

1



2

**Document Identifier: DSP1012**

3

**Date: 2014-09-02**

4

5

**Version: 1.1.0**

## 6 **Boot Control Profile**

7 **Document Type: Specification**

8 **Document Status: DMTF Standard**

9 **Document Language: en-US**

10 Copyright Notice

11 Copyright © 2008–2010, 2014 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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 .....	14
45	7 Implementation.....	15
46	7.1 CIM_BootService .....	15
47	7.2 CIM_ComputerSystem.....	16
48	7.3 Representing boot service capabilities .....	16
49	7.4 Boot configurations .....	17
50	7.5 Applying the boot configuration .....	18
51	7.6 Creating a boot configuration.....	19
52	7.7 Deleting a boot configuration .....	20
53	7.8 Identifying boot sources .....	20
54	7.9 Changing the boot order .....	24
55	7.10 Representing a set of aggregated boot sources.....	24
56	7.11 Boot order during the boot process .....	25
57	7.12 Settings to apply during the boot process.....	26
58	8 Methods.....	27
59	8.1 CIM_BootService.CreateBootConfigSetting( ) .....	27
60	8.2 CIM_BootService.ApplyBootConfigSetting( ) .....	28
61	8.3 CIM_BootConfigSetting.ChangeBootOrder( ) .....	30
62	8.4 Profile conventions for operations .....	30
63	8.5 CIM_BootService.....	31
64	8.6 CIM_BootConfigSetting .....	31
65	8.7 CIM_BootSettingData .....	32
66	8.8 CIM_BootSourceSetting .....	33
67	8.9 CIM_ConcreteComponent .....	33
68	8.10 CIM_ConcreteDependency .....	33
69	8.11 CIM_ElementCapabilities .....	34
70	8.12 CIM_ElementSettingData .....	34
71	8.13 CIM_BootServiceCapabilities .....	35
72	8.14 CIM_HostedService .....	36
73	8.15 CIM_LogicalIdentity .....	36
74	8.16 CIM_OrderedComponent .....	36
75	8.17 CIM_ServiceAffectsElement .....	37
76	9 Use cases.....	37
77	9.1 Advertising the profile conformance .....	37
78	9.2 Object diagram for a monolithic server .....	37
79	9.3 Object diagram for a monolithic server with service processor .....	38
80	9.4 Object diagram for a modular system.....	39
81	9.5 PXE boot source .....	40
82	9.6 Disk boot source .....	41
83	9.7 Local CDROM and floppy boot sources .....	42
84	9.8 Representing IPL and Boot Control Vector (BCV) lists .....	42
85	9.9 Representing settings and boot settings.....	44

86 9.10 Representing the same boot device ..... 44

87 9.11 Representing the default boot configuration for a computer system ..... 45

88 9.12 Representing the next boot configuration for a computer system ..... 46

89 9.13 Representing the current boot configuration for a booted computer system..... 47

90 9.14 Create a new boot configuration ..... 48

91 9.15 Apply an existing boot configuration ..... 49

92 9.16 Find the boot service for a computer system..... 50

93 9.17 Find the boot configuration for a computer system ..... 50

94 9.18 Find the boot service capabilities for a computer system..... 50

95 9.19 Find the current boot configuration for a computer system ..... 50

96 9.20 Find the default boot configuration for a computer system..... 50

97 9.21 Find the boot configuration that will be used during the next reboot for a computer  
98 system..... 51

99 9.22 Make a boot configuration applicable for subsequent reboots ..... 51

100 9.23 Make a boot configuration applicable for the next reboot only ..... 51

101 9.24 Determine whether the computer system supports PXE boot..... 51

102 9.25 Find the boot order for a computer system for the next reboot ..... 51

103 9.26 Change the boot order for a computer system ..... 52

104 9.27 Determine whether BootService.ElementName is modifiable ..... 52

105 9.28 Determine whether a new boot configuration can be created ..... 53

106 9.29 Determine whether a boot configuration can be applied ..... 53

107 10 CIM Elements ..... 53

108 10.1 CIM\_RegisteredProfile..... 54

109 10.2 CIM\_BootService ..... 54

110 10.3 CIM\_BootServiceCapabilities ..... 54

111 10.4 CIM\_BootConfigSetting ..... 55

112 10.5 CIM\_BootSettingData ..... 55

113 10.6 CIM\_BootSourceSetting ..... 55

114 10.7 CIM\_ConcreteComponent ..... 56

115 10.8 CIM\_ConcreteDependency ..... 57

116 10.9 CIM\_ElementCapabilities ..... 58

117 10.10 CIM\_ElementSettingData ..... 58

118 10.11 CIM\_HostedService ..... 58

119 10.12 CIM\_LogicalIdentity ..... 59

120 10.13 CIM\_OrderedComponent ..... 59

121 10.14 CIM\_ServiceAffectsElement ..... 60

122 ANNEX A (informative) Change log..... 61

123

124 **Figures**

125 Figure 1 – Boot Control Profile: Class diagram..... 14

126 Figure 2 – Registered Profile ..... 37

127 Figure 3 – Monolithic server object diagram ..... 38

128 Figure 4 – Monolithic server with service processor object diagram ..... 38

129 Figure 5 – Modular system object diagram ..... 39

130 Figure 6 – Modular system object diagram..... 40

131 Figure 7 – PXE boot sources object diagram ..... 41

132 Figure 8 – Booting from disk ..... 41

133 Figure 9 – Booting from CDROM and floppy ..... 42

134 Figure 10 – Booting from IPL and BCV devices ..... 43

135 Figure 11 – Setting data and boot setting data..... 44

136 Figure 12 - Representing the same boot device ..... 45

137 Figure 13 – Default boot configuration object diagram ..... 46  
 138 Figure 14 – Next boot configuration object diagram ..... 47  
 139 Figure 15 – Boot configuration for a booted system object diagram ..... 48  
 140 Figure 16 – System with new CIM\_BootConfigSetting ..... 49  
 141

142 **Tables**

143 Table 1 – Related Profiles ..... 12  
 144 Table 2 – Structured name identifiers ..... 23  
 145 Table 3 – CreateBootConfigSetting( ) Method: Return code values ..... 28  
 146 Table 4 – CreateBootConfigSetting( ) Method: Parameters ..... 28  
 147 Table 5 – ApplyBootConfigSetting( ) Method: Return code values ..... 29  
 148 Table 6 – ApplyBootConfigSetting( ) Method: Parameters ..... 29  
 149 Table 7 – ChangeBootOrder( ) Method: Return code values ..... 30  
 150 Table 8 – ChangeBootOrder( ) Method: Parameters ..... 30  
 151 Table 9 – Operations: CIM\_BootService ..... 31  
 152 Table 10 – Operations: CIM\_BootConfigSetting ..... 31  
 153 Table 11 – Operations: CIM\_BootSettingData ..... 32  
 154 Table 12 – Operations: CIM\_BootSourceSetting ..... 33  
 155 Table 13 – Operations: CIM\_ConcreteComponent ..... 33  
 156 Table 14 – Operations: CIM\_ConcreteDependency ..... 33  
 157 Table 15 – Operations: CIM\_ElementCapabilities ..... 34  
 158 Table 16 – Operations: CIM\_ElementSettingData ..... 34  
 159 Table 17 – Operations: CIM\_BootServiceCapabilities ..... 36  
 160 Table 18 – Operations: CIM\_HostedService ..... 36  
 161 Table 19 – Operations: CIM\_LogicalIdentity ..... 36  
 162 Table 20 – Operations: CIM\_OrderedComponent ..... 36  
 163 Table 21 – Operations: CIM\_ServiceAffectsElement ..... 37  
 164 Table 22 CIM Elements – Boot Control Profile ..... 53  
 165 Table 23 – Class: CIM\_RegisteredProfile ..... 54  
 166 Table 24 – Class: CIM\_BootService ..... 54  
 167 Table 25 – Class: CIM\_BootServiceCapabilities ..... 55  
 168 Table 26 – Class: CIM\_BootConfigSetting ..... 55  
 169 Table 27 – Class: CIM\_BootSettingData ..... 55  
 170 Table 28 – Class: CIM\_BootSourceSetting ..... 56  
 171 Table 29 – Class: CIM\_ConcreteComponent – Use 1 ..... 56  
 172 Table 30 – Class: CIM\_ConcreteComponent – Use 2 ..... 56  
 173 Table 31 – Class: CIM\_ConcreteComponent – Use 3 ..... 57  
 174 Table 32 – Class: CIM\_ConcreteComponent – Use 4 ..... 57  
 175 Table 33 – Class: CIM\_ConcreteDependency ..... 57  
 176 Table 34 – Class: CIM\_ElementCapabilities ..... 58  
 177 Table 35 – Class: CIM\_ElementSettingData ..... 58  
 178 Table 36 – Class: CIM\_HostedService ..... 58  
 179 Table 37 – Class: CIM\_LogicalIdentity ..... 59  
 180 Table 38 – Class: CIM\_OrderedComponent ..... 59  
 181 Table 39 – Class: CIM\_ServiceAffectsElement ..... 60

182



184

## Foreword

185 The *Boot Control Profile* (DSP1012) was prepared by the Server Desktop Mobile Platforms Working  
186 Group.

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

## 189 Acknowledgments

190 The authors wish to acknowledge the following people.

### 191 Editors:

- 192 • John Leung – Intel
- 193 • Khachatur Papanyan – Dell
- 194 • Hemal Shah – Broadcom
- 195 • David Simpson – IBM

### 196 Contributors:

- 197 • Jon Hass – Dell Aaron Merkin – IBM
- 198 • Jeff Hilland – Hewlett-Packard Company
- 199 • Arvind Kumar – Intel
- 200 • Khachatur Papanyan – Dell
- 201 • Christina Shaw – Hewlett-Packard Company
- 202 • Enoch Suen – Dell
- 203 • Perry Vincent – Intel

205

## Introduction

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

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



212

# Boot Control Profile

## 213 1 Scope

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

## 216 2 Normative references

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

### 220 2.1 Approved references

221 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,  
222 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.5.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf)

223 DMTF DSP0200, *CIM Operations over HTTP 1.2*,  
224 <http://www.dmtf.org/sites/default/files/standards/documents/DSP200.html>

225 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,  
226 [http://www.dmtf.org/standards/published\\_documents/DSP1001\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf)

227 DMTF DSP1033, *Profile Registration Profile 1.0*,  
228 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf)

### 229 2.2 Other references

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

232 *BIOS Boot Specification 1.01* (January 11, 1996),  
233 <http://www.phoenix.com/resources/specs-bbs101.pdf>

## 234 3 Terms and definitions

### 235 3.1

#### 236 can

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

### 238 3.2

#### 239 cannot

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

### 241 3.3

#### 242 conditional

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

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

286 **3.16**287 **Boot Configuration Representation**

288 the CIM representation of a boot configuration, which consists of an instance of class  
289 CIM\_BootConfigSetting and, optionally, all of the instances of classes CIM\_BootSourceSetting,  
290 CIM\_BootSettingData and CIM\_SettingData that it directly or indirectly aggregates

291 **3.17**292 **Current Boot Configuration**

293 the instance of CIM\_BootConfigSetting that was used the last time the managed system was successfully  
294 booted

295 **3.18**296 **Default Boot Configuration**

297 the instance of CIM\_BootConfigSetting that the computer system manufacturer or a client has  
298 informatively tagged as its default boot configuration

299 **3.19**300 **Next Boot Configuration**

301 the instance of CIM\_BootConfigSetting that will be used during the next boot of the Boot Configurable  
302 System

303 **3.20**304 **Next Single Use Boot Configuration**

305 the instance of CIM\_BootConfigSetting that will only be used during the next boot of the Boot  
306 Configurable System and then not used again

307 **3.21**308 **Not Next Boot Configuration**

309 an instance of CIM\_BootConfigSetting that will not be used during the next boot

310 **3.22**311 **Template Boot Configuration**

312 an existing instance of CIM\_BootConfigSetting that is to be used as the template for creating a new boot  
313 configuration

314 **4 Symbols and abbreviated terms**315 **4.1**316 **BCV**

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

318 **4.2**319 **IPL**

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

321 **4.3**322 **PXE**

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

324 **5 Synopsis**

325 **Profile Name:** *Boot Control*

326 **Version:** 1.1.0

327 **Organization:** DMTF

328 **CIM Schema Version:** 2.19

329 **Central Class:** CIM\_BootService

330 **Scoping Class:** CIM\_ComputerSystem

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

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

335 CIM\_BootService shall be the Central Class of this profile. The instance of CIM\_BootService shall be the  
 336 Central Instance of this profile.

337 CIM\_ComputerSystem shall be the Scoping Class of this profile. The instance of CIM\_ComputerSystem  
 338 with which the Central Instance is associated through an instance of CIM\_HostedService shall be the  
 339 Scoping Instance of this profile.

340

**Table 1 – Related Profiles**

Profile Name	Organization	Version	Relationship
<a href="#">Profile Registration</a>	DMTF	1.0	Mandatory

341 **6 Description**

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

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

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

355 A typical boot configuration could have the following characteristics:

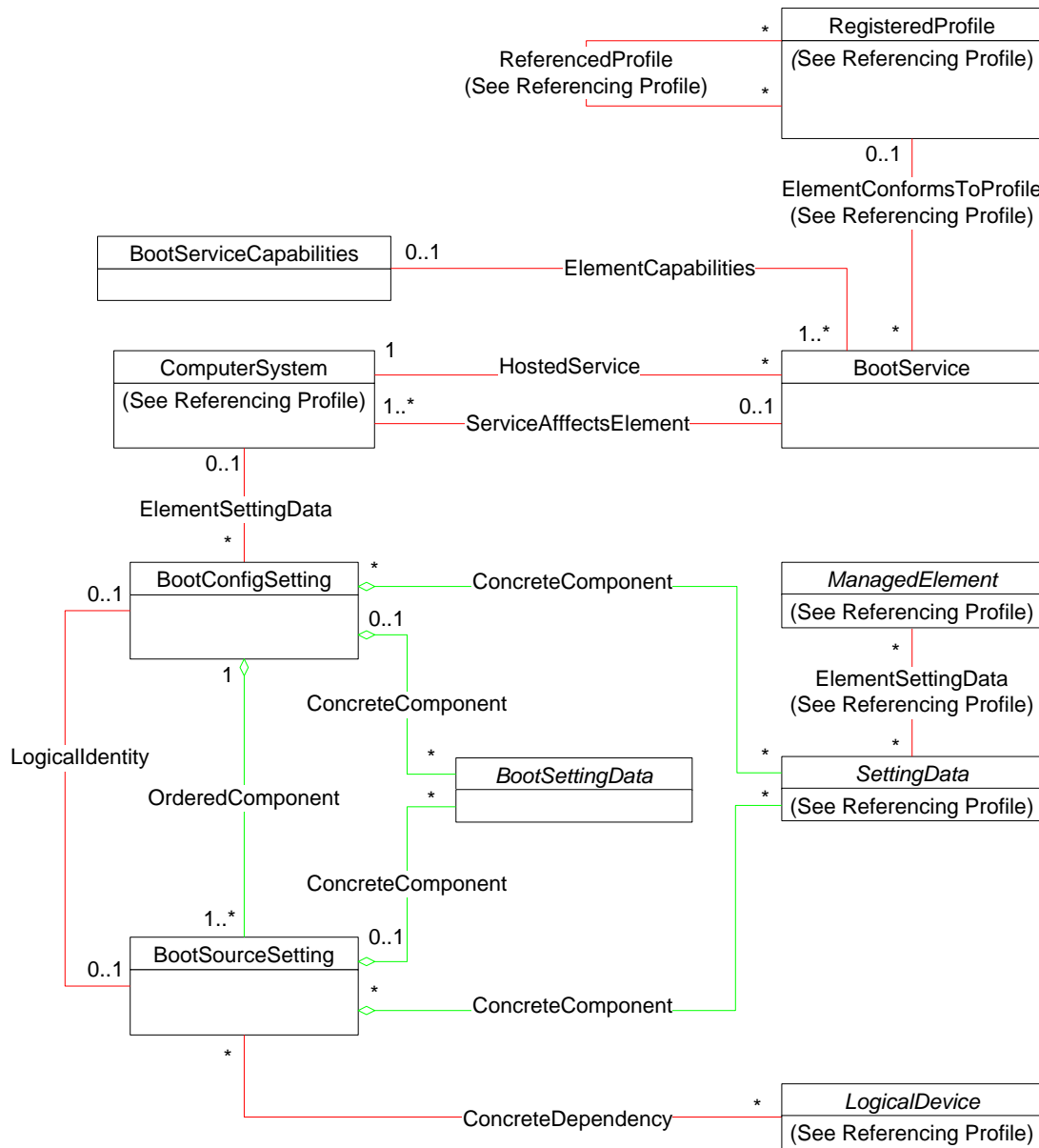
- 356 • A boot configuration can contain a boot order that specifies the order in which boot devices are  
357 accessed. The boot devices include, but are not limited to, floppy device, CD device, hard disks,  
358 network controllers (using the PXE protocol), and BCV devices composed of additional boot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

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

368 **6.1 Class diagram**

369 Figure 1 represents the class schema for the *Boot Control Profile*. For simplicity, the prefix CIM\_ has  
 370 been removed from the name of the classes.

371 In Figure 1, CIM\_ManagedElement, CIM\_LogicalDevice, CIM\_SettingData, and CIM\_BootSettingData  
 372 are abstract



373 classes.

374 **Figure 1 – Boot Control Profile: Class diagram**

375 A computer system can have multiple boot configurations. Each boot configuration is modeled by a Boot  
 376 Configuration Representation, which consists of an instance of CIM\_BootConfigSetting class and,  
 377 optionally, all of the instances of classes CIM\_BootSourceSetting, CIM\_BootSettingData and  
 378 CIM\_SettingData that the instance of CIM\_BootConfigSetting aggregates

379 The usage of each Boot Configuration Representation during the boot process is determined by the  
380 IsNext property of the CIM\_ElementSettingData association between the Boot Configuration  
381 Representation and Boot Configurable System whose boot configuration is being managed.

382 Each Boot Configuration Representation contains an ordered list of boot sources, which indicate the  
383 logical devices to use during the boot process. The boot order is defined by interpreting a property in the  
384 CIM\_OrderedComponent association between the instance of CIM\_BootConfigSetting representing a  
385 boot configuration and instances of CIM\_BootSourceSetting representing the boot sources.

386 In some cases a single boot source might, in turn, represent additional ordered boot sources. This set of  
387 aggregated boot sources is represented by an instance of CIM\_BootConfigSetting, which is associated to  
388 the instance of CIM\_BootSourceSetting through an instance of CIM\_LogicalIdentity.

389 Settings that apply to a managed element during the boot process are represented by instances of a  
390 concrete subclass of the CIM\_SettingData class.

391 Settings that apply to the boot process, itself, are represented by instances of a concrete subclass of the  
392 CIM\_BootSettingData class.

393 These settings can apply to either the entire boot configuration or to a specific boot source within a boot  
394 configuration. This scoping is determined by traversing the CIM\_ConcreteComponent association to  
395 either an instance of CIM\_BootConfigSetting representing the boot configuration or  
396 CIM\_BootSourceSetting representing the boot source, respectively.

## 397 **7 Implementation**

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

### 401 **7.1 CIM\_BootService**

402 At least one instance of the Central Class, CIM\_BootService, shall exist.

#### 403 **7.1.1 CIM\_BootService.ElementName**

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

#### 405 **7.1.2 Modifying ElementName is supported**

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

408 Conditional Requirement:

- 409 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
410 through an instance of CIM\_ElementCapabilities.
- 411 2) The CIM\_BootServiceCapabilities.ElementNameEditSupport property has the value of TRUE.
- 412 3) The CIM\_BootServiceCapabilities.MaxElementNameLen property has a non-zero value

413 The implementation shall allow the CIM\_BootService.ModifyInstance intrinsic operation to change the  
414 value of the ElementName property. The ModifyInstance operation shall enforce the length restriction  
415 specified in the MaxElementNameLen property.

### 416 7.1.3 Modifying ElementName is not supported

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

419 Conditional Requirement 1:

- 420 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
421 through an instance of CIM\_ElementCapabilities.
- 422 2) The CIM\_BootServiceCapabilities.ElementNameEditSupport property has the value of FALSE.

423 Conditional Requirement 2:

- 424 1) An instance of CIM\_BootServiceCapabilities is not associated with the CIM\_BootService  
425 instance through an instance of CIM\_ElementCapabilities.

426 The implementation shall not allow the CIM\_BootService.ModifyInstance intrinsic operation to change the  
427 value of the ElementName property.

## 428 7.2 CIM\_ComputerSystem

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

432 One Scoping Instance shall exist. Clause 5 describes the process for determining the Scoping Instance  
433 from the Central Instance.

434 Each instance of CIM\_ComputerSystem representing a Boot Configurable System shall be associated to  
435 the Central Instance through an instance of the CIM\_ServiceAffectsElement association. At least one  
436 instance of a Boot Configurable System shall exist.

## 437 7.3 Representing boot service capabilities

438 Subclause 7.3 describes optional behavior.

439 An instance of CIM\_BootServiceCapabilities may exist, which represents the capabilities of the boot  
440 service.

441 If an instance of CIM\_BootServiceCapabilities is instantiated, then it shall be associated with an instance  
442 of CIM\_BootService using an instance of CIM\_ElementCapabilities.

### 443 7.3.1 Representing implementation specific boot service capabilities

444 Subclause 7.3.1 describes optional behavior.

445 An implementation may identify method-related boot configuration capabilities, other than those explicitly  
446 defined in this profile, by setting the BootConfigCapabilities and OtherBootConfigCapabilities property  
447 arrays of the CIM\_BootServiceCapabilities class.

448 The additional boot configuration capability shall be identified by setting an entry in the  
449 CIM\_BootServiceCapabilities.BootConfigCapabilities property array to a value of 1 (Other) for each  
450 additional boot configuration capability.

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



## 454 **7.4 Boot configurations**

455 An instance of CIM\_BootConfigSetting shall represent a boot configuration that may be used during the  
456 boot process.

457 Each Boot Configurable System shall have at least one instance of CIM\_BootConfigSetting associated to  
458 it through an instance of CIM\_ElementSettingData.

### 459 **7.4.1 CIM\_ElementSettingData**

460 An instance of CIM\_ElementSettingData shall be used to associate each instance of  
461 CIM\_BootConfigSetting, representing a boot configuration, to each instance of CIM\_ComputerSystem,  
462 representing a Boot Configurable System to which the boot configuration applies.

463 When the CIM\_ElementSettingData association is used in this manner, its ManagedElement property  
464 shall reference the CIM\_ComputerSystem instance and its SettingData property shall reference the  
465 CIM\_BootConfigSetting instance.

466 For an instance of CIM\_ElementSettingData, the IsNext property shall determine how the associated  
467 instance of CIM\_BootConfigSetting is used, if at all, during the boot of the Boot Configurable System.

### 468 **7.4.2 Default boot configuration**

469 Subclause 7.4.2 describes optional behavior.

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

474 The Default Boot Configuration shall be the instance of CIM\_BootConfigSetting that is associated by the  
475 instance of CIM\_ElementSettingData when the IsDefault property has a value of 1 (Is Default).

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

### 479 **7.4.3 Current boot configuration**

480 Subclause 7.4.3 describes optional behavior.

481 The Current Boot Configuration is the instance of CIM\_BootConfigSetting that was used the last time the  
482 system represented by the Boot Configurable System was successfully booted.

483 The Current Boot Configuration shall be the instance of CIM\_BootConfigSetting that is associated by the  
484 instance of CIM\_ElementSettingData when the IsCurrent property has a value of 1 (Is Current).

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

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

#### 491 **7.4.4 Next boot configuration**

492 Subclause 7.4.4 describes optional behavior.

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

496 The Next Boot Configuration is the instance of CIM\_BootConfigSetting that shall be used during the next  
497 boot of the system represented by the Boot Configurable System, unless there is a Next Single Use Boot  
498 Configuration associated to the same Boot Configurable System.

499 The Next Boot Configuration shall be the instance of CIM\_BootConfigSetting that is associated by the  
500 instance of CIM\_ElementSettingData when the IsNext property has a value of 1 (Is Next).

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

#### 502 **7.4.5 Next single use boot configuration**

503 Subclause 7.4.5 describes optional behavior.

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

507 The Next Single Use Boot Configuration is the instance of CIM\_BootConfigSetting that shall only be used  
508 during the next boot of the system represented by the Boot Configurable System.

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

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

513 The Next Single Use Boot Configuration shall be the instance of CIM\_BootConfigSetting that is  
514 associated by the instance of CIM\_ElementSettingData when the IsNext property has a value of 3 (Is  
515 Next For Single Use).

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

#### 518 **7.4.6 Not next boot configuration**

519 The Not Next Boot Configuration is an instance of CIM\_BootConfigSetting that will not be used during the  
520 next boot.

521 The Not Next Boot Configuration shall be a CIM\_BootConfigSetting whose  
522 CIM\_ElementSettingData.IsNext property has the value of 2 (Is Not Next).

### 523 **7.5 Applying the boot configuration**

524 The CIM\_BootService associated to the Boot Configurable System may support the explicit application of  
525 a Boot Configuration Representation through the ApplyBootConfigSetting() method.

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

### 529 **7.5.1 Apply boot configuration is supported**

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

532 Conditional Requirement:

- 533 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
534 through an instance of CIM\_ElementCapabilities.
- 535 2) The CIM\_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 3  
536 (Applies Boot Configuration).

537 The implementation shall support the CIM\_BootService.ApplyBootConfigSetting() method.

### 538 **7.5.2 Apply boot configuration is not supported**

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

541 Conditional Requirement 1:

- 542 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
543 through an instance of CIM\_ElementCapabilities.
- 544 2) The CIM\_BootServiceCapabilities.BootConfigCapabilities property array does not contain a  
545 value of 3 (Applies Boot Configuration).

546 Conditional Requirement 2:

- 547 1) An instance of CIM\_BootServiceCapabilities is not associated with the CIM\_BootService  
548 instance through an instance of CIM\_ElementCapabilities.

549 The implementation shall not support the CIM\_BootService.ApplyBootConfigSetting() method.

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

## 552 **7.6 Creating a boot configuration**

553 The CIM\_BootService may support the client creation of a new boot configuration from an existing boot  
554 configuration through the CreateBootConfigSetting() method.

### 555 **7.6.1 Creating boot configuration is supported**

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

558 Conditional Requirement:

- 559 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
560 through an instance of CIM\_ElementCapabilities.
- 561 2) The CIM\_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 2  
562 (Creates Boot Configuration).

563 The implementation shall support the CreateBootConfigSetting() method.

## 564 7.6.2 Creating boot configuration is not supported

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

567 Conditional Requirement 1:

- 568 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
569 through an instance of CIM\_ElementCapabilities.
- 570 2) The CIM\_BootServiceCapabilities.BootConfigCapabilities property array does not contain a  
571 value of 2 (Creates Boot Configuration).

572 Conditional Requirement 2:

- 573 1) An instance of CIM\_BootServiceCapabilities is not associated with the CIM\_BootService  
574 instance through an instance of CIM\_ElementCapabilities.

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

## 577 7.7 Deleting a boot configuration

578 Subclause 7.7 describes conditional behavior.

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

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

## 584 7.8 Identifying boot sources

585 Subclause 7.8 describes optional behavior.

586 An instance of CIM\_BootSourceSetting represents a source from which a boot image can be loaded  
587 during the boot process.

588 An instance of CIM\_BootSourceSetting shall be associated to one or more instances of  
589 CIM\_BootConfigSetting.

590 The CIM\_BootSourceSetting class has three boot string properties: BootString, BIOSBootString and  
591 StructuredBootString. The BootString and BIOSBootString properties may be supported. The  
592 StructuredBootString property should be supported.

### 593 7.8.1 CIM\_BootServiceCapabilities

594 When no instance of CIM\_BootServiceCapabilities exists, it is not possible to determine, via the  
595 CIM\_BootServiceCapabilities, which boot string properties are supported.

#### 596 7.8.1.1 CIM\_BootServiceCapabilities.BootStringsSupported

597 When an instance of CIM\_BootServiceCapabilities exists, its BootStringsSupported property array shall  
598 contain one or more of the values 2 (BootString), 3 (BIOSBootString) and 4 (StructuredBootString).

599 The presence of a value in the property array means that the specified boot string in each instance of  
600 CIM\_BootSourceSettings which are associated to an instance of CIM\_BootConfigSetting, which in turn is  
601 associated to the CIM\_BootService, shall not be NULL.

## 602 **7.8.2 CIM\_BootSourceSetting.ElementName property**

603 The CIM\_BootSourceSetting.ElementName property shall be a character string of variable length  
604 (pattern ".\*").

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

606 When the CIM\_BootSourceSetting.BIOSBootString property is not null, the ElementName property shall  
607 match the BIOSBootString property.

## 608 **7.8.3 CIM\_BootSourceSetting.BootString property**

609 An implementation may support the CIM\_BootSourceSetting.BootString property.

### 610 **7.8.3.1 CIM\_BootSourceSetting.BootString property is supported**

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

613 Conditional Requirement:

614 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
615 through an instance of CIM\_ElementCapabilities.

616 2) The CIM\_BootServiceCapabilities.BootStringsSupported property array contains a value of 2  
617 (BootString).

618 The CIM\_BootSourceSetting.BootString property shall contain a character string.

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

### 623 **7.8.3.2 CIM\_BootSourceSetting.BootString property is not supported**

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

626 Conditional Requirement 1:

627 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
628 through an instance of CIM\_ElementCapabilities.

629 2) The CIM\_BootServiceCapabilities.BootStringsSupported property array does not contain a  
630 value of 2 (BootString).

631 Conditional Requirement 2:

632 1) An instance of CIM\_BootServiceCapabilities is not associated with the CIM\_BootService  
633 instance through an instance of CIM\_ElementCapabilities.

634 The CIM\_BootSourceSetting.BootString property may be NULL.

## 635 **7.8.4 CIM\_BootSourceSetting.BIOSBootString property**

636 An implementation may support the CIM\_BootSourceSetting.BIOSBootString property.

**637 7.8.4.1 CIM\_BootSourceSetting.BIOSBootString property is supported**

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

640 Conditional Requirement:

- 641 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
642 through an instance of CIM\_ElementCapabilities.
- 643 2) The CIM\_BootServiceCapabilities.BootStringsSupported property array contains a value of 3  
644 (BIOSBootString).

645 The CIM\_BootSourceSetting.BIOSBootString property shall contain a character string of variable length  
646 (pattern ".\*").

647 The CIM\_BootSourceSetting.BIOSBootString property shall contain a string that identifies the boot  
648 source. The property shall match the string used by the BIOS to uniquely name the boot source in its  
649 namespace.

650 For an UEFI BIOS, the BIOSBootString property should match the output of the  
651 EFI\_DEVICE\_PATH\_TO\_TEXT\_PROTOCOL service.

**652 7.8.4.2 CIM\_BootSourceSetting.BIOSBootString property is not supported**

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

655 Conditional Requirement 1:

- 656 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
657 through an instance of CIM\_ElementCapabilities.
- 658 2) The CIM\_BootServiceCapabilities.BootStringsSupported property array does not contain a  
659 value of 3 (BIOSBootString).

660 Conditional Requirement 2:

- 661 1) An instance of CIM\_BootServiceCapabilities is not associated with the CIM\_BootService  
662 instance through an instance of CIM\_ElementCapabilities.

663 The CIM\_BootSourceSetting.BIOSBootString property may be NULL.

**664 7.8.5 CIM\_BootSourceSetting.StructuredBootString property**

665 An implementation should support the CIM\_BootSourceSetting.StructuredBootString property.

**666 7.8.5.1 CIM\_BootSourceSetting.StructuredBootString property is supported**

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

669 Conditional Requirement:

- 670 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
671 through an instance of CIM\_ElementCapabilities.
- 672 2) The CIM\_BootServiceCapabilities.BootStringsSupported property array contains a value of 4  
673 (StructuredBootString).

674 The CIM\_BootSourceSetting.StructuredBootString property shall contain a string that identifies the boot  
675 source using the following format:

676 "`<OrgID>:<identifier>:<index>`"

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

681 The value of the CIM\_BootSourceSetting.StructuredBootString for instances of CIM\_BootSourceSetting  
 682 that associate to the same CIM\_LogicalDevice should have the same OrgId,Identifier, and Index.

683 The value of the CIM\_BootSourceSetting.StructuredBootString should be unique so that this string can be  
 684 used as a reference.

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

687 **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

688 **7.8.5.2 CIM\_BootSourceSetting.StructuredBootString property is not supported**

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

691 Conditional Requirement 1:

- 692 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
 693 through an instance of CIM\_ElementCapabilities.
- 694 2) The CIM\_BootServiceCapabilities.BootStringsSupported property array does not contain a  
 695 value of 4 (StructuredBootString).

696 Conditional Requirement 2:

- 697 1) An instance of CIM\_BootServiceCapabilities is not associated with the CIM\_BootService  
 698 instance through an instance of CIM\_ElementCapabilities.

699 The CIM\_BootSourceSetting.StructuredBootString property may be NULL.

700 **7.8.6 CIM\_ConcreteDependency association**

701 An instance of a concrete subclass of CIM\_LogicalDevice may exist, which represents the boot source  
 702 device.

703 If such an instance of CIM\_LogicalDevice is instantiated, then it shall be associated with an instance of  
 704 CIM\_BootSourceSetting using an instance of CIM\_ConcreteDependency.

705 When the association is used in this manner, its Antecedent property shall reference the instance of a  
 706 concrete subclass of CIM\_LogicalDevice and its Dependent property shall reference the  
 707 CIM\_BootSourceSetting instance.

## 708 **7.9 Changing the boot order**

709 A Boot Configuration Representation may support the client changing the boot order of the boot sources  
710 associated to an instance of CIM\_BootConfigSetting through the  
711 CIM\_BootConfigSetting.ChangeBootOrder() method.

### 712 **7.9.1 Changing boot order is supported**

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

715 Conditional Requirement 1:

- 716 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
717 through an instance of CIM\_ElementCapabilities.
- 718 2) The CIM\_BootServiceCapabilities.BootConfigCapabilities property array does not contain a  
719 value of 6 (Change Boot Order Not Supported).

720 Conditional Requirement 2:

- 721 1) An instance of CIM\_BootServiceCapabilities is not associated with the CIM\_BootService  
722 instance through an instance of CIM\_ElementCapabilities.

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

### 725 **7.9.2 Changing boot order is not supported**

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

728 Conditional Requirement:

- 729 1) An instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance  
730 through an instance of CIM\_ElementCapabilities.
- 731 2) The CIM\_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 6  
732 (Change Boot Order Not Supported).

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

## 734 **7.10 Representing a set of aggregated boot sources**

735 Subclause 7.10 describes optional behavior.

736 An instance of CIM\_BootSourceSetting may represent an aggregated boot source. An example of an  
737 aggregated boot source is a BCV.

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

### 740 **7.10.1 Aggregated boot sources**

741 An instance of CIM\_BootSourceSetting shall exist representing an aggregated boot source.

### 742 **7.10.2 Aggregated boot configuration**

743 An instance of CIM\_BootConfigSetting shall exist representing the set of aggregated boot sources.



744 The ElementName property for the instance of CIM\_BootConfigSetting representing the set of  
745 aggregated boot sources shall match the value of the ElementName property of the instance of  
746 CIM\_BootSourceSetting that represents the aggregated boot source.

### 747 **7.10.3 Logical identity relationship**

748 An instance of CIM\_LogicalIdentity shall associate the instance of CIM\_BootSourceSetting with the  
749 instance of CIM\_BootConfigSetting.

#### 750 **7.10.3.1 CIM\_LogicalIdentity.SystemElement**

751 The value of the SystemElement reference shall be the instance of CIM\_BootSourceSetting that  
752 represents the aggregated boot source.

#### 753 **7.10.3.2 CIM\_LogicalIdentity.SameElement**

754 The value of the SameElement reference shall be the instance of CIM\_BootConfigSetting that represents  
755 the set of aggregated boot sources.

## 756 **7.11 Boot order during the boot process**

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

### 759 **7.11.1 CIM\_OrderedComponent association**

760 The CIM\_OrderedComponent association class shall be used to associate instance of  
761 CIM\_BootConfigSetting to each instance of CIM\_BootSourceSetting representing one of the boot sources  
762 in the boot configuration.

763 When the association is used in this manner, its GroupComponent property shall reference the  
764 CIM\_BootConfigSetting instance and its PartComponent property shall reference the  
765 CIM\_BootSourceSetting instance.

#### 766 **7.11.1.1 CIM\_OrderedComponent.AssignedSequence property**

767 When a CIM\_BootConfigSetting instance has multiple CIM\_BootSourceSetting instances associated to it  
768 through instances of the CIM\_OrderedComponent association, the value of the  
769 CIM\_OrderedComponent.AssignedSequence property shall be used to determine the sequence in which  
770 the associated CIM\_BootSourceSetting instances are used during the boot process.

771 The value of the AssignedSequence property across instances of CIM\_OrderedComponent that  
772 reference the same CIM\_BootConfigSetting shall be unique when it is not equal to zero.

773 The boot order shall be interpreted as follows:

- 774 • The AssignedSequence properties are compared across instances of CIM\_OrderedComponent  
775 that reference the same CIM\_BootConfigSetting.
- 776 • A CIM\_BootSourceSetting whose associated CIM\_OrderedComponent.AssignedSequence  
777 property is equal to zero shall be ignored and not considered part of the boot order.
- 778 • The boot order shall proceed from the lowest to the highest non-zero integer value of the  
779 AssignedSequence properties.

### 780 **7.11.2 CIM\_BootSourceSetting.FailThroughSupported**

781 The FailThroughSupported property shall describe the behavior of the boot process when the attempt to  
782 boot from a boot device represented by an instance of CIM\_BootSourceSetting is not successful.

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

785 When the FailThroughSupported property has a value of 2 (Is Not Supported), then an unsuccessful boot  
786 attempt shall result in the termination of the boot order for the remaining instances of  
787 CIM\_BootSourceSetting associated to the same instance of CIM\_BootConfigSetting.

## 788 **7.12 Settings to apply during the boot process**

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

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

### 793 **7.12.1 Settings that apply to a managed element**

794 An instance of a concrete subclass of CIM\_SettingData represents a setting that is applied to a managed  
795 element during the boot process. The instance shall be associated to either an instance of  
796 CIM\_BootConfigSetting or an instance of CIM\_BootSourceSetting through an instance of  
797 CIM\_ConcreteComponent.

798 When a setting to a managed element is applicable to an entire boot configuration, an instance of a  
799 concrete subclass of CIM\_SettingData shall be associated to the instance of CIM\_BootConfigSetting  
800 representing 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\_BootConfigSetting instance and its PartComponent property shall reference the  
803 CIM\_SettingData instance.

804 When a setting to a managed element is applicable to a specific boot source, an instance of a concrete  
805 subclass of CIM\_SettingData shall be associated to the instance of CIM\_BootSourceSetting representing  
806 the boot configuration through an instance of CIM\_ConcreteComponent.

807 When the CIM\_ConcreteComponent association is used in this manner, its GroupComponent property  
808 shall reference the CIM\_BootSourceSetting instance and its PartComponent property shall reference the  
809 CIM\_SettingData instance.

### 810 **7.12.2 Settings that apply to the boot process**

811 An instance of a concrete subclass of CIM\_BootSettingData represents a setting that is applied during the  
812 boot process but does not apply to a managed element. The setting can apply to an entire boot  
813 configuration or to a specific boot source.

814 When an instance of CIM\_BootSettingData is instantiated, then it shall be associated with an instance of  
815 CIM\_BootConfigSetting or CIM\_BootSourceSetting using an instance of 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 CIM\_BootSettingData instance.

819 When an instance of a concrete subclass of CIM\_SettingData is instantiated, then it shall be associated  
820 with an instance of CIM\_BootConfigSetting or CIM\_BootSourceSetting using an instance of  
821 CIM\_ConcreteComponent.

822 When the CIM\_ConcreteComponent association is used in this manner, its GroupComponent property  
823 shall reference the CIM\_BootConfigSetting or CIM\_BootSourceSetting instance and its PartComponent  
824 property shall reference the instance of a concrete subclass of CIM\_SettingData.

## 825 8 Methods

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

### 828 8.1 CIM\_BootService.CreateBootConfigSetting()

829 This method is conditional on the CIM\_BootServiceCapabilities.BootConfigCapabilities property array  
830 containing a value of 2 (Creates Boot Configuration). For more information, see subclause 7.6.1.

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

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

837 The input parameter, ScopingComputerSystem, shall be used to reference the Boot Configurable  
838 System, an existing CIM\_ComputerSystem, to which the new CIM\_BootConfigSetting instance shall be  
839 associated through an instance of CIM\_ElementSettingData.

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

843 When the StartingBootConfig parameter has a NULL value and the ScopingComputerSystem parameter  
844 has a non-NULL value, the implementation shall find the Default Boot Configuration associated to the  
845 CIM\_ComputerSystem instance referenced by the ScopingComputerSystem and use it as the Template  
846 Boot Configuration for the new boot configuration. If a Default Boot Configuration is not found, a return  
847 value or an exception shall be returned. When a return value is returned, it shall have a value of 2 (Error  
848 Occurred).

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

852 Upon successful completion of this method, a new Boot Configuration Representation shall exist and be a  
853 replica of the Template Boot Configuration. The new instance of CIM\_BootConfigSetting shall be  
854 associated to the instance representing the Boot Configurable System through an instance of  
855 CIM\_ElementSettingData. All properties in the new Boot Configuration Representation and Template  
856 Boot Configuration representations are expected to have the same value, except for the key properties,  
857 unless otherwise mandated in the requirements below.

- 858 • A new instance of CIM\_BootConfigSetting shall exist and be referenced by the output  
859 NewBootConfig parameter. The new CIM\_BootConfigSetting.InstanceID property shall be set to  
860 a unique value.
- 861 • A new instance of CIM\_ElementSettingData shall exist that associates the new  
862 CIM\_BootConfigSetting to the instance of the Boot Configurable System, which is specified by  
863 the ScopingComputerSystem parameter when it is non-NULL or implied by the  
864 StartingBootConfig parameter when the ScopingComputerSystem parameter is NULL.
- 865 • The CIM\_ElementSettingData.IsDefault property shall be set to 2 (Is Not Default). The  
866 CIM\_ElementSettingData.IsCurrent property shall be set to 2 (Is Not Current). The  
867 CIM\_ElementSettingData.IsNext property shall be set to 2 (Is Not Next).

- 868 • New instances of CIM\_BootSourceSetting shall exist, along with instances of  
869 CIM\_OrderedComponent, when they are present in the boot configuration represented by the  
870 Template Boot Configuration. The new instances shall be duplicates of those found in the boot  
871 configuration represented by the Template Boot Configuration, except for the key property  
872 value.
- 873 • New instances of CIM\_BootSettingData shall exist when they are present in the boot  
874 configuration represented by the Template Boot Configuration. The new instances shall be  
875 duplicates of those found in the boot configuration represented by the Template Boot  
876 Configuration, except for the key property value.
- 877 • New instances of CIM\_ConcreteComponent shall exist when they are present in the boot  
878 configuration represented by the Template Boot Configuration.
- 879 • New instances of CIM\_ConcreteDependency shall exist when they are present in the boot  
880 configuration represented by the Template Boot Configuration.
- 881 • CIM elements that are defined in a Referencing Profile are not copied.

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

884 No standard messages are defined.

885 **Table 3 – CreateBootConfigSetting() Method: Return code values**

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

886 **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

887 **8.2 CIM\_BootService.ApplyBootConfigSetting()**

888 This method is conditional on the CIM\_BootServiceCapabilities.BootConfigCapabilities property array  
889 containing a value of 3 (Applies Boot Configuration). See subclause 7.5.1 for more information.

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

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

897 The input parameter, ScopingComputerSystem, shall be used to reference the Boot Configurable  
 898 System, an existing CIM\_ComputerSystem with instances CIM\_BootConfigSetting associated to it  
 899 through an instance of CIM\_ElementSettingData.

900 When the ScopingComputerSystem parameter is NULL, the boot configuration shall be applied to each  
 901 CIM\_ComputerSystem which is associated to the instance of CIM\_BootConfigSetting referenced by the  
 902 ApplyBootConfig parameter via an instance of CIM\_ElementSettingData.

903 When the instance of CIM\_ComputerSystem referenced by ScopingComputerSystem parameter is not  
 904 associated to an instance of CIM\_BootService, a return value or an exception shall be returned. When a  
 905 return value is returned, it shall have a value of 2 (Error Occurred).

906 The input parameter, ApplyBootConfig, shall be used to provide a reference to an instance of  
 907 CIM\_BootConfigSetting associated to the Boot Configurable System for use in the boot process.

908 When the ApplyBootConfig parameter is NULL, 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 When the instance of CIM\_BootConfigSetting referenced by ApplyBootConfig parameter is not found, a  
 911 return value or an exception shall be returned. When a return value is returned, it shall have a value of 2  
 912 (Error Occurred).

913 When the instance of CIM\_BootConfigSetting referenced by ApplyBootConfig parameter is not  
 914 associated with the ScopingComputerSystem, a return value or an exception shall be returned. When a  
 915 return value is returned, it shall have a value of 2 (Error Occurred).

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

918 The return code values and parameters for the ApplyBootConfigSetting() method are specified in Table 5  
 919 and Table 6, respectively.

920 No standard messages are defined.

**Table 5 – ApplyBootConfigSetting() Method: Return code values**

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

**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

923 **8.3 CIM\_BootConfigSetting.ChangeBootOrder()**

924 The ChangeBootOrder() method shall set the order in which the instances of CIM\_BootSourceSetting are  
 925 associated to a CIM\_BootConfigSetting instance. The method has one input parameter: Source.

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

927 The input parameter, Source, is an ordered array of references to CIM\_BootSourceSetting instances that  
 928 defines the new sequence of the CIM\_BootSourceSetting instances associated to the instance of  
 929 CIM\_BootConfigSetting. Each CIM\_BootSourceSetting instance in the array shall already be associated  
 930 with this CIM\_BootConfigSetting instance through an instance of CIM\_OrderedComponent. This  
 931 parameter is required.

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

933 When any of the CIM\_BootSourceSetting instance in the Source array are not associated to the instance  
 934 of CIM\_BootConfigSetting, the implementation shall return a value of 2 (Error Occurred).

935 Upon successful completion of this method, the value of the AssignedSequence property on each  
 936 instance of CIM\_OrderedComponent shall be updated such that the values are monotonically increasing  
 937 in correlation with the position of the referenced CIM\_BootSourceSetting instance in the Source input  
 938 parameter. That is, the first position in the array shall have the lowest non-zero value for  
 939 AssignedSequence. The second position will have the second lowest value, and so on.

940 Upon successful completion of this method, the value of the AssignedSequence property on each  
 941 instance of CIM\_OrderedComponent, that associates the target CIM\_BootConfigSetting instance to a  
 942 CIM\_BootSourceSetting instance that is not present in the input array, shall be assigned a value of 0.

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

945 No standard messages are defined.

946 **Table 7 – ChangeBootOrder() Method: Return code values**

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

947 **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.

948 **8.4 Profile conventions for operations**

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

952 **8.5 CIM\_BootService**

953 Compliant implementations of this profile shall support the operations listed in Table 9 for  
 954 CIM\_BootService. Each operation shall be supported as defined in [DSP0200](#).

955 **Table 9 – Operations: CIM\_BootService**

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

956 **8.5.1 CIM\_BootService — ModifyInstance operation**

957 Subclause 8.5.1 details the specific requirements for the ModifyInstance operation applied to an instance  
 958 of CIM\_BootService.

959 When an instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance and  
 960 the CIM\_BootServiceCapabilities.ElementNameEditSupported property has a value of TRUE, the  
 961 implementation shall allow the ModifyInstance operation to change the value of the ElementName  
 962 property of the CIM\_BootService instance. The ModifyInstance operation shall enforce the length  
 963 restriction specified in the MaxElementNameLen property of the CIM\_BootServiceCapabilities instance.

964 When no instance of CIM\_BootServiceCapabilities is associated with the CIM\_BootService instance, or  
 965 the ElementNameEditSupported property of the CIM\_BootServiceCapabilities has a value of FALSE, the  
 966 implementation shall not allow the ModifyInstance operation to change the value of the ElementName  
 967 property of the CIM\_BootService instance.

968 **8.6 CIM\_BootConfigSetting**

969 Compliant implementations of this profile shall support the operations listed in Table 10 for the  
 970 CIM\_BootConfigSetting class. Each operation shall be supported as defined in [DSP0200](#).

971 **Table 10 – Operations: CIM\_BootConfigSetting**

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

972 **8.6.1 CIM\_BootConfigSetting – DeleteInstance**

973 Subclause 8.6.1 describes conditional behavior.

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

977 When the DeleteInstance operation is supported for an instance of CIM\_BootConfigSetting, upon  
 978 completion of this operation, the following instances shall be deleted:

- 979 • The target instance of CIM\_BootConfigSetting shall no longer exist.
- 980 • The instance of CIM\_ElementSettingData that associated the target CIM\_BootConfigSetting to  
 981 the instance of CIM\_ComputerSystem shall no longer exist.
- 982 • The instances of CIM\_ConcreteComponent, which associate the target instance of  
 983 CIM\_BootConfigSetting to instances of a concrete subclass of CIM\_SettingData, shall no longer  
 984 exist.
- 985 • The instances of CIM\_ConcreteComponent, which associate the target instance of  
 986 CIM\_BootConfigSetting to instances of a concrete subclass of CIM\_BootSettingData, shall no  
 987 longer exist. The instances of the associated concrete subclass of CIM\_BootSettingData shall  
 988 no longer exist.
- 989 • The instances of CIM\_OrderedComponent, which associate the target instance of  
 990 CIM\_BootConfigSetting to instances of CIM\_BootSourceSetting, shall no longer exist. The  
 991 instances of the associated CIM\_BootSourceSetting shall no longer exist.
- 992 • The instances of CIM\_ConcreteComponent, which associate instances of a concrete subclass  
 993 of CIM\_SettingData to instances of CIM\_BootSourceSetting, which in turn are associated to the  
 994 target instance of CIM\_BootConfigSetting, shall no longer exist.
- 995 • The instances of CIM\_ConcreteComponent, which associate instances of a concrete subclass  
 996 of CIM\_BootSettingData to instances of CIM\_BootSourceSetting, which in turn are associated  
 997 to the target instance of CIM\_BootConfigSetting, shall no longer exist. The instances of the  
 998 associated concrete subclass of CIM\_BootSettingData shall no longer exist.
- 999 • The instances of CIM\_ConcreteDependency, which associate instances of a concrete subclass  
 1000 of CIM\_LogicalDevice to instances of CIM\_BootSourceSetting, shall no longer exist.
- 1001 • The instance of CIM\_LogicalIdentity, which associates a deleted instance of  
 1002 CIM\_BootSourceSetting to an instance of CIM\_BootConfigSetting, shall no longer exist. The  
 1003 associated instance of CIM\_BootConfigSetting shall no longer exist. The requirements in this  
 1004 subclause shall be applied recursively to the deleted CIM\_BootConfigSetting instance.

1005 **8.7 CIM\_BootSettingData**

1006 Compliant implementations of this profile shall support the operations listed in Table 11 for the  
 1007 CIM\_BootSettingData class. Each operation shall be supported as defined in [DSP0200](#).

1008 **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

1009 **8.8 CIM\_BootSourceSetting**

1010 Compliant implementations of this profile shall support the operations listed in Table 12 for the  
 1011 CIM\_BootSourceSetting class. Each operation shall be supported as defined in [DSP0200](#).

1012 **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

1013 **8.9 CIM\_ConcreteComponent**

1014 Compliant implementations of this profile shall support the operations listed in Table 13 for the  
 1015 CIM\_ConcreteComponent class. Each operation shall be supported as defined in [DSP0200](#).

1016 **Table 13 – Operations: CIM\_ConcreteComponent**

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

1017 **8.10 CIM\_ConcreteDependency**

1018 Compliant implementations of this profile shall support the operations listed in Table 14 for the  
 1019 CIM\_ConcreteDependency class. Each operation shall be supported as defined in [DSP0200](#).

1020 **Table 14 – Operations: CIM\_ConcreteDependency**

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

1021 **8.11 CIM\_ElementCapabilities**

1022 Compliant implementations of this profile shall support the operations listed in Table 15 for the  
 1023 CIM\_ElementCapabilities class. Each operation shall be supported as defined in [DSP0200](#).

1024 **Table 15 – Operations: CIM\_ElementCapabilities**

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

1025 **8.12 CIM\_ElementSettingData**

1026 Compliant implementations of this profile shall support the operations listed in Table 16 for the  
 1027 CIM\_ElementSettingData class. Each operation shall be supported as defined in [DSP0200](#).

1028 **Table 16 – Operations: CIM\_ElementSettingData**

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

1029 **8.12.1 CIM\_ElementSettingData – ModifyInstance operation**

1030 The behavior of the ModifyInstance operation varies depending on the property of the association  
 1031 modified.

1032 **8.12.1.1 CIM\_ElementSettingData.IsDefault property**

1033 When the ModifyInstance operation is used to set the IsDefault property to a value of 1 (Is Default), the  
 1034 ModifyInstance operation shall implement the following behavior.

1035 The behavior described insures that there is at most one instance of CIM\_ElementSettingData associated  
 1036 to the Boot Configurable System whose IsDefault property has a value of 1 (Is Default) as specified in  
 1037 subclause 7.4.2, by first finding any existing instance of CIM\_ElementSettingData whose IsDefault  
 1038 property already has a value of 1 (Is Default) and modifying the value to 2 (Is Not Default).

- 1039 • Search for an instance of CIM\_ElementSettingData that associates an instance of  
 1040 CIM\_BootConfigSetting with the instance of CIM\_ComputerSystem, which is referenced by the  
 1041 target instance of CIM\_ElementSettingData where the IsDefault property has a value of 1 (Is  
 1042 Default).
- 1043 • If such an instance of CIM\_ElementSettingData is found, the ModifyInstance operation shall set  
 1044 the value of the IsDefault property to 2 (Is Not Default).
- 1045 • For the target instance of CIM\_ElementSettingData, when the IsDefault property already has a  
 1046 value of 1 (Is Default), the ModifyInstance operation shall complete successfully.
- 1047 • For the target instance of CIM\_ElementSettingData, set the value of the IsDefault property to 1  
 1048 (Is Default).

### 1049 8.12.1.2 CIM\_ElementSettingData.IsNext property

1050 When the ModifyInstance operation is used to set the IsNext property to a value of 1 (Is Next), the  
1051 ModifyInstance operation shall implement the following behavior.

1052 The behavior described insures that there is at most one instance of CIM\_ElementSettingData associated  
1053 to the Boot Configurable System whose IsNext property has a value of 1 (Is Next) as specified in  
1054 subclause 7.4.4, by first finding any existing instance of CIM\_ElementSettingData whose IsNext property  
1055 already has a value of 1 (Is Next) and modifying the value to 2 (Is Not Next).

- 1056 • Search for an instance of CIM\_ElementSettingData that associates an instance of  
1057 CIM\_BootConfigSetting with the instance of CIM\_ComputerSystem, which is referenced by the  
1058 target instance of CIM\_ElementSettingData where the IsNext property has a value of 1 (Is  
1059 Next).
- 1060 • If such an instance of CIM\_ElementSettingData is found, the ModifyInstance operation shall set  
1061 the value of the IsNext property to 2 (Is Not Next).
- 1062 • For the target instance of CIM\_ElementSettingData, when the IsNext property already has a  
1063 value of 1 (Is Next), the ModifyInstance operation shall complete successfully.
- 1064 • For the target instance of CIM\_ElementSettingData, when the IsNext property has a value other  
1065 than 1 (Is Next), set the value of the IsNext property to 1 (Is Next).

1066 When the ModifyInstance operation is used to set the IsNext property to a value of 3 (Is Next For Single  
1067 Use), the ModifyInstance operation shall implement the following behavior.

1068 The behavior described insures that there is at most one instance of CIM\_ElementSettingData associated  
1069 to the Boot Configurable System whose IsNext property has a value of 3 (Is Next For Single Use) as  
1070 specified in subclause 7.4.5, by first finding any existing instance of CIM\_ElementSettingData whose  
1071 IsNext property already has a value of 3 (Is Next For Single Use) and modifying the value to 2 (Is Not  
1072 Next).

- 1073 • For the target instance of CIM\_ElementSettingData, when the IsNext property has a value of 1  
1074 (Is Next), the ModifyInstance operation shall fail.
- 1075 • Search for an instance of CIM\_ElementSettingData that associates an instance of  
1076 CIM\_BootConfigSetting with the instance of CIM\_ComputerSystem referenced by the target  
1077 instance of CIM\_ElementSettingData where the IsNext property has a value of 3 (Is Next For  
1078 Single Use).
- 1079 • If such an instance of CIM\_ElementSettingData is found, the ModifyInstance operation shall set  
1080 the value of the IsNext property to 2 (Is Not Next).
- 1081 • For the target instance of CIM\_ElementSettingData, when the IsNext property already has a  
1082 value of 3 (Is Next For Single Use), the ModifyInstance operation shall complete successfully.
- 1083 • For the target instance of CIM\_ElementSettingData, when the IsNext property has a value  
1084 neither 1 (Is Next) nor 3 (Is Next For Single Use), set the value of the IsNext property to 3 (Is  
1085 Next For Single Use).

### 1086 8.12.1.3 CIM\_ElementSettingData.IsCurrent property

1087 The ModifyInstance operation shall not be used to set the IsCurrent property.

## 1088 8.13 CIM\_BootServiceCapabilities

1089 Compliant implementations of this profile shall support the operations listed in Table 17 for the  
1090 CIM\_BootServiceCapabilities class. Each operation shall be supported as defined in [DSP0200](#).

1091

**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

1092

**8.14 CIM\_HostedService**

1093

Compliant implementations of this profile shall support the operations listed in Table 18 for the CIM\_HostedService class. Each operation shall be supported as defined in [DSP0200](#).

1094

1095

**Table 18 – Operations: CIM\_HostedService**

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

1096

**8.15 CIM\_LogicalIdentity**

1097

Compliant implementations of this profile shall support the operations listed in Table 19 for the CIM\_LogicalIdentity class. Each operation shall be supported as defined in [DSP0200](#).

1098

1099

**Table 19 – Operations: CIM\_LogicalIdentity**

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

1100

**8.16 CIM\_OrderedComponent**

1101

Compliant implementations of this profile shall support the operations listed in Table 20 for the CIM\_OrderedComponent class. Each operation shall be supported as defined in [DSP0200](#).

1102

1103

**Table 20 – Operations: CIM\_OrderedComponent**

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

1104 **8.17 CIM\_ServiceAffectsElement**

1105 Compliant implementations of this profile shall support the operations listed in Table 21 for the  
 1106 CIM\_ServiceAffectsElement class. Each operation shall be supported as defined in [DSP0200](#).

1107 **Table 21 – Operations: CIM\_ServiceAffectsElement**

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

1108 **9 Use cases**

1109 Clause 9 contains object diagrams and use cases for the *Boot Control Profile*.

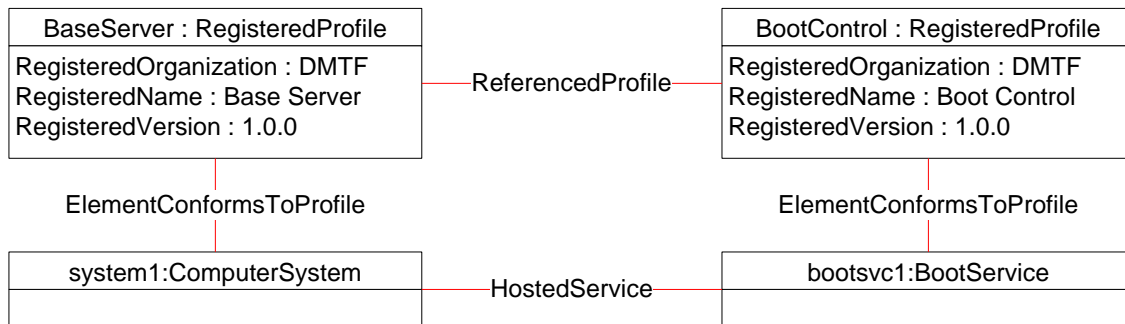
1110 **9.1 Advertising the profile conformance**

1111 The object diagram in Figure 2 shows how instances of CIM\_RegisteredProfile are used to identify the  
 1112 version of the *Boot Control Profile* with which an instance of CIM\_BootService and its associated  
 1113 instances are conformant. An instance of CIM\_RegisteredProfile exists for each profile that is  
 1114 instrumented in the system. One instance of CIM\_RegisteredProfile identifies the DMTF *Base Server*  
 1115 *Profile*, version 1.0.0. The other instance identifies the DMTF *Boot Control Profile*, version 1.0.0. The  
 1116 Central Instance is the CIM\_BootService. The Scoping Instance is the CIM\_ComputerSystem instance.

1117 This instance of CIM\_ComputerSystem is conformant with the *Base Server Profile* version 1.0.0 as  
 1118 indicated by the CIM\_ElementConformsToProfile association to the CIM\_RegisteredProfile instance.

1119 This instance of CIM\_BootService is conformant with the *Boot Control Profile* version 1.0.0 as indicated  
 1120 by the CIM\_ElementConformsToProfile association to the CIM\_RegisteredProfile instance.

1121 The CIM\_ReferencedProfile relationship between *BaseServer* and *BootControl* places the  
 1122 CIM\_BootService instance within the scope of *BaseServer*.



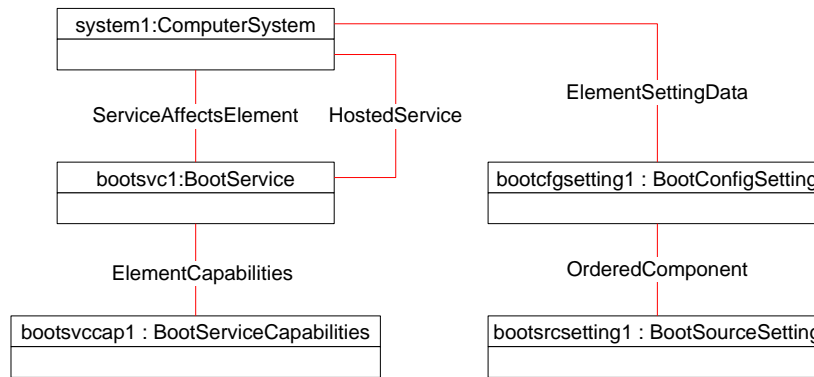
1123

1124 **Figure 2 – Registered Profile**

1125 **9.2 Object diagram for a monolithic server**

1126 Figure 3 shows the CIM instances required to control the boot configuration for a single, monolithic  
 1127 server, *system1*. *System1* hosts the boot service, *bootsvc1*, which is used to control the boot  
 1128 configuration, *bootcfgsetting1*, for *system1*. *System1* is also identified as the Boot Configurable System  
 1129 through the CIM\_ServiceAffectsElement association. The capabilities of *bootsvc1* are defined by  
 1130 *bootsvccap1*.

1131 The boot configuration, *bootcfgsetting1*, has one boot source, *bootsrcsetting1*.



1132

1133 **Figure 3 – Monolithic server object diagram**

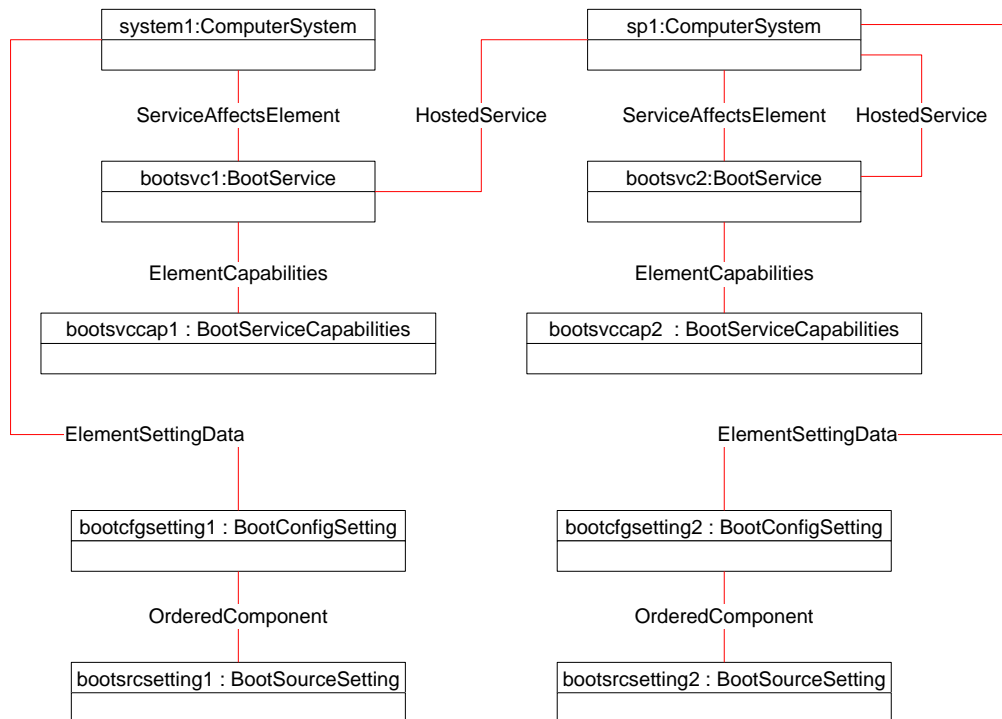
1134 **9.3 Object diagram for a monolithic server with service processor**

1135 Figure 4 shows the CIM instances required to control the boot configuration for a single, monolithic  
 1136 server, *system1*, with an attached service processor, *sp1*. The boot service, *bootsvc1*, is hosted by the  
 1137 service processor and is responsible for managing the boot configuration, *bootcfgsetting1*, for *system1*.

1138 Optionally, the service processor may host another boot configuration service, *bootsvc2*, to control its  
 1139 own boot configuration, *bootcfgsetting2*.

1140 The capabilities of *bootsvc1* and *bootsvc2* are defined by *bootsvccap1* and *bootsvccap2* respectively.

1141 Each boot configuration (*bootcfgsetting1*, *bootcfgsetting2*) has one boot source (*bootsrcsetting1*,  
 1142 *bootsrcsetting2*), respectively.



1143

1144 **Figure 4 – Monolithic server with service processor object diagram**

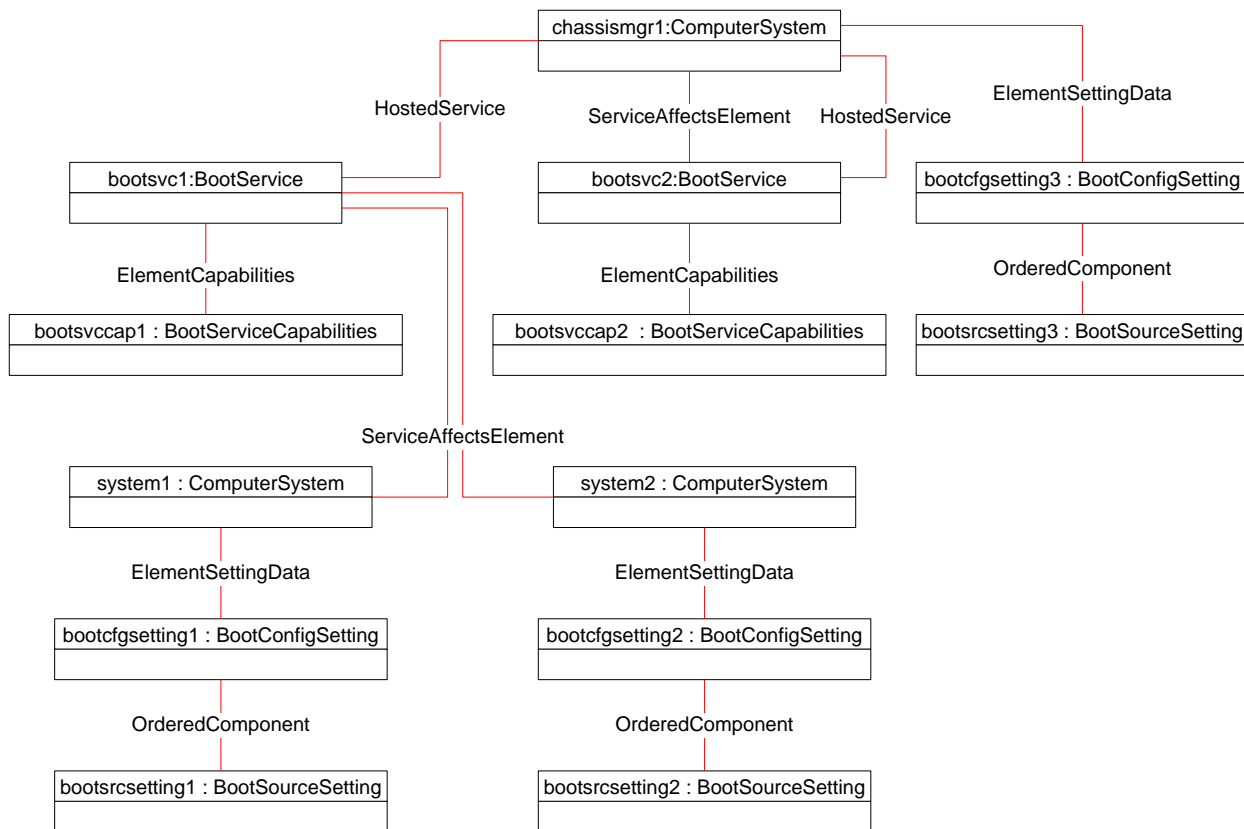
1145 **9.4 Object diagram for a modular system**

1146 Figure 5 shows the CIM instances required to control the boot configuration for a modular system. The  
 1147 boot service, *bootsvc1*, is hosted by the chassis manager, *chassismgr1*, and is responsible for managing  
 1148 the boot configuration for two blade systems, *system1* and *system2*. *System1* and *system2* each have  
 1149 one boot configuration, *bootcfgsetting1* and *bootcfgsetting2* respectively.

1150 Optionally, the chassis manager may host another boot configuration service, *bootsvc2*, to control its own  
 1151 boot configuration, *bootcfgsetting3*.

1152 The capabilities of *bootsvc1* and *bootsvr2* are defined by *bootsvccap1* and *bootsvccap2* respectively.

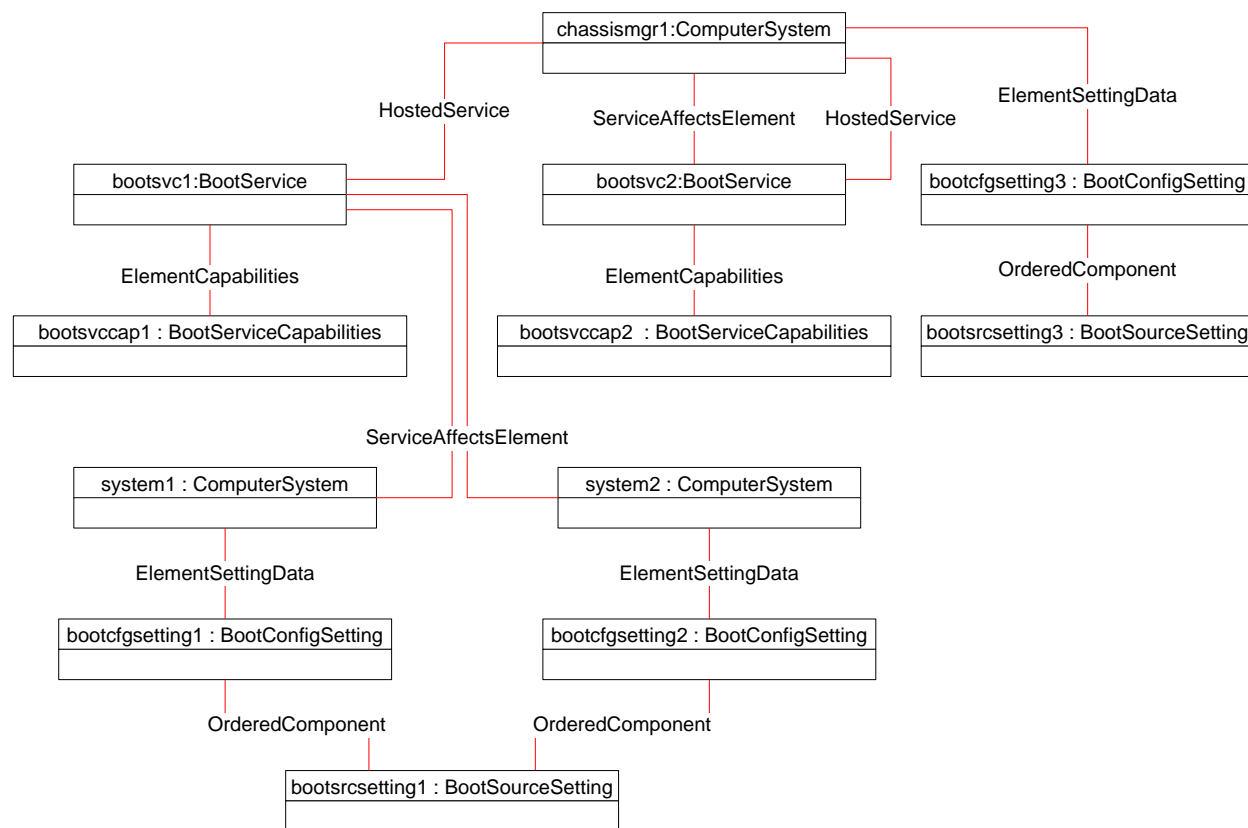
1153 Each boot configuration (*bootcfgsetting1*, *bootcfgsetting2*, *bootcfgsetting3*) has one boot source  
 1154 (*bootsrcsetting1*, *bootsrcsetting2*, *bootsrcsetting3*), respectively.



1155

1156 **Figure 5 – Modular system object diagram**

1157 Alternative representation of modular blade system boot sources is shown in the Figure 6. The blade  
 1158 computer systems, in this example, have the same boot sources, and thus, the representation of boot  
 1159 sources can be optimized by instantiating only one CIM\_BootSourceSetting shared between the  
 1160 respective CIM\_BootConfigSetting instances. This optimization is especially useful when modular system  
 1161 contains many blade computer systems with the similar boot sources. Each blade boot configuration  
 1162 (*bootcfgsetting1*, *bootcfgsetting2*) has one boot source (*bootsrcsetting1*).



1163

1164

Figure 6 – Modular system object diagram

1165 **9.5 PXE boot source**

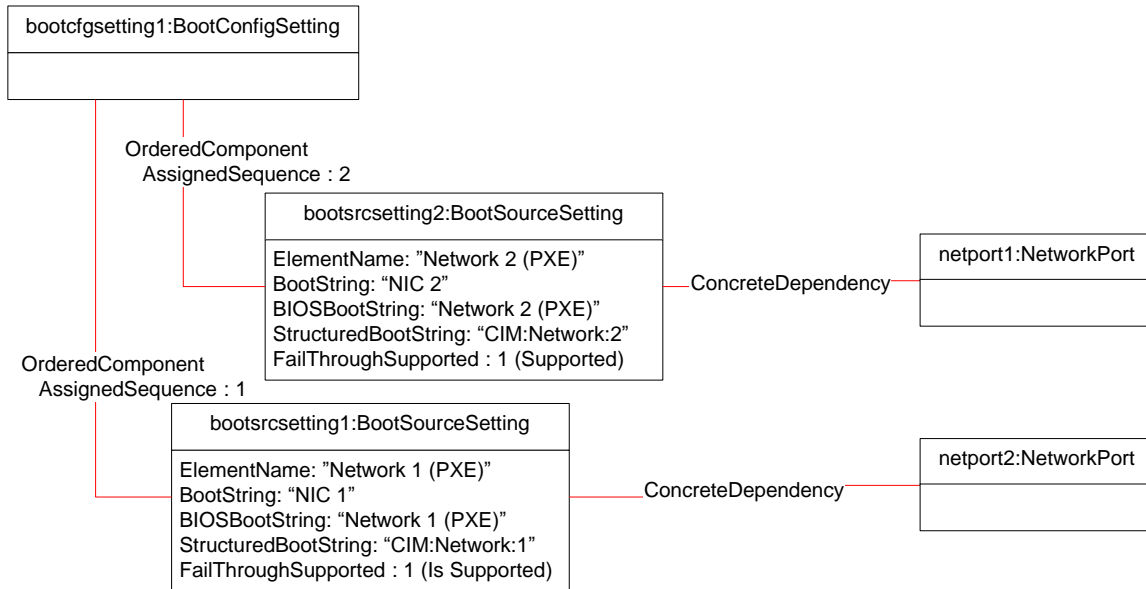
1166 Figure 7 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has two boot sources  
 1167 associated to it, *bootsrcsetting1* and *bootsrcsetting2*, which are both network ports.

1168 A network port can support various protocols. Both *bootsrcsetting1* and *bootsrcsetting2* designate the  
 1169 PXE protocol in their `BIOSBootString` property. The two `CIM_ConcreteDependency` associations to  
 1170 instances of `CIM_NetworkPort` are *netport2* and *netport1*, respectively.

1171 The `AssignedSequence` property values on the `OrderedComponent` associations indicate that the boot  
 1172 order is *bootsrcsetting1* followed by *bootsrcsetting2*.

1173 On *bootsrcsetting1*, the `FailThroughSupported` property value of 1 (Is Supported) indicates that if the  
 1174 *bootsrcsetting1* fails or times out, the boot process should proceed to *bootsrcsetting2* on *netport1*.





1175

1176

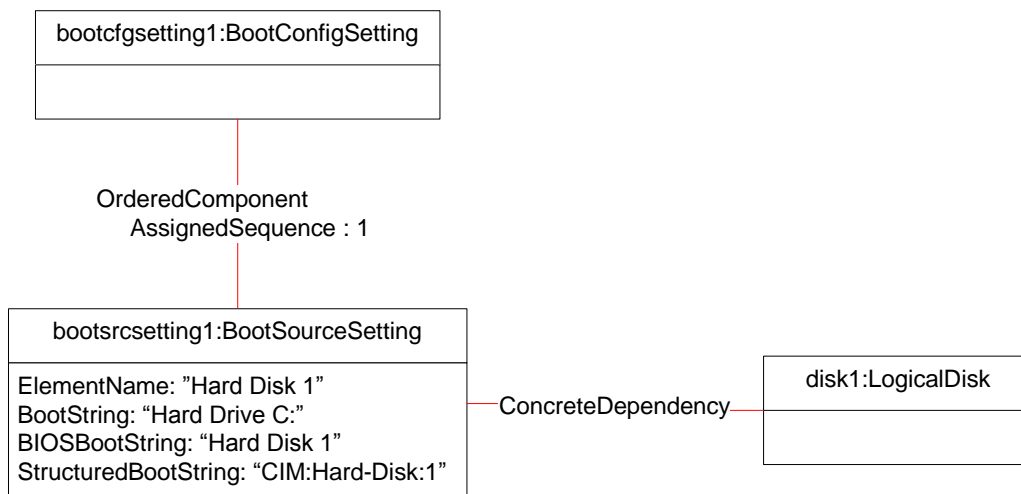
Figure 7 – PXE boot sources object diagram

1177 **9.6 Disk boot source**

1178 Figure 8 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has a single boot  
 1179 source associated to it, *bootsrcsetting1*, which is a disk device.

1180 The CIM\_BootSourceSetting.ElementName property identifies "Hard Disk 1" as the boot source, which  
 1181 matches the BIOSBootString property. The BootString property contains the string "C:", which could be  
 1182 interpreted by the boot process to assign the hard drive the letter "C". The CIM\_ConcreteDependency  
 1183 association relates *bootsrcsetting1* to a CIM\_LogicalDisk (*disk1*).

1184 Because there is only one boot source, the value of the CIM\_BootSourceSetting.FailThroughSupported is  
 1185 not meaningful.



1186

1187

Figure 8 – Booting from disk

1188 **9.7 Local CDROM and floppy boot sources**

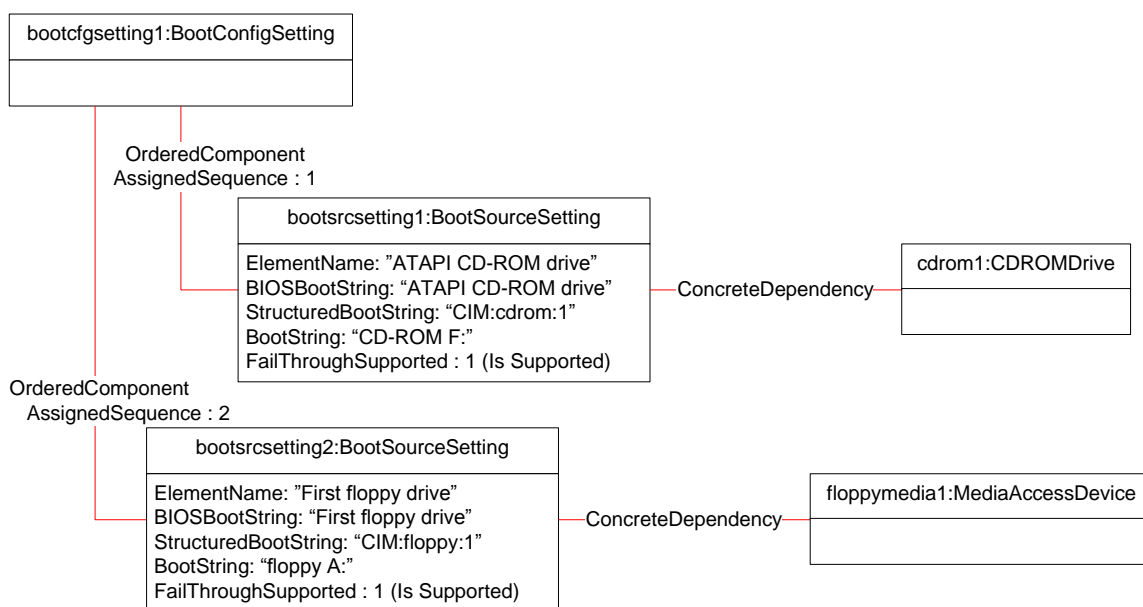
1189 Figure 9 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has two boot sources  
 1190 associated to it, *bootsrcsetting1* and *bootsrcsetting2*. *Bootsrcsetting1* is a CD-ROM device;  
 1191 *bootsrcsetting2* is a floppy drive.

1192 The AssignedSequence property of the CIM\_OrderedComponent associations is set such that booting is  
 1193 attempted from the CD-ROM drive first and then the floppy drive.

1194 The BootString property for the CD-ROM drive, *bootsrcsetting1*, contains the string "F:", which could be  
 1195 interpreted by the boot process to assign the floppy drive the letter "F". The CIM\_ConcreteDependency  
 1196 association relates *bootsrcsetting1* to a CIM\_CDROMDrive (*cdrom1*).

1197 The BootString property for the floppy drive, *bootsrcsetting2*, contains the string "A:", which could be  
 1198 interpreted by the boot process to assign the floppy drive the letter "A". The CIM\_ConcreteDependency  
 1199 association relates *bootsrcsetting2* to a CIM\_DisketteDrive (*floppymedia1*).

1200 On *bootsrcsetting1*, the value of the FailThroughSupported property set to 1 (Is Supported) specifies that  
 1201 if the *bootsrcsetting1*, the CD-ROM device, fails or times out, then the boot process should proceed to  
 1202 *bootsrcsetting2*, the floppy device.



1203

1204 **Figure 9 – Booting from CDROM and floppy**

1205 **9.8 Representing IPL and Boot Control Vector (BCV) lists**

1206 Figure 10 shows an instance diagram for a boot configuration, *bootcfgsetting1*, composed of an IPL and  
 1207 BCV list of boot devices.

1208 To represent the IPL list, *bootcfgsetting1* has three boot sources associated to it, *bootsrcsetting1*,  
 1209 *bootsrcsetting2*, and *bootsrcsetting3*. *Bootsrcsetting1* is a CD-ROM device. *Bootsrcsetting2* is a floppy  
 1210 drive. *Bootsrcsetting3* is a BCV device (boot control vector).

1211 The AssignedSequence property of the CIM\_OrderedComponent associations is set such that booting is  
 1212 attempted from the CD-ROM drive first and then the BCV device. Booting from the floppy device is not  
 1213 attempted because the AssignedSequence property is set to 0. The

1214 CIM\_BootConfigSetting.FailThroughSupported property value of 1 (Is Supported) specifies that the boot  
 1215 process should proceed to the second boot source if the first boot source fails or times out.

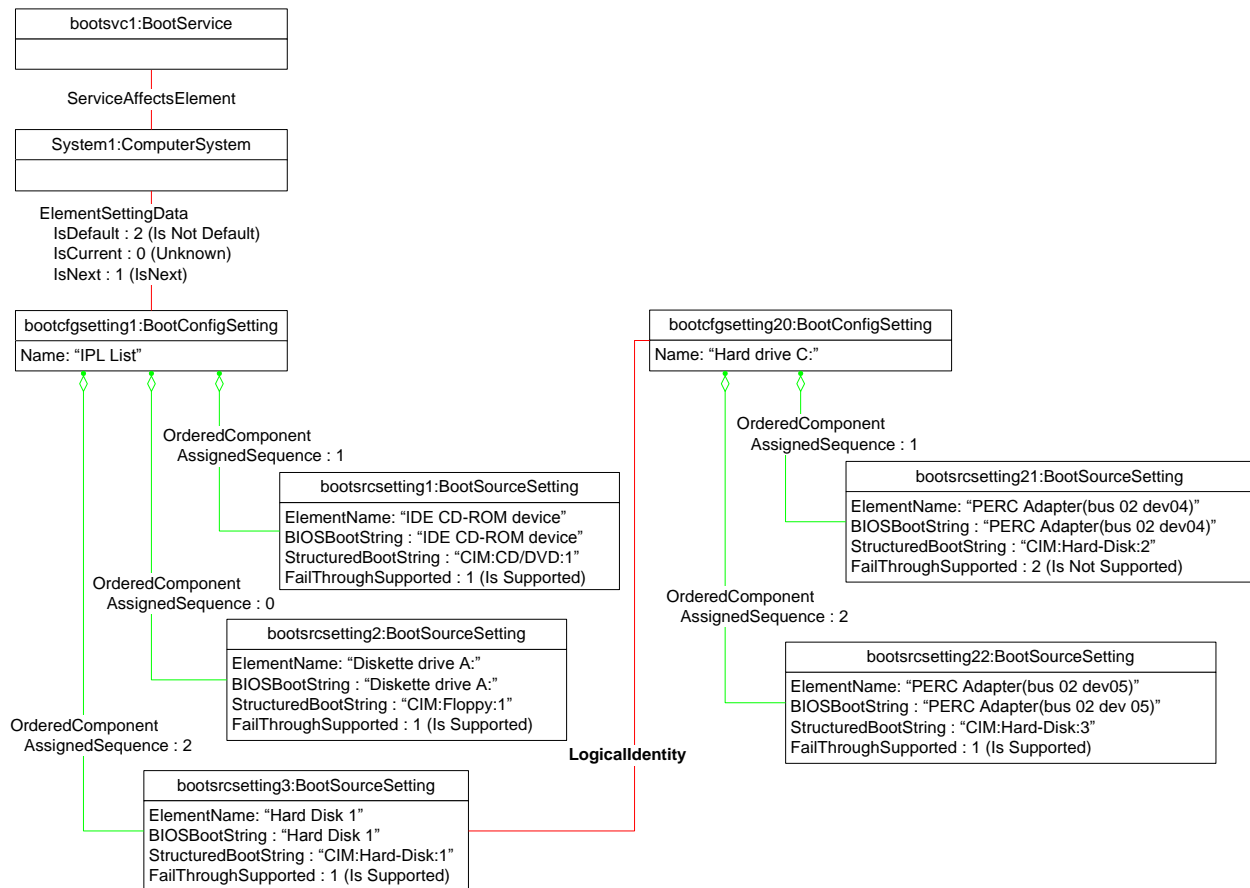
1216 In the diagram, the BCV device is a SCSI controller that may have multiple bootable SCSI devices  
 1217 attached to it. This relationship is represented by an instance of CIM\_LogicalIdentity between  
 1218 *bootsrcsetting3* and an instance of CIM\_BootConfigSetting, *bootcfgsetting20*.

1219 The boot configuration, *bootcfgsetting20*, has two boot sources associated to it, *bootsrcsetting21* and  
 1220 *bootsrcsetting22*. Both boot sources are hard disk devices.

1221 The AssignedSequence property of the CIM\_OrderedComponent associations is set such that booting is  
 1222 attempted from *bootsrcsetting21* first and from *bootsrcsetting22*.

1223 On *bootsrcsetting21*, the FailThroughSupported property value of 2 (Is Not Supported) specifies that if the  
 1224 *bootsrcsetting21*, "CIM:Hard-Disk:2", fails or times out, then the boot process should terminate the boot  
 1225 order for *bootconfigsetting20*.

1226 In total, this use case describes a source boot order that proceeds from *bootsrcsetting1* to  
 1227 *bootsrcsetting21*. *bootsrcsetting2* will never be used because of its AssignedSequence value of 0 and  
 1228 *bootsrcsetting22* will never be used because of the FailThroughSupported value on *bootsrcsetting21*.



1229

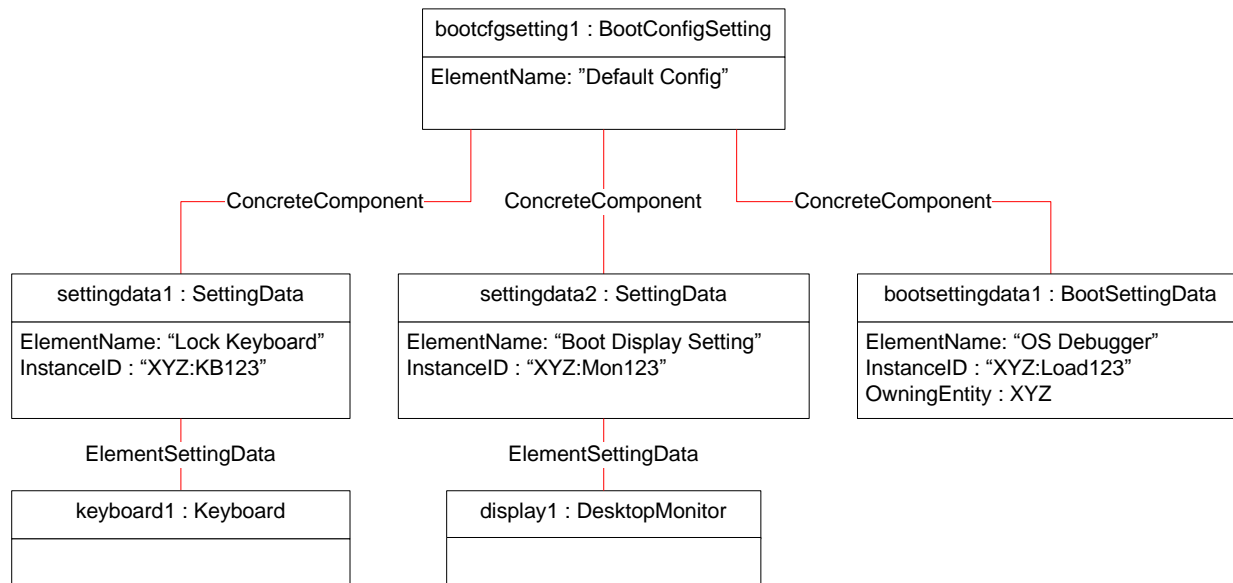
1230

Figure 10 – Booting from IPL and BCV devices

1231 **9.9 Representing settings and boot settings**

1232 Figure 11 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has settings that  
 1233 need to be applied to the computer system during the boot process. Two example concrete subclasses of  
 1234 CIM\_SettingData, *settingdata1* and *settingdata2*, apply to concrete subclasses of CIM\_LogicalDevice,  
 1235 *keyboard1* and *display1*. The instance of an example concrete subclass of CIM\_BootSettingData is  
 1236 *bootsettingdata1*.

1237 Being associated to the instance of CIM\_BootConfigSetting, the settings apply to the entire boot process  
 1238 that uses *bootcfgsetting1*. Note that any of these settings could be associated to an instance of  
 1239 CIM\_BootSourceSetting, which would reduce the scope of the settings to just the specified boot source.



1240

1241 **Figure 11 – Setting data and boot setting data**

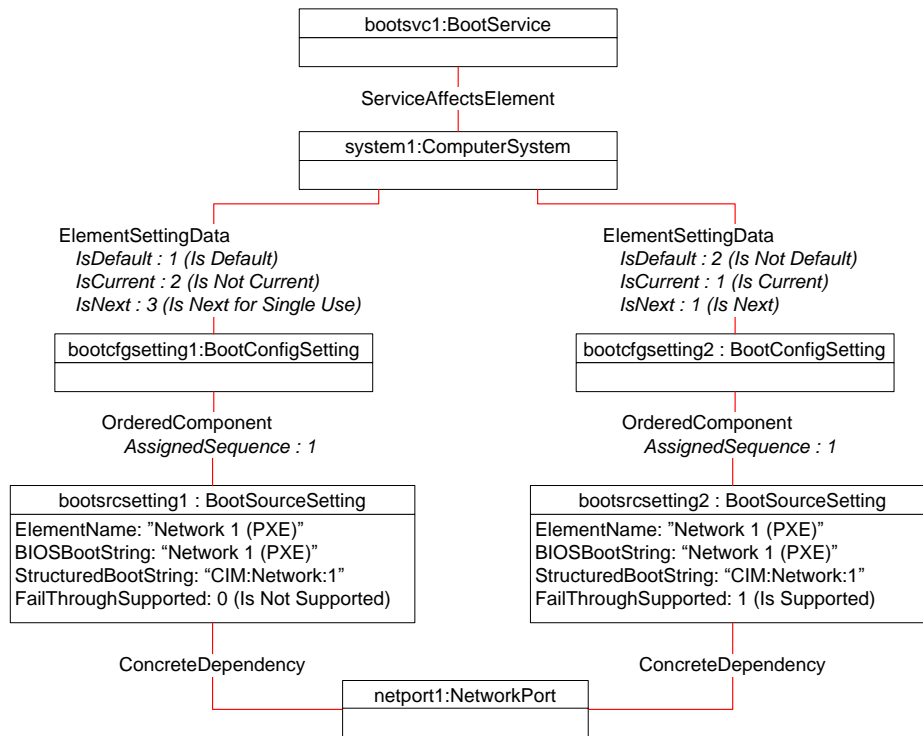
1242 **9.10 Representing the same boot device**

1243 Figure 12 shows an instance diagram for two boot configurations, *bootcfgsetting1* and *bootcfgsetting2*,  
 1244 which used the same boot device, *netport1*.

1245 The *bootcfgsetting1* instance represents the next one time boot configuration and is associated to an  
 1246 instance of CIM\_BootSourceSetting, *bootsrcsetting1*. The instance *bootsrcsetting1* is associated to the  
 1247 boot device, *netport1*.

1248 There are two different CIM\_BootSourceSetting instances *bootsrcsetting1* and *bootsrcsetting2* are  
 1249 associated to the same boot device, *netport1*.

1250 Since both instances of CIM\_BootSourceSetting are associated to the same boot device, the  
 1251 StructuredBootString property in each instance is set to the same string (“CIM:Network:1”), namely, both  
 1252 strings have the same OrgId, Identifier, and Index.



1253

1254

Figure 12 - Representing the same boot device

1255

### 9.11 Representing the default boot configuration for a computer system

1256

1257

1258

1259

Figure 13 shows an instance diagram for a Boot Configurable System, *system1*. *System1* has a single boot configuration, *bootcfgsetting2*. This boot configuration is a Default Boot Configuration, because the value of the *ElementSettingData.IsDefault* property is set to 1 (Is Default). There are no Next Boot Configuration or Current Boot Configuration.

1260

1261

1262

1263

*Bootcfgsetting2* is associated with two instances of *CIM\_BootSourceSetting* (*bootsrcsetting1* and *bootsrcsetting2*), through instances of *CIM\_OrderedComponent*. The respective *CIM\_OrderedComponent.AssignedSequence* properties designate the order in which the boot process should use the boot sources (*bootsrcsetting1* followed by *bootsrcsetting2*).

1264

1265

1266

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

1267

1268

1269

1270

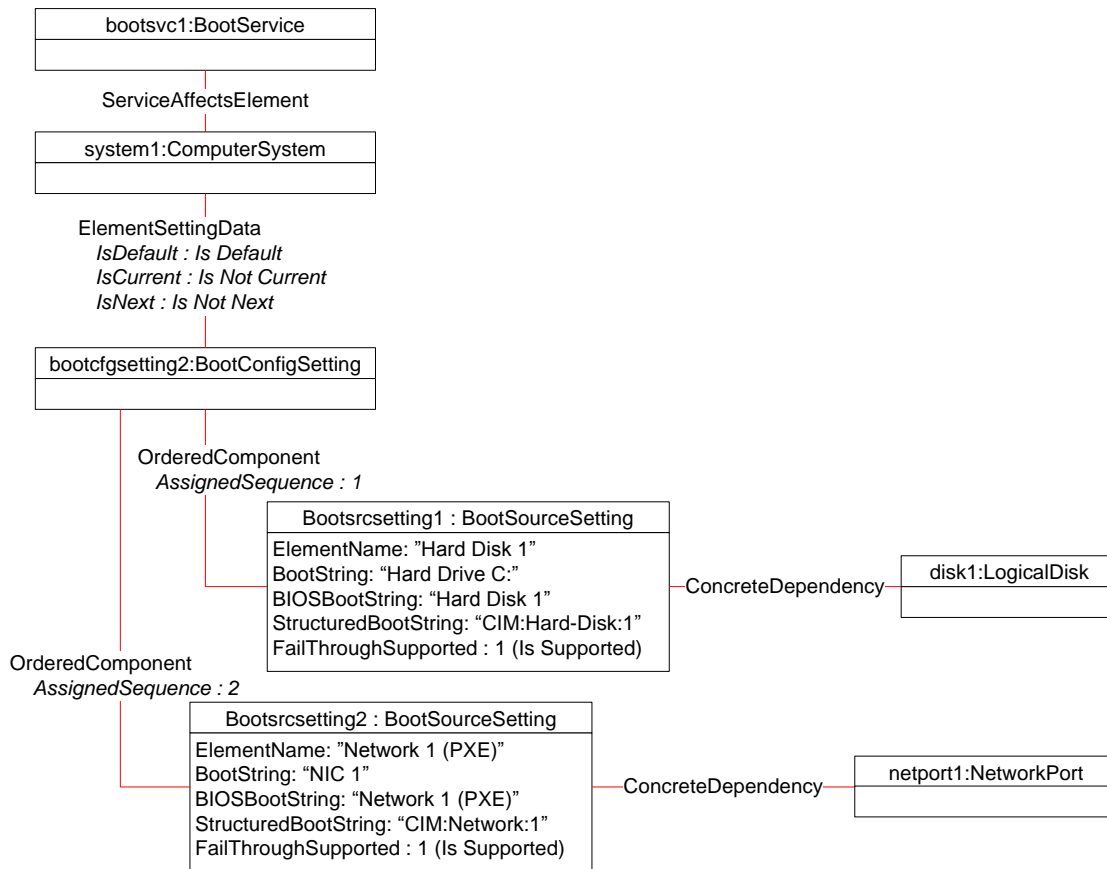
When the system represented by *system1* is enabled, the boot process will not be initiated because there is no Next Boot Configuration for the boot process to use. The system, *system1*, will be in an enabled, but not booted, state. One could manually boot the system from this state by applying an existing boot configuration (see subclause 9.15).

1271

1272

1273

*System1* would initiate the boot process if the Default Boot Configuration were also the Next Boot Configuration (see subclause 9.12) or a new boot configuration is created as the Next Boot Configuration (see subclause 9.14).



1274

1275

Figure 13 – Default boot configuration object diagram

1276

### 9.12 Representing the next boot configuration for a computer system

1277

Figure 14 shows an instance diagram for a Boot Configurable System, *system1*. *System1* has a single boot configuration, *bootcfgsetting2*. This boot configuration is a Default Boot Configuration, because the value of the *ElementSettingData.IsDefault* property is set to 1 (Is Default). This boot configuration is also the Next Boot Configuration, because the value of the *ElementSettingData.IsNext* property is set to 1 (Is Next).

1282

*Bootcfgsetting2* is associated with two instances of *CIM\_BootSourceSetting* (*bootsrcsetting1* and *bootsrcsetting2*), through instances of *CIM\_OrderedComponent*. The respective *CIM\_OrderedComponent.AssignedSequence* properties designate the order in which the boot process should use the boot sources (*bootsrcsetting1* followed by *bootsrcsetting2*).

1286

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

1289

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

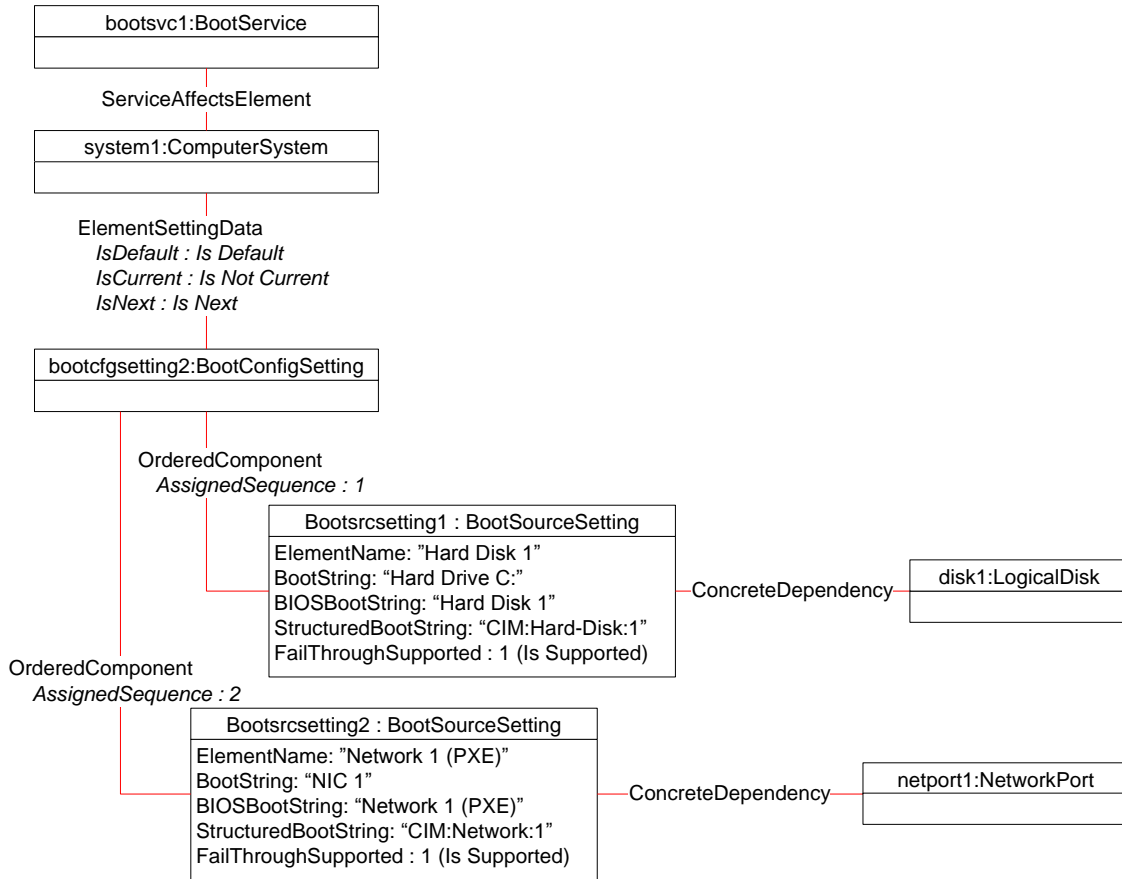
1290

1291

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

1292

1293



1294

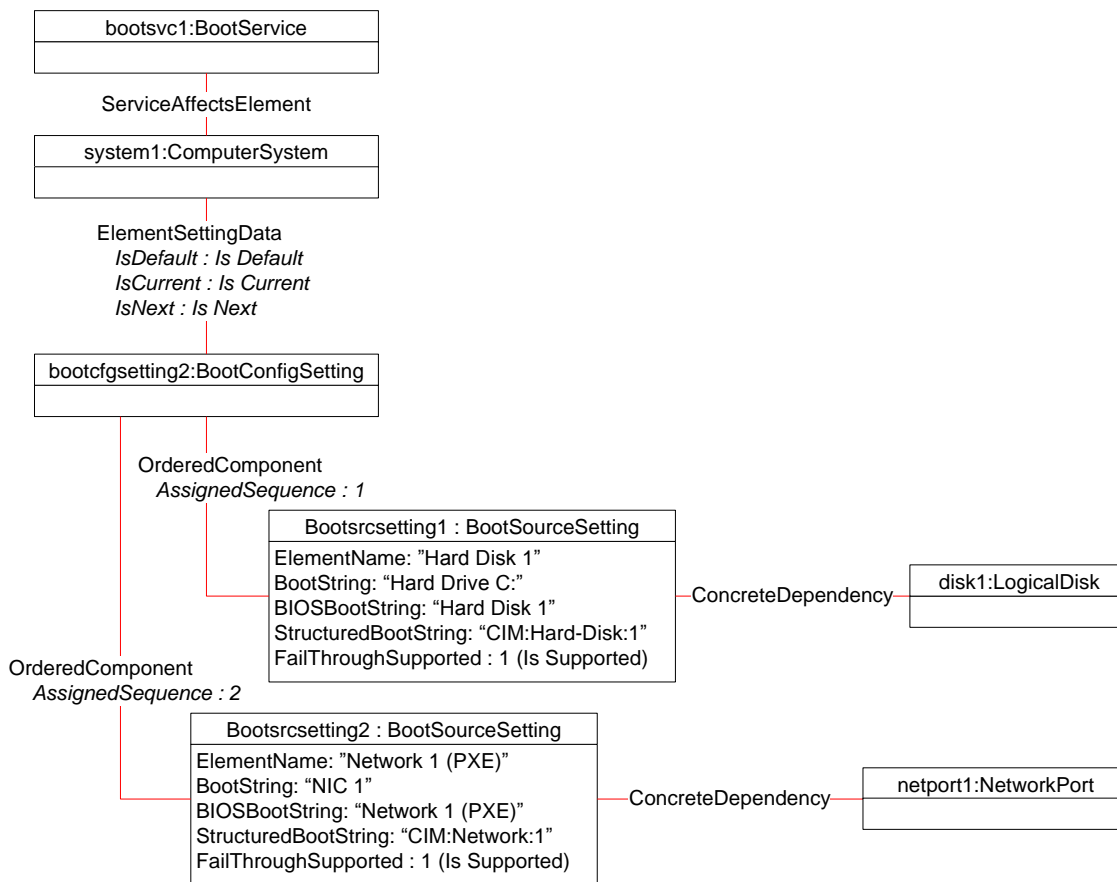
1295

Figure 14 – Next boot configuration object diagram

1296 **9.13 Representing the current boot configuration for a booted computer system**

1297 Figure 15 shows an instance diagram for the Boot Configurable System, *system1*, described in the  
 1298 previous use case (see Figure 14) after it has been successfully booted.

1299 The boot configuration, *bootcfgsetting1*, is now the Current Boot Configuration, because the value of the  
 1300 ElementSettingData.IsCurrent property is set to 1 (Is Current). *Bootcfgsetting1* is still concurrently the  
 1301 Default Boot Configuration and the Next Boot Configuration.



1302

1303

Figure 15 – Boot configuration for a booted system object diagram

1304 **9.14 Create a new boot configuration**

1305 Referencing the object diagram in Figure 13, a client could create a new boot configuration as follows:

- 1306 1) From the Boot Configurable System, *system1*, find the instance of `CIM_BootService` that
- 1307 manages the boot configurable system by traversing the `CIM_ServiceAffectsElement`
- 1308 association.
- 1309 2) Verify that the `CreateBootConfigSetting()` method is supported (see subclause 9.28). If not, a
- 1310 new boot configuration cannot be created.
- 1311 3) Find an existing instance of `CIM_BootConfigSetting` to use as the template. For this use case,
- 1312 *bootcfgsetting2* is the only template configuration available.
- 1313 4) Create the new boot configuration, *bootcfgsetting4*, by invoking the
- 1314 `CIM_BootService.CreateBootConfigSetting()` method. The `ScopingComputerSystem` parameter
- 1315 is set to *system1* and the `StartingBootConfig` parameter is set to *bootcfgsetting2*.

1316 Figure 16 shows the instance diagram after the `CreateBootConfigSetting()` method has been invoked and

1317 successfully completed on the computer system, *system1*, shown in Figure 15. The new boot

1318 configuration, *bootcfgsetting4*, is associated to *system1* through a new instance of

1319 `CIM_ElementSettingData`.

1320 In the new instance of `CIM_ElementSettingData`, the `IsDefault` property is set to 2 (Is Not Default); the

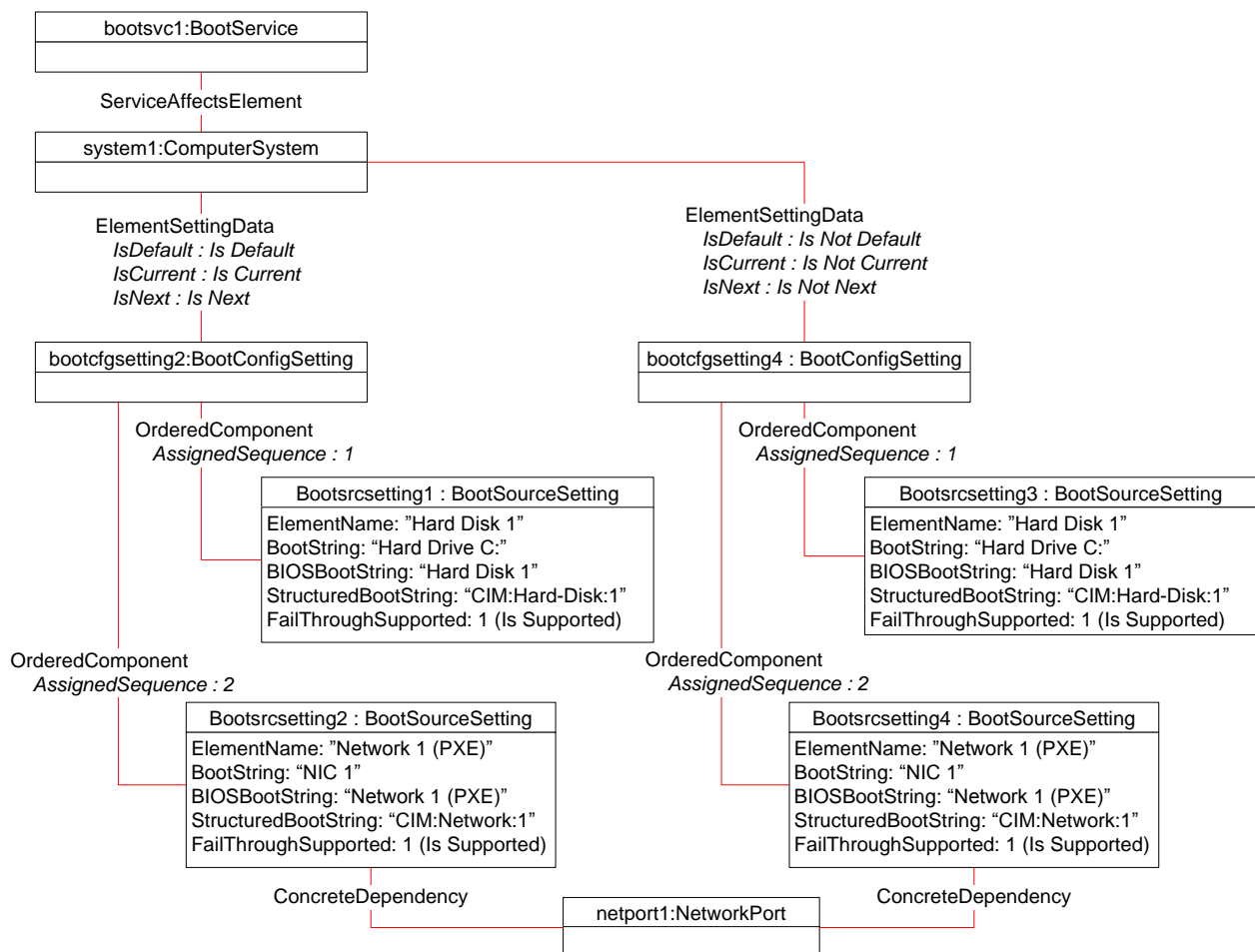
1321 `IsCurrent` property is set to 2 (Is Not Current); and the `IsNext` property is set to 2 (Is Not Next).



1322 *Bootcfgsetting4* is associated through instances of *CIM\_OrderedComponent* to two instances of  
 1323 *CIM\_BootSourceSetting* (*bootsrcsetting3* and *bootsrcsetting4*), which are copies of *bootsrcsetting1* and  
 1324 *bootsrcsetting2*, respectively.

1325 The instance of *CIM\_NetworkPort* is not copied. *CIM\_NetworkPort* is a concrete subclass of  
 1326 *CIM\_LogicalDevice*, which is not part of the *Boot Control Profile*. However, an instance of  
 1327 *CIM\_ConcreteDependency* has been created that associates the instance of *CIM\_NetworkPort* to the  
 1328 new instance of *CIM\_BootSourceSetting* (*bootsrcsetting4*).

1329 *CIM\_LogicalDisk* has been elided from the object diagram to make the diagram less cluttered, but the  
 1330 instance of *CIM\_LogicalDisk* is also not copied. An instance of *CIM\_ConcreteDependency* is created that  
 1331 associates the existing instance of *CIM\_LogicalDisk* to the new instance of *CIM\_BootSourceSetting*  
 1332 (*bootsrcsetting3*).



1333

1334

Figure 16 – System with new *CIM\_BootConfigSetting*

1335 **9.15 Apply an existing boot configuration**

1336 Referencing the object diagram in Figure 13, a client could apply a boot configuration as follows:

- 1337 1) Find the instance of *CIM\_BootService* for the boot configurable system as outlined in subclause
- 1338 9.16.
- 1339 2) Verify that the *ApplyBootConfigSetting()* method is supported (see subclause 9.29). If not, a
- 1340 boot configuration cannot be applied.

- 1341           3) Find the existing instances of CIM\_BootConfigSetting for *system1* (see subclause 9.19). In this  
1342           example, this results in *bootcfgsetting2*. Pick one of them to use as the boot configuration to  
1343           apply.
- 1344           4) Apply the selected boot configuration, *bootcfgsetting2*, by invoking the  
1345           CIM\_BootService.ApplyBootConfigSetting() method. The ScopingComputerSystem parameter  
1346           is set to *system1* and the BootConfigSetting parameter is set to *bootcfgsetting2*.

1347           The ApplyBootConfigSetting() method will boot *system1* by applying the boot configuration specified in  
1348           *bootcfgsetting2*. If *system1* is currently booted, an implementation has the option of rejecting the  
1349           ApplyBootConfigSetting() request or of rebooting the system.

## 1350   **9.16 Find the boot service for a computer system**

1351           A client can find the boot service for a given computer system as follows:

- 1352           1) For the instance of CIM\_ComputerSystem, representing the given computer system, select the  
1353           referencing instance of CIM\_BootService, representing the boot control service for the server,  
1354           through the CIM\_ServiceAffectsElement association.

## 1355   **9.17 Find the boot configuration for a computer system**

1356           A client can find the boot configurations for a computer system as follows:

- 1357           1) From the instance of CIM\_ComputerSystem, enumerate the CIM\_ElementSettingData  
1358           associations with CIM\_BootConfigSetting as the SettingData reference.

## 1359   **9.18 Find the boot service capabilities for a computer system**

1360           A client can find the boot service capabilities for a computer system as follows:

- 1361           1) Find the boot service for the computer system as specified in subclause 9.16.
- 1362           2) Select the instance of CIM\_BootServiceCapabilities through the CIM\_ElementCapabilities  
1363           association.

## 1364   **9.19 Find the current boot configuration for a computer system**

1365           A client can find the current boot configuration for a computer system as follows:

- 1366           1) From the instance of CIM\_ComputerSystem, enumerate the CIM\_ElementSettingData  
1367           associations with CIM\_BootConfigSetting as the SettingData reference.
- 1368           2) Find the instance of CIM\_ElementSettingData whose IsCurrent property is set to 1 (Is Current).
- 1369           3) The CIM\_BootConfigSetting instance referenced by this association instance represents the  
1370           current boot configuration.

## 1371   **9.20 Find the default boot configuration for a computer system**

1372           A client can find the default boot configuration for a computer system as follows:

- 1373           1) From the instance of CIM\_ComputerSystem, enumerate the CIM\_ElementSettingData  
1374           associations with CIM\_BootConfigSetting as the SettingData reference.
- 1375           2) Find the instance of CIM\_ElementSettingData whose IsDefault property is set to 1 (Is Default).
- 1376           3) The CIM\_BootConfigSetting instance referenced by this association instance represents the  
1377           default boot configuration.

### 1378 **9.21 Find the boot configuration that will be used during the next reboot for a** 1379 **computer system**

1380 A client can find the boot configuration that will be used during a computer system's next reboot as  
1381 follows:

- 1382 1) For the instance of CIM\_ComputerSystem, enumerate the CIM\_ElementSettingData  
1383 associations with CIM\_BootConfigSetting as the SettingData reference.
- 1384 2) Find the CIM\_ElementSettingData instances for the instance whose IsNext property is set to 3  
1385 (Is Next For Single Use). The CIM\_BootConfigSetting instance referenced by this association  
1386 instance represents the next boot configuration.
- 1387 3) If no instance is found, find the instance of CIM\_ElementSettingData whose IsNext property is  
1388 set to 1 (Is Next). The CIM\_BootConfigSetting instance referenced by this association instance  
1389 represents the next boot configuration.

### 1390 **9.22 Make a boot configuration applicable for subsequent reboots**

1391 A client can make a boot configuration apply to a computer system for subsequent reboots as follows:

- 1392 1) From the instance of CIM\_ComputerSystem, find the CIM\_BootConfigSetting of interest as  
1393 outlined in subclauses 9.9 through 9.13.
- 1394 2) On the instance of the CIM\_ElementSettingData association that associates the instance of  
1395 CIM\_ComputerSystem to the instance of CIM\_BootConfigSetting, use the intrinsic  
1396 ModifyInstance() to change the IsNext property to 1 (Is Next).

1397 Note that this boot configuration applies for all subsequence reboots, unless it is overridden by a Next  
1398 Single Use Boot Configuration that is associated to the CIM\_ComputerSystem of interest.

### 1399 **9.23 Make a boot configuration applicable for the next reboot only**

1400 A client can make a boot configuration apply to a computer system for only the next reboot as follows:

- 1401 1) From the instance of CIM\_ComputerSystem, find the CIM\_BootConfigSetting of interest as  
1402 outlined in subclauses 9.9 through 9.13.
- 1403 2) On the instance of the CIM\_ElementSettingData association that associates the instance of  
1404 CIM\_ComputerSystem to the instance of CIM\_BootConfigSetting, use the intrinsic  
1405 ModifyInstance() to change the IsNext property to 3 (Is Next For Single Use).

1406 The behavior of this property after the next boot is specified in subclause 7.4.5.

### 1407 **9.24 Determine whether the computer system supports PXE boot**

1408 A client can determine if the computer system supports PXE boot as follows:

- 1409 1) For the instance of CIM\_ComputerSystem enumerate its instances of CIM\_BootConfigSetting  
1410 as outline in subclause 9.19.
- 1411 2) For each instance of CIM\_BootConfigSetting, enumerate the instances of  
1412 CIM\_BootSourceSetting.
- 1413 3) For each CIM\_BootSourceSetting, inspect the BootString, BIOSBootString, or  
1414 StructuredBootString property to determine if PXE is supported.

### 1415 **9.25 Find the boot order for a computer system for the next reboot**

1416 This use case references the object diagram in Figure 10, which represents a complex boot order.

1417 A client can find the boot order for the next reboot of a computer system as follows:

- 1418 1) From the instance of CIM\_ComputerSystem, *system1*, find the CIM\_BootConfigSetting that will  
1419 be used during the next reboot, *bootcfgsetting1* (see subclause 3)).
- 1420 2) Determine the boot order for *bootcfgsetting1* by enumerating the CIM\_OrderedComponent  
1421 associations with *bootcfgsetting1* as the GroupComponent reference. The results in this  
1422 example would be *bootsrcsetting1*, *bootsrcsetting2* and *bootsrcsetting3*.
- 1423 3) Use the CIM\_OrderedComponent.AssignedSequence property to determine the boot order. The  
1424 boot order in this example will be *bootsrcsetting1* followed by *bootsrcsetting3*. The boot source  
1425 represented by *bootsrcsetting2* will be ignored because its associated AssignedSequence  
1426 property value is 0.
- 1427 4) For each boot source, determine whether any it contains additional boot sources by checking  
1428 for a CIM\_LogicalIdentity association to an instance of CIM\_BootConfigSetting; in this example,  
1429 *bootcfgsetting20*, and repeat steps in this subclause recursively to find the boot order of the  
1430 associated boot sources.

## 1431 9.26 Change the boot order for a computer system

1432 This use case references the object diagram in Figure 14.

1433 A client can change the boot order for a computer system as follows:

- 1434 1) Find the boot configuration of interest from the set of boot configurations for the computer  
1435 system as outlined in subclause 9.19.
- 1436 2) Find the set of boot sources for the boot configuration by following the OrderedComponent  
1437 associations from the selected boot configuration representation (*bootcfgsetting2*) to all  
1438 instances of CIM\_BootSourceSetting. In this example, this results in *bootsrcsetting1* and  
1439 *bootsrcsetting2*.
- 1440 3) Determine the desired boot order.
- 1441 4) Create an array of CIM\_BootSourceSetting references. Assign the existing boot sources to the  
1442 array in the new order. For instance, if one wanted to reverse the boot order in this example, the  
1443 array would contain *bootsrcsetting2* at index 0 and *bootsrcsetting1* at index 1.
- 1444 5) Invoke the ChangeBootOrder() method on the selected instance of CIM\_BootConfigSetting. The  
1445 Source parameter is set to the array created above.

1446 NOTE The order of each boot configuration must be changed independently. Thus if the computer system has a  
1447 complex boot structure, such as that illustrated in Figure 10, changing the boot order for the system may require  
1448 changing the boot order for multiple CIM\_BootConfigSetting instances.

## 1449 9.27 Determine whether BootService.ElementName is modifiable

1450 A client can determine whether the ElementName can be modified as follows:

- 1451 1) Find the CIM\_BootServiceCapabilities instance associated with the CIM\_BootService instance  
1452 through the CIM\_ElementCapabilities association.
- 1453 2) If a CIM\_BootConfigCapabilities instance cannot be found, then the  
1454 CIM\_BootService.ElementName property cannot be modified.
- 1455 3) Query the value of the CIM\_BootServiceCapabilities.ElementNameEditSupported.
- 1456 4) If the value is TRUE, the CIM\_BootService.ElementName property can be modified
- 1457 5) If the value of ElementNameEditSupported has a value of FALSE, then the  
1458 CIM\_BootService.ElementName property cannot be modified.

1459 **9.28 Determine whether a new boot configuration can be created**

1460 A client can determine whether a new boot configuration can be created as follows:

- 1461 1) Find the CIM\_BootServiceCapabilities instance that is associated with the CIM\_BootService  
1462 instance through the CIM\_ElementCapabilities association.
- 1463 2) Query the value of the CIM\_BootServiceCapabilities.BootConfigCapabilities property array. If  
1464 the array contains the value 2 (Creates Boot Configuration), the client's ability to create a boot  
1465 configuration is supported.
- 1466 3) If the BootConfigCapabilities property array does not contain the value 2 (Creates Boot  
1467 Configuration), or there is not an instance of CIM\_BootServiceCapabilities associated with the  
1468 CIM\_BootService instance, a boot configuration cannot be created.

1469 **9.29 Determine whether a boot configuration can be applied**

1470 A client can determine whether a boot configuration can be manually applied to the boot configurable  
1471 systems as follows:

- 1472 1) Find the CIM\_BootServiceCapabilities instance that is associated with the CIM\_BootService  
1473 instance through the CIM\_ElementCapabilities association.
- 1474 2) Query the value of the CIM\_BootServiceCapabilities.BootConfigCapabilities property array. If  
1475 the array contains the value 3 (Applies Boot Configuration), the client's ability to manually apply  
1476 a boot configuration is supported.
- 1477 3) If the BootConfigCapabilities property array does not contain the value 3 (Applies Boot  
1478 Configuration), or there is not an instance of CIM\_BootServiceCapabilities associated with the  
1479 CIM\_BootService instance, a boot configuration cannot be manually applied.

1480 **10 CIM Elements**

1481 Table 22 shows the instances of CIM Elements for this profile. Instances of the following CIM Elements  
1482 shall be implemented as described in Table 22. Clauses 7 (“Implementation”) and 8 (“Methods”) may  
1483 impose additional requirements on these elements.

1484 **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.

1485 **10.1 CIM\_RegisteredProfile**

1486 CIM\_RegisteredProfile identifies the *Boot Control Profile* in order for a client to determine whether an  
 1487 instance of CIM\_ComputerSystem is conformant with this profile. The CIM\_RegisteredProfile class is  
 1488 defined by the [Profile Registration Profile](#). With the exception of the mandatory values specified for the  
 1489 properties below, the behavior of the CIM\_RegisteredProfile instance is per the [Profile Registration](#)  
 1490 [Profile](#). Table 23 contains the requirements for elements of this class.

1491 **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).

1492 NOTE Previous versions of this document included the suffix 'Profile' for the RegisteredName value. If  
 1493 implementations querying for RegisteredName value find the suffix 'Profile', they should ignore the suffix, with any  
 1494 surrounding white spaces, before any comparison is done with the value as specified in this document.

1495 **10.2 CIM\_BootService**

1496 The CIM\_BootService class represents the ability to view and control the boot settings of a computer  
 1497 system. Table 24 contains the requirements for elements of this class.

1498 **Table 24 – Class: CIM\_BootService**

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

1499 **10.3 CIM\_BootServiceCapabilities**

1500 Support of the CIM\_BootServiceCapabilities class is optional.

1501 When supported, CIM\_BootServiceCapabilities is used to indicate the capabilities of the boot service.  
 1502 Table 25 contains the requirements for elements of this class.

1503

**Table 25 – Class: CIM\_BootServiceCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	<b>Key</b>
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.

1504 **10.4 CIM\_BootConfigSetting**

1505 The CIM\_BootConfigSetting class represents a boot configuration of a computer system. Table 26  
 1506 contains the requirements for elements of this class.

1507

**Table 26 – Class: CIM\_BootConfigSetting**

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

1508 **10.5 CIM\_BootSettingData**

1509 Support of the CIM\_BootSettingData class is optional.

1510 The CIM\_BootSettingData class represents the settings that apply during booting of a computer system.  
 1511 Table 27 contains the requirements for elements of this class.

1512 For each property added in a concrete subclass of CIM\_BootSettingData, there shall be a Description  
 1513 qualifier that contains a string which describes the setting. When the range of the setting is bounded and  
 1514 discrete, the Values and ValueMap qualifiers should contain the values and name of the values,  
 1515 respectively, which are applicable for the setting.

1516

**Table 27 – Class: CIM\_BootSettingData**

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

1517 **10.6 CIM\_BootSourceSetting**

1518 Support of the CIM\_BootSourceSetting class is mandatory.

1519 The CIM\_BootSourceSetting class represents a boot source, from which booting is attempted during the  
 1520 boot process. Table 28 contains the requirements for elements of this class.

1521

**Table 28 – Class: CIM\_BootSourceSetting**

Elements	Requirement	Notes
InstanceID	Mandatory	<b>Key</b>
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.

1522 **10.7 CIM\_ConcreteComponent**

1523 Subclause 10.7 describes optional behavior.

1524 **10.7.1 Relating CIM\_BootConfigSetting to a concrete subclass of CIM\_SettingData**

1525 When supported, the CIM\_ConcreteComponent association is used to relate an instance of a concrete  
 1526 subclass of CIM\_SettingData to a CIM\_BootConfigSetting instance. Table 29 contains the requirements  
 1527 for elements of this class.

1528

**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 "*".

1529 **10.7.2 Relating CIM\_BootConfigSetting to a concrete subclass of CIM\_BootSettingData**

1530 When supported, the CIM\_ConcreteComponent association is used to relate an instance of a concrete  
 1531 subclass of CIM\_BootSettingData to a CIM\_BootConfigSetting instance. Table 30 contains the  
 1532 requirements for elements of this class.

1533

**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 "*".



1534 **10.7.3 Relating CIM\_BootSourceSetting to a concrete subclass of CIM\_SettingData**

1535 When supported, the CIM\_ConcreteComponent association is used to relate an instance of a concrete  
 1536 subclass of CIM\_SettingData to a CIM\_BootSourceSetting instance. Table 31 contains the requirements  
 1537 for elements of this class.

1538 **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 "**".

1539 **10.7.4 Relating CIM\_BootSourceSetting to a concrete subclass of CIM\_BootSettingData**

1540 When supported, the CIM\_ConcreteComponent association is used to relate an instance a concrete  
 1541 subclass of CIM\_BootSettingData to a CIM\_BootSourceSetting instance. Table 32 contains the  
 1542 requirements for elements of this class.

1543 **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 "**".

1544 **10.8 CIM\_ConcreteDependency**

1545 Subclause 10.8 describes optional behavior.

1546 When supported, the CIM\_ConcreteDependency association is used to relate the dependency of a  
 1547 CIM\_BootSourceSetting instance on an instance of a concrete subclass of CIM\_LogicalDevice. Table 33  
 1548 contains the requirements for elements of this class.

1549 **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 "**".

1550 **10.9 CIM\_ElementCapabilities**

1551 Subclause 10.9 describes optional behavior.

1552 When supported, the CIM\_ElementCapabilities association is used to relate an instance of  
 1553 CIM\_BootServiceCapabilities with an instance of CIM\_BootService. Table 34 contains the requirements  
 1554 for elements of this class.

1555 **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".

1556 **10.10 CIM\_ElementSettingData**

1557 The CIM\_ElementSettingData association is used to relate the CIM\_BootConfigSetting instance to the  
 1558 CIM\_ComputerSystem instance to which it applies. Table 35 contains the requirements for elements of  
 1559 this class.

1560 **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 7.4.
IsCurrent	Mandatory	See subclause 7.4.
IsNext	Mandatory	See subclause 7.4.

1561 **10.11 CIM\_HostedService**

1562 The CIM\_HostedService association is used to relate the CIM\_BootService to the CIM\_ComputerSystem  
 1563 on which it is hosted. Table 36 contains the requirements for elements of this class.

1564 **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 "*".

1565 **10.12 CIM\_LogicalIdentity**

1566 Support of the CIM\_LogicalIdentity association is conditional.

1567 Conditional Requirement: The support is required if instances of CIM\_BootSourceSetting are used to  
 1568 represent aggregated boot sources; see subclause 7.10.

1569 When supported, CIM\_LogicalIdentity is used to associate an instance of CIM\_BootSourceSetting with an  
 1570 instance of CIM\_BootConfigSetting. Table 37 contains the requirements for elements of this class.

1571 **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"

1572 **10.13 CIM\_OrderedComponent**

1573 Support of the CIM\_OrderedComponent association is mandatory.

1574 When supported, the CIM\_OrderedComponent association is used to indicate the order in which  
 1575 CIM\_BootSourceSetting instances should be attempted for a CIM\_BootConfigSetting instance. Table 38  
 1576 contains the requirements for elements of this class.

1577 **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.

1578 **10.14 CIM\_ServiceAffectsElement**

1579 The CIM\_ServiceAffectsElement association is used to associate the CIM\_BootService instance with a  
 1580 CIM\_ComputerSystem instance that it affects. Table 39 contains the requirements for elements of this  
 1581 class.

1582 **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)

1583

**ANNEX A  
(informative)**

**Change log**

1584  
1585  
1586  
1587

Version	Date	Description
1.0.0	11/03/2008	Final Standard
1.0.1	06/22/2009	DMTF Standard Release
1.0.2	05/03/2010	Errata release to correct the class diagram. Fix mantis errata.
1.1.0	04/28/2014	Phrase 7.8.5.1, add wording describing when the StructuredBootString should unique and when should be the same (1799, 2015). Also added a use case (clause 9.10) to show the usage.  Fix incorrect usage of 'Messages' column in operations table – remove references to subclauses (2061). Fix subclause reference (1586). Fix table reference to add Table 6 (1587).

1588