system components
- 359 during the boot process.
- 360 • A boot configuration can contain data that can be passed to the booted image (for example,
- 361 second-stage boot loader or bootblock) in the form of a boot string.
- 362 • Boot devices can be local to the computer system or remote to the computer system.

363 A boot configuration can be applied when the computer system starts the boot process. The boot process
 364 can be started automatically as part of the enablement of the computer system or by a specific request
 365 when the computer system is enabled but not booted.

366 6.1 Class Diagram

367 Figure 1 represents the class schema for the *Boot Control Profile*. For simplicity, the prefix CIM_ has
 368 been removed from the name of the classes.

369 In Figure 1, CIM_ManagedElement, CIM_LogicalDevice, CIM_SettingData, and CIM_BootSettingData
 370 are abstract classes.



371

372 **Figure 1 – Boot Control Profile: Class Diagram**

373 A computer system can have multiple boot configurations. Each boot configuration is modeled by a Boot
 374 Configuration Representation, which consists of an instance of CIM_BootConfigSetting class and,
 375 optionally, all of the instances of classes CIM_BootSourceSetting, CIM_BootSettingData and
 376 CIM_SettingData that the instance of CIM_BootConfigSetting aggregates

377 The usage of each Boot Configuration Representation during the boot process is determined by the
378 IsNext property of the CIM_ElementSettingData association between the Boot Configuration
379 Representation and Boot Configurable System whose boot configuration is being managed.

380 Each Boot Configuration Representation contains an ordered list of boot sources, which indicate the
381 logical devices to use during the boot process. The boot order is defined by interpreting a property in the
382 CIM_OrderedComponent association between the instance of CIM_BootConfigSetting representing a
383 boot configuration and instances of CIM_BootSourceSetting representing the boot sources.

384 In some cases a single boot source might, in turn, represent additional ordered boot sources. This set of
385 aggregated boot sources is represented by an instance of CIM_BootConfigSetting, which is associated to
386 the instance of CIM_BootSourceSetting through an instance of CIM_LogicalIdentity.

387 Settings that apply to a managed element during the boot process are represented by instances of a
388 concrete subclass of the CIM_SettingData class.

389 Settings that apply to the boot process, itself, are represented by instances of a concrete subclass of the
390 CIM_BootSettingData class.

391 These settings can apply to either the entire boot configuration or to a specific boot source within a boot
392 configuration. This scoping is determined by traversing the CIM_ConcreteComponent association to
393 either an instance of CIM_BootConfigSetting representing the boot configuration or
394 CIM_BootSourceSetting representing the boot source, respectively.

395 **7 Implementation**

396 This clause contains normative information about the model and the relationship between the model and
397 underlying instrumentation. Normative text for properties is included in this clause. Normative text for
398 methods is contained in clause 8.

399 **7.1 CIM_BootService**

400 At least one instance of the Central Class, CIM_BootService, shall exist.

401 **7.1.1 CIM_BootService.ElementName**

402 ElementName shall be formatted as a free-form string of variable length (pattern “.*/”).

403 **7.1.2 Modifying ElementName Is Supported**

404 Subclause 7.1.2 describes conditional behavior. Subclause 7.1.2 describes the CIM elements and
405 behaviors that shall be implemented when the following conditions are met.

406 Conditional Requirement:

- 407 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
408 through an instance of CIM_ElementCapabilities.
- 409 2) The CIM_BootServiceCapabilities.ElementNameEditSupport property has the value of TRUE.
- 410 3) The CIM_BootServiceCapabilities.MaxElementNameLen property has a non-zero value

411 The implementation shall allow the CIM_BootService.ModifyInstance intrinsic operation to change the
412 value of the ElementName property. The ModifyInstance operation shall enforce the length restriction
413 specified in the MaxElementNameLen property.

414 **7.1.3 Modifying ElementName Is Not Supported**

415 Subclause 7.1.3 describes conditional behavior, Subclause 7.1.3 describes the CIM elements and
416 behaviors that shall be implemented when either of the following conditions are met.

417 Conditional Requirement 1:

- 418 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
419 through an instance of CIM_ElementCapabilities.
- 420 2) The CIM_BootServiceCapabilities.ElementNameEditSupport property has the value of FALSE.

421 Conditional Requirement 2:

- 422 1) An instance of CIM_BootServiceCapabilities is not associated with the CIM_BootService
423 instance through an instance of CIM_ElementCapabilities.

424 The implementation shall not allow the CIM_BootService.ModifyInstance intrinsic operation to change the
425 value of the ElementName property.

426 **7.2 CIM_ComputerSystem**

427 An instance of CIM_ComputerSystem shall represent either a Scoping Instance or a Boot Configurable
428 System, or both. The Scoping Instance is used to determine profile conformance. The Boot Configurable
429 System represents a computer system whose boot configurations are being managed.

430 One Scoping Instance shall exist. Clause 5 describes the process for determining the Scoping Instance
431 from the Central Instance.

432 Each instance of CIM_ComputerSystem representing a Boot Configurable System shall be associated to
433 the Central Instance through an instance of the CIM_ServiceAffectsElement association. At least one
434 instance of a Boot Configurable System shall exist.

435 **7.3 Representing Boot Service Capabilities**

436 Subclause 7.3 describes optional behavior.

437 An instance of CIM_BootServiceCapabilities may exist, which represents the capabilities of the boot
438 service.

439 If an instance of CIM_BootServiceCapabilities is instantiated, then it shall be associated with an instance
440 of CIM_BootService using an instance of CIM_ElementCapabilities.

441 **7.3.1 Representing Implementation Specific Boot Service Capabilities**

442 Subclause 7.3.1 describes optional behavior.

443 An implementation may identify method-related boot configuration capabilities, other than those explicitly
444 defined in this profile, by setting the BootConfigCapabilities and OtherBootConfigCapabilities property
445 arrays of the CIM_BootServiceCapabilities class.

446 The additional boot configuration capability shall be identified by setting an entry in the
447 CIM_BootServiceCapabilities.BootConfigCapabilities property array to a value of 1 (Other) for each
448 additional boot configuration capability.

449 For each entry in the BootConfigCapabilities array property with the value 1 (Other), the corresponding
450 entry in the CIM_BootServiceCapabilities.OtherBootConfigCapabilities array property shall contain a non-
451 NULL, non-empty string that provides a short description of the capability.

452 **7.4 Boot Configurations**

453 An instance of CIM_BootConfigSetting shall represent a boot configuration that may be used during the
454 boot process.

455 Each Boot Configurable System shall have at least one instance of CIM_BootConfigSetting associated to
456 it through an instance of CIM_ElementSettingData.

457 **7.4.1 CIM_ElementSettingData**

458 An instance of CIM_ElementSettingData shall be used to associate each instance of
459 CIM_BootConfigSetting, representing a boot configuration, to each instance of CIM_ComputerSystem,
460 representing a Boot Configurable System to which the boot configuration applies.

461 When the CIM_ElementSettingData association is used in this manner, its ManagedElement property
462 shall reference the CIM_ComputerSystem instance and its SettingData property shall reference the
463 CIM_BootConfigSetting instance.

464 For an instance of CIM_ElementSettingData, the IsNext property shall determine how the associated
465 instance of CIM_BootConfigSetting is used, if at all, during the boot of the Boot Configurable System.

466 **7.4.2 Default Boot Configuration**

467 Subclause 7.4.2 describes optional behavior.

468 The Default Boot Configuration is the instance of CIM_BootConfigSetting that the computer system
469 manufacturer or a client has informatively tagged as the default configuration for the Boot Configurable
470 System. The Default Boot Configuration does not impact which boot configuration applies during the boot
471 process.

472 The Default Boot Configuration shall be the instance of CIM_BootConfigSetting that is associated by the
473 instance of CIM_ElementSettingData when the IsDefault property has a value of 1 (Is Default).

474 For a given Boot Configurable System, at most one Default Boot Configuration shall be associated. The
475 IsDefault property of instances of CIM_ElementSettingData associating the Boot Configurable System to
476 all other Boot Configuration Representations shall have a value of 2 (Is Not Default).

477 **7.4.3 Current Boot Configuration**

478 Subclause 7.4.3 describes optional behavior.

479 The Current Boot Configuration is the instance of CIM_BootConfigSetting that was used the last time the
480 system represented by the Boot Configurable System was successfully booted.

481 The Current Boot Configuration shall be the instance of CIM_BootConfigSetting that is associated by the
482 instance of CIM_ElementSettingData when the IsCurrent property has a value of 1 (Is Current).

483 For a given Boot Configurable System, zero or one Current Boot Configuration shall be associated. The
484 IsCurrent property of instances of CIM_ElementSettingData associating the Boot Configurable System to
485 all other Boot Configuration Representations shall have a value of 2 (Is Not Current).

486 An implementation may support the Current Boot Configuration when it is able to determine the
487 configuration last used during a successful boot. When an implementation supports the Current Boot
488 Configuration, the Current Boot Configuration shall exist after a successful boot.

489 **7.4.4 Next Boot Configuration**

490 Subclause 7.4.4 describes optional behavior.

491 NOTE: Successful execution of the ApplyBootConfigSetting() method can independently apply a boot configuration to
492 a Boot Configurable System regardless of the Next Boot Configuration. The requirements in this subclause shall not
493 apply when a Boot Configurable System is booted using the ApplyBootConfigSetting() method.

494 The Next Boot Configuration is the instance of CIM_BootConfigSetting that shall be used during the next
495 boot of the system represented by the Boot Configurable System, unless there is a Next Single Use Boot
496 Configuration associated to the same Boot Configurable System.

497 The Next Boot Configuration shall be the instance of CIM_BootConfigSetting that is associated by the
498 instance of CIM_ElementSettingData when the IsNext property has a value of 1 (Is Next).

499 For a given Boot Configurable System, at most one Next Boot Configuration shall be associated.

500 **7.4.5 Next Single Use Boot Configuration**

501 Subclause 7.4.5 describes optional behavior.

502 NOTE: Successful execution of the ApplyBootConfigSetting() method can independently apply a boot configuration to
503 a Boot Configurable System regardless of the Next Single Use Boot Configuration. The requirements in this
504 subclause shall not apply when a Boot Configurable System is booted using the ApplyBootConfigSetting() method.

505 The Next Single Use Boot Configuration is the instance of CIM_BootConfigSetting that shall only be used
506 during the next boot of the system represented by the Boot Configurable System.

507 When a Next Boot Configuration is also associated to the Boot Configurable System, the Next Single Use
508 Boot Configuration shall take precedence over the Next Boot Configuration.

509 Upon a successful usage during a boot, the Next Single Use Boot Configuration shall become a Not Next
510 Boot Configuration.

511 The Next Single Use Boot Configuration shall be the instance of CIM_BootConfigSetting that is
512 associated by the instance of CIM_ElementSettingData when the IsNext property has a value of 3 (Is
513 Next For Single Use).

514 For a given Boot Configurable System, there shall be at most one Next Single Use Boot Configuration
515 associated.

516 **7.4.6 Not Next Boot Configuration**

517 The Not Next Boot Configuration is an instance of CIM_BootConfigSetting that will not be used during the
518 next boot.

519 The Not Next Boot Configuration shall be a CIM_BootConfigSetting whose
520 CIM_ElementSettingData.IsNext property has the value of 2 (Is Not Next).

521 **7.5 Applying the Boot Configuration**

522 The CIM_BootService associated to the Boot Configurable System may support the explicit application of
523 a Boot Configuration Representation through the ApplyBootConfigSetting() method.

524 NOTE: Successful execution of the ApplyBootConfigSetting() method can independently apply a boot configuration to
525 a Boot Configurable System regardless of the Next Boot Configuration. The requirements in subclause 7.4.4 shall not
526 apply when a Boot Configurable System is booted using the ApplyBootConfigSetting() method.

527 **7.5.1 Apply Boot Configuration Is Supported**

528 Subclause 7.5.1 describes conditional behavior. Subclause 7.5.1 describes the CIM elements and
529 behaviors that shall be implemented when the following conditions are met.

530 Conditional Requirement:

- 531 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
532 through an instance of CIM_ElementCapabilities.
- 533 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 3
534 (Applies Boot Configuration).

535 The implementation shall support the CIM_BootService.ApplyBootConfigSetting() method.

536 **7.5.2 Apply Boot Configuration Is Not Supported**

537 Subclause 7.5.2 describes conditional behavior. Subclause 7.5.2 describes the CIM elements and
538 behaviors that shall be implemented when either of the following conditions are met.

539 Conditional Requirement 1:

- 540 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
541 through an instance of CIM_ElementCapabilities.
- 542 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array does not contain a
543 value of 3 (Applies Boot Configuration).

544 Conditional Requirement 2:

- 545 1) An instance of CIM_BootServiceCapabilities is not associated with the CIM_BootService
546 instance through an instance of CIM_ElementCapabilities.

547 The implementation shall not support the CIM_BootService.ApplyBootConfigSetting() method.

548 When a Boot Configurable System, that is not associated to a Next Boot Configuration or Next Single Use
549 Boot Configuration, transitions to the Enabled state, then the normal boot process shall be initiated.

550 **7.6 Creating a Boot Configuration**

551 The CIM_BootService may support the client creation of a new boot configuration from an existing boot
552 configuration through the CreateBootConfigSetting() method.

553 **7.6.1 Creating Boot Configuration Is Supported**

554 Subclause 7.6.1 describes conditional behavior. Subclause 7.6.1 describes the CIM elements and
555 behaviors that shall be implemented when the following conditions are met.

556 Conditional Requirement:

- 557 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
558 through an instance of CIM_ElementCapabilities.
- 559 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 2
560 (Creates Boot Configuration).

561 The implementation shall support the CreateBootConfigSetting() method.

562 **7.6.2 Creating Boot Configuration Is Not Supported**

563 Subclause 7.6.2 describes conditional behavior. Subclause 7.6.2 describes the CIM elements and
564 behaviors that shall be implemented when either of the following conditions are met.

565 Conditional Requirement 1:

- 566 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
567 through an instance of CIM_ElementCapabilities.
- 568 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array does not contain a
569 value of 2 (Creates Boot Configuration).

570 Conditional Requirement 2:

- 571 1) An instance of CIM_BootServiceCapabilities is not associated with the CIM_BootService
572 instance through an instance of CIM_ElementCapabilities.

573 When either of the preceding conditions are met, the implementation shall not support the
574 CreateBootConfigSetting() method.

575 **7.7 Deleting a Boot Configuration**

576 Subclause 7.7 describes conditional behavior.

577 Conditional Requirement: The implementation shall support the client deleting or removing an existing
578 boot configuration through the DeleteInstance() intrinsic operation, when the implementation supports the
579 creation of a new boot configuration.

580 This conditional behavior shall be determined with the same mechanism used to determine that an
581 implementation supports the creation of a new boot configuration. See subclause 7.6.

582 **7.8 Identifying Boot Sources**

583 Subclause 7.8 describes optional behavior.

584 An instance of CIM_BootSourceSetting represents a source from which a boot image can be loaded
585 during the boot process.

586 An instance of CIM_BootSourceSetting shall be associated to one or more instances of
587 CIM_BootConfigSetting.

588 The CIM_BootSourceSetting class has three boot string properties: BootString, BIOSBootString and
589 StructuredBootString. The BootString and BIOSBootString properties may be supported. The
590 StructuredBootString property should be supported.

591 **7.8.1 CIM_BootServiceCapabilities**

592 When no instance of CIM_BootServiceCapabilities exists, it is not possible to determine, via the
593 CIM_BootServiceCapabilities, which boot string properties are supported.

594 **7.8.1.1 CIM_BootServiceCapabilities.BootStringsSupported**

595 When an instance of CIM_BootServiceCapabilities exists, its BootStringsSupported property array shall
596 contain one or more of the values 1 (BootString), 2 (BIOSBootString) and 3 (StructuredBootString).

597 The presence of a value in the property array means that the specified boot string in each instance of
598 CIM_BootSourceSettings which are associated to an instance of CIM_BootConfigSetting, which in turn is
599 associated to the CIM_BootService, shall not be NULL.

600 **7.8.2 CIM_BootSourceSetting.ElementName Property**

601 The CIM_BootSourceSetting.ElementName property shall be a character string of variable length
602 (pattern ".*").

603 The ElementName property shall contain a string that identifies the boot source.

604 When the CIM_BootSourceSetting.BIOSBootString property is not null, the ElementName property shall
605 match the BIOSBootString property.

606 **7.8.3 CIM_BootSourceSetting.BootString Property**

607 An implementation may support the CIM_BootSourceSetting.BootString property.

608 **7.8.3.1 CIM_BootSourceSetting.BootString Property Is Supported**

609 Subclause 7.8.3.1 describes conditional behavior. Subclause 7.8.3.1 describes the CIM elements and
610 behaviors that shall be implemented when either of the following conditions are met.

611 Conditional Requirement:

612 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
613 through an instance of CIM_ElementCapabilities.

614 2) The CIM_BootServiceCapabilities.BootStringsSupported property array contains a value of 2
615 (BootString).

616 The CIM_BootSourceSetting.BootString property shall contain a character string.

617 The CIM_BootSourceSetting.BootString property shall contain a string that identifies the boot source. The
618 property may include additional information to be used during the boot process. Examples include a
619 specific address of a bootable partition, flags to request the loading of a kernel debugger, or name of the
620 kernel image.

621 **7.8.3.2 CIM_BootSourceSetting.BootString Property Is Not Supported**

622 Subclause 7.8.3.2 describes conditional behavior. Subclause 7.8.3.2 describes the CIM elements and
623 behaviors that shall be implemented when either of the following conditions are met.

624 Conditional Requirement 1:

625 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
626 through an instance of CIM_ElementCapabilities.

627 2) The CIM_BootServiceCapabilities.BootStringsSupported property array does not contain a
628 value of 2 (BootString).

629 Conditional Requirement 2:

630 1) An instance of CIM_BootServiceCapabilities is not associated with the CIM_BootService
631 instance through an instance of CIM_ElementCapabilities.

632 The CIM_BootSourceSetting.BootString property may be NULL.

633 **7.8.4 CIM_BootSourceSetting.BIOSBootString Property**

634 An implementation may support the CIM_BootSourceSetting.BIOSBootString property.

635 **7.8.4.1 CIM_BootSourceSetting.BIOSBootString Property Is Supported**

636 Subclause 7.8.4.1 describes conditional behavior. Subclause 7.8.4.1 describes the CIM elements and
637 behaviors that shall be implemented when either of the following conditions are met.

638 Conditional Requirement:

- 639 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
640 through an instance of CIM_ElementCapabilities.
- 641 2) The CIM_BootServiceCapabilities.BootStringsSupported property array contains a value of 3
642 (BIOSBootString).

643 The CIM_BootSourceSetting.BIOSBootString property shall contain a character string of variable length
644 (pattern ".*").

645 The CIM_BootSourceSetting.BIOSBootString property shall contain a string that identifies the boot
646 source. The property shall match the string used by the BIOS to uniquely name the boot source in its
647 namespace.

648 For an UEFI BIOS, the BIOSBootString property should match the output of the
649 EFI_DEVICE_PATH_TO_TEXT_PROTOCOL service.

650 **7.8.4.2 CIM_BootSourceSetting.BIOSBootString Property Is Not Supported**

651 Subclause 7.8.4.2 describes conditional behavior. Subclause 7.8.4.2 describes the CIM elements and
652 behaviors that shall be implemented when either of the following conditions are met.

653 Conditional Requirement 1:

- 654 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
655 through an instance of CIM_ElementCapabilities.
- 656 2) The CIM_BootServiceCapabilities.BootStringsSupported property array does not contain a
657 value of 3 (BIOSBootString).

658 Conditional Requirement 2:

- 659 1) An instance of CIM_BootServiceCapabilities is not associated with the CIM_BootService
660 instance through an instance of CIM_ElementCapabilities.

661 The CIM_BootSourceSetting.BIOSBootString property may be NULL.

662 **7.8.5 CIM_BootSourceSetting.StructuredBootString Property**

663 An implementation should support the CIM_BootSourceSetting.StructuredBootString property.

664 **7.8.5.1 CIM_BootSourceSetting.StructuredBootString Property Is Supported**

665 Subclause 7.8.5.1 describes conditional behavior. Subclause 7.8.5.1 describes the CIM elements and
666 behaviors that shall be implemented when either of the following conditions are met.

667 Conditional Requirement:

- 668 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
669 through an instance of CIM_ElementCapabilities.
- 670 2) The CIM_BootServiceCapabilities.BootStringsSupported property array contains a value of 4
671 (StructuredBootString).

672 The CIM_BootSourceSetting.StructuredBootString property shall contain a string that identifies the boot
673 source using the following format:

674 " <OrgID>:<identifier>:<index> "

675 The value of <OrgID> shall include a copyrighted, trademarked or otherwise unique name that is owned
 676 by the entity creating or defining the CIM_BootSourceSetting, or is a registered ID that is assigned to the
 677 entity by a recognized global authority. In addition, <OrgID> shall not contain a colon (:). For DMTF
 678 defined instances, the algorithm shall be used with the <OrgID> set to "CIM".

679 The value of <index> shall be an unsigned integer. When the value of <OrgID> matches "CIM", the value
 680 of the <identifier> shall be one of the identifiers listed in Table 2.

681 **Table 2 – Structured Name Identifiers**

Identifier	Description
"Unknown"	The boot device type is unknown
"Floppy"	Boot from a floppy device
"Hard-Disk"	Boot from a hard drive device
"CD/DVD"	Boot from a CD or DVD device
"Network"	Boot from a network device
"PCMCIA"	Boot from a PCMCIA device
"BEV"	Boot from a Boot Entry Vector device
"USB"	Boot from a USB device

682 **7.8.5.2 CIM_BootSourceSetting.StructuredBootString Property Is Not Supported**

683 Subclause 7.8.5.2 describes conditional behavior. Subclause 7.8.5.2 describes the CIM elements and
 684 behaviors that shall be implemented when either of the following conditions are met.

685 Conditional Requirement 1:

- 686 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
 687 through an instance of CIM_ElementCapabilities.
- 688 2) The CIM_BootServiceCapabilities.BootStringsSupported property array does not contain a
 689 value of 4 (StructuredBootString).

690 Conditional Requirement 2:

- 691 1) An instance of CIM_BootServiceCapabilities is not associated with the CIM_BootService
 692 instance through an instance of CIM_ElementCapabilities.

693 The CIM_BootSourceSetting.StructuredBootString property may be NULL.

694 **7.8.6 CIM_ConcreteDependency Association**

695 An instance of a concrete subclass of CIM_LogicalDevice may exist, which represents the boot source
 696 device.

697 If such an instance of CIM_LogicalDevice is instantiated, then it shall be associated with an instance of
 698 CIM_BootSourceSetting using an instance of CIM_ConcreteDependency.

699 When the association is used in this manner, its Antecedent property shall reference the instance of a
 700 concrete subclass of CIM_LogicalDevice and its Dependent property shall reference the
 701 CIM_BootSourceSetting instance.

702 **7.9 Changing the Boot Order**

703 A Boot Configuration Representation may support the client changing the boot order of the boot sources
 704 associated to an instance of CIM_BootConfigSetting through the
 705 CIM_BootConfigSetting.ChangeBootOrder() method.

706 **7.9.1 Changing Boot Order Is Supported**

707 Subclause 7.9.1 describes conditional behavior. Subclause 7.9.1 describes the CIM elements and
708 behaviors that shall be implemented when either of the following conditions are met.

709 Conditional Requirement 1:

- 710 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
711 through an instance of CIM_ElementCapabilities.
- 712 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array does not contain a
713 value of 6 (Change Boot Order Not Supported).

714 Conditional Requirement 2:

- 715 1) An instance of CIM_BootServiceCapabilities is not associated with the CIM_BootService
716 instance through an instance of CIM_ElementCapabilities.

717 When either of the preceding conditions are met, the implementation shall support the
718 ChangeBootOrder() method.

719 **7.9.2 Changing Boot Order Is Not Supported**

720 Subclause 7.9.2 describes conditional behavior. Subclause 7.9.2 describes the CIM elements and
721 behaviors that shall be implemented when the following conditions are met.

722 Conditional Requirement:

- 723 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
724 through an instance of CIM_ElementCapabilities.
- 725 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 6
726 (Change Boot Order Not Supported).

727 The implementation shall not support the ChangeBootOrder() method.

728 **7.10 Representing a Set of Aggregated Boot Sources**

729 Subclause 7.9 describes optional behavior.

730 An instance of CIM_BootSourceSetting may represent an aggregated boot source. An example of an
731 aggregated boot source is a BCV.

732 When an aggregated boot source is represented, it shall be associated to a representation of the set of
733 aggregated boot sources. The following requirements shall apply.

734 **7.10.1 Aggregated Boot Sources**

735 An instance of CIM_BootSourceSetting shall exist representing an aggregated boot source.

736 **7.10.2 Aggregated Boot Configuration**

737 An instance of CIM_BootConfigSetting shall exist representing the set of aggregated boot sources.

738 The ElementName property for the instance of CIM_BootConfigSetting representing the set of
739 aggregated boot sources shall match the value of the ElementName property of the instance of
740 CIM_BootSourceSetting that represents the aggregated boot source.

741 **7.10.3 Logical Identity Relationship**

742 An instance of CIM_LogicalIdentity shall associate the instance of CIM_BootSourceSetting with the
743 instance of CIM_BootConfigSetting.

744 **7.10.3.1 CIM_LogicalIdentity.SystemElement**

745 The value of the SystemElement reference shall be the instance of CIM_BootSourceSetting that
746 represents the aggregated boot source.

747 **7.10.3.2 CIM_LogicalIdentity.SameElement**

748 The value of the SameElement reference shall be the instance of CIM_BootConfigSetting that represents
749 the set of aggregated boot sources.

750 **7.11 Boot Order During the Boot Process**

751 Subclause 7.11 describes the CIM elements and behaviors that shall be implemented to define the order
752 or sequence in which the boot sources are used during the boot process.

753 **7.11.1 CIM_OrderedComponent Association**

754 The CIM_OrderedComponent association class shall be used to associate instance of
755 CIM_BootConfigSetting to each instance of CIM_BootSourceSetting representing one of the boot sources
756 in the boot configuration.

757 When the association is used in this manner, its GroupComponent property shall reference the
758 CIM_BootConfigSetting instance and its PartComponent property shall reference the
759 CIM_BootSourceSetting instance.

760 **7.11.1.1 CIM_OrderedComponent.AssignedSequence Property**

761 When a CIM_BootConfigSetting instance has multiple CIM_BootSourceSetting instances associated to it
762 through instances of the CIM_OrderedComponent association, the value of the
763 CIM_OrderedComponent.AssignedSequence property shall be used to determine the sequence in which
764 the associated CIM_BootSourceSetting instances are used during the boot process.

765 The value of the AssignedSequence property across instances of CIM_OrderedComponent that
766 reference the same CIM_BootConfigSetting shall be unique when it is not equal to zero.

767 The boot order shall be interpreted as follows:

- 768 • The AssignedSequence properties are compared across instances of CIM_OrderedComponent
769 that reference the same CIM_BootConfigSetting.
- 770 • A CIM_BootSourceSetting whose associated CIM_OrderedComponent.AssignedSequence
771 property is equal to zero shall be ignored and not considered part of the boot order.
- 772 • The boot order shall proceed from the lowest to the highest non-zero integer value of the
773 AssignedSequence properties.

774 **7.11.2 CIM_BootSourceSetting.FailThroughSupported**

775 The FailThroughSupported property shall describe the behavior of the boot process when the attempt to
776 boot from a boot device represented by an instance of CIM_BootSourceSetting is not successful.

777 When the FailThroughSupported property has a value of 1 (Is Supported), an unsuccessful boot attempt
778 shall result in continuing through the ordered list for boot sources from which to attempt to boot.

779 When the FailThroughSupported property has a value of 2 (Is Not Supported), then an unsuccessful boot
780 attempt shall result in the termination of the boot order for the remaining instances of
781 CIM_BootSourceSetting associated to the same instance of CIM_BootConfigSetting.

782 **7.12 Settings to Apply During the Boot Process**

783 Subclause 7.12 describes optional behavior. Subclause 7.12 describes the CIM elements and behaviors
784 that may be implemented to apply settings during the boot process.

785 During the boot process, settings can be applied to managed elements or the boot process itself. A
786 setting can be applicable to an entire configuration or to a specific boot source.

787 **7.12.1 Settings that Apply to a Managed Element**

788 An instance of a concrete subclass of CIM_SettingData represents a setting that is applied to a managed
789 element during the boot process. The instance shall be associated to either an instance of
790 CIM_BootConfigSetting or an instance of CIM_BootSourceSetting through an instance of
791 CIM_ConcreteComponent.

792 When a setting to a managed element is applicable to an entire boot configuration, an instance of a
793 concrete subclass of CIM_SettingData shall be associated to the instance of CIM_BootConfigSetting
794 representing the boot configuration through an instance of CIM_ConcreteComponent.

795 When the CIM_ConcreteComponent association is used in this manner, its GroupComponent property
796 shall reference the CIM_BootConfigSetting instance and its PartComponent property shall reference the
797 CIM_SettingData instance.

798 When a setting to a managed element is applicable to a specific boot source, an instance of a concrete
799 subclass of CIM_SettingData shall be associated to the instance of CIM_BootSourceSetting representing
800 the boot configuration through an instance of CIM_ConcreteComponent.

801 When the CIM_ConcreteComponent association is used in this manner, its GroupComponent property
802 shall reference the CIM_BootSourceSetting instance and its PartComponent property shall reference the
803 CIM_SettingData instance.

804 **7.12.2 Settings that Apply to the Boot Process**

805 An instance of a concrete subclass of CIM_BootSettingData represents a setting that is applied during the
806 boot process but does not apply to a managed element. The setting can apply to an entire boot
807 configuration or to a specific boot source.

808 When an instance of CIM_BootSettingData is instantiated, then it shall be associated with an instance of
809 CIM_BootConfigSetting or CIM_BootSourceSetting using an instance of CIM_ConcreteComponent.

810 When the CIM_ConcreteComponent association is used in this manner, its GroupComponent property
811 shall reference the CIM_BootConfigSetting or CIM_BootSourceSetting instance and its PartComponent
812 property shall reference the CIM_BootSettingData instance.

813 When an instance of a concrete subclass of CIM_SettingData is instantiated, then it shall be associated
814 with an instance of CIM_BootConfigSetting or CIM_BootSourceSetting using an instance of
815 CIM_ConcreteComponent.

816 When the CIM_ConcreteComponent association is used in this manner, its GroupComponent property
817 shall reference the CIM_BootConfigSetting or CIM_BootSourceSetting instance and its PartComponent
818 property shall reference the instance of a concrete subclass of CIM_SettingData.

819 **8 Methods**

820 Clause 8 details the requirements for supporting intrinsic operations and extrinsic methods for the CIM
821 elements defined by this profile.

822 **8.1 CIM_BootService.CreateBootConfigSetting()**

823 This method is conditional on the CIM_BootServiceCapabilities.BootConfigCapabilities property array
824 containing a value of 2 (Creates Boot Configuration). For more information, see subclause 7.6.1.

825 The CreateBootConfigSetting() method shall create a clone of an existing Boot Configuration using a
826 Template Boot Configuration and associate the new Boot Configuration to the Boot Configurable System.
827 The method has two input parameters: StartingBootConfig and ScopingComputerSystem. At least one of
828 the two parameters shall be non-null for the method to be successfully invoked.

829 The input parameter, StartingBootConfig, shall be used to provide a reference to the Template Boot
830 Configuration to use as the template for the new Boot Configuration Representation.

831 The input parameter, ScopingComputerSystem, shall be used to reference the Boot Configurable
832 System, an existing CIM_ComputerSystem, to which the new CIM_BootConfigSetting instance shall be
833 associated through an instance of CIM_ElementSettingData.

834 When the StartingBootConfig parameter and the ScopingComputerSystem parameter are both NULL, a
835 return value or an exception shall be returned. When a return value is returned, it shall have a value of 2
836 (Error Occurred).

837 When the StartingBootConfig parameter has a NULL value and the ScopingComputerSystem parameter
838 has a non-NULL value, the implementation shall find the Default Boot Configuration associated to the
839 CIM_ComputerSystem instance referenced by the ScopingComputerSystem and use it as the Template
840 Boot Configuration for the new boot configuration. If a Default Boot Configuration is not found, a return
841 value or an exception shall be returned. When a return value is returned, it shall have a value of 2 (Error
842 Occurred).

843 When the StartingBootConfig parameter has a non-NULL value and the ScopingComputerSystem
844 parameter is NULL, the implementation shall associate the new boot configuration to the Boot
845 Configurable System of the Template Boot Configuration.

846 Upon successful completion of this method, a new Boot Configuration Representation shall exist and be a
847 replica of the Template Boot Configuration. The new instance of CIM_BootConfigSetting shall be
848 associated to the instance representing the Boot Configurable System through an instance of
849 CIM_ElementSettingData. All properties in the new Boot Configuration Representation and Template
850 Boot Configuration representations are expected to have the same value, except for the key properties,
851 unless otherwise mandated in the requirements below.

- 852 • A new instance of CIM_BootConfigSetting shall exist and be referenced by the output
853 NewBootConfig parameter. The new CIM_BootConfigSetting.InstanceID property shall be set to
854 a unique value.
- 855 • A new instance of CIM_ElementSettingData shall exist that associates the new
856 CIM_BootConfigSetting to the instance of the Boot Configurable System, which is specified by
857 the ScopingComputerSystem parameter when it is non-NULL or implied by the
858 StartingBootConfig parameter when the ScopingComputerSystem parameter is NULL.
- 859 • The CIM_ElementSettingData.IsDefault property shall be set to 2 (Is Not Default). The
860 CIM_ElementSettingData.IsCurrent property shall be set to 2 (Is Not Current). The
861 CIM_ElementSettingData.IsNext property shall be set to 2 (Is Not Next).

- 862 • New instances of CIM_BootSourceSetting shall exist, along with instances of
863 CIM_OrderedComponent, when they are present in the boot configuration represented by the
864 Template Boot Configuration. The new instances shall be duplicates of those found in the boot
865 configuration represented by the Template Boot Configuration, except for the key property
866 value.
- 867 • New instances of CIM_BootSettingData shall exist when they are present in the boot
868 configuration represented by the Template Boot Configuration. The new instances shall be
869 duplicates of those found in the boot configuration represented by the Template Boot
870 Configuration, except for the key property value.
- 871 • New instances of CIM_ConcreteComponent shall exist when they are present in the boot
872 configuration represented by the Template Boot Configuration.
- 873 • New instances of CIM_ConcreteDependency shall exist when they are present in the boot
874 configuration represented by the Template Boot Configuration.
- 875 • CIM elements that are defined in a Referencing Profile are not copied.

876 The return code values and parameters for the CreateBootConfigSetting() method are specified in
877 Table 3 and Table 4.

878 No standard messages are defined.

879 **Table 3 – CreateBootConfigSetting() Method: Return Code Values**

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred
4096	Job started

880 **Table 4 – CreateBootConfigSetting() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	ScopingComputerSystem	CIM_ComputerSystem REF	Reference to an existing CIM_ComputerSystem instance
IN, REQ	StartingBootConfig	CIM_BootConfigSetting REF	Reference to an existing CIM_BootConfigSetting instance
OUT	NewBootConfig	CIM_BootConfigSetting REF	Reference to the newly created CIM_BootConfigSetting
OUT	Job	CIM_ConcreteJob REF	Reference to a CIM_ConcreteJob returned if job started

881 **8.2 CIM_BootService.ApplyBootConfigSetting()**

882 This method is conditional on the CIM_BootServiceCapabilities.BootConfigCapabilities property array
883 containing a value of 3 (Applies Boot Configuration). See subclause 7.5.1 for more information.

884 NOTE: Successful execution of the ApplyBootConfigSetting() method can independently apply a boot configuration to
885 a Boot Configurable System regardless of the Next Boot Configuration. The requirements in subclause 7.4.4 shall not
886 apply when a Boot Configurable System is booted using the ApplyBootConfigSetting() method.

887 The ApplyBootConfigSetting() method shall start the boot process on a specified Boot Configurable
 888 System, using the specified boot configuration of the Boot Configurable System. The boot process may
 889 be started from a pause in the boot flow or from a reboot of the Boot Configurable System. The method
 890 has two input parameters, ScopingComputerSystem and ApplyBootConfig.

891 The input parameter, ScopingComputerSystem, shall be used to reference the Boot Configurable
 892 System, an existing CIM_ComputerSystem with instances CIM_BootConfigSetting associated to it
 893 through an instance of CIM_ElementSettingData.

894 When the ScopingComputerSystem parameter is NULL, the boot configuration shall be applied to each
 895 CIM_ComputerSystem which is associated to the instance of CIM_BootConfigSetting referenced by the
 896 ApplyBootConfig parameter via an instance of CIM_ElementSettingData.

897 When the instance of CIM_ComputerSystem referenced by ScopingComputerSystem parameter is not
 898 associated to an instance of CIM_BootService, a return value or an exception shall be returned. When a
 899 return value is returned, it shall have a value of 2 (Error Occurred).

900 The input parameter, ApplyBootConfig, shall be used to provide a reference to an instance of
 901 CIM_BootConfigSetting associated to the Boot Configurable System for use in the boot process.

902 When the ApplyBootConfig parameter is NULL, a return value or an exception shall be returned. When a
 903 return value is returned, it shall have a value of 2 (Error Occurred).

904 When the instance of CIM_BootConfigSetting referenced by ApplyBootConfig parameter is not found, a
 905 return value or an exception shall be returned. When a return value is returned, it shall have a value of 2
 906 (Error Occurred).

907 When the instance of CIM_BootConfigSetting referenced by ApplyBootConfig parameter is not
 908 associated with the ScopingComputerSystem, a return value or an exception shall be returned. When a
 909 return value is returned, it shall have a value of 2 (Error Occurred).

910 Upon successful completion of this method, the boot process shall have started using the boot
 911 configuration referenced by the ApplyBootConfig parameter.

912 The return code values and parameters for the ApplyBootConfigSetting() method are specified in Table 5,
 913 respectively.

914 No standard messages are defined.

915 **Table 5 – ApplyBootConfigSetting() Method: Return Code Values**

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred
4096	Job started

916 **Table 6 – ApplyBootConfigSetting() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	ScopingComputerSystem	CIM_ComputerSystem REF	Reference to an existing CIM_ComputerSystem instance
IN, REQ	ApplyBootConfig	CIM_BootConfigSetting REF	Reference to an existing CIM_BootConfigSetting instance
OUT	Job	CIM_ConcreteJob REF	Reference to a CIM_ConcreteJob returned if job started

917 **8.3 CIM_BootConfigSetting.ChangeBootOrder()**

918 The ChangeBootOrder() method shall set the order in which the instances of CIM_BootSourceSetting are
 919 associated to a CIM_BootConfigSetting instance. The method has one input parameter: Source.

920 When the ChangeBootOrder() method is not supported, a return value or an exception shall be returned.

921 The input parameter, Source, is an ordered array of references to CIM_BootSourceSetting instances that
 922 defines the new sequence of the CIM_BootSourceSetting instances associated to the instance of
 923 CIM_BootConfigSetting. Each CIM_BootSourceSetting instance in the array shall already be associated
 924 with this CIM_BootConfigSetting instance through an instance of CIM_OrderedComponent. This
 925 parameter is required.

926 When the Source parameter is NULL, a return value of 2 (Error Occurred) shall be returned.

927 When any of the CIM_BootSourceSetting instance in the Source array are not associated to the instance
 928 of CIM_BootConfigSetting, the implementation shall return a value of 2 (Error Occurred).

929 Upon successful completion of this method, the value of the AssignedSequence property on each
 930 instance of CIM_OrderedComponent shall be updated such that the values are monotonically increasing
 931 in correlation with the position of the referenced CIM_BootSourceSetting instance in the Source input
 932 parameter. That is, the first position in the array shall have the lowest non-zero value for
 933 AssignedSequence. The second position will have the second lowest value, and so on.

934 Upon successful completion of this method, the value of the AssignedSequence property on each
 935 instance of CIM_OrderedComponent, that associates the target CIM_BootConfigSetting instance to a
 936 CIM_BootSourceSetting instance that is not present in the input array, shall be assigned a value of 0.

937 The return code values and parameters for the ChangeBootOrder() method are specified in Table 7 and
 938 Table 8, respectively.

939 No standard messages are defined.

940 **Table 7 – ChangeBootOrder() Method: Return Code Values**

Value	Description
0	Completed with No Error
1	Not Supported
2	Error Occurred
4096	Job Started

941 **Table 8 – ChangeBootOrder() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	Source[]	CIM_BootSourceSetting REF	An ordered array of references to CIM_BootSourceSetting instances
OUT	Job	CIM_ConcreteJob REF	Reference to a CIM_ConcreteJob Returned if job started.

942 **8.4 Profile Conventions for Operations**

943 Support for operations for each profile class (including associations) is specified in the following
 944 subclauses. Each of these subclauses includes a table listing all the operations supported by this profile.
 945 Compliant implementations of this profile shall support all these operations.

946 **8.5 CIM_BootService**

947 Compliant implementations of this profile shall support the operations listed in Table 9 for
 948 CIM_BootService. Each operation shall be supported as defined in [DSP0200](#).

949 **Table 9 – Operations: CIM_BootService**

Operation	Requirement	Messages
GetInstance	Mandatory	None
ModifyInstance	Optional	See subclause 8.5.1.
Associators	Mandatory	None
AssociatorNames	Mandatory	None
References	Mandatory	None
ReferenceNames	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

950 **8.5.1 CIM_BootService — ModifyInstance Operation**

951 Subclause 8.5.1 details the specific requirements for the ModifyInstance operation applied to an instance
 952 of CIM_BootService.

953 When an instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance and
 954 the CIM_BootServiceCapabilities.ElementNameEditSupported property has a value of TRUE, the
 955 implementation shall allow the ModifyInstance operation to change the value of the ElementName
 956 property of the CIM_BootService instance. The ModifyInstance operation shall enforce the length
 957 restriction specified in the MaxElementNameLen property of the CIM_BootServiceCapabilities instance.

958 When no instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance, or
 959 the ElementNameEditSupported property of the CIM_BootServiceCapabilities has a value of FALSE, the
 960 implementation shall not allow the ModifyInstance operation to change the value of the ElementName
 961 property of the CIM_BootService instance.

962 **8.6 CIM_BootConfigSetting**

963 Compliant implementations of this profile shall support the operations listed in Table 10 for the
 964 CIM_BootConfigSetting class. Each operation shall be supported as defined in [DSP0200](#).

965 **Table 10 – Operations: CIM_BootConfigSetting**

Operation	Requirement	Messages
DeleteInstance	Conditional	See subclause 8.6.1.
GetInstance	Mandatory	None
Associators	Mandatory	None
AssociatorNames	Mandatory	None
References	Mandatory	None
ReferenceNames	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

966 **8.6.1 CIM_BootConfigSetting – DeleteInstance**

967 Subclause 8.6.1 describes conditional behavior.

968 Conditional requirement: Subclause 7.7 describes the conditions when the DeleteInstance operation shall
 969 be supported. Implementations may choose to support the DeleteInstance operation even when the
 970 conditions described in subclause 7.7 are not met.

971 When the DeleteInstance operation is supported for an instance of CIM_BootConfigSetting, upon
 972 completion of this operation, the following instances shall be deleted:

- 973 • The target instance of CIM_BootConfigSetting shall no longer exist.
- 974 • The instance of CIM_ElementSettingData that associated the target CIM_BootConfigSetting to
 975 the instance of CIM_ComputerSystem shall no longer exist.
- 976 • The instances of CIM_ConcreteComponent, which associate the target instance of
 977 CIM_BootConfigSetting to instances of a concrete subclass of CIM_SettingData, shall no longer
 978 exist.
- 979 • The instances of CIM_ConcreteComponent, which associate the target instance of
 980 CIM_BootConfigSetting to instances of a concrete subclass of CIM_BootSettingData, shall no
 981 longer exist. The instances of the associated concrete subclass of CIM_BootSettingData shall
 982 no longer exist.
- 983 • The instances of CIM_OrderedComponent, which associate the target instance of
 984 CIM_BootConfigSetting to instances of CIM_BootSourceSetting, shall no longer exist. The
 985 instances of the associated CIM_BootSourceSetting shall no longer exist.
- 986 • The instances of CIM_ConcreteComponent, which associate instances of a concrete subclass
 987 of CIM_SettingData to instances of CIM_BootSourceSetting, which in turn are associated to the
 988 target instance of CIM_BootConfigSetting, shall no longer exist.
- 989 • The instances of CIM_ConcreteComponent, which associate instances of a concrete subclass
 990 of CIM_BootSettingData to instances of CIM_BootSourceSetting, which in turn are associated
 991 to the target instance of CIM_BootConfigSetting, shall no longer exist. The instances of the
 992 associated concrete subclass of CIM_BootSettingData shall no longer exist.
- 993 • The instances of CIM_ConcreteDependency, which associate instances of a concrete subclass
 994 of CIM_LogicalDevice to instances of CIM_BootSourceSetting, shall no longer exist.
- 995 • The instance of CIM_LogicalIdentity, which associates a deleted instance of
 996 CIM_BootSourceSetting to an instance of CIM_BootConfigSetting, shall no longer exist. The
 997 associated instance of CIM_BootConfigSetting shall no longer exist. The requirements in this
 998 subclause shall be applied recursively to the deleted CIM_BootConfigSetting instance.

999 **8.7 CIM_BootSettingData**

1000 Compliant implementations of this profile shall support the operations listed in Table 11 for the
 1001 CIM_BootSettingData class. Each operation shall be supported as defined in [DSP0200](#).

1002 **Table 11 – Operations: CIM_BootSettingData**

Operation	Requirement	Messages
GetInstance	Mandatory	None
Associators	Mandatory	None
AssociatorNames	Mandatory	None

Operation	Requirement	Messages
References	Mandatory	None
ReferenceNames	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

1003 **8.8 CIM_BootSourceSetting**

1004 Compliant implementations of this profile shall support the operations listed in Table 12 for the
 1005 CIM_BootSourceSetting class. Each operation shall be supported as defined in [DSP0200](#).

1006 **Table 12 – Operations: CIM_BootSourceSetting**

Operation	Requirement	Messages
GetInstance	Mandatory	None
Associators	Mandatory	None
AssociatorNames	Mandatory	None
References	Mandatory	None
ReferenceNames	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

1007 **8.9 CIM_ConcreteComponent**

1008 Compliant implementations of this profile shall support the operations listed in Table 13 for the
 1009 CIM_ConcreteComponent class. Each operation shall be supported as defined in [DSP0200](#).

1010 **Table 13 – Operations: CIM_ConcreteComponent**

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

1011 **8.10 CIM_ConcreteDependency**

1012 Compliant implementations of this profile shall support the operations listed in Table 14 for the
 1013 CIM_ConcreteDependency class. Each operation shall be supported as defined in [DSP0200](#).

1014 **Table 14 – Operations: CIM_ConcreteDependency**

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

1015 **8.11 CIM_ElementCapabilities**

1016 Compliant implementations of this profile shall support the operations listed in Table 15 for the
 1017 CIM_ElementCapabilities class. Each operation shall be supported as defined in [DSP0200](#).

1018 **Table 15 – Operations: CIM_ElementCapabilities**

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

1019 **8.12 CIM_ElementSettingData**

1020 Compliant implementations of this profile shall support the operations listed in Table 16 for the
 1021 CIM_ElementSettingData class. Each operation shall be supported as defined in [DSP0200](#).

1022 **Table 16 – Operations: CIM_ElementSettingData**

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None
ModifyInstance	Optional	See subclause 8.12.1.

1023 **8.12.1 CIM_ElementSettingData – ModifyInstance Operation**

1024 The behavior of the ModifyInstance operation varies depending on the property of the association
 1025 modified.

1026 **8.12.1.1 CIM_ElementSettingData.IsDefault Property**

1027 When the ModifyInstance operation is used to set the IsDefault property to a value of 1 (Is Default), the
 1028 ModifyInstance operation shall implement the following behavior.

1029 The behavior described insures that there is at most one instance of CIM_ElementSettingData associated
 1030 to the Boot Configurable System whose IsDefault property has a value of 1 (Is Default) as specified in
 1031 subclause 7.4.2, by first finding any existing instance of CIM_ElementSettingData whose IsDefault
 1032 property already has a value of 1 (Is Default) and modifying the value to 2 (Is Not Default).

- 1033 • Search for an instance of CIM_ElementSettingData that associates an instance of
 1034 CIM_BootConfigSetting with the instance of CIM_ComputerSystem, which is referenced by the
 1035 target instance of CIM_ElementSettingData where the IsDefault property has a value of 1 (Is
 1036 Default).
- 1037 • If such an instance of CIM_ElementSettingData is found, the ModifyInstance operation shall set
 1038 the value of the IsDefault property to 2 (Is Not Default).
- 1039 • For the target instance of CIM_ElementSettingData, when the IsDefault property already has a
 1040 value of 1 (Is Default), the ModifyInstance operation shall complete successfully.
- 1041 • For the target instance of CIM_ElementSettingData, set the value of the IsDefault property to 1
 1042 (Is Default).

1043 8.12.1.2 CIM_ElementSettingData.IsNext Property

1044 When the ModifyInstance operation is used to set the IsNext property to a value of 1 (Is Next), the
1045 ModifyInstance operation shall implement the following behavior.

1046 The behavior described insures that there is at most one instance of CIM_ElementSettingData associated
1047 to the Boot Configurable System whose IsNext property has a value of 1 (Is Next) as specified in
1048 subclause 7.4.4, by first finding any existing instance of CIM_ElementSettingData whose IsNext property
1049 already has a value of 1 (Is Next) and modifying the value to 2 (Is Not Next).

- 1050 • Search for an instance of CIM_ElementSettingData that associates an instance of
1051 CIM_BootConfigSetting with the instance of CIM_ComputerSystem, which is referenced by the
1052 target instance of CIM_ElementSettingData where the IsNext property has a value of 1 (Is
1053 Next).
- 1054 • If such an instance of CIM_ElementSettingData is found, the ModifyInstance operation shall set
1055 the value of the IsNext property to 2 (Is Not Next).
- 1056 • For the target instance of CIM_ElementSettingData, when the IsNext property already has a
1057 value of 1 (Is Next), the ModifyInstance operation shall complete successfully.
- 1058 • For the target instance of CIM_ElementSettingData, when the IsNext property has a value other
1059 than 1 (Is Next), set the value of the IsNext property to 1 (Is Next).

1060 When the ModifyInstance operation is used to set the IsNext property to a value of 3 (Is Next For Single
1061 Use), the ModifyInstance operation shall implement the following behavior.

1062 The behavior described insures that there is at most one instance of CIM_ElementSettingData associated
1063 to the Boot Configurable System whose IsNext property has a value of 3 (Is Next For Single Use) as
1064 specified in subclause 7.4.5, by first finding any existing instance of CIM_ElementSettingData whose
1065 IsNext property already has a value of 3 (Is Next For Single Use) and modifying the value to 2 (Is Not
1066 Next).

- 1067 • For the target instance of CIM_ElementSettingData, when the IsNext property has a value of 1
1068 (Is Next), the ModifyInstance operation shall fail.
- 1069 • Search for an instance of CIM_ElementSettingData that associates an instance of
1070 CIM_BootConfigSetting with the instance of CIM_ComputerSystem referenced by the target
1071 instance of CIM_ElementSettingData where the IsNext property has a value of 3 (Is Next For
1072 Single Use).
- 1073 • If such an instance of CIM_ElementSettingData is found, the ModifyInstance operation shall set
1074 the value of the IsNext property to 2 (Is Not Next).
- 1075 • For the target instance of CIM_ElementSettingData, when the IsNext property already has a
1076 value of 3 (Is Next For Single Use), the ModifyInstance operation shall complete successfully.
- 1077 • For the target instance of CIM_ElementSettingData, when the IsNext property has a value
1078 neither 1 (Is Next) nor 3 (Is Next For Single Use), set the value of the IsNext property to 3 (Is
1079 Next For Single Use).

1080 8.12.1.3 CIM_ElementSettingData.IsCurrent Property

1081 The ModifyInstance operation shall not be used to set the IsCurrent property.

1082 8.13 CIM_BootServiceCapabilities

1083 Compliant implementations of this profile shall support the operations listed in Table 17 for the
1084 CIM_BootServiceCapabilities class. Each operation shall be supported as defined in [DSP0200](#).

1085

Table 17 – Operations: CIM_BootServiceCapabilities

Operation	Requirement	Messages
GetInstance	Mandatory	None
Associators	Mandatory	None
AssociatorNames	Mandatory	None
References	Mandatory	None
ReferenceNames	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

1086

8.14 CIM_HostedService

1087

Compliant implementations of this profile shall support the operations listed in Table 18 for the

1088

CIM_HostedService class. Each operation shall be supported as defined in [DSP0200](#).

1089

Table 18 – Operations: CIM_HostedService

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

1090

8.15 CIM_LogicalIdentity

1091

Compliant implementations of this profile shall support the operations listed in Table 19 for the

1092

CIM_LogicalIdentity class. Each operation shall be supported as defined in [DSP0200](#).

1093

Table 19 – Operations: CIM_LogicalIdentity

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

1094

8.16 CIM_OrderedComponent

1095

Compliant implementations of this profile shall support the operations listed in Table 20 for the

1096

CIM_OrderedComponent class. Each operation shall be supported as defined in [DSP0200](#).

1097

Table 20 – Operations: CIM_OrderedComponent

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

1098 **8.17 CIM_ServiceAffectsElement**

1099 Compliant implementations of this profile shall support the operations listed in Table 21 for the
 1100 CIM_ServiceAffectsElement class. Each operation shall be supported as defined in [DSP0200](#).

1101 **Table 21 – Operations: CIM_ServiceAffectsElement**

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

1102 **9 Use Cases**

1103 Clause 9 contains object diagrams and use cases for the *Boot Control Profile*.

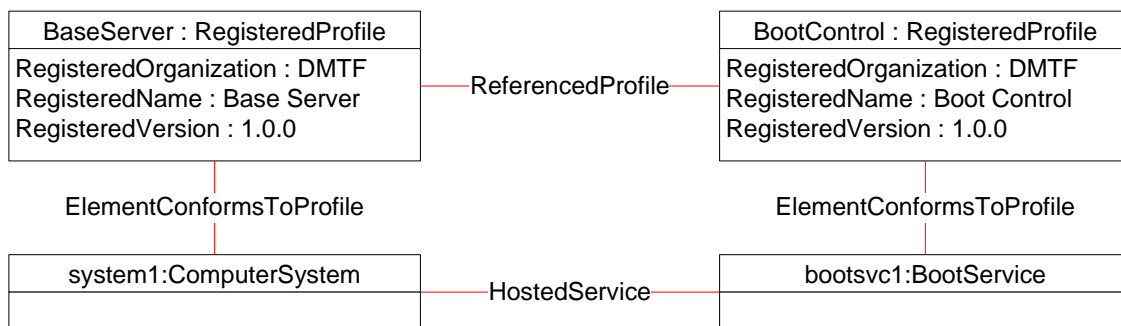
1104 **9.1 Advertising the Profile Conformance**

1105 The object diagram in Figure 2 shows how instances of CIM_RegisteredProfile are used to identify the
 1106 version of the *Boot Control Profile* with which an instance of CIM_BootService and its associated
 1107 instances are conformant. An instance of CIM_RegisteredProfile exists for each profile that is
 1108 instrumented in the system. One instance of CIM_RegisteredProfile identifies the DMTF *Base Server*
 1109 *Profile*, version 1.0.0. The other instance identifies the DMTF *Boot Control Profile*, version 1.0.0. The
 1110 Central Instance is the CIM_BootService. The Scoping Instance is the CIM_ComputerSystem instance.

1111 This instance of CIM_ComputerSystem is conformant with the *Base Server Profile* version 1.0.0 as
 1112 indicated by the CIM_ElementConformsToProfile association to the CIM_RegisteredProfile instance.

1113 This instance of CIM_BootService is conformant with the *Boot Control Profile* version 1.0.0 as indicated
 1114 by the CIM_ElementConformsToProfile association to the CIM_RegisteredProfile instance.

1115 The CIM_ReferencedProfile relationship between *BaseServer* and *BootControl* places the
 1116 CIM_BootService instance within the scope of *BaseServer*.



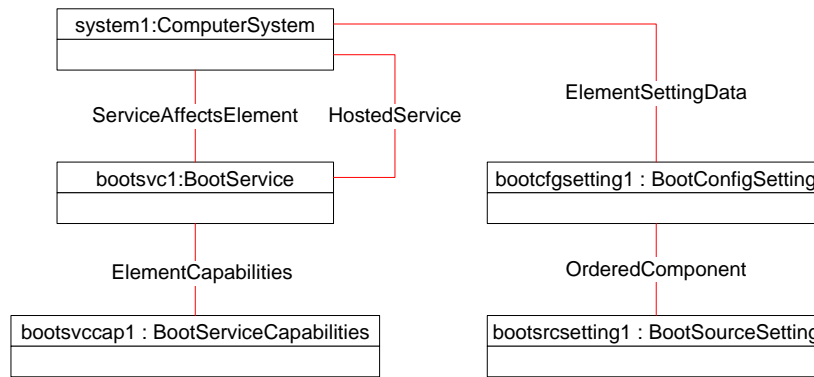
1117

1118 **Figure 2 – Registered Profile**

1119 **9.2 Object Diagram for a Monolithic Server**

1120 Figure 3 shows the CIM instances required to control the boot configuration for a single, monolithic
 1121 server, *system1*. *System1* hosts the boot service, *bootsvc1*, which is used to control the boot
 1122 configuration, *bootcfgsetting1*, for *system1*. *System1* is also identified as the Boot Configurable System
 1123 through the CIM_ServiceAffectsElement association. The capabilities of *bootsvc1* are defined by
 1124 *bootsvccap1*.

1125 The boot configuration, *bootcfgsetting1*, has one boot source, *bootsrcsetting1*.



1126

1127

Figure 3 – Monolithic Server Object Diagram

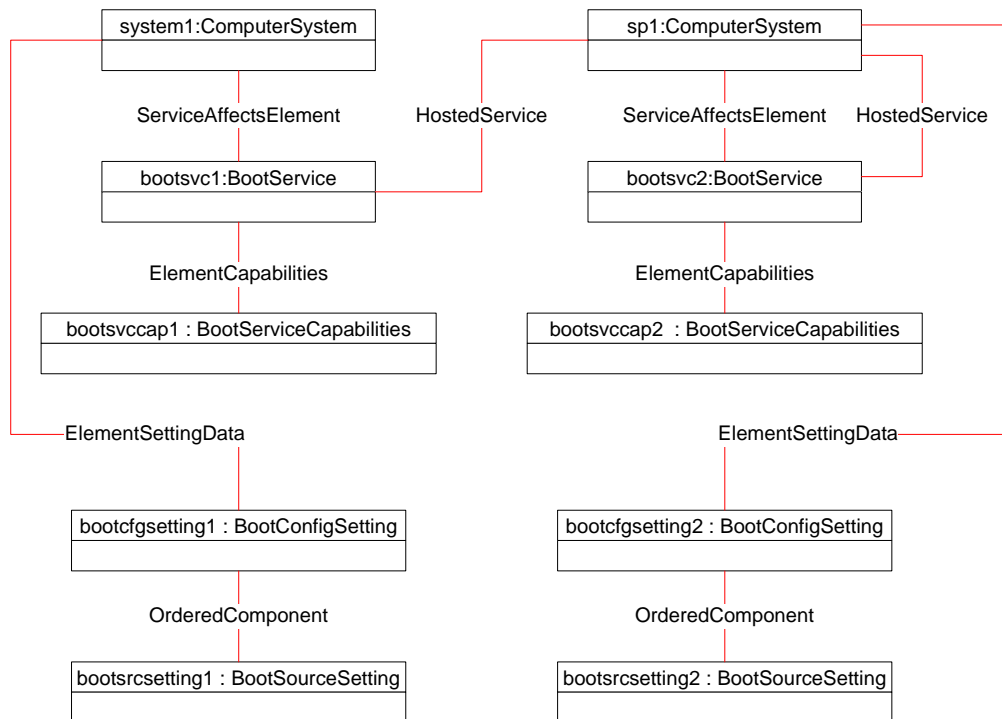
1128 **9.3 Object Diagram for a Monolithic Server with Service Processor**

1129 Figure 4 shows the CIM instances required to control the boot configuration for a single, monolithic
 1130 server, *system1*, with an attached service processor, *sp1*. The boot service, *bootsvc1*, is hosted by the
 1131 service processor and is responsible for managing the boot configuration, *bootcfgsetting1*, for *system1*.

1132 Optionally, the service processor may host another boot configuration service, *bootsvc2*, to control its
 1133 own boot configuration, *bootcfgsetting2*.

1134 The capabilities of *bootsvc1* and *bootsvc2* are defined by *bootsvccap1* and *bootsvccap2* respectively.

1135 Each boot configuration (*bootcfgsetting1*, *bootcfgsetting2*) has one boot source (*bootsrcsetting1*,
 1136 *bootsrcsetting2*), respectively.



1137

1138

Figure 4 – Monolithic Server with Service Processor Object Diagram

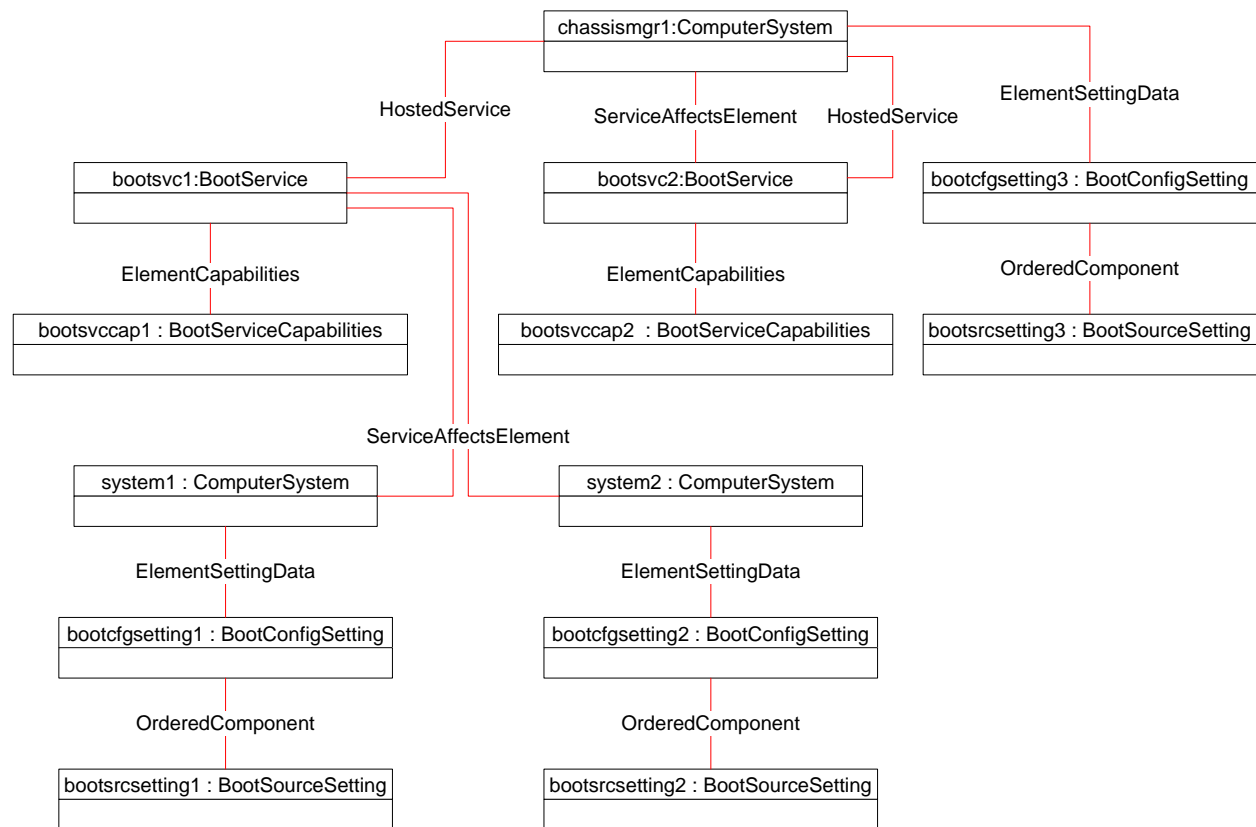
1139 **9.4 Object Diagram for a Modular System**

1140 Figure 5 shows the CIM instances required to control the boot configuration for a modular system. The
 1141 boot service, *bootsvc1*, is hosted by the chassis manager, *chassismgr1*, and is responsible for managing
 1142 the boot configuration for two blade systems, *system1* and *system2*. *System1* and *system2* each have
 1143 one boot configuration, *bootcfgsetting1* and *bootcfgsetting2* respectively.

1144 Optionally, the chassis manager may host another boot configuration service, *bootsvc2*, to control its own
 1145 boot configuration, *bootcfgsetting3*.

1146 The capabilities of *bootsvc1* and *bootsvr2* are defined by *bootsvccap1* and *bootsvccap2* respectively.

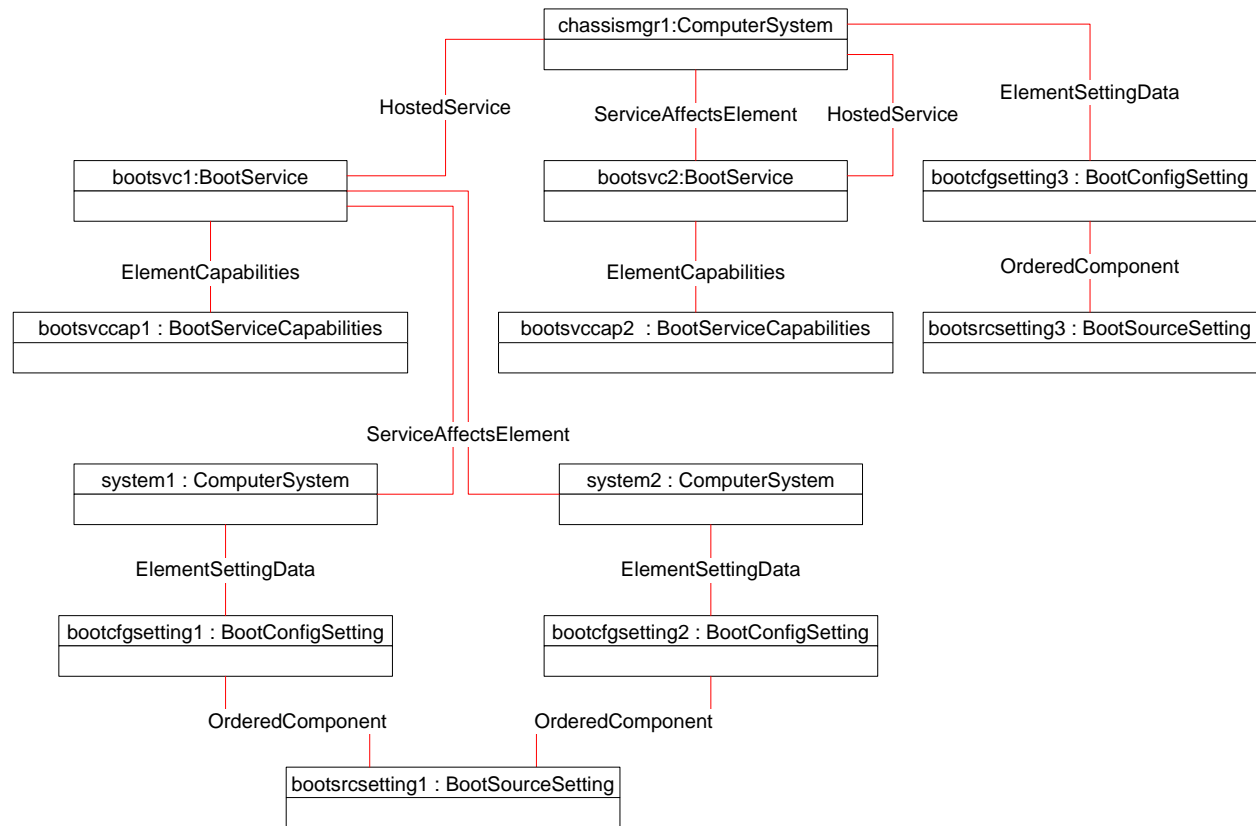
1147 Each boot configuration (*bootcfgsetting1*, *bootcfgsetting2*, *bootcfgsetting3*) has one boot source
 1148 (*bootsrcsetting1*, *bootsrcsetting2*, *bootsrcsetting3*), respectively.



1149

1150 **Figure 5 – Modular System Object Diagram**

1151 Alternative representation of modular blade system boot sources is shown in the Figure 6. The blade
 1152 computer systems, in this example, have the same boot sources, and thus, the representation of boot
 1153 sources can be optimized by instantiating only one CIM_BootSourceSetting shared between the
 1154 respective CIM_BootConfigSetting instances. This optimization is especially useful when modular system
 1155 contains many blade computer systems with the similar boot sources. Each blade boot configuration
 1156 (*bootcfgsetting1*, *bootcfgsetting2*) has one boot source (*bootsrcsetting1*).



1157

1158

Figure 6 – Modular System Object Diagram

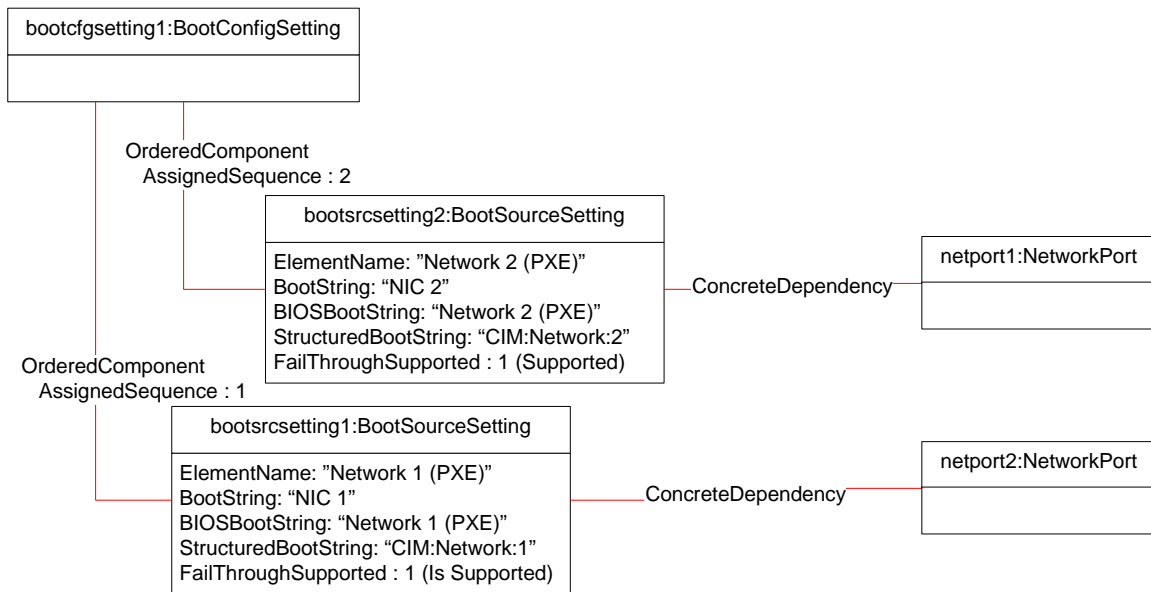
1159 **9.5 PXE Boot Source**

1160 Figure 7 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has two boot sources
 1161 associated to it, *bootsrcsetting1* and *bootsrcsetting2*, which are both network ports.

1162 A network port can support various protocols. Both *bootsrcsetting1* and *bootsrcsetting2* designate the
 1163 PXE protocol in their `BIOSBootString` property. The two `CIM_ConcreteDependency` associations to
 1164 instances of `CIM_NetworkPort` are *netport2* and *netport1*, respectively.

1165 The `AssignedSequence` property values on the `OrderedComponent` associations indicate that the boot
 1166 order is *bootsrcsetting1* followed by *bootsrcsetting2*.

1167 On *bootsrcsetting1*, the `FailThroughSupported` property value of 1 (Is Supported) indicates that if the
 1168 *bootsrcsetting1* fails or times out, the boot process should proceed to *bootsrcsetting2* on *netport1*.



1169

1170

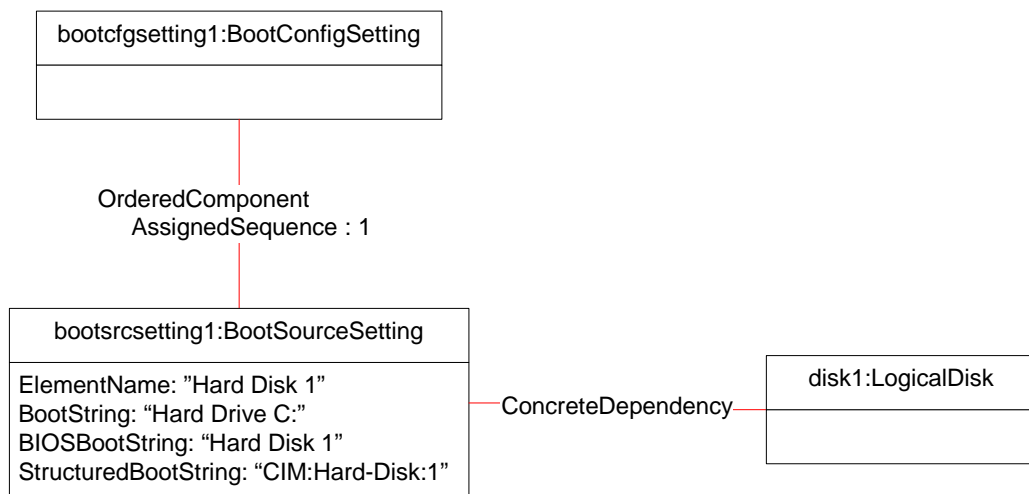
Figure 7 – PXE Boot Sources Object Diagram

1171 **9.6 Disk Boot Source**

1172 Figure 8 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has a single boot
 1173 source associated to it, *bootsrcsetting1*, which is a disk device.

1174 The CIM_BootSourceSetting.ElementName property identifies "Hard Disk 1" as the boot source, which
 1175 matches the BIOSBootString property. The BootString property contains the string "C:", which could be
 1176 interpreted by the boot process to assign the hard drive the letter "C". The CIM_ConcreteDependency
 1177 association relates *bootsrcsetting1* to a CIM_LogicalDisk (*disk1*).

1178 Because there is only one boot source, the value of the CIM_BootSourceSetting.FailThroughSupported is
 1179 not meaningful.



1180

1181

Figure 8 – Booting from Disk

1182 **9.7 Local CDROM and Floppy Boot Sources**

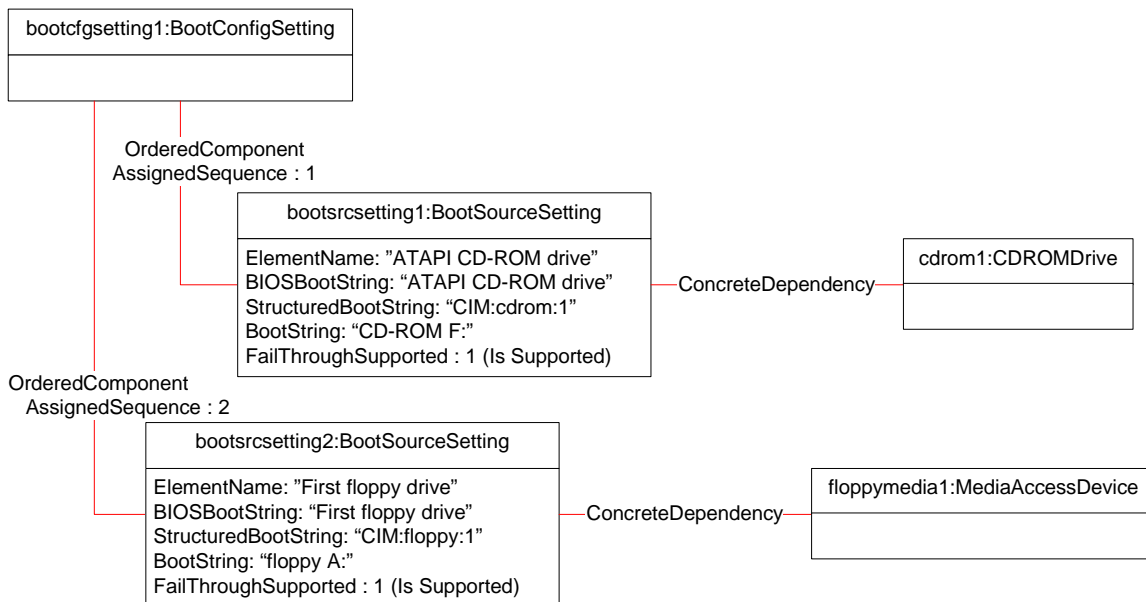
1183 Figure 9 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has two boot sources
 1184 associated to it, *bootsrcsetting1* and *bootsrcsetting2*. *Bootsrcsetting1* is a CD-ROM device;
 1185 *bootsrcsetting2* is a floppy drive.

1186 The AssignedSequence property of the CIM_OrderedComponent associations is set such that booting is
 1187 attempted from the CD-ROM drive first and then the floppy drive.

1188 The BootString property for the CD-ROM drive, *bootsrcsetting1*, contains the string "F:", which could be
 1189 interpreted by the boot process to assign the floppy drive the letter "F". The CIM_ConcreteDependency
 1190 association relates *bootsrcsetting1* to a CIM_CDROMDrive (*cdrom1*).

1191 The BootString property for the floppy drive, *bootsrcsetting2*, contains the string "A:", which could be
 1192 interpreted by the boot process to assign the floppy drive the letter "A". The CIM_ConcreteDependency
 1193 association relates *bootsrcsetting2* to a CIM_DisketteDrive (*floppymedia1*).

1194 On *bootsrcsetting1*, the value of the FailThroughSupported property set to 1 (Is Supported) specifies that
 1195 if the *bootsrcsetting1*, the CD-ROM device, fails or times out, then the boot process should proceed to
 1196 *bootsrcsetting2*, the floppy device.



1197

1198 **Figure 9 – Booting from CDROM and Floppy**

1199 **9.8 Representing IPL and Boot Control Vector (BCV) Lists**

1200 Figure 10 shows an instance diagram for a boot configuration, *bootcfgsetting1*, composed of an IPL and
 1201 BCV list of boot devices.

1202 To represent the IPL list, *bootcfgsetting1* has three boot sources associated to it, *bootsrcsetting1*,
 1203 *bootsrcsetting2*, and *bootsrcsetting3*. *Bootsrcsetting1* is a CD-ROM device. *Bootsrcsetting2* is a floppy
 1204 drive. *Bootsrcsetting3* is a BCV device (boot control vector).

1205 The AssignedSequence property of the CIM_OrderedComponent associations is set such that booting is
 1206 attempted from the CD-ROM drive first and then the BCV device. Booting from the floppy device is not
 1207 attempted because the AssignedSequence property is set to 0. The

1208 CIM_BootConfigSetting.FailThroughSupported property value of 1 (Is Supported) specifies that the boot
 1209 process should proceed to the second boot source if the first boot source fails or times out.

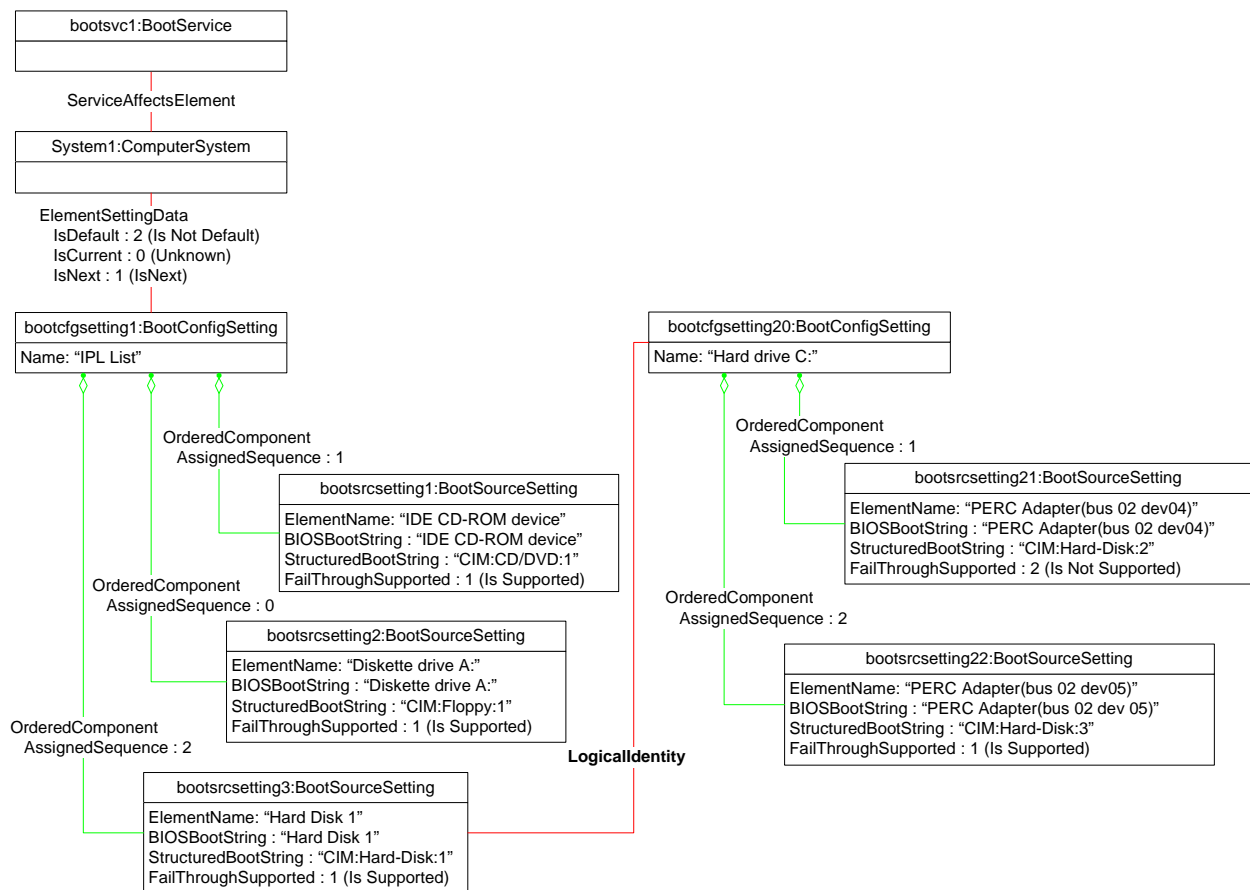
1210 In the diagram, the BCV device is a SCSI controller that may have multiple bootable SCSI devices
 1211 attached to it. This relationship is represented by an instance of CIM_LogicalIdentity between
 1212 *bootsrcsetting3* and an instance of CIM_BootConfigSetting, *bootcfgsetting20*.

1213 The boot configuration, *bootcfgsetting20*, has two boot sources associated to it, *bootsrcsetting21* and
 1214 *bootsrcsetting22*. Both boot sources are hard disk devices.

1215 The AssignedSequence property of the CIM_OrderedComponent associations is set such that booting is
 1216 attempted from *bootsrcsetting21* first and from *bootsrcsetting22*.

1217 On *bootsrcsetting21*, the FailThroughSupported property value of 2 (Is Not Supported) specifies that if the
 1218 *bootsrcsetting21*, "CIM:Hard-Disk:2", fails or times out, then the boot process should terminate the boot
 1219 order for *bootconfigsetting20*.

1220 In total, this use case describes a source boot order that proceeds from *bootsrcsetting1* to
 1221 *bootsrcsetting21*. *bootsrcsetting22* will never be used because of its AssignedSequence value of 0 and
 1222 *bootsrcsetting22* will never be used because of the FailThroughSupported value on *bootsrcsetting21*.



1223

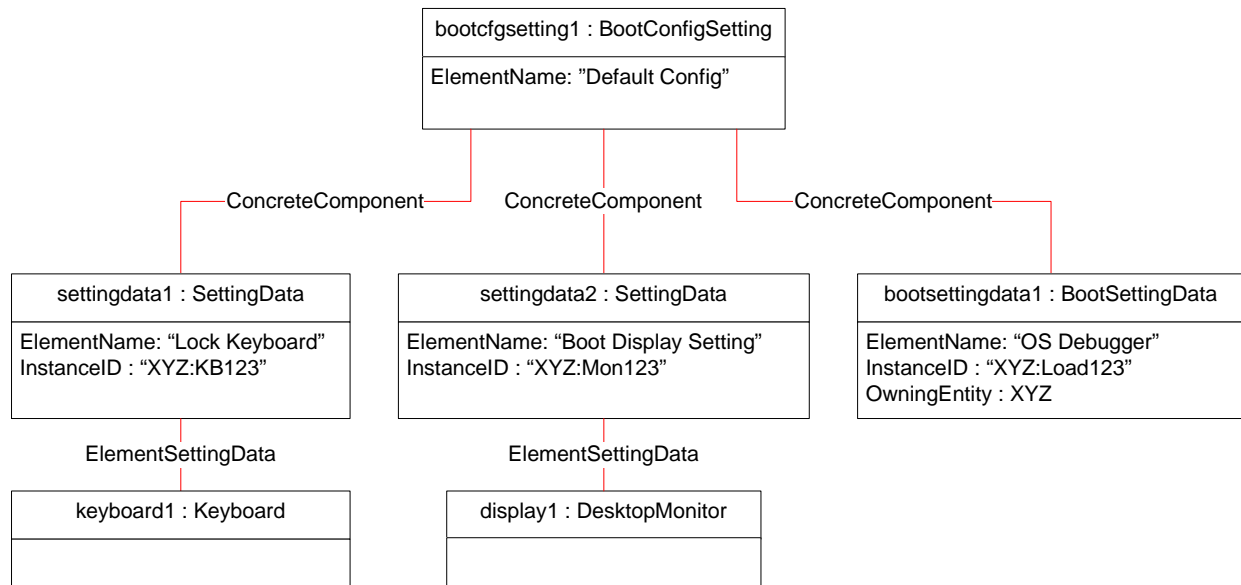
1224

Figure 10 – Booting from IPL and BCV Devices

1225 **9.9 Representing Settings and Boot Settings**

1226 Figure 11 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has settings that
 1227 need to be applied to the computer system during the boot process. Two example concrete subclasses of
 1228 CIM_SettingData, *settingdata1* and *settingdata2*, apply to concrete subclasses of CIM_LogicalDevice,
 1229 *keyboard1* and *display1*. The instance of an example concrete subclass of CIM_BootSettingData is
 1230 *bootsettingdata1*.

1231 Being associated to the instance of CIM_BootConfigSetting, the settings apply to the entire boot process
 1232 that uses *bootcfgsetting1*. Note that any of these settings could be associated to an instance of
 1233 CIM_BootSourceSetting, which would reduce the scope of the settings to just the specified boot source.



1234

1235 **Figure 11 – Setting Data and Boot Setting Data**

1236 **9.10 Representing the Default Boot Configuration for a Computer System**

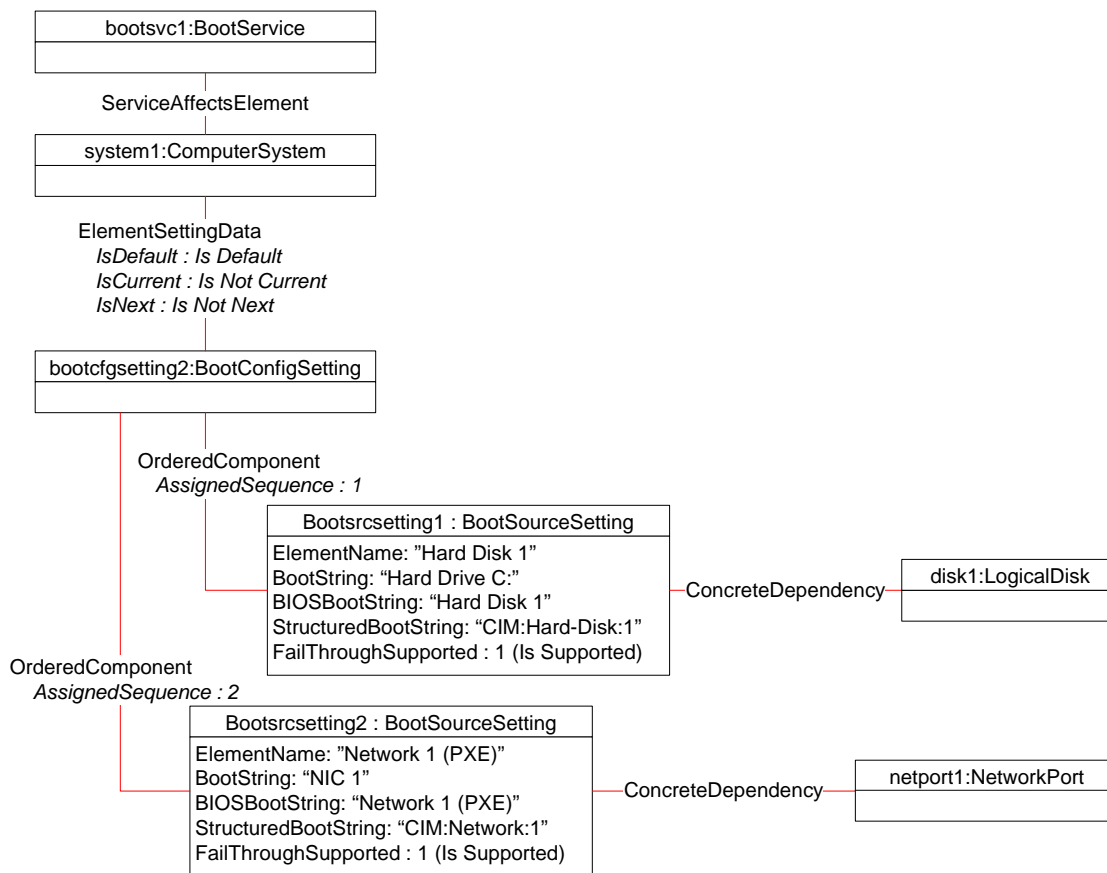
1237 Figure 12 shows an instance diagram for a Boot Configurable System, *system1*. *System1* has a single
 1238 boot configuration, *bootcfgsetting2*. This boot configuration is a Default Boot Configuration, because the
 1239 value of the ElementSettingData.IsDefault property is set to 1 (Is Default). There are no Next Boot
 1240 Configuration or Current Boot Configuration.

1241 *Bootcfgsetting2* is associated with two instances of CIM_BootSourceSetting (*bootsrcsetting1* and
 1242 *bootsrcsetting2*), through instances of CIM_OrderedComponent. The respective
 1243 CIM_OrderedComponent.AssignedSequence properties designate the order in which the boot process
 1244 should use the boot sources (*bootsrcsetting1* followed by *bootsrcsetting2*).

1245 On *bootsrcsetting1*, the FailThroughSupported property value of 1 (Is Supported) specifies that if, during
 1246 the boot of *bootsrcsetting1*, the hard disk fails or times out, then the boot process should proceed to
 1247 *bootsrcsetting2*, the network port using PXE.

1248 When the system represented by *system1* is enabled, the boot process will not be initiated because there
 1249 is no Next Boot Configuration for the boot process to use. The system, *system1*, will be in an enabled, but
 1250 not booted, state. One could manually boot the system from this state by applying an existing boot
 1251 configuration (see subclause 9.14).

1252 *System1* would initiate the boot process if the Default Boot Configuration were also the Next Boot
 1253 Configuration (see subclause 9.11) or a new boot configuration is created as the Next Boot Configuration
 1254 (see subclause 9.13).



1255

1256

Figure 12 – Default Boot Configuration Object Diagram

1257 **9.11 Representing the Next Boot Configuration for a Computer System**

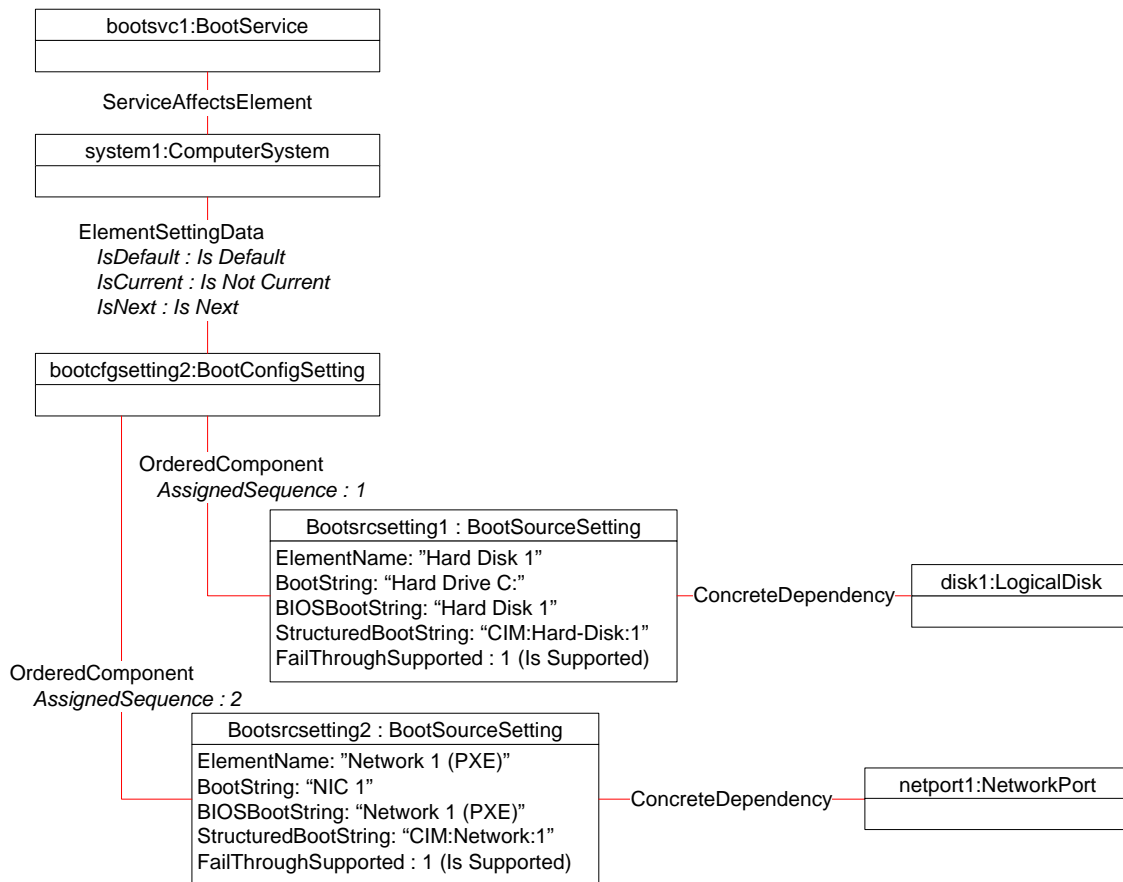
1258 Figure 13 shows an instance diagram for a Boot Configurable System, *system1*. *System1* has a single
 1259 boot configuration, *bootcfgsetting2*. This boot configuration is a Default Boot Configuration, because the
 1260 value of the *ElementSettingData.IsDefault* property is set to 1 (Is Default). This boot configuration is also
 1261 the Next Boot Configuration, because the value of the *ElementSettingData.IsNext* property is set to 1 (Is
 1262 Next).

1263 *Bootcfgsetting2* is associated with two instances of *CIM_BootSourceSetting* (*bootsrcsetting1* and
 1264 *bootsrcsetting2*), through instances of *CIM_OrderedComponent*. The respective
 1265 *CIM_OrderedComponent.AssignedSequence* properties designate the order in which the boot process
 1266 should use the boot sources (*bootsrcsetting1* followed by *bootsrcsetting2*).

1267 On *bootsrcsetting1*, the *FailThroughSupported* property value of 1 (Is Supported) specifies that if the
 1268 *bootsrcsetting1*, the hard disk fails or times out during the boot process, then the boot process should
 1269 proceed to *bootsrcsetting2*, the network port using PXE.

1270 When the system represented by *system1* is enabled, the boot process will find a Next Boot
 1271 Configuration, *bootcfgsetting2* and proceed to use it to boot.

1272 When the system represented by *system1* is an enabled, but not booted, state. The
 1273 `BootService.ApplyBootConfigSetting()` method can be invoked referencing *system1* as the
 1274 `BootConfigurableSystem` parameter.



1275

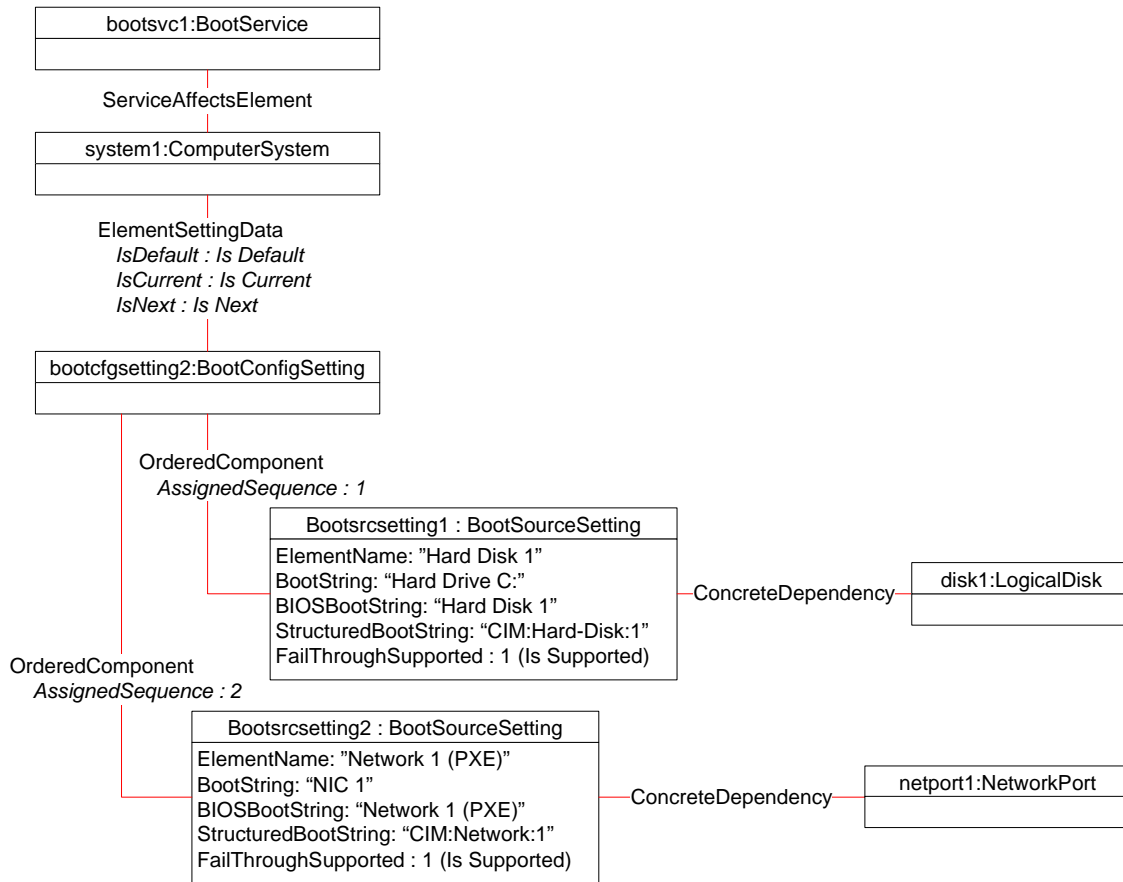
1276

Figure 13 – Next Boot Configuration Object Diagram

1277 **9.12 Representing the Current Boot Configuration for a Booted Computer System**

1278 Figure 14 shows an instance diagram for the Boot Configurable System, *system1*, described in the
 1279 previous use case (see Figure 13) after it has been successfully booted.

1280 The boot configuration, *bootcfgsetting1*, is now the Current Boot Configuration, because the value of the
 1281 `ElementSettingData.IsCurrent` property is set to 1 (Is Current). *Bootcfgsetting1* is still concurrently the
 1282 Default Boot Configuration and the Next Boot Configuration.



1283

1284

Figure 14 – Boot Configuration for a Booted System Object Diagram

1285 **9.13 Create a New Boot Configuration**

1286 Referencing the object diagram in Figure 12, a client could create a new boot configuration as follows:

- 1287 1) From the Boot Configurable System, *system1*, find the instance of `CIM_BootService` that
- 1288 manages the boot configurable system by traversing the `CIM_ServiceAffectsElement`
- 1289 association.
- 1290 2) Verify that the `CreateBootConfigSetting()` method is supported (see subclause 9.27). If not, a
- 1291 new boot configuration cannot be created.
- 1292 3) Find an existing instance of `CIM_BootConfigSetting` to use as the template. For this use case,
- 1293 *bootcfgsetting2* is the only template configuration available.
- 1294 4) Create the new boot configuration, *bootcfgsetting4*, by invoking the
- 1295 `CIM_BootService.CreateBootConfigSetting()` method. The `ScopingComputerSystem` parameter
- 1296 is set to *system1* and the `StartingBootConfig` parameter is set to *bootcfgsetting2*.

1297 Figure 15 shows the instance diagram after the `CreateBootConfigSetting()` method has been invoked and

1298 successfully completed on the computer system, *system1*, shown in Figure 14. The new boot

1299 configuration, *bootcfgsetting4*, is associated to *system1* through a new instance of

1300 `CIM_ElementSettingData`.

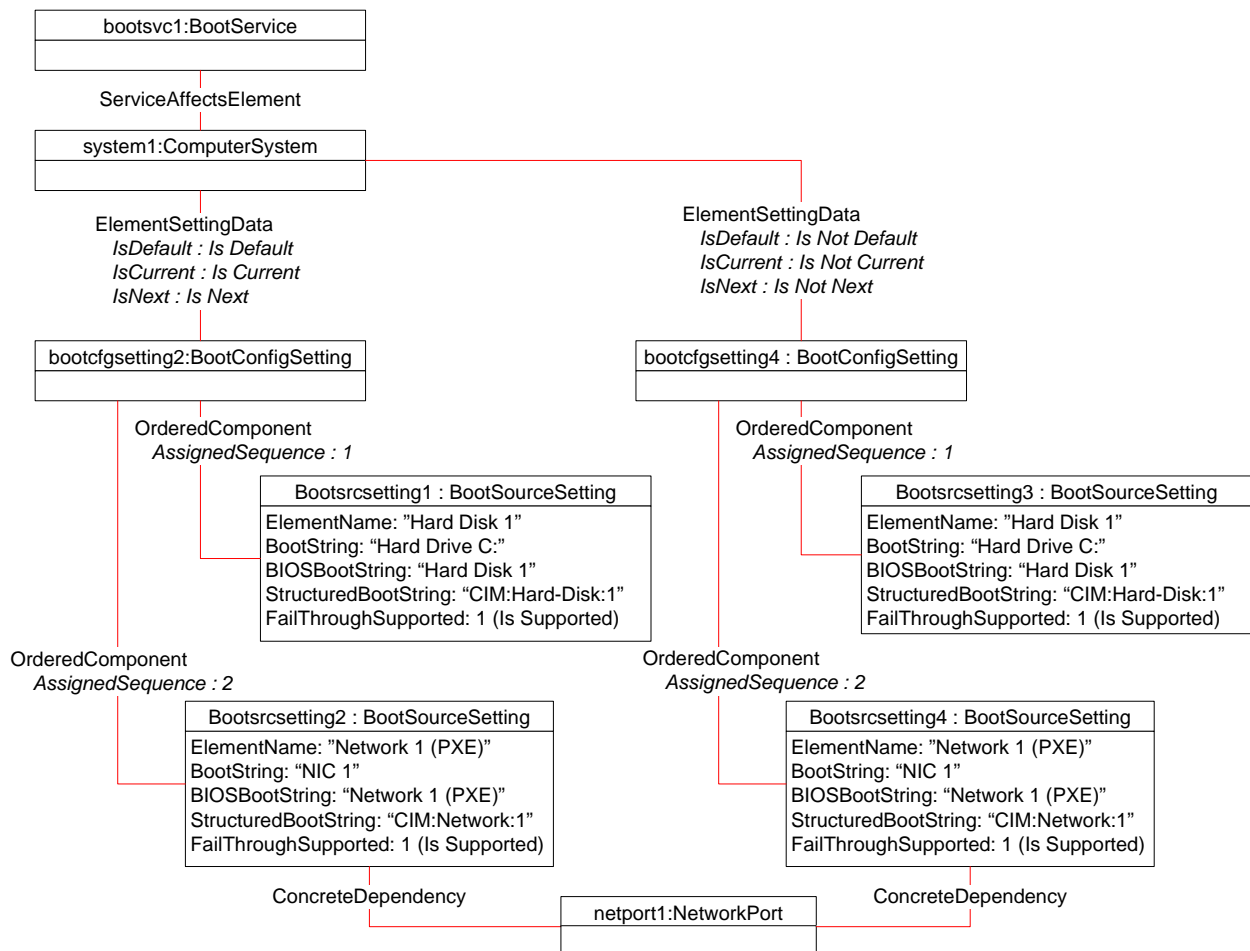
1301 In the new instance of `CIM_ElementSettingData`, the `IsDefault` property is set to 2 (Is Not Default); the

1302 `IsCurrent` property is set to 2 (Is Not Current); and the `IsNext` property is set to 2 (Is Not Next).

1303 *Bootcfgsetting4* is associated through instances of *CIM_OrderedComponent* to two instances of
 1304 *CIM_BootSourceSetting* (*bootsrcsetting3* and *bootsrcsetting4*), which are copies of *bootsrcsetting1* and
 1305 *bootsrcsetting2*, respectively.

1306 The instance of *CIM_NetworkPort* is not copied. *CIM_NetworkPort* is a concrete subclass of
 1307 *CIM_LogicalDevice*, which is not part of the *Boot Control Profile*. However, an instance of
 1308 *CIM_ConcreteDependency* has been created that associates the instance of *CIM_NetworkPort* to the
 1309 new instance of *CIM_BootSourceSetting* (*bootsrcsetting4*).

1310 *CIM_LogicalDisk* has been elided from the object diagram to make the diagram less cluttered, but the
 1311 instance of *CIM_LogicalDisk* is also not copied. An instance of *CIM_ConcreteDependency* is created that
 1312 associates the existing instance of *CIM_LogicalDisk* to the new instance of *CIM_BootSourceSetting*
 1313 (*bootsrcsetting3*).



1314

1315

Figure 15 – System with New CIM_BootConfigSetting

1316 **9.14 Apply an Existing Boot Configuration**

1317 Referencing the object diagram in Figure 12, a client could apply a boot configuration as follows:

- 1318 1) Find the instance of *CIM_BootService* for the boot configurable system as outlined in subclause
 1319 9.15.
- 1320 2) Verify that the *ApplyBootConfigSetting()* method is supported (see subclause 9.28). If not, a
 1321 boot configuration cannot be applied.

- 1322 3) Find the existing instances of CIM_BootConfigSetting for *system1* (see subclause 9.18). In this
 1323 example, this results in *bootcfgsetting2*. Pick one of them to use as the boot configuration to
 1324 apply.
- 1325 4) Apply the selected boot configuration, *bootcfgsetting2*, by invoking the
 1326 CIM_BootService.ApplyBootConfigSetting() method. The ScopingComputerSystem parameter
 1327 is set to *system1* and the BootConfigSetting parameter is set to *bootcfgsetting2*.

1328 The ApplyBootConfigSetting() method will boot *system1* by applying the boot configuration specified in
 1329 *bootcfgsetting2*. If *system1* is currently booted, an implementation has the option of rejecting the
 1330 ApplyBootConfigSetting() request or of rebooting the system.

1331 **9.15 Find the Boot Service for a Computer System**

1332 A client can find the boot service for a given computer system as follows:

- 1333 1) For the instance of CIM_ComputerSystem, representing the given computer system, select the
 1334 referencing instance of CIM_BootService, representing the boot control service for the server,
 1335 through the CIM_ServiceAffectsElement association.

1336 **9.16 Find the Boot Configuration for a Computer System**

1337 A client can find the boot configurations for a computer system as follows:

- 1338 1) From the instance of CIM_ComputerSystem, enumerate the CIM_ElementSettingData
 1339 associations with CIM_BootConfigSetting as the SettingData reference.

1340 **9.17 Find the Boot Service Capabilities for a Computer System**

1341 A client can find the boot service capabilities for a computer system as follows:

- 1342 1) Find the boot service for the computer system as specified in subclause 9.15.
- 1343 2) Select the instance of CIM_BootServiceCapabilities through the CIM_ElementCapabilities
 1344 association.

1345 **9.18 Find the Current Boot Configuration for a Computer System**

1346 A client can find the current boot configuration for a computer system as follows:

- 1347 1) From the instance of CIM_ComputerSystem, enumerate the CIM_ElementSettingData
 1348 associations with CIM_BootConfigSetting as the SettingData reference.
- 1349 2) Find the instance of CIM_ElementSettingData whose IsCurrent property is set to 1 (Is Current).
- 1350 3) The CIM_BootConfigSetting instance referenced by this association instance represents the
 1351 current boot configuration.

1352 **9.19 Find the Default Boot Configuration for a Computer System**

1353 A client can find the default boot configuration for a computer system as follows:

- 1354 1) From the instance of CIM_ComputerSystem, enumerate the CIM_ElementSettingData
 1355 associations with CIM_BootConfigSetting as the SettingData reference.
- 1356 2) Find the instance of CIM_ElementSettingData whose IsDefault property is set to 1 (Is Default).
- 1357 3) The CIM_BootConfigSetting instance referenced by this association instance represents the
 1358 default boot configuration.

1359 **9.20 Find the Boot Configuration that Will Be Used during the Next Reboot for a** 1360 **Computer System**

1361 A client can find the boot configuration that will be used during a computer system's next reboot as
1362 follows:

- 1363 1) For the instance of CIM_ComputerSystem, enumerate the CIM_ElementSettingData
1364 associations with CIM_BootConfigSetting as the SettingData reference.
- 1365 2) Find the CIM_ElementSettingData instances for the instance whose IsNext property is set to 3
1366 (Is Next For Single Use). The CIM_BootConfigSetting instance referenced by this association
1367 instance represents the next boot configuration.
- 1368 3) If no instance is found, find the instance of CIM_ElementSettingData whose IsNext property is
1369 set to 1 (Is Next). The CIM_BootConfigSetting instance referenced by this association instance
1370 represents the next boot configuration.

1371 **9.21 Make a Boot Configuration Applicable for Subsequent Reboots**

1372 A client can make a boot configuration apply to a computer system for subsequent reboots as follows:

- 1373 1) From the instance of CIM_ComputerSystem, find the CIM_BootConfigSetting of interest as
1374 outlined in subclauses 9.9 through 9.12.
- 1375 2) On the instance of the CIM_ElementSettingData association that associates the instance of
1376 CIM_ComputerSystem to the instance of CIM_BootConfigSetting, use the intrinsic
1377 ModifyInstance() to change the IsNext property to 1 (Is Next).

1378 Note that this boot configuration applies for all subsequence reboots, unless it is overridden by a Next
1379 Single Use Boot Configuration that is associated to the CIM_ComputerSystem of interest.

1380 **9.22 Make a Boot Configuration Applicable for the Next Reboot Only**

1381 A client can make a boot configuration apply to a computer system for only the next reboot as follows:

- 1382 1) From the instance of CIM_ComputerSystem, find the CIM_BootConfigSetting of interest as
1383 outlined in subclauses 9.9 through 9.12.
- 1384 2) On the instance of the CIM_ElementSettingData association that associates the instance of
1385 CIM_ComputerSystem to the instance of CIM_BootConfigSetting, use the intrinsic
1386 ModifyInstance() to change the IsNext property to 3 (Is Next For Single Use).

1387 The behavior of this property after the next boot is specified in subclause 7.4.5.

1388 **9.23 Determine If the Computer System Supports PXE Boot**

1389 A client can determine if the computer system supports PXE boot as follows:

- 1390 1) For the instance of CIM_ComputerSystem enumerate its instances of CIM_BootConfigSetting
1391 as outline in subclause 9.18.
- 1392 2) For each instance of CIM_BootConfigSetting, enumerate the instances of
1393 CIM_BootSourceSetting.
- 1394 3) For each CIM_BootSourceSetting, inspect the BootString, BIOSBootString, or
1395 StructuredBootString property to determine if PXE is supported.

1396 **9.24 Find the Boot Order for a Computer System for the Next Reboot**

1397 This use case references the object diagram in Figure 10, which represents a complex boot order.

1398 A client can find the boot order for the next reboot of a computer system as follows:

- 1399 1) From the instance of CIM_ComputerSystem, *system1*, find the CIM_BootConfigSetting that will
1400 be used during the next reboot, *bootcfgsetting1* (see subclause 3)).
- 1401 2) Determine the boot order for *bootcfgsetting1* by enumerating the CIM_OrderedComponent
1402 associations with *bootcfgsetting1* as the GroupComponent reference. The results in this
1403 example would be *bootsrcsetting1*, *bootsrcsetting2* and *bootsrcsetting3*.
- 1404 3) Use the CIM_OrderedComponent.AssignedSequence property to determine the boot order. The
1405 boot order in this example will be *bootsrcsetting1* followed by *bootsrcsetting3*. The boot source
1406 represented by *bootsrcsetting2* will be ignored because its associated AssignedSequence
1407 property value is 0.
- 1408 4) For each boot source, determine whether any it contains additional boot sources by checking
1409 for a CIM_LogicalIdentity association to an instance of CIM_BootConfigSetting; in this example,
1410 *bootcfgsetting20*, and repeat steps in this subclause recursively to find the boot order of the
1411 associated boot sources.

1412 9.25 Change the Boot Order for a Computer System

1413 This use case references the object diagram in Figure 13.

1414 A client can change the boot order for a computer system as follows:

- 1415 1) Find the boot configuration of interest from the set of boot configurations for the computer
1416 system as outlined in subclause 9.18.
- 1417 2) Find the set of boot sources for the boot configuration by following the OrderedComponent
1418 associations from the selected boot configuration representation (*bootcfgsetting2*) to all
1419 instances of CIM_BootSourceSetting. In this example, this results in *bootsrcsetting1* and
1420 *bootsrcsetting2*.
- 1421 3) Determine the desired boot order.
- 1422 4) Create an array of CIM_BootSourceSetting references. Assign the existing boot sources to the
1423 array in the new order. For instance, if one wanted to reverse the boot order in this example, the
1424 array would contain *bootsrcsetting2* at index 0 and *bootsrcsetting1* at index 1.
- 1425 5) Invoke the ChangeBootOrder() method on the selected instance of CIM_BootConfigSetting. The
1426 Source parameter is set to the array created above.

1427 NOTE: The order of each boot configuration must be changed independently. Thus if the computer system has a
1428 complex boot structure, such as that illustrated in Figure 10, changing the boot order for the system may require
1429 changing the boot order for multiple CIM_BootConfigSetting instances.

1430 9.26 Determine Whether BootService.ElementName Is Modifiable

1431 A client can determine whether the ElementName can be modified as follows:

- 1432 1) Find the CIM_BootServiceCapabilities instance associated with the CIM_BootService instance
1433 through the CIM_ElementCapabilities association.
- 1434 2) If a CIM_BootConfigCapabilities instance cannot be found, then the
1435 CIM_BootService.ElementName property cannot be modified.
- 1436 3) Query the value of the CIM_BootServiceCapabilities.ElementNameEditSupported.
- 1437 4) If the value is TRUE, the CIM_BootService.ElementName property can be modified
- 1438 5) If the value of ElementNameEditSupported has a value of FALSE, then the
1439 CIM_BootService.ElementName property cannot be modified.

1440 **9.27 Determine Whether a New Boot Configuration Can Be Created**

1441 A client can determine whether a new boot configuration can be created as follows:

- 1442 1) Find the CIM_BootServiceCapabilities instance that is associated with the CIM_BootService
1443 instance through the CIM_ElementCapabilities association.
- 1444 2) Query the value of the CIM_BootServiceCapabilities.BootConfigCapabilities property array. If
1445 the array contains the value 2 (Creates Boot Configuration), the client's ability to create a boot
1446 configuration is supported.
- 1447 3) If the BootConfigCapabilities property array does not contain the value 2 (Creates Boot
1448 Configuration), or there is not an instance of CIM_BootServiceCapabilities associated with the
1449 CIM_BootService instance, a boot configuration cannot be created.

1450 **9.28 Determine Whether a Boot Configuration Can Be Applied**

1451 A client can determine whether a boot configuration can be manually applied to the boot configurable
1452 systems as follows:

- 1453 1) Find the CIM_BootServiceCapabilities instance that is associated with the CIM_BootService
1454 instance through the CIM_ElementCapabilities association.
- 1455 2) Query the value of the CIM_BootServiceCapabilities.BootConfigCapabilities property array. If
1456 the array contains the value 3 (Applies Boot Configuration), the client's ability to manually apply
1457 a boot configuration is supported.
- 1458 3) If the BootConfigCapabilities property array does not contain the value 3 (Applies Boot
1459 Configuration), or there is not an instance of CIM_BootServiceCapabilities associated with the
1460 CIM_BootService instance, a boot configuration cannot be manually applied.

1461 **10 CIM Elements**

1462 Table 22 shows the instances of CIM Elements for this profile. Instances of the following CIM Elements
1463 shall be implemented as described in Table 22. Clauses 7 ("Implementation") and 8 ("Methods") may
1464 impose additional requirements on these elements.

1465 **Table 22 CIM Elements – Boot Control Profile**

Element Name	Requirement	Description
CIM_RegisteredProfile	Mandatory	See subclause 10.1.
CIM_BootService	Mandatory	See subclause 10.2.
CIM_BootServiceCapabilities	Optional	See subclause 10.3.
CIM_BootConfigSetting	Mandatory	See subclause 10.4.
CIM_BootSettingData	Optional	See subclause 10.5.
CIM_BootSourceSetting	Mandatory	See subclause 10.6.
CIM_ConcreteComponent	Optional	See subclause 10.7.
CIM_ConcreteDependency	Optional	See subclause 10.8.
CIM_ElementCapabilities	Optional	See subclause 10.9.
CIM_ElementSettingData	Mandatory	See subclause 10.10.
CIM_HostedService	Mandatory	See subclause 10.11.
CIM_LogicalIdentity	Conditional	See subclause 10.12.
CIM_OrderedComponent	Mandatory	See subclause 10.13.
CIM_ServiceAffectsElement	Mandatory	See subclause 10.14.

1466 **10.1 CIM_RegisteredProfile**

1467 CIM_RegisteredProfile identifies the *Boot Control Profile* in order for a client to determine whether an
 1468 instance of CIM_ComputerSystem is conformant with this profile. The CIM_RegisteredProfile class is
 1469 defined by the [Profile Registration Profile](#). With the exception of the mandatory values specified for the
 1470 properties below, the behavior of the CIM_RegisteredProfile instance is per the [Profile Registration](#)
 1471 [Profile](#). Table 23 contains the requirements for elements of this class.

1472 **Table 23 – Class: CIM_RegisteredProfile**

Elements	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Boot Control".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.1".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

1473 NOTE: Previous versions of this document included the suffix 'Profile' for the RegisteredName value. If
 1474 implementations querying for RegisteredName value find the suffix 'Profile', they should ignore the suffix, with any
 1475 surrounding white spaces, before any comparison is done with the value as specified in this document.

1476 **10.2 CIM_BootService**

1477 The CIM_BootService class represents the ability to view and control the boot settings of a computer
 1478 system. Table 24 contains the requirements for elements of this class.

1479 **Table 24 – Class: CIM_BootService**

Elements	Requirement	Notes
CreationClassName	Mandatory	Key
Name	Mandatory	Key
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
ElementName	Mandatory	See subclause 7.1.1.
CreateBootConfigSetting()	Conditional	See subclause 8.1.
ApplyBootConfigSetting()	Conditional	See subclause 8.2.

1480 **10.3 CIM_BootServiceCapabilities**

1481 Support of the CIM_BootServiceCapabilities class is optional.

1482 When supported, CIM_BootServiceCapabilities is used to indicate the capabilities of the boot service.
 1483 Table 25 contains the requirements for elements of this class.

1484

Table 25 – Class: CIM_BootServiceCapabilities

Elements	Requirement	Notes
InstanceID	Mandatory	Key
ElementName	Mandatory	This property shall be a character string of variable length (pattern “. *”).
ElementNameEditSupported	Mandatory	See subclause 7.1.2
BootConfigCapabilities	Mandatory	See subclauses 7.5, 7.6, and 7.7.
OtherBootConfigCapabilities	Conditional	See subclause 7.3.1.
BootStringsSupported	Optional	See subclause 7.8.

1485 **10.4 CIM_BootConfigSetting**

1486 The CIM_BootConfigSetting class represents a boot configuration of a computer system. Table 26
 1487 contains the requirements for elements of this class.

1488

Table 26 – Class: CIM_BootConfigSetting

Elements	Requirement	Notes
InstanceID	Mandatory	Key
ElementName	Mandatory	This property shall be a character string of variable length (pattern “. *”).
ChangeBootOrder()	Conditional	See subclause 7.9 and 8.3.

1489 **10.5 CIM_BootSettingData**

1490 Support of the CIM_BootSettingData class is optional.

1491 The CIM_BootSettingData class represents the settings that apply during booting of a computer system.
 1492 Table 27 contains the requirements for elements of this class.

1493 For each property added in a concrete subclass of CIM_BootSettingData, there shall be a Description
 1494 qualifier that contains a string which describes the setting. When the range of the setting is bounded and
 1495 discrete, the Values and ValueMap qualifiers should contain the values and name of the values,
 1496 respectively, which are applicable for the setting.

1497

Table 27 – Class: CIM_BootSettingData

Elements	Requirement	Notes
InstanceID	Mandatory	Key
ElementName	Mandatory	This property shall be a character string of variable length (pattern “. *”).
OwningEntity	Mandatory	None

1498 **10.6 CIM_BootSourceSetting**

1499 Support of the CIM_BootSourceSetting class is optional.

1500 The CIM_BootSourceSetting class represents a boot source, from which booting is attempted during the
 1501 boot process. Table 28 contains the requirements for elements of this class.

1502

Table 28 – Class: CIM_BootSourceSetting

Elements	Requirement	Notes
InstanceID	Mandatory	Key
ElementName	Mandatory	See subclause 7.8.2.
BootString	Conditional	See subclause 7.8.3.
BIOSBootString	Conditional	See subclause 7.8.4.
StructuredBootString	Conditional	See subclause 7.8.5.
FailThroughSupported	Mandatory	See subclause 7.11.2.

1503 **10.7 CIM_ConcreteComponent**

1504 Subclause 10.7 describes optional behavior.

1505 **10.7.1 Relating CIM_BootConfigSetting to a Concrete Subclass of CIM_SettingData**

1506 When supported, the CIM_ConcreteComponent association is used to relate an instance of a concrete
 1507 subclass of CIM_SettingData to a CIM_BootConfigSetting instance. Table 29 contains the requirements
 1508 for elements of this class.

1509

Table 29 – Class: CIM_ConcreteComponent – Use 1

Elements	Requirement	Notes
GroupComponent	Mandatory	This property shall be a reference to an instance of the CIM_BootConfigSetting class. See subclause 7.12. Cardinality is "**".
PartComponent	Mandatory	This property shall be a reference to an instance of a concrete subclass of the CIM_SettingData class. See subclause 7.12. Cardinality is "**".

1510 **10.7.2 Relating CIM_BootConfigSetting to a Concrete Subclass of CIM_BootSettingData**

1511 When supported, the CIM_ConcreteComponent association is used to relate an instance of a concrete
 1512 subclass of CIM_BootSettingData to a CIM_BootConfigSetting instance. Table 30 contains the
 1513 requirements for elements of this class.

1514

Table 30 – Class: CIM_ConcreteComponent – Use 2

Elements	Requirement	Notes
GroupComponent	Mandatory	This property shall be a reference to an instance of the CIM_BootConfigSetting class. See subclause 7.12. Cardinality is "0..1".
PartComponent	Mandatory	This property shall be a reference to an instance of a concrete subclass of the CIM_BootSettingData class. See subclause 7.12. Cardinality is "**".

1515 **10.7.3 Relating CIM_BootSourceSetting to a Concrete Subclass of CIM_SettingData**

1516 When supported, the CIM_ConcreteComponent association is used to relate an instance of a concrete
 1517 subclass of CIM_SettingData to a CIM_BootSourceSetting instance. Table 31 contains the requirements
 1518 for elements of this class.

1519 **Table 31 – Class: CIM_ConcreteComponent – Use 3**

Elements	Requirement	Notes
GroupComponent	Mandatory	This property shall be a reference to an instance of the CIM_BootSourceSetting class. See subclause 7.12. Cardinality is "**".
PartComponent	Mandatory	This property shall be a reference to an instance of a concrete subclass of the CIM_SettingData class. See subclause 7.12. Cardinality is "**".

1520 **10.7.4 Relating CIM_BootSourceSetting to a Concrete Subclass of CIM_BootSettingData**

1521 When supported, the CIM_ConcreteComponent association is used to relate an instance a concrete
 1522 subclass of CIM_BootSettingData to a CIM_BootSourceSetting instance. Table 32 contains the
 1523 requirements for elements of this class.

1524 **Table 32 – Class: CIM_ConcreteComponent – Use 4**

Elements	Requirement	Notes
GroupComponent	Mandatory	This property shall be a reference to an instance of the CIM_BootSourceSetting class. See subclause 7.12. Cardinality is "0..1".
PartComponent	Mandatory	This property shall be a reference to an instance of a concrete subclass of the CIM_BootSettingData class. See subclause 7.12. Cardinality is "**".

1525 **10.8 CIM_ConcreteDependency**

1526 Subclause 10.8 describes optional behavior.

1527 When supported, the CIM_ConcreteDependency association is used to relate the dependency of a
 1528 CIM_BootSourceSetting instance on an instance of a concrete subclass of CIM_LogicalDevice. Table 33
 1529 contains the requirements for elements of this class.

1530 **Table 33 – Class: CIM_ConcreteDependency**

Elements	Requirement	Notes
Antecedent	Mandatory	This property shall be a reference to an instance of a concrete subclass of the CIM_LogicalDevice class. See subclause 7.8.5.2. Cardinality is "**".
Dependent	Mandatory	This property shall be a reference to an instance of the CIM_BootSourceSetting. See subclause 7.8.5.2. Cardinality is "**".

1531 **10.9 CIM_ElementCapabilities**

1532 Subclause 10.9 describes optional behavior.

1533 When supported, the CIM_ElementCapabilities association is used to relate an instance of
 1534 CIM_BootServiceCapabilities with an instance of CIM_BootService. Table 34 contains the requirements
 1535 for elements of this class.

1536 **Table 34 – Class: CIM_ElementCapabilities**

Elements	Requirement	Notes
ManagedElement	Mandatory	This property shall be a reference to an instance of CIM_BootService. See subclause 7.1. Cardinality is "1..*".
Capabilities	Mandatory	This property shall be a reference to an instance of CIM_BootServiceCapabilities. See subclause 7.1. Cardinality is "0..1".

1537 **10.10 CIM_ElementSettingData**

1538 The CIM_ElementSettingData association is used to relate the CIM_BootConfigSetting instance to the
 1539 CIM_ComputerSystem instance to which it applies. Table 35 contains the requirements for elements of
 1540 this class.

1541 **Table 35 – Class: CIM_ElementSettingData**

Elements	Requirement	Notes
ManagedElement	Mandatory	This property shall be a reference to an instance of the CIM_ComputerSystem class. See subclause 7.4.1. Cardinality is "0..1".
SettingData	Mandatory	This property shall be a reference to an instance of the CIM_BootConfigSetting class. See subclause 7.4.1. Cardinality is "**".
IsDefault	Mandatory	See subclause 7.3.
IsCurrent	Mandatory	See subclause 7.3.
IsNext	Mandatory	See subclause 7.3.

1542 **10.11 CIM_HostedService**

1543 The CIM_HostedService association is used to relate the CIM_BootService to the CIM_ComputerSystem
 1544 on which it is hosted. Table 36 contains the requirements for elements of this class.

1545 **Table 36 – Class: CIM_HostedService**

Elements	Requirement	Notes
Antecedent	Mandatory	This property shall be a reference to the scoping instance of the CIM_ComputerSystem class. See subclause 5. Cardinality is "1".
Dependent	Mandatory	This property shall be a reference to an instance of the CIM_BootService. See subclause 5. Cardinality is "**".

1546 **10.12 CIM_LogicalIdentity**

1547 Support of the CIM_LogicalIdentity association is conditional.

1548 Conditional Requirement: The support is required if instances of CIM_BootSourceSetting are used to
 1549 represent aggregated boot sources; see subclause 7.9.

1550 When supported, CIM_LogicalIdentity is used to associate an instance of CIM_BootSourceSetting with an
 1551 instance of CIM_BootConfigSetting. Table 37 contains the requirements for elements of this class.

1552 **Table 37 – Class: CIM_LogicalIdentity**

Elements	Requirement	Notes
SystemElement	Mandatory	This property shall be a reference to an instance of the CIM_BootSourceSetting class. See subclause 7.10.3. Cardinality is "0..1"
SameElement	Mandatory	This property shall be a reference to an instance of the CIM_BootConfigSetting class. See subclause 7.10.3. Cardinality is "0..1"

1553 **10.13 CIM_OrderedComponent**

1554 Support of the CIM_OrderedComponent association is conditional.

1555 Conditional Requirement: The support is required if the CIM_BootSourceSetting instances are used to
 1556 represent boot sources; see 7.11.1.

1557 When supported, the CIM_OrderedComponent association is used to indicate the order in which
 1558 CIM_BootSourceSetting instances should be attempted for a CIM_BootConfigSetting instance. Table 38
 1559 contains the requirements for elements of this class.

1560 **Table 38 – Class: CIM_OrderedComponent**

Elements	Requirement	Notes
GroupComponent	Mandatory	This property shall be a reference to an instance of the CIM_BootConfigSetting class. See subclause 7.11.1. Cardinality is "1..*"
PartComponent	Mandatory	This property shall be a reference to an instance of the CIM_BootSourceSetting class. See subclause 7.11.1. Cardinality is "1..*"
AssignedSequence	Mandatory	See subclause 7.11.1.1.

1561 **10.14 CIM_ServiceAffectsElement**

1562 The CIM_ServiceAffectsElement association is used to associate the CIM_BootService instance with a
 1563 CIM_ComputerSystem instance that it affects. Table 39 contains the requirements for elements of this
 1564 class.

1565 **Table 39 – Class: CIM_ServiceAffectsElement**

Elements	Requirement	Notes
AffectingElement	Mandatory	This property shall be a reference to an instance of the CIM_BootService class. See subclause 7.2. Cardinality is "0..1".
AffectedElement	Mandatory	This property shall be a reference to an instance of the CIM_ComputerSystem class. See subclause 7.2. Cardinality is "1..*".
ElementEffects	Mandatory	Matches 5 (Manages)

1566

**ANNEX A
(informative)****Change Log**1567
1568
1569
1570

Version	Date	Description
1.0.0a	10/10/2006	Preliminary Standard
1.0.0	11/03/2008	Final Standard
1.0.1	06/22/2009	DMTF Standard Release

1571