

1



2

Document Number: DSP1012

3

Date: 2008-11-03

4

5

Version: 1.0.0

6 **Boot Control Profile**

7 **Document Type: Specification**

8 **Document Status: Final Standard**

9 **Document Language: E**

1 Copyright Notice

2 Copyright © 2008 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

3 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
4 management and interoperability. Members and non-members may reproduce DMTF specifications and
5 documents for uses consistent with this purpose, provided that correct attribution is given. As DMTF
6 specifications may be revised from time to time, the particular version and release date should always be
7 noted.

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18 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

Contents

1			
2	1	Scope	9
3	2	Normative References.....	9
4	2.1	Approved References	9
5	2.2	Other References.....	9
6	3	Terms and Definitions	9
7	4	Symbols and Abbreviated Terms	11
8	5	Synopsis.....	11
9	6	Description	12
10	6.1	Class Diagram	14
11	7	Implementation.....	15
12	7.1	CIM_BootService	15
13	7.2	CIM_ComputerSystem.....	16
14	7.3	Representing Boot Service Capabilities	16
15	7.4	Boot Configurations	17
16	7.5	Applying the Boot Configuration	18
17	7.6	Creating a Boot Configuration	19
18	7.7	Deleting a Boot Configuration	20
19	7.8	Identifying Boot Sources.....	20
20	7.9	Changing the Boot Order.....	23
21	7.10	Representing a Set of Aggregated Boot Sources.....	24
22	7.11	Boot Order During the Boot Process	25
23	7.12	Settings to Apply During the Boot Process.....	26
24	8	Methods.....	27
25	8.1	CIM_BootService.CreateBootConfigSetting()	27
26	8.2	CIM_BootService.ApplyBootConfigSetting()	28
27	8.3	CIM_BootConfigSetting.ChangeBootOrder()	29
28	8.4	Profile Conventions for Operations.....	30
29	8.5	CIM_BootService	31
30	8.6	CIM_BootConfigSetting	31
31	8.7	CIM_BootSettingData	33
32	8.8	CIM_BootSourceSetting	33
33	8.9	CIM_ConcreteComponent	33
34	8.10	CIM_ConcreteDependency	33
35	8.11	CIM_ElementCapabilities	34
36	8.12	CIM_ElementSettingData	34
37	8.13	CIM_BootServiceCapabilities	36
38	8.14	CIM_HostedService	36
39	8.15	CIM_LogicalIdentity	36
40	8.16	CIM_OrderedComponent	36
41	8.17	CIM_ServiceAffectsElement	38
42	9	Use Cases.....	39
43	9.1	Advertising the Profile Conformance	39
44	9.2	Object Diagram for a Monolithic Server.....	39
45	9.3	Object Diagram for a Monolithic Server with Service Processor.....	40
46	9.4	Object Diagram for a Modular System.....	41
47	9.5	PXE Boot Source	41
48	9.6	Disk Boot Source	42
49	9.7	Local CDROM and Floppy Boot Sources	43
50	9.8	Representing IPL and Boot Control Vector (BCV) Lists	44
51	9.9	Representing Settings and Boot Settings.....	45
52	9.10	Representing the Default Boot Configuration for a Computer System.....	46
53	9.11	Representing the Next Boot Configuration for a Computer System.....	47
54	9.12	Representing the Current Boot Configuration for a Booted Computer System.....	48

Boot Control Profile

1	9.13	Create a New Boot Configuration	49
2	9.14	Apply an Existing Boot Configuration	50
3	9.15	Find the Boot Service for a Computer System	51
4	9.16	Find the Boot Configuration for a Computer System.....	51
5	9.17	Find the Boot Service Capabilities for a Computer System.....	51
6	9.18	Find the Current Boot Configuration for a Computer System.....	51
7	9.19	Find the Default Boot Configuration for a Computer System	51
8	9.20	Find the Boot Configuration that Will Be Used during the Next Reboot for a Computer	
9		System	52
10	9.21	Make a Boot Configuration Applicable for Subsequent Reboots	52
11	9.22	Make a Boot Configuration Applicable for the Next Reboot Only.....	52
12	9.23	Determine If the Computer System Supports PXE Boot	52
13	9.24	Find the Boot Order for a Computer System for the Next Reboot.....	52
14	9.25	Change the Boot Order for a Computer System	53
15	9.26	Determine Whether BootService.ElementName Is Modifiable	53
16	9.27	Determine Whether a New Boot Configuration Can Be Created	54
17	9.28	Determine Whether a Boot Configuration Can Be Applied.....	54
18	10	CIM Elements.....	55
19	10.1	CIM_RegisteredProfile.....	55
20	10.2	CIM_BootService	56
21	10.3	CIM_BootServiceCapabilities	56
22	10.4	CIM_BootConfigSetting	57
23	10.5	CIM_BootSettingData	57
24	10.6	CIM_BootSourceSetting	57
25	10.7	CIM_ConcreteComponent	58
26	10.8	CIM_ConcreteDependency	59
27	10.9	CIM_ElementCapabilities	59
28	10.10	CIM_ElementSettingData	60
29	10.11	CIM_HostedService	60
30	10.12	CIM_LogicalIdentity	60
31	10.13	CIM_OrderedComponent	61
32	10.14	CIM_ServiceAffectsElement	61
33			

1

List of Figures

2	Figure 1 – Boot Control Profile: Class Diagram.....	14
3	Figure 2 – Registered Profile	39
4	Figure 3 – Monolithic Server Object Diagram.....	40
5	Figure 4 – Monolithic Server with Service Processor Object Diagram	41
6	Figure 5 – Modular System Object Diagram.....	41
7	Figure 6 – PXE Boot Sources Object Diagram.....	42
8	Figure 7 – Booting from Disk	43
9	Figure 8 – Booting from CDROM and Floppy.....	44
10	Figure 9 – Booting from IPL and BCV Devices.....	45
11	Figure 10 – Setting Data and Boot Setting Data.....	46
12	Figure 11 – Default Boot Configuration Object Diagram	47
13	Figure 12 – Next Boot Configuration Object Diagram	48
14	Figure 13 – Boot Configuration for a Booted System Object Diagram	49
15	Figure 14 – System with New CIM_BootConfigSetting	50
16		

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

List of Tables

Table 1 – Related Profiles	12
Table 2 – Structured Name Identifiers	23
Table 3 – CreateBootConfigSetting() Method: Return Code Values	28
Table 4 – CreateBootConfigSetting() Method: Parameters	28
Table 5 – ApplyBootConfigSetting() Method: Return Code Values	29
Table 6 – ApplyBootConfigSetting() Method: Parameters	29
Table 7 – ChangeBootOrder() Method: Return Code Values	30
Table 8 – ChangeBootOrder() Method: Parameters	30
Table 9 – Operations: CIM_BootService	31
Table 10 – Operations: CIM_BootConfigSetting	31
Table 11 – Operations: CIM_BootSettingData	33
Table 12 – Operations: CIM_BootSourceSetting	33
Table 13 – Operations: CIM_ConcreteComponent	33
Table 14 – Operations: CIM_ConcreteDependency	34
Table 15 – Operations: CIM_ElementCapabilities	34
Table 16 – Operations: CIM_ElementSettingData	34
Table 17 – Operations: CIM_BootServiceCapabilities	36
Table 18 – Operations: CIM_HostedService	36
Table 19 – Operations: CIM_LogicalIdentity	36
Table 20 – Operations: CIM_OrderedComponent	36
Table 21 – Operations: CIM_ServiceAffectsElement	38
Table 22 CIM Elements – Boot Control Profile	55
Table 23 – Class: CIM_RegisteredProfile	55
Table 24 – Class: CIM_BootService	56
Table 25 – Class: CIM_BootServiceCapabilities	56
Table 26 – Class: CIM_BootConfigSetting	57
Table 27 – Class: CIM_BootSettingData	57
Table 28 – Class: CIM_BootSourceSetting	57
Table 29 – Class: CIM_ConcreteComponent – Use 1	58
Table 30 – Class: CIM_ConcreteComponent – Use 2	58
Table 31 – Class: CIM_ConcreteComponent – Use 3	58
Table 32 – Class: CIM_ConcreteComponent – Use 4	59
Table 33 – Class: CIM_ConcreteDependency	59
Table 34 – Class: CIM_ElementCapabilities	60
Table 35 – Class: CIM_ElementSettingData	60
Table 36 – Class: CIM_HostedService	60
Table 37 – Class: CIM_LogicalIdentity	61
Table 38 – Class: CIM_OrderedComponent	61
Table 39 – Class: CIM_ServiceAffectsElement	61

1

Foreword

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

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

1

Introduction

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

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

8

Boot Control Profile

9 1 Scope

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

12 2 Normative References

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

16 2.1 Approved References

17 DMTF [DSP0200](#), *CIM Operations over HTTP 1.2.0*

18 DMTF [DSP0004](#), *CIM Infrastructure Specification 2.5.0 (Preliminary)*

19 DMTF [DSP1000](#), *Management Profile Specification Template*

20 DMTF [DSP1001](#), *Management Profile Specification Usage Guide*

21 DMTF [DSP1033](#), *Profile Registration Profile*

22 2.2 Other References

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

25 OMG Unified Modeling Language (OMG UML) Superstructure v2.1.2,
26 <http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF/>

27 *BIOS Boot Specification v1.01* (January 11, 1996), <http://www.phoenix.com/NR/rdonlyres/56E38DE2-3E6F-4743-835F-B4A53726ABED/0/specsbbs101.pdf>

29 3 Terms and Definitions

30 3.1

31 can

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

33 3.2

34 cannot

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

36 3.3

37 conditional

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

- 40 **3.4**
41 **mandatory**
42 used to indicate requirements strictly to be followed, in order to conform to the document and from which
43 no deviation is permitted
- 44 **3.5**
45 **may**
46 used to indicate a course of action permissible within the limits of the document
- 47 **3.6**
48 **need not**
49 used to indicate a course of action permissible within the limits of the document
- 50 **3.7**
51 **optional**
52 used to indicate a course of action permissible within the limits of the document
- 53 **3.8**
54 **referencing profile**
55 indicates a profile that owns the definition of a class used, but not defined, in this document and can be
56 included in the "Referenced Profiles" table
- 57 **3.9**
58 **shall**
59 used to indicate requirements strictly to be followed, in order to conform to the document and from which
60 no deviation is permitted
- 61 **3.10**
62 **shall not**
63 used to indicate requirements strictly to be followed, in order to conform to the document and from which
64 no deviation is permitted
- 65 **3.11**
66 **should**
67 used to indicate that among several possibilities, one is recommended as particularly suitable, without
68 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 69 **3.12**
70 **should not**
71 used to indicate that a certain possibility or course of action is deprecated but not prohibited
- 72 **3.13**
73 **unspecified**
74 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 75 **3.14**
76 **Boot Configurable System**
77 an instance of CIM_ComputerSystem whose boot configurations are being managed
- 78 **3.15**
79 **Boot Configuration**
80 a collection of settings that are applied to a boot configurable system during the boot process

81 3.16**82 Boot Configuration Representation**

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

86 3.17**87 Current Boot Configuration**

88 the instance of CIM_BootConfigSetting that was used the last time the managed system was successfully
89 booted

90 3.18**91 Default Boot Configuration**

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

94 3.19**95 Next Boot Configuration**

96 the instance of CIM_BootConfigSetting that will be used during the next boot of the Boot Configurable
97 System

98 3.20**99 Next Single Use Boot Configuration**

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

102 3.21**103 Not Next Boot Configuration**

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

105 3.22**106 Template Boot Configuration**

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

109 4 Symbols and Abbreviated Terms**110 4.1****111 BCV**

112 Boot Control Vector. See the BIOS Boot Specification for additional information.

113 4.2**114 IPL**

115 Initial Program Load. See the BIOS Boot Specification for additional information.

116 4.3**117 PXE**

118 Preboot Execution Environment. See the BIOS Boot Specification for additional information.

119 5 Synopsis

120 **Profile Name:** *Boot Control*

Boot Control Profile

121 **Version:** 1.0.0

122 **Organization:** DMTF

123 **CIM Schema Version:** 2.19.1

124 **Central Class:** CIM_BootService

125 **Scoping Class:** CIM_ComputerSystem

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

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

130 CIM_BootService shall be the Central Class of this profile. The instance of CIM_BootService shall be the
131 Central Instance of this profile.

132 CIM_ComputerSystem shall be the Scoping Class of this profile. The instance of CIM_ComputerSystem
133 with which the Central Instance is associated through an instance of CIM_HostedService shall be the
134 Scoping Instance of this profile.

135 **Table 1 – Related Profiles**

Profile Name	Organization	Version	Relationship
<i>Profile Registration</i>	DMTF	1.0.0	Mandatory

136 **6 Description**

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

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

- 140 • A computer system can have one or more boot configurations.
- 141 • A computer system can contain a boot configuration that is used during each boot.
- 142 • A computer system can contain a single-use boot configuration that is used only during the next boot
143 and then not used again.
- 144 • A computer system can contain a current boot configuration that represents the boot configuration
145 successfully used in the last boot.
- 146 • A computer system can contain a default boot configuration that is set by the computer system
147 manufacturer or a client.
- 148 • A computer system can create new boot configurations.
- 149 • A computer system can apply a boot configuration to an active or inactive computer system.

150 A typical boot configuration could have the following characteristics:

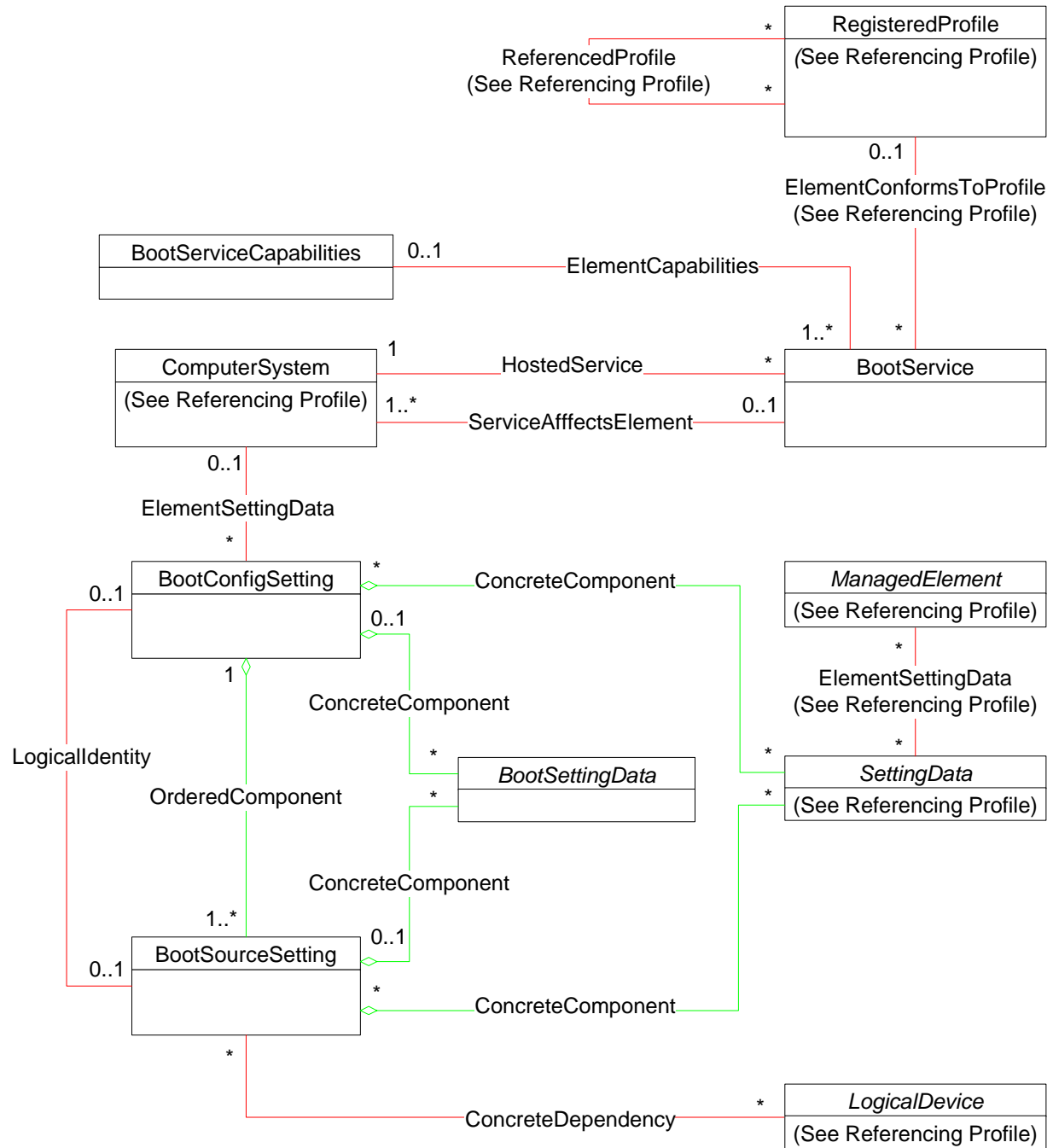
- 151 • A boot configuration can contain a boot order that specifies the order in which boot devices are
152 accessed. The boot devices include, but are not limited to, floppy device, CD device, hard disks,
153 network controllers (using the PXE protocol), and BCV devices composed of additional boot sources.
- 154 • A boot configuration can contain data that can affect various computer system components during the
155 boot process.

- 156 • A boot configuration can contain data that can be passed to the booted image (for example, second-
157 stage boot loader or bootblock) in the form of a boot string.
 - 158 • Boot devices can be local to the computer system or remote to the computer system.
- 159 A boot configuration can be applied when the computer system starts the boot process. The boot process
160 can be started automatically as part of the enablement of the computer system or by a specific request
161 when the computer system is enabled but not booted.

162 **6.1 Class Diagram**

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

165 In Figure 1, CIM_ManagedElement, CIM_LogicalDevice, CIM_SettingData, and CIM_BootSettingData
 166 are abstract classes.



167

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

169 A computer system can have multiple boot configurations. Each boot configuration is modeled by a Boot
 170 Configuration Representation, which consists of an instance of CIM_BootConfigSetting class and,

171 optionally, all of the instances of classes CIM_BootSourceSetting, CIM_BootSettingData and
 172 CIM_SettingData that the instance of CIM_BootConfigSetting aggregates

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

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

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

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

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

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

191 **7 Implementation**

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

195 **7.1 CIM_BootService**

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

197 **7.1.1 CIM_BootService.ElementName**

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

199 **7.1.2 Modifying ElementName Is Supported**

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

202 Conditional Requirement:

- 203 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
 204 through an instance of CIM_ElementCapabilities.
- 205 2) The CIM_BootServiceCapabilities.ElementNameEditSupport property has the value of TRUE.
- 206 3) The CIM_BootServiceCapabilities.MaxElementNameLen property has a non-zero value

207 The implementation shall allow the CIM_BootService.ModifyInstance intrinsic operation to change the
 208 value of the ElementName property. The ModifyInstance operation shall enforce the length restriction
 209 specified in the MaxElementNameLen property.

210

211

212 **7.1.3 Modifying ElementName Is Not Supported**

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

215 Conditional Requirement 1:

216 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
217 through an instance of CIM_ElementCapabilities.

218 2) The CIM_BootServiceCapabilities.ElementNameEditSupport property has the value of FALSE.

219 Conditional Requirement 2:

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

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

224 **7.2 CIM_ComputerSystem**

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

228 One Scoping Instance shall exist. Clause 5 describes the process for determining the Scoping Instance
229 from the Central Instance.

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

233 **7.3 Representing Boot Service Capabilities**

234 Subclause 7.3 describes optional behavior.

235 An instance of CIM_BootServiceCapabilities may exist, which represents the capabilities of the boot
236 service.

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

239 **7.3.1 Representing Implementation Specific Boot Service Capabilities**

240 Subclause 7.3.1 describes optional behavior.

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

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

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

250 **7.4 Boot Configurations**

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

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

255 **7.4.1 CIM_ElementSettingData**

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

259 When the CIM_ElementSettingData association is used in this manner, its ManagedElement property
260 shall reference the CIM_ComputerSystem instance and its SettingData property shall reference the
261 CIM_BootConfigSetting instance.

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

264 **7.4.2 Default Boot Configuration**

265 Subclause 7.4.2 describes optional behavior.

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

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

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

275 **7.4.3 Current Boot Configuration**

276 Subclause 7.4.3 describes optional behavior.

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

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

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

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

287 **7.4.4 Next Boot Configuration**

288 Subclause 7.4.4 describes optional behavior.

Boot Control Profile

289 Note: Successful execution of the ApplyBootConfigSetting() method can independently apply a boot
290 configuration to a Boot Configurable System regardless of the Next Boot Configuration. The requirements
291 in this subclause shall not apply when a Boot Configurable System is booted using the
292 ApplyBootConfigSetting() method.

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

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

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

299 **7.4.5 Next Single Use Boot Configuration**

300 Subclause 7.4.5 describes optional behavior.

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

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

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

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

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

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

316 **7.4.6 Not Next Boot Configuration**

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

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

321 **7.5 Applying the Boot Configuration**

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

324 Note: Successful execution of the ApplyBootConfigSetting() method can independently apply a boot
325 configuration to a Boot Configurable System regardless of the Next Boot Configuration. The requirements
326 in subclause 7.4.4 shall not apply when a Boot Configurable System is booted using the
327 ApplyBootConfigSetting() method.

328 **7.5.1 Apply Boot Configuration Is Supported**

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

331 Conditional Requirement:

- 332 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
333 through an instance of CIM_ElementCapabilities.
- 334 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 3
335 (Applies Boot Configuration).

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

337 **7.5.2 Apply Boot Configuration Is Not Supported**

338 Subclause 0 describes conditional behavior. Subclause 0 describes the CIM elements and behaviors that
339 shall be implemented when either of the following conditions are met.

340 Conditional Requirement 1:

- 341 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
342 through an instance of CIM_ElementCapabilities.
- 343 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array does not contain a
344 value of 3 (Applies Boot Configuration).

345 Conditional Requirement 2:

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

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

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

351 **7.6 Creating a Boot Configuration**

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

354 **7.6.1 Creating Boot Configuration Is Supported**

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

357 Conditional Requirement:

- 358 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
359 through an instance of CIM_ElementCapabilities.
- 360 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 2
361 (Creates Boot Configuration).

362 The implementation shall support the CreateBootConfigSetting() method.

363 **7.6.2 Creating Boot Configuration Is Not Supported**

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

Boot Control Profile

366 Conditional Requirement 1:

367 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
368 through an instance of CIM_ElementCapabilities.

369 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array does not contain a
370 value of 2 (Creates Boot Configuration).

371 Conditional Requirement 2:

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

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

376 **7.7 Deleting a Boot Configuration**

377 Subclause 7.7 describes conditional behavior.

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

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

383 **7.8 Identifying Boot Sources**

384 Subclause 7.8 describes optional behavior.

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

387 One or more instances of CIM_BootSourceSetting shall be associated to an instance of
388 CIM_BootConfigSetting.

389 The CIM_BootSourceSetting class has three boot string properties: BootString, BIOSBootString and
390 StructuredBootString. The BootString and BIOSBootString properties may be supported. The
391 StructuredBootString property should be supported.

392 **7.8.1 CIM_BootServiceCapabilities**

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

395 **7.8.1.1 CIM_BootServiceCapabilities.BootStringsSupported**

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

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

401 **7.8.2 CIM_BootSourceSetting.ElementName Property**

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

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

405 When the CIM_BootSourceSetting.BIOSBootString property is not null, the ElementName property shall
406 match the BIOSBootString property.

407 **7.8.3 CIM_BootSourceSetting.BootString Property**

408 An implementation may support the CIM_BootSourceSetting.BootString property.

409 **7.8.3.1 CIM_BootSourceSetting.BootString Property is Supported**

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

412 Conditional Requirement:

- 413 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
414 through an instance of CIM_ElementCapabilities.
- 415 2) The CIM_BootServiceCapabilities.BootStringsSupported property array contains a value of 2
416 (BootString).

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

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

422 **7.8.3.2 CIM_BootSourceSetting.BootString Property is Not Supported**

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

425 Conditional Requirement 1:

- 426 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
427 through an instance of CIM_ElementCapabilities.
- 428 2) The CIM_BootServiceCapabilities.BootStringsSupported property array does not contain a
429 value of 2 (BootString).

430 Conditional Requirement 2:

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

433 The CIM_BootSourceSetting.BootString property may be NULL.

434 **7.8.4 CIM_BootSourceSetting.BIOSBootString Property**

435 An implementation may support the CIM_BootSourceSetting.BIOSBootString property.

436 **7.8.4.1 CIM_BootSourceSetting.BIOSBootString Property is Supported**

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

439 Conditional Requirement:

- 440 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
441 through an instance of CIM_ElementCapabilities.
- 442 2) The CIM_BootServiceCapabilities.BootStringsSupported property array contains a value of 3
443 (BIOSBootString).

Boot Control Profile

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

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

449 **7.8.4.2 CIM_BootSourceSetting.BIOSBootString Property is Not Supported**

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

452 Conditional Requirement 1:

453 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
454 through an instance of CIM_ElementCapabilities.

455 2) The CIM_BootServiceCapabilities.BootStringsSupported property array does not contain a
456 value of 3 (BIOSBootString).

457 Conditional Requirement 2:

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

460 The CIM_BootSourceSetting.BIOSBootString property may be NULL.

461 **7.8.5 CIM_BootSourceSetting.StructuredBootString Property**

462 An implementation should support the CIM_BootSourceSetting.StructuredBootString property.

463 **7.8.5.1 CIM_BootSourceSetting.StructuredBootString Property is Supported**

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

466 Conditional Requirement:

467 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
468 through an instance of CIM_ElementCapabilities.

469 2) The CIM_BootServiceCapabilities.BootStringsSupported property array contains a value of 4
470 (StructuredBootString).

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

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

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

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

480 **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

481 **7.8.5.2 CIM_BootSourceSetting.StructuredBootString Property is Not Supported**

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

484 Conditional Requirement 1:

- 485 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
 486 through an instance of CIM_ElementCapabilities.
- 487 2) The CIM_BootServiceCapabilities.BootStringsSupported property array does not contain a
 488 value of 4 (StructuredBootString).

489 Conditional Requirement 2:

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

492 The CIM_BootSourceSetting.StructuredBootString property may be NULL.

493 **7.8.6 CIM_ConcreteDependency Association**

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

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

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

501 **7.9 Changing the Boot Order**

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

505 **7.9.1 Changing Boot Order Is Supported**

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

508 Conditional Requirement 1:

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

513 Conditional Requirement 2:

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

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

518 **7.9.2 Changing Boot Order Is Not Supported**

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

521 Conditional Requirement:

- 522 1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance
523 through an instance of CIM_ElementCapabilities.
- 524 2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 6
525 (Change Boot Order Not Supported).

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

527 **7.10 Representing a Set of Aggregated Boot Sources**

528 Subclause 7.9 describes optional behavior.

529 An instance of CIM_BootSourceSetting may represent an aggregated boot source. An example of an
530 aggregated boot source is a BCV.

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

533 **7.10.1 Aggregated Boot Sources**

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

535 **7.10.2 Aggregated Boot Configuration**

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

537 The ElementName property for the instance of CIM_BootConfigSetting representing the set of
538 aggregated boot sources shall match the value of the ElementName property of the instance of
539 CIM_BootSourceSetting that represents the aggregated boot source.

540 **7.10.3 Logical Identity Relationship**

541 An instance of CIM_LogicalIdentity shall associate the instance of CIM_BootSourceSetting with the
542 instance of CIM_BootConfigSetting.

543 **7.10.3.1 CIM_LogicalIdentity.SystemElement**

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

546 **7.10.3.2 CIM_LogicalIdentity.SameElement**

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

549 **7.11 Boot Order During the Boot Process**

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

552 **7.11.1 CIM_OrderedComponent Association**

553 The CIM_OrderedComponent association class shall be used to associate an instance of
554 CIM_BootConfigSetting to each instance of CIM_BootSourceSetting representing one of the boot sources
555 in the boot configuration.

556 When the association is used in this manner, its GroupComponent property shall reference the
557 CIM_BootConfigSetting instance and its PartComponent property shall reference the
558 CIM_BootSourceSetting instance.

559 **7.11.1.1 CIM_OrderedComponent.AssignedSequence Property**

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

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

566 The boot order shall be interpreted as follows:

- 567 • The AssignedSequence properties are compared across instances of CIM_OrderedComponent that
568 reference the same CIM_BootConfigSetting.
- 569 • A CIM_BootSourceSetting whose associated CIM_OrderedComponent.AssignedSequence property
570 is equal to zero shall be ignored and not considered part of the boot order.
- 571 • The boot order shall proceed from the lowest to the highest non-zero integer value of the
572 AssignedSequence properties.

573 **7.11.2 CIM_BootSourceSetting.FailThroughSupported**

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

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

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

581 **7.12 Settings to Apply During the Boot Process**

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

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

586 **7.12.1 Settings that Apply to a Managed Element**

587 An instance of a concrete subclass of CIM_SettingData represents a setting that is applied to a managed
588 element during the boot process. The instance shall be associated to either an instance of
589 CIM_BootConfigSetting or an instance of CIM_BootSourceSetting through an instance of
590 CIM_ConcreteComponent.

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

594 When the CIM_ConcreteComponent association is used in this manner, its GroupComponent property
595 shall reference the CIM_BootConfigSetting instance and its PartComponent property shall reference the
596 CIM_SettingData instance.

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

600 When the CIM_ConcreteComponent association is used in this manner, its GroupComponent property
601 shall reference the CIM_BootSourceSetting instance and its PartComponent property shall reference the
602 CIM_SettingData instance.

603 **7.12.2 Settings that Apply to the Boot Process**

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

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

609 When the CIM_ConcreteComponent association is used in this manner, its GroupComponent property
610 shall reference the CIM_BootConfigSetting or CIM_BootSourceSetting instance and its PartComponent
611 property shall reference the CIM_BootSettingData instance.

612 When an instance of a concrete subclass of CIM_SettingData is instantiated, then it shall be associated
613 with an instance of CIM_BootConfigSetting or CIM_BootSourceSetting using an instance of
614 CIM_ConcreteComponent.

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

618 8 Methods

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

621 8.1 CIM_BootService.CreateBootConfigSetting()

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

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

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

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

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

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

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

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

651 • A new instance of CIM_BootConfigSetting shall exist and be referenced by the output NewBootConfig
652 parameter. The new CIM_BootConfigSetting.InstanceID property shall be set to a unique value.

653 • A new instance of CIM_ElementSettingData shall exist that associates the new
654 CIM_BootConfigSetting to the instance of the Boot Configurable System, which is specified by the
655 ScopingComputerSystem parameter when it is non-NULL or implied by the StartingBootConfig
656 parameter when the ScopingComputerSystem parameter is NULL.

657 • The CIM_ElementSettingData.IsDefault property shall be set to 2 (Is Not Default). The
658 CIM_ElementSettingData.IsCurrent property shall be set to 2 (Is Not Current). The
659 CIM_ElementSettingData.IsNext property shall be set to 2 (Is Not Next).

660 • New instances of CIM_BootSourceSetting shall exist, along with instances of
661 CIM_OrderedComponent, when they are present in the boot configuration represented by the

Boot Control Profile

662 Template Boot Configuration. The new instances shall be duplicates of those found in the boot
663 configuration represented by the Template Boot Configuration, except for the key property value.

664 • New instances of CIM_BootSettingData shall exist when they are present in the boot configuration
665 represented by the Template Boot Configuration. The new instances shall be duplicates of those
666 found in the boot configuration represented by the Template Boot Configuration, except for the key
667 property value.

668 • New instances of CIM_ConcreteComponent shall exist when they are present in the boot
669 configuration represented by the Template Boot Configuration.

670 • New instances of CIM_ConcreteDependency shall exist when they are present in the boot
671 configuration represented by the Template Boot Configuration.

672 • CIM elements that are defined in a Referencing Profile are not copied.

673 The return code values and parameters for the CreateBootConfigSetting() method are specified in Table
674 3 and Table 4.

675 No standard messages are defined.

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

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

677 **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

678 **8.2 CIM_BootService.ApplyBootConfigSetting()**

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

681 Note: Successful execution of the ApplyBootConfigSetting() method can independently apply a boot
682 configuration to a Boot Configurable System regardless of the Next Boot Configuration. The requirements
683 in subclause 7.4.4 shall not apply when a Boot Configurable System is booted using the
684 ApplyBootConfigSetting() method.

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

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

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

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

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

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

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

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

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

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

712 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

8.3 CIM_BootConfigSetting.ChangeBootOrder()

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

Boot Control Profile

- 718 When the ChangeBootOrder() method is not supported, a return value or an exception shall be returned.
- 719 The input parameter, Source, is an ordered array of references to CIM_BootSourceSetting instances that
720 defines the new sequence of the CIM_BootSourceSetting instances associated to the instance of
721 CIM_BootConfigSetting. Each CIM_BootSourceSetting instance in the array shall already be associated
722 with this CIM_BootConfigSetting instance through an instance of CIM_OrderedComponent. This
723 parameter is required.
- 724 When the Source parameter is NULL, a return value of 2 (Error Occurred) may be returned. When a
725 return value or an exception shall be returned.
- 726 When any of the CIM_BootSourceSetting instance in the Source array are not associated to the instance
727 of CIM_BootConfigSetting, the implementation may return a value of 2 (Error Occurred). When a return
728 value or an exception shall be returned.
- 729 Upon successful completion of this method, the value of the AssignedSequence property on each
730 instance of CIM_OrderedComponent shall be updated such that the values are monotonically increasing
731 in correlation with the position of the referenced CIM_BootSourceSetting instance in the Source input
732 parameter. That is, the first position in the array shall have the lowest non-zero value for
733 AssignedSequence. The second position will have the second lowest value, and so on.
- 734 Upon successful completion of this method, the value of the AssignedSequence property on each
735 instance of CIM_OrderedComponent, that associates the target CIM_BootConfigSetting instance to a
736 CIM_BootSourceSetting instance that is not present in the input array, shall be assigned a value of 0.
- 737 The return code values and parameters for the ChangeBootOrder() method are specified in Table 7 and
738 Table 8, respectively.
- 739 No standard messages are defined.

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

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

741 **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.

742 **8.4 Profile Conventions for Operations**

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

746 **8.5 CIM_BootService**

747 Compliant implementations of this profile shall support the operations listed in Table 9 for
 748 CIM_BootService. Each operation shall be supported as defined in DSP0200.

749 **Table 9 – Operations: CIM_BootService**

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

750 **8.5.1 CIM_BootService – ModifyInstance Operation**

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

753 When an instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance and
 754 the CIM_BootServiceCapabilities.ElementNameEditSupported property has a value of TRUE, the
 755 implementation shall allow the ModifyInstance operation to change the value of the ElementName
 756 property of the CIM_BootService instance. The ModifyInstance operation shall enforce the length
 757 restriction specified in the MaxElementNameLen property of the CIM_BootServiceCapabilities instance.

758 When no instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance, or
 759 the ElementNameEditSupported property of the CIM_BootServiceCapabilities has a value of FALSE, the
 760 implementation shall not allow the ModifyInstance operation to change the value of the ElementName
 761 property of the CIM_BootService instance.

762 **8.6 CIM_BootConfigSetting**

763 Compliant implementations of this profile shall support the operations listed in Table 10 for the
 764 CIM_BootConfigSetting class. Each operation shall be supported as defined in DSP0200.

765 **Table 10 – Operations: CIM_BootConfigSetting**

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

766 **8.6.1 CIM_BootConfigSetting – DeleteInstance**

767 Subclause 8.6.1 describes conditional behavior.

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

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

- 773 • The target instance of CIM_BootConfigSetting shall no longer exist.
- 774 • The instance of CIM_ElementSettingData that associated the target CIM_BootConfigSetting to the
775 instance of CIM_ComputerSystem shall no longer exist.
- 776 • The instances of CIM_ConcreteComponent, which associate the target instance of
777 CIM_BootConfigSetting to instances of a concrete subclass of CIM_SettingData, shall no longer exist.
- 778 • The instances of CIM_ConcreteComponent, which associate the target instance of
779 CIM_BootConfigSetting to instances of a concrete subclass of CIM_BootSettingData, shall no longer
780 exist. The instances of the associated concrete subclass of CIM_BootSettingData shall no longer
781 exist.
- 782 • The instances of CIM_OrderedComponent, which associate the target instance of
783 CIM_BootConfigSetting to instances of CIM_BootSourceSetting, shall no longer exist. The instances
784 of the associated CIM_BootSourceSetting shall no longer exist.
- 785 • The instances of CIM_ConcreteComponent, which associate instances of a concrete subclass of
786 CIM_SettingData to instances of CIM_BootSourceSetting, which in turn are associated to the target
787 instance of CIM_BootConfigSetting, shall no longer exist.
- 788 • The instances of CIM_ConcreteComponent, which associate instances of a concrete subclass of
789 CIM_BootSettingData to instances of CIM_BootSourceSetting, which in turn are associated to the
790 target instance of CIM_BootConfigSetting, shall no longer exist. The instances of the associated
791 concrete subclass of CIM_BootSettingData shall no longer exist.
- 792 • The instances of CIM_ConcreteDependency, which associate instances of a concrete subclass of
793 CIM_LogicalDevice to instances of CIM_BootSourceSetting, shall no longer exist.
- 794 • The instance of CIM_LogicalIdentity, which associates a deleted instance of CIM_BootSourceSetting
795 to an instance of CIM_BootConfigSetting, shall no longer exist. The associated instance of
796 CIM_BootConfigSetting shall no longer exist. The requirements in this subclause shall be applied
797 recursively to the deleted CIM_BootConfigSetting instance.

798 **8.7 CIM_BootSettingData**

799 Compliant implementations of this profile shall support the operations listed in Table 11 for the
800 CIM_BootSettingData class. Each operation shall be supported as defined in DSP0200.

801 **Table 11 – Operations: CIM_BootSettingData**

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

802 **8.8 CIM_BootSourceSetting**

803 Compliant implementations of this profile shall support the operations listed in Table 12 for the
804 CIM_BootSourceSetting class. Each operation shall be supported as defined in DSP0200.

805 **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

806 **8.9 CIM_ConcreteComponent**

807 Compliant implementations of this profile shall support the operations listed in Table 13 for the
808 CIM_ConcreteComponent class. Each operation shall be supported as defined in DSP0200.

809 **Table 13 – Operations: CIM_ConcreteComponent**

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

810 **8.10 CIM_ConcreteDependency**

811 Compliant implementations of this profile shall support the operations listed in Table 14 for the
812 CIM_ConcreteDependency class. Each operation shall be supported as defined in DSP0200.

813 **Table 14 – Operations: CIM_ConcreteDependency**

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

814 **8.11 CIM_ElementCapabilities**

815 Compliant implementations of this profile shall support the operations listed in Table 15 for the
 816 CIM_ElementCapabilities class. Each operation shall be supported as defined in DSP0200.

817 **Table 15 – Operations: CIM_ElementCapabilities**

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

818 **8.12 CIM_ElementSettingData**

819 Compliant implementations of this profile shall support the operations listed in Table 16 for the
 820 CIM_ElementSettingData class. Each operation shall be supported as defined in DSP0200.

821 **Table 16 – Operations: CIM_ElementSettingData**

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

822 **8.12.1 CIM_ElementSettingData – ModifyInstance Operation**

823 The behavior of the ModifyInstance operation varies depending on the property of the association
 824 modified.

825 **8.12.1.1 CIM_ElementSettingData.IsDefault Property**

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

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

- 832 • Search for an instance of CIM_ElementSettingData that associates an instance of
 833 CIM_BootConfigSetting with the instance of CIM_ComputerSystem, which is referenced by the target
 834 instance of CIM_ElementSettingData where the IsDefault property has a value of 1 (Is Default).

- 835 • If such an instance of CIM_ElementSettingData is found, the ModifyInstance operation shall set the
836 value of the IsDefault property to 2 (Is Not Default).
- 837 • For the target instance of CIM_ElementSettingData, when the IsDefault property already has a value
838 of 1 (Is Default), the ModifyInstance operation shall complete successfully.
- 839 • For the target instance of CIM_ElementSettingData, set the value of the IsDefault property to 1 (Is
840 Default).

841 **8.12.1.2 CIM_ElementSettingData.IsNext Property**

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

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

- 848 • Search for an instance of CIM_ElementSettingData that associates an instance of
849 CIM_BootConfigSetting with the instance of CIM_ComputerSystem, which is referenced by the target
850 instance of CIM_ElementSettingData where the IsNext property has a value of 1 (Is Next).
- 851 • If such an instance of CIM_ElementSettingData is found, the ModifyInstance operation shall set the
852 value of the IsNext property to 2 (Is Not Next).
- 853 • For the target instance of CIM_ElementSettingData, when the IsNext property already has a value of
854 1 (Is Next), the ModifyInstance operation shall complete successfully.
- 855 • For the target instance of CIM_ElementSettingData, when the IsNext property has a value other than
856 1 (Is Next), set the value of the IsNext property to 1 (Is Next).

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

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

- 864 • For the target instance of CIM_ElementSettingData, when the IsNext property has a value of 1 (Is
865 Next), the ModifyInstance operation shall fail.
- 866 • Search for an instance of CIM_ElementSettingData that associates an instance of
867 CIM_BootConfigSetting with the instance of CIM_ComputerSystem referenced by the target instance
868 of CIM_ElementSettingData where the IsNext property has a value of 3 (Is Next For Single Use).
- 869 • If such an instance of CIM_ElementSettingData is found, the ModifyInstance operation shall set the
870 value of the IsNext property to 2 (Is Not Next).
- 871 • For the target instance of CIM_ElementSettingData, when the IsNext property already has a value of
872 3 (Is Next For Single Use), the ModifyInstance operation shall complete successfully.
- 873 • For the target instance of CIM_ElementSettingData, when the IsNext property has a value neither 1
874 (Is Next) nor 3 (Is Next For Single Use), set the value of the IsNext property to 3 (Is Next For Single
875 Use).

876 **8.12.1.3 CIM_ElementSettingData.IsCurrent Property**

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

878 **8.13 CIM_BootServiceCapabilities**

879 Compliant implementations of this profile shall support the operations listed in Table 17 for the
880 CIM_BootServiceCapabilities class. Each operation shall be supported as defined in DSP0200.

881 **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

882 **8.14 CIM_HostedService**

883 Compliant implementations of this profile shall support the operations listed in Table 18 for the
884 CIM_HostedService class. Each operation shall be supported as defined in DSP0200.

885 **Table 18 – Operations: CIM_HostedService**

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

886 **8.15 CIM_LogicalIdentity**

887 Compliant implementations of this profile shall support the operations listed in Table 19 for the
888 CIM_LogicalIdentity class. Each operation shall be supported as defined in DSP0200.

889 **Table 19 – Operations: CIM_LogicalIdentity**

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

890 **8.16 CIM_OrderedComponent**

891 Compliant implementations of this profile shall support the operations listed in Table 20 for the
892 CIM_OrderedComponent class. Each operation shall be supported as defined in DSP0200.

893 **Table 20 – Operations: CIM_OrderedComponent**

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

894 **8.17 CIM_ServiceAffectsElement**

895 Compliant implementations of this profile shall support the operations listed in Table 21 for the
896 CIM_ServiceAffectsElement class. Each operation shall be supported as defined in DSP0200.

897 **Table 21 – Operations: CIM_ServiceAffectsElement**

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

898

899 **9 Use Cases**

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

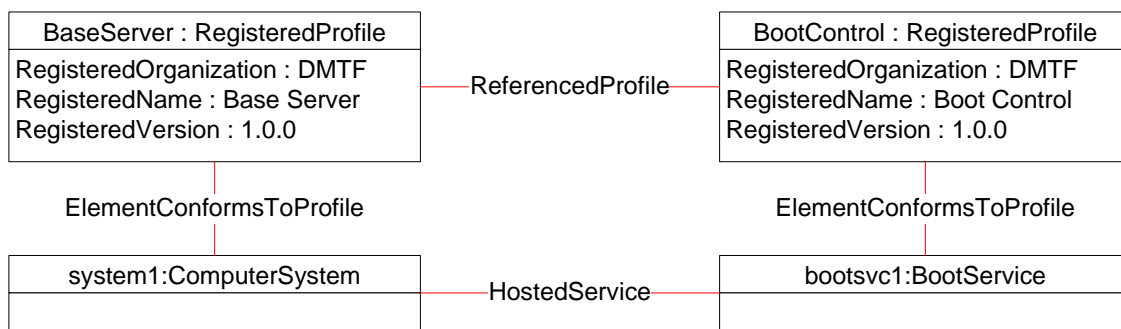
901 **9.1 Advertising the Profile Conformance**

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

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

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

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



914

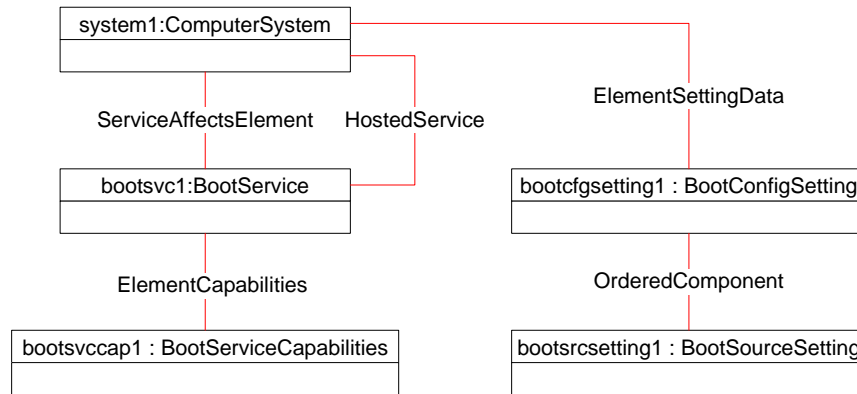
915

Figure 2 – Registered Profile

916 **9.2 Object Diagram for a Monolithic Server**

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

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



923

924

Figure 3 – Monolithic Server Object Diagram

925 **9.3 Object Diagram for a Monolithic Server with Service Processor**

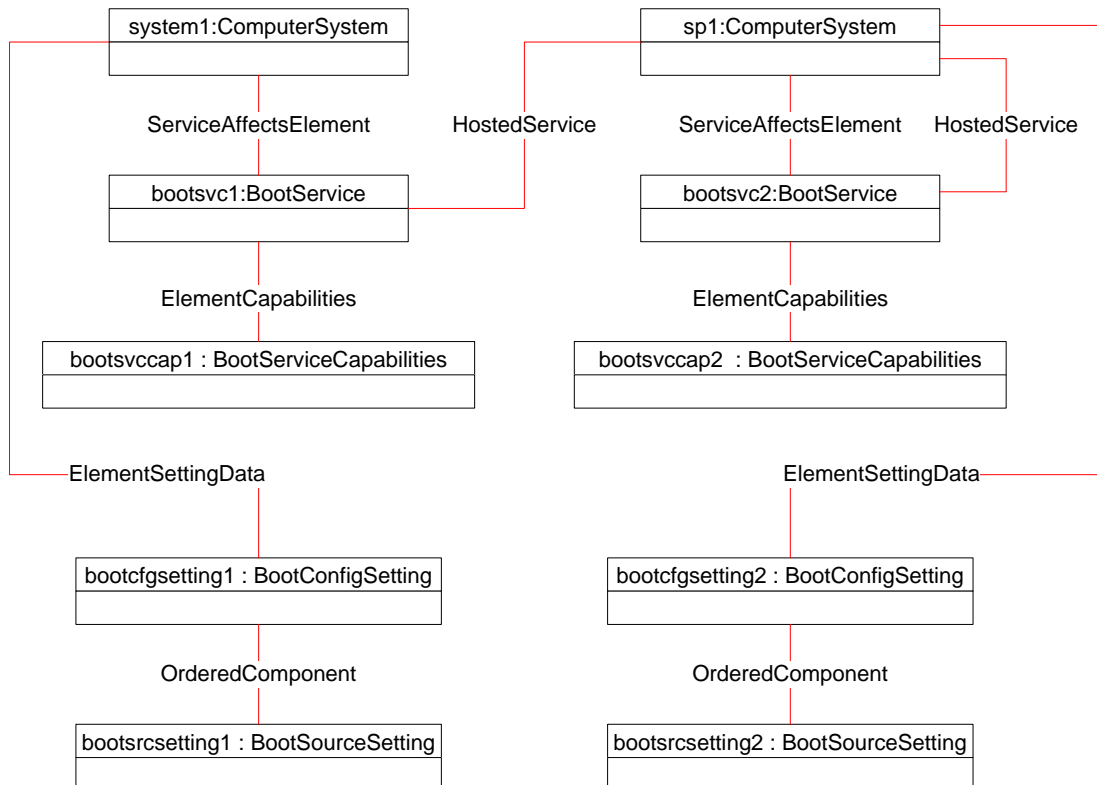
926 Figure 4 shows the CIM instances required to control the boot configuration for a single, monolithic
 927 server, *system1*, with an attached service processor, *sp1*. The boot service, *bootsvc1*, is hosted by the
 928 service processor and is responsible for managing the boot configuration, *bootcfgsetting1*,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

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

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

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

934



935

936

Figure 4 – Monolithic Server with Service Processor Object Diagram

937 **9.4 Object Diagram for a Modular System**

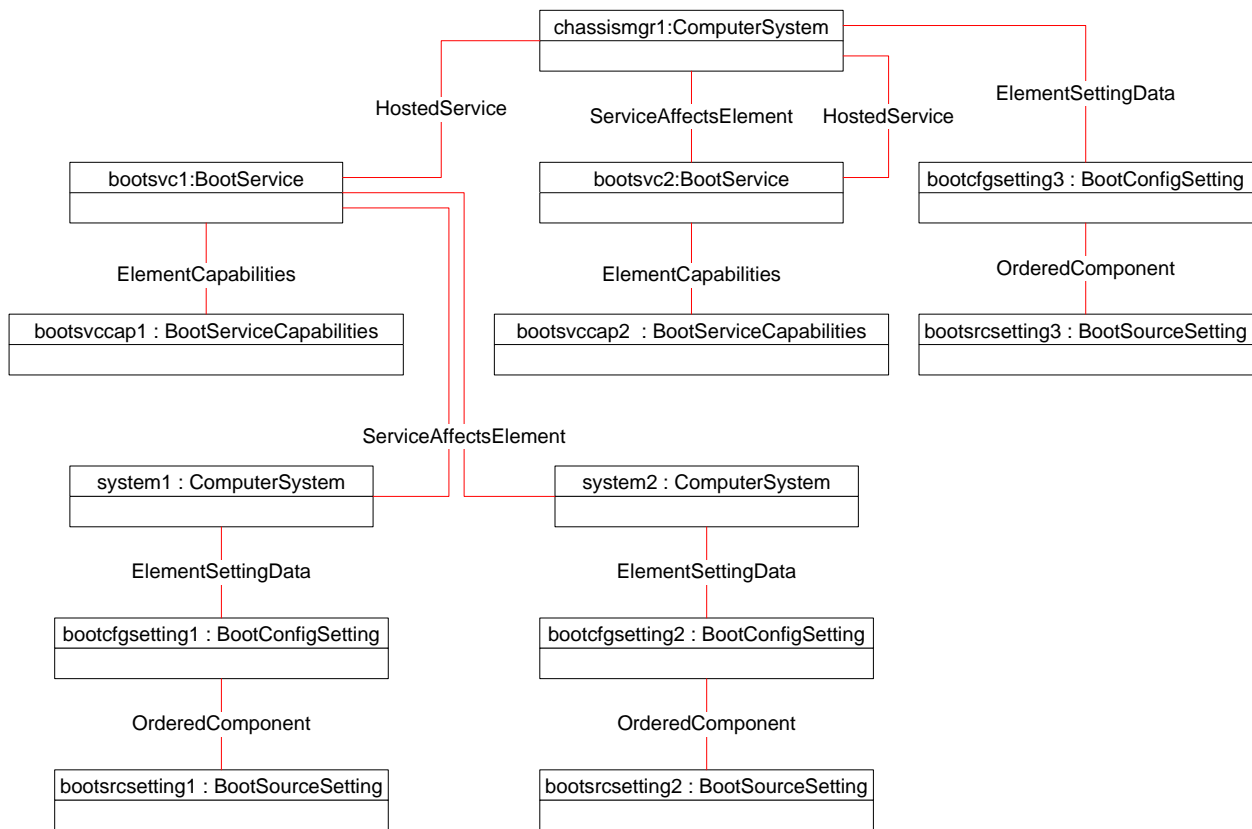
938 Figure 5 shows the CIM instances required to control the boot configuration for a modular system. The
 939 boot service, *bootsvc1*, is hosted by the chassis manager, *chassismgr1*, and is responsible for managing
 940 the boot configuration for two blade systems, *system1* and *system2*. *System1*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

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

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

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

947



948

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

950 **9.5 PXE Boot Source**

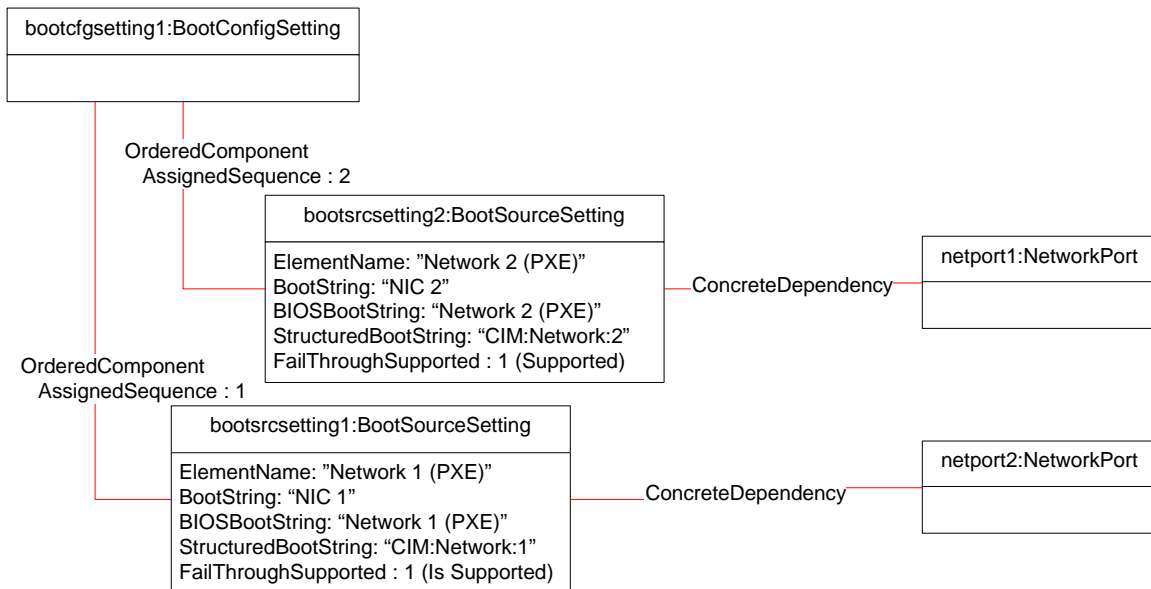
951 Figure 6 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has two boot sources
 952 associated to it, *bootsrcsetting1* and *bootsrcsetting2*, which are both network ports.

Boot Control Profile

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

956 The AssignedSequence property values on the OrderedComponent associations indicate that the boot
957 order is *bootsrcsetting1* followed by *bootsrcsetting2*.

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



960

961

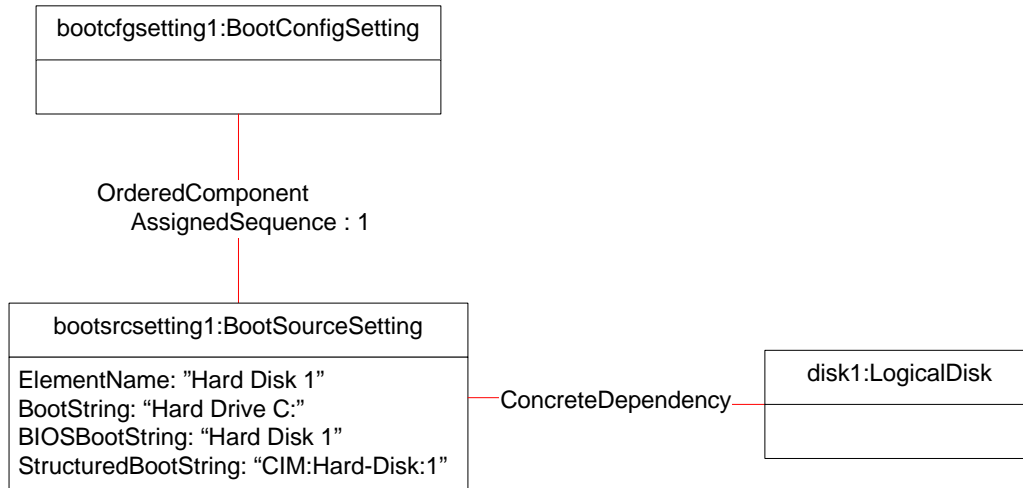
Figure 6 – PXE Boot Sources Object Diagram

962 9.6 Disk Boot Source

963 Figure 7 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has a single boot
964 source associated to it, *bootsrcsetting1*, which is a disk device.

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

969 Because there is only one boot source, the value of the `CIM_BootSourceSetting.FailThroughSupported` is
970 not meaningful.



971

972

Figure 7 – Booting from Disk

973 9.7 Local CDROM and Floppy Boot Sources

974 Figure 8 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has two boot sources
 975 associated to it, *bootsrcsetting1* and *bootsrcsetting2*. *Bootsrcsetting1* is a CD-ROM device;
 976 *bootsrcsetting2* is a floppy drive.

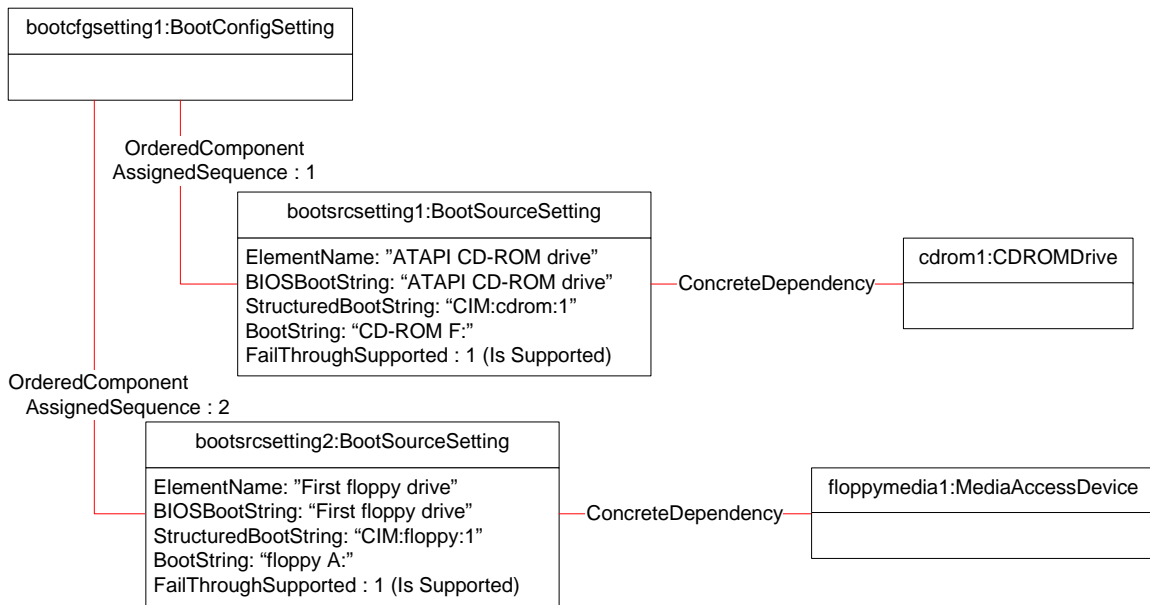
977 The AssignedSequence property of the CIM_OrderedComponent associations is set such that booting is
 978 attempted from the CD-ROM drive first and then the floppy drive.

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

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

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

Boot Control Profile



988

989

Figure 8 – Booting from CDROM and Floppy

989 9.8 Representing IPL and Boot Control Vector (BCV) Lists

991 Figure 9 shows an instance diagram for a boot configuration, *bootcfgsetting1*, composed of an IPL and
 992 BCV list of boot devices.

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

996 The `AssignedSequence` property of the `CIM_OrderedComponent` associations is set such that booting is
 997 attempted from the CD-ROM drive first and then the BCV device. Booting from the floppy device is not
 998 attempted because the `AssignedSequence` property is set to 0. The
 999 `CIM_BootConfigSetting.FailThroughSupported` property value of 1 (Is Supported) specifies that the boot
 1000 process should proceed to the second boot source if the first boot source fails or times out.

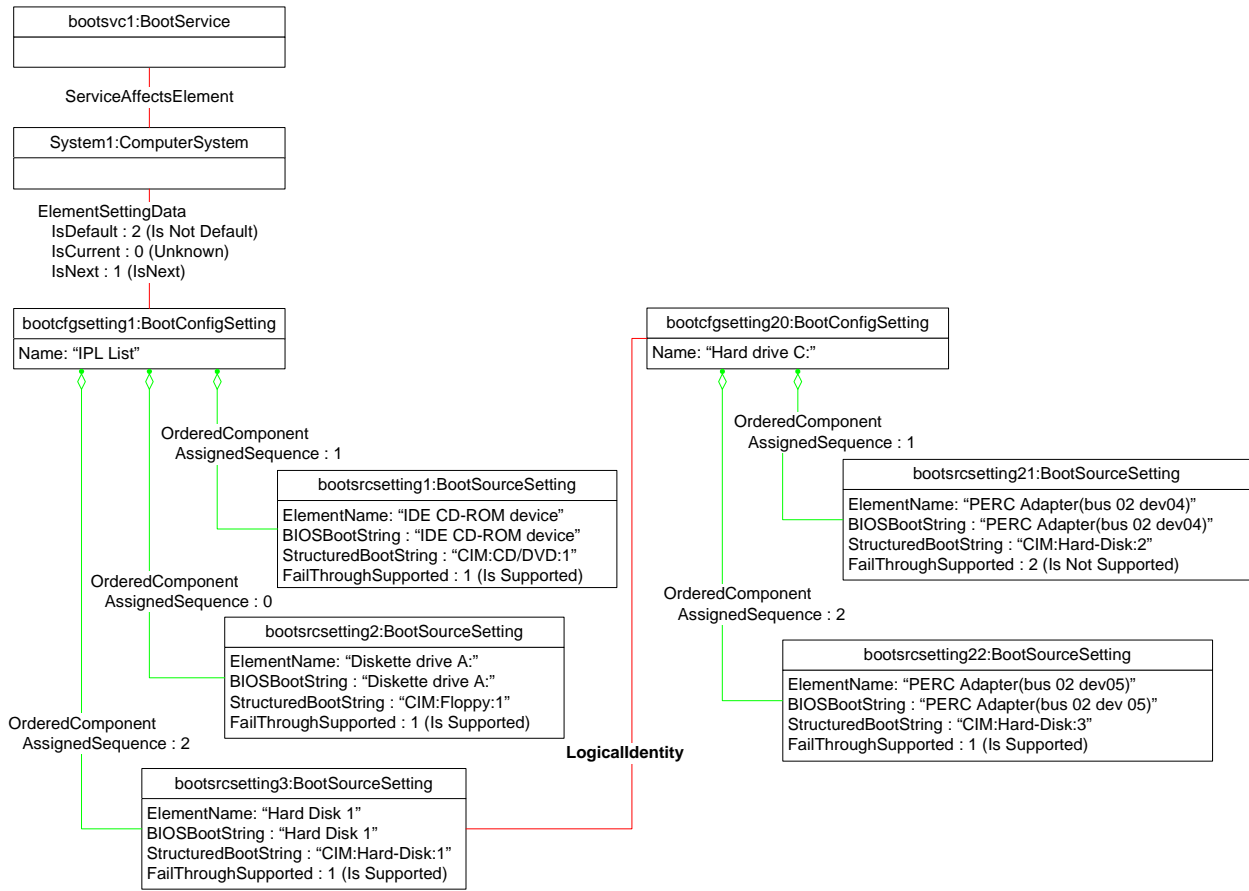
1001 In the diagram, the BCV device is a SCSI controller that may have multiple bootable SCSI devices
 1002 attached to it. This relationship is represented by an instance of `CIM_LogicalIdentity` between
 1003 *bootsrcsetting3* and an instance of `CIM_BootConfigSetting`, *bootcfgsetting20*.

1004 The boot configuration, *bootcfgsetting20*, has two boot sources associated to it, *bootsrcsetting21* and
 1005 *bootsrcsetting22*. Both boot sources are hard disk devices.

1006 The `AssignedSequence` property of the `CIM_OrderedComponent` associations is set such that booting is
 1007 attempted from *bootsrcsetting21* first and from *bootsrcsetting22*.

1008 On *bootsrcsetting21*, the `FailThroughSupported` property value of 2 (Is Not Supported) specifies that if the
 1009 *bootsrcsetting21*, "CIM:Hard-Disk:2", fails or times out, then the boot process should terminate the boot
 1010 order for *bootconfigsetting20*.

1011 In total, this use case describes a source boot order that proceeds from *bootsrcsetting1* to
 1012 *bootsrcsetting21*. *Bootsrcsetting2* will never be used because of its `AssignedSequence` value of 0 and
 1013 *bootsrcsetting22* will never be used because of the `FailThroughSupported` value on *bootsrcsetting21*.



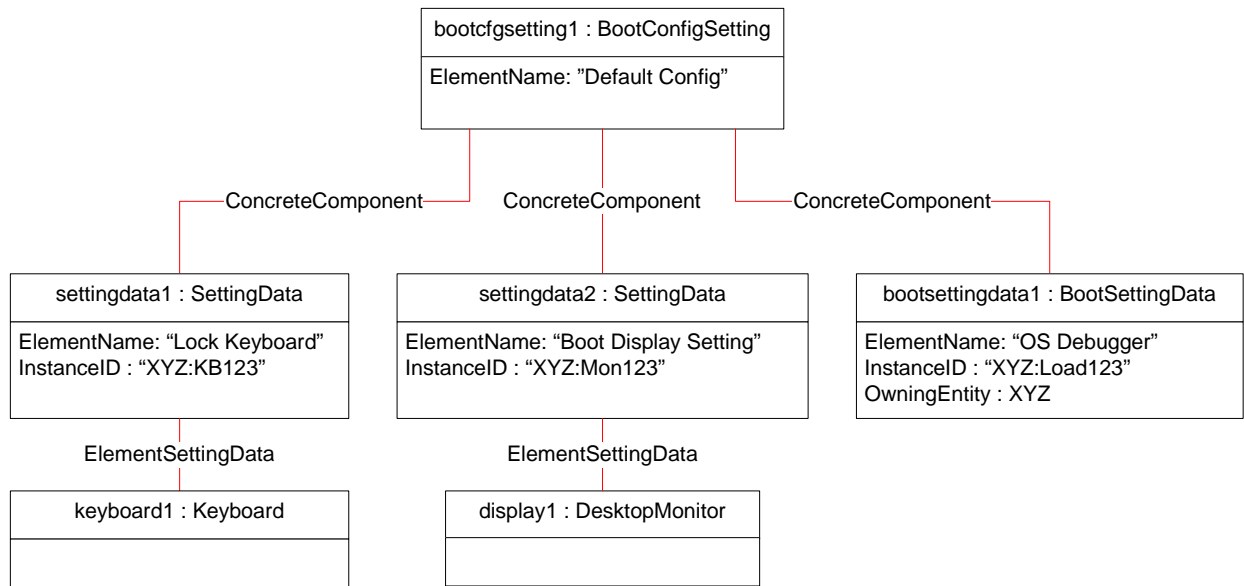
1014
1015

Figure 9 – Booting from IPL and BCV Devices

1016 **9.9 Representing Settings and Boot Settings**

1017 Figure 10 shows an instance diagram for a boot configuration, *bootcfgsetting1*, which has settings that
 1018 need to be applied to the computer system during the boot process. Two example concrete subclasses of
 1019 CIM_SettingData, *settingdata1* and *settingdata2*, apply to concrete subclasses of CIM_LogicalDevice,
 1020 *keyboard1* and *display1*. The instance of an example concrete subclass of CIM_BootSettingData is
 1021 *bootsettingdata1*.

1022 Being associated to the instance of CIM_BootConfigSetting, the settings apply to the entire boot process
 1023 that uses *bootcfgsetting1*. Note that any of these settings could be associated to an instance of
 1024 CIM_BootSourceSetting, which would reduce the scope of the settings to just the specified boot source.



1025
1026

Figure 10 – Setting Data and Boot Setting Data

1027 **9.10 Representing the Default Boot Configuration for a Computer System**

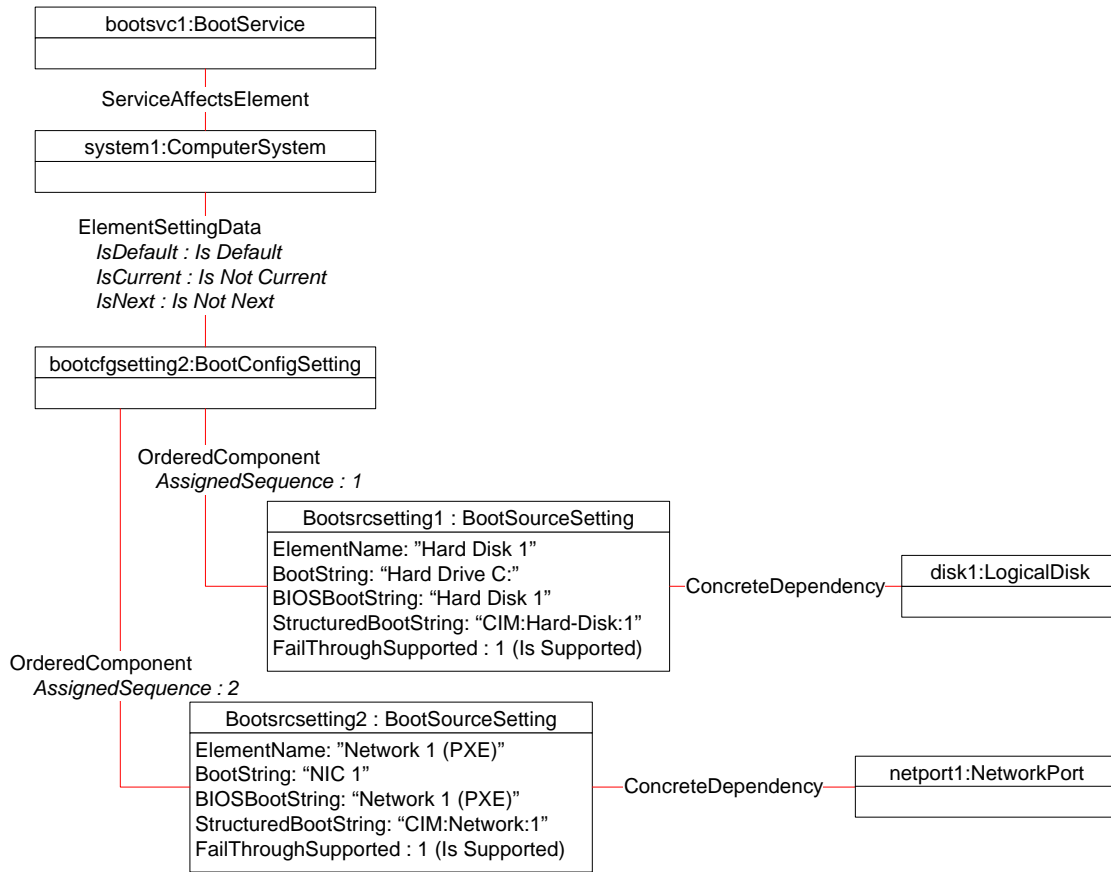
1028 Figure 11 shows an instance diagram for a Boot Configurable System, *system1*. *System1* has a single
1029 boot configuration, *bootcfgsetting2*. This boot configuration is a Default Boot Configuration, because the
1030 value of the ElementSettingData.IsDefault property is set to 1 (Is Default). There are no Next Boot
1031 Configuration or Current Boot Configuration.

1032 *Bootcfgsetting2* is associated with two instances of CIM_BootSourceSetting (*bootsrcsetting1* and
1033 *bootsrcsetting2*), through instances of CIM_OrderedComponent. The respective
1034 CIM_OrderedComponent.AssignedSequence properties designate the order in which the boot process
1035 should use the boot sources (*bootsrcsetting1* followed by *bootsrcsetting2*).

1036 On *bootsrcsetting1*, the FailThroughSupported property value of 1 (Is Supported) specifies that if, during
1037 the boot of *bootsrcsetting1*, the hard disk fails or times out, then the boot process should proceed to
1038 *bootsrcsetting2*, the network port using PXE.

1039 When the system represented by *system1* is enabled, the boot process will not be initiated because there
1040 is no Next Boot Configuration for the boot process to use. The system, *system1*, will be in an enabled, but
1041 not booted, state. One could manually boot the system from this state by applying an existing boot
1042 configuration (see subclause 9.14).

1043 *System1* would initiate the boot process if the Default Boot Configuration were also the Next Boot
1044 Configuration (see subclause 9.11) or a new boot configuration is created as the Next Boot Configuration
1045 (see subclause 9.13).



1046

1047

Figure 11 – Default Boot Configuration Object Diagram

1048 **9.11 Representing the Next Boot Configuration for a Computer System**

1049 Figure 12 shows an instance diagram for a Boot Configurable System, *system1*. *System1* has a single
 1050 boot configuration, *bootcfgsetting2*. This boot configuration is a Default Boot Configuration, because the
 1051 value of the *ElementSettingData.IsDefault* property is set to 1 (Is Default). This boot configuration is also
 1052 the Next Boot Configuration, because the value of the *ElementSettingData.IsNext* property is set to 1 (Is
 1053 Next).

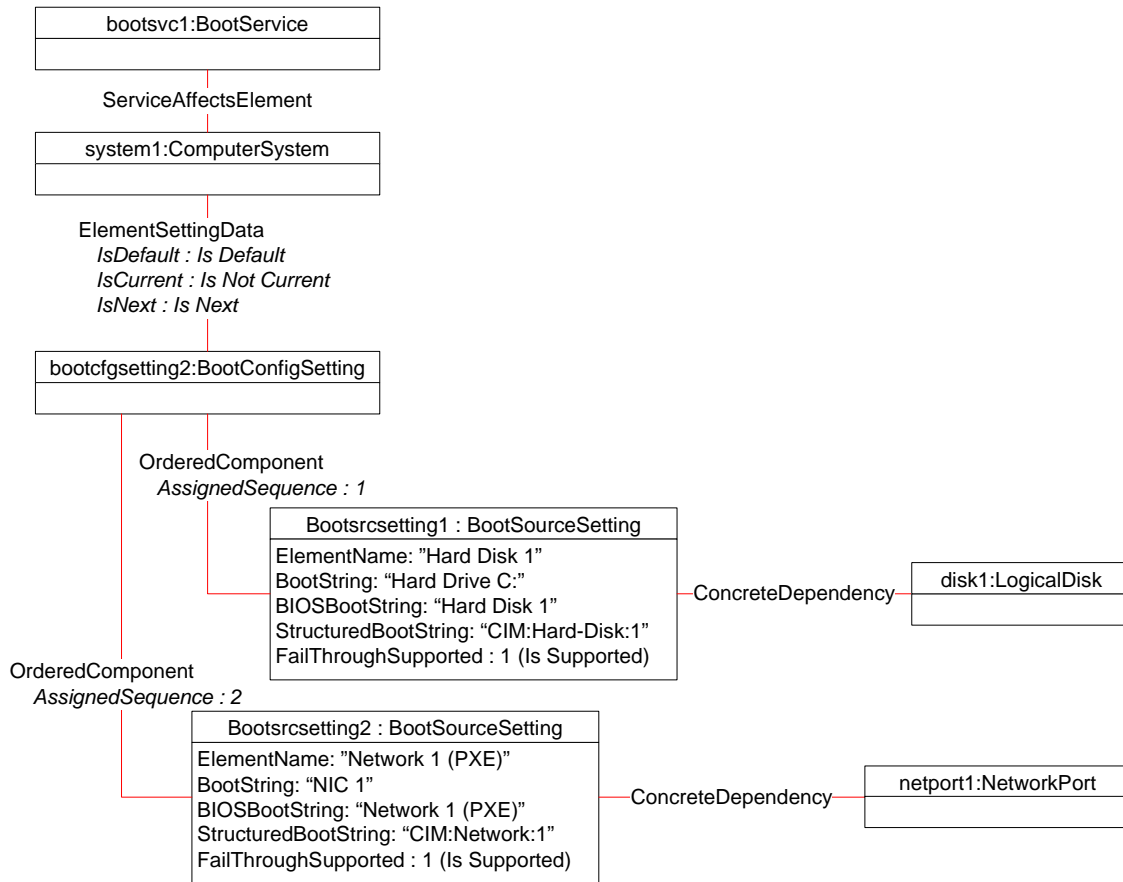
1054 *Bootcfgsetting2* is associated with two instances of *CIM_BootSourceSetting* (*bootsrcsetting1* and
 1055 *bootsrcsetting2*), through instances of *CIM_OrderedComponent*. The respective
 1056 *CIM_OrderedComponent.AssignedSequence* properties designate the order in which the boot process
 1057 should use the boot sources (*bootsrcsetting1* followed by *bootsrcsetting2*).

1058 On *bootsrcsetting1*, the *FailThroughSupported* property value of 1 (Is Supported) specifies that if the
 1059 *bootsrcsetting1*, the hard disk fails or times out during the boot process, then the boot process should
 1060 proceed to *bootsrcsetting2*, the network port using PXE.

1061 When the system represented by *system1* is enabled, the boot process will find a Next Boot
 1062 Configuration, *bootcfgsetting2* and proceed to use it to boot.

1063 When the system represented by *system1* is an enabled, but not booted, state. The
 1064 *BootService.ApplyBootConfigSetting()* method can be invoked referencing *system1* as the
 1065 *BootConfigurableSystem* parameter.

Boot Control Profile



1066

1067

Figure 12 – Next Boot Configuration Object Diagram

1068

9.12 Representing the Current Boot Configuration for a Booted Computer System

1069

1070

Figure 13 shows an instance diagram for the Boot Configurable System, *system1*, described in the previous use case (see Figure 12) after it has been successfully booted.

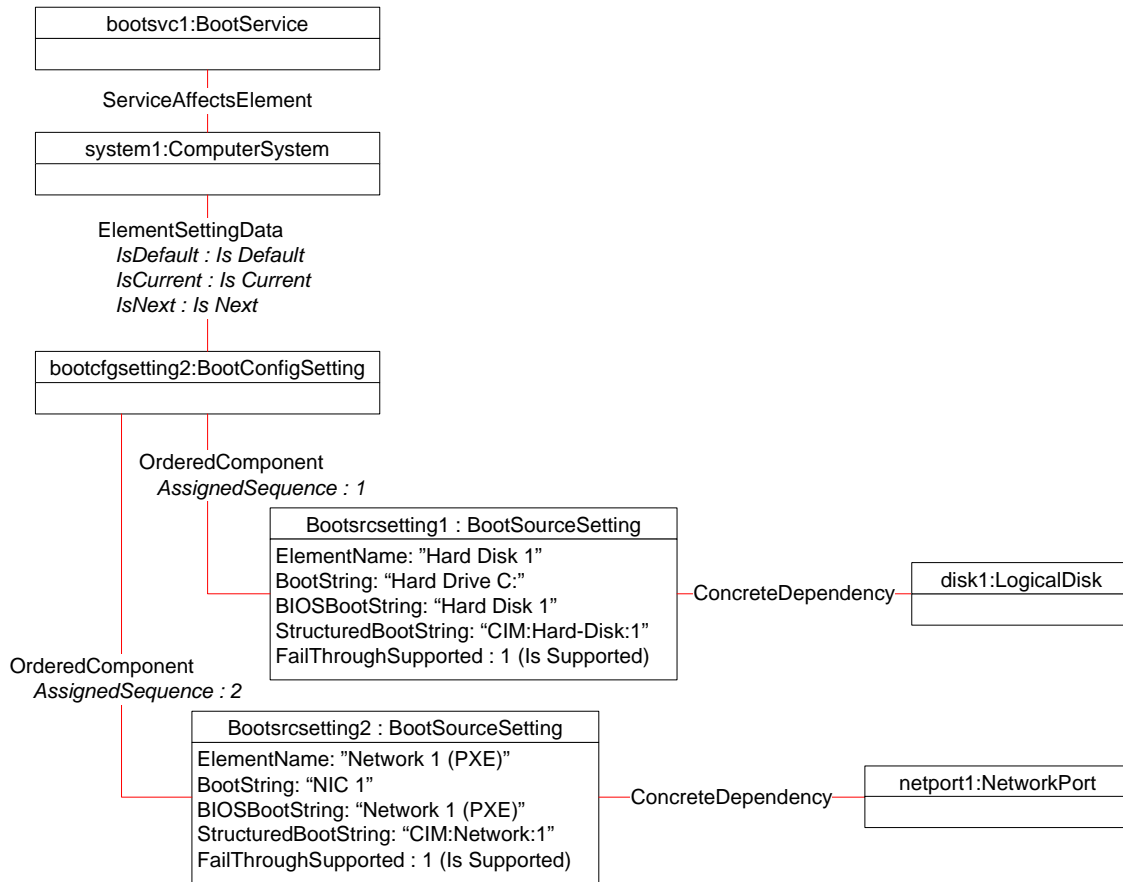
1071

1072

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

1073

1074



1075

1076

Figure 13 – Boot Configuration for a Booted System Object Diagram

1077 9.13 Create a New Boot Configuration

1078 Referencing the object diagram in Figure 11, a client could create a new boot configuration as follows:

- 1079 1. From the Boot Configurable System, *system1*, find the instance of CIM_BootService that manages
1080 the boot configurable system by traversing the CIM_ServiceAffectsElement association.
- 1081 2. Verify that the CreateBootConfigSetting() method is supported (see subclause 9.27). If not, a new
1082 boot configuration cannot be created.
- 1083 3. Find an existing instance of CIM_BootConfigSetting to use as the template. For this use case,
1084 *bootcfgsetting2* is the only template configuration available.
- 1085 4. Create the new boot configuration, *bootcfgsetting4*, by invoking the
1086 CIM_BootService.CreateBootConfigSetting() method. The ScopingComputerSystem parameter is set
1087 to *system1* and the StartingBootConfig parameter is set to *bootcfgsetting2*.

1088 Figure 14 shows the instance diagram after the CreateBootConfigSetting() method has been invoked and
1089 successfully completed on the computer system, *system1*, shown in Figure 13. The new boot
1090 configuration, *bootcfgsetting4*, is associated to *system1* through a new instance of
1091 CIM_ElementSettingData.

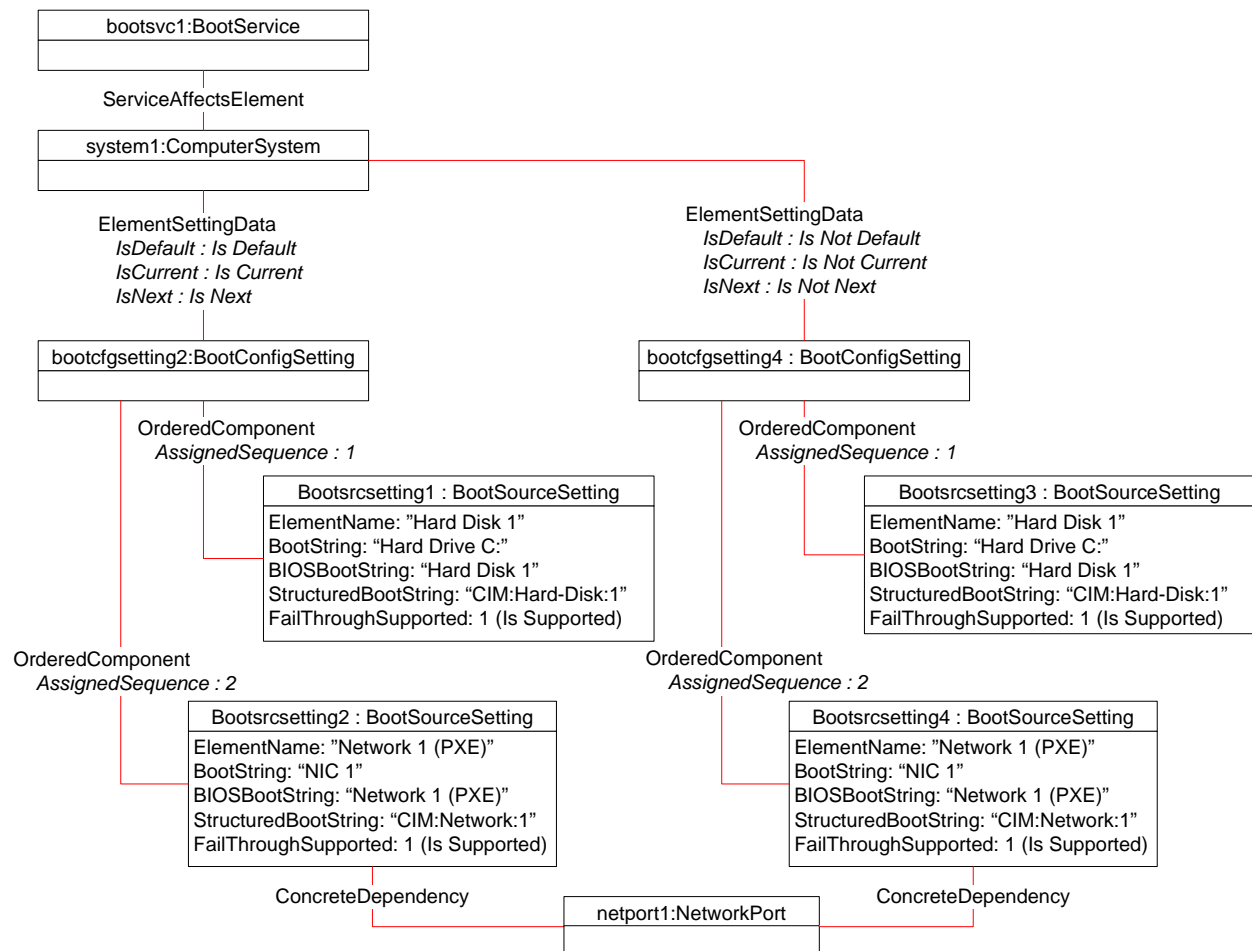
1092 In the new instance of CIM_ElementSettingData, the `IsDefault` property is set to 2 (Is Not Default); the
1093 `IsCurrent` property is set to 2 (Is Not Current); and the `IsNext` property is set to 2 (Is Not Next).

Boot Control Profile

1094 *bootcfgsetting4* is associated through instances of *CIM_OrderedComponent* to two instances of
 1095 *CIM_BootSourceSetting* (*bootsrcsetting3* and *bootsrcsetting4*), which are copies of *bootsrcsetting1* and
 1096 *bootsrcsetting2*, respectively.

1097 The instance of *CIM_NetworkPort* is not copied. *CIM_NetworkPort* is a concrete subclass of
 1098 *CIM_LogicalDevice*, which is not part of the *Boot Control Profile*. However, an instance of
 1099 *CIM_ConcreteDependency* has been created that associates the instance of *CIM_NetworkPort* to the
 1100 new instance of *CIM_BootSourceSetting* (*bootsrcsetting4*).

1101 *CIM_LogicalDisk* has been elided from the object diagram to make the diagram less cluttered, but the
 1102 instance of *CIM_LogicalDisk* is also not copied. An instance of *CIM_ConcreteDependency* is created that
 1103 associates the existing instance of *CIM_LogicalDisk* to the new instance of *CIM_BootSourceSetting*
 1104 (*bootsrcsetting3*).



1105

1106

Figure 14 – System with New CIM_BootConfigSetting

1107 9.14 Apply an Existing Boot Configuration

1108 Referencing the object diagram in Figure 11, a client could apply a boot configuration as follows:

- 1109 1. Find the instance of *CIM_BootService* for the boot configurable system as outlined in subclause 9.15.
- 1110 2. Verify that the *ApplyBootConfigSetting()* method is supported (see subclause 9.28). If not, a boot
 1111 configuration cannot be applied.

- 1112 3. Find the existing instances of CIM_BootConfigSetting for *system1* (see subclause 9.18). In this
1113 example, this results in *bootcfgsetting2*. Pick one of them to use as the boot configuration to apply.
- 1114 4. Apply the selected boot configuration, *bootcfgsetting2*, by invoking the
1115 CIM_BootService.ApplyBootConfigSetting() method. The ScopingComputerSystem parameter is set
1116 to *system1* and the BootConfigSetting parameter is set to *bootcfgsetting2*.
- 1117 The ApplyBootConfigSetting() method will boot *system1* by applying the boot configuration specified in
1118 *bootcfgsetting2*. If *system1* is currently booted, an implementation has the option of rejecting the
1119 ApplyBootConfigSetting() request or of rebooting the system.

1120 **9.15 Find the Boot Service for a Computer System**

1121 A client can find the boot service for a given computer system as follows:

- 1122 1. For the instance of CIM_ComputerSystem, representing the given computer system, select the
1123 referencing instance of CIM_BootService, representing the boot control service for the server,
1124 through the CIM_ServiceAffectsElement association.

1125 **9.16 Find the Boot Configuration for a Computer System**

1126 A client can find the boot configurations for a computer system as follows:

- 1127 1. From the instance of CIM_ComputerSystem, enumerate the CIM_ElementSettingData associations
1128 with CIM_BootConfigSetting as the SettingData reference.

1129 **9.17 Find the Boot Service Capabilities for a Computer System**

- 1130 1. Find the boot service for the computer system as specified in subclause 9.15 above.
- 1131 2. Select the instance of CIM_BootServiceCapabilities through the CIM_ElementCapabilities
1132 association.

1133 **9.18 Find the Current Boot Configuration for a Computer System**

1134 A client can find the current boot configuration for a computer system as follows:

- 1135 1. From the instance of CIM_ComputerSystem, enumerate the CIM_ElementSettingData associations
1136 with CIM_BootConfigSetting as the SettingData reference.
- 1137 2. Find the instance of CIM_ElementSettingData whose IsCurrent property is set to 1 (Is Current).
- 1138 3. The CIM_BootConfigSetting instance referenced by this association instance represents the current
1139 boot configuration.

1140 **9.19 Find the Default Boot Configuration for a Computer System**

1141 A client can find the default boot configuration for a computer system as follows:

- 1142 1. From the instance of CIM_ComputerSystem, enumerate the CIM_ElementSettingData associations
1143 with CIM_BootConfigSetting as the SettingData reference.
- 1144 2. Find the instance of CIM_ElementSettingData whose IsDefault property is set to 1 (Is Default).
- 1145 3. The CIM_BootConfigSetting instance referenced by this association instance represents the default
1146 boot configuration.

1147 **9.20 Find the Boot Configuration that Will Be Used during the Next Reboot for a**
1148 **Computer System**

1149 A client can find the boot configuration that will be used during a computer system's next reboot as
1150 follows:

- 1151 1. For the instance of CIM_ComputerSystem, enumerate the CIM_ElementSettingData associations
1152 with CIM_BootConfigSetting as the SettingData reference.
- 1153 2. Find the CIM_ElementSettingData instances for the instance whose IsNext property is set to 3 (Is
1154 Next For Single Use). The CIM_BootConfigSetting instance referenced by this association instance
1155 represents the next boot configuration.
- 1156 3. If no instance is found, find the instance of CIM_ElementSettingData whose IsNext property is set to
1157 1 (Is Next). The CIM_BootConfigSetting instance referenced by this association instance represents
1158 the next boot configuration.

1159 **9.21 Make a Boot Configuration Applicable for Subsequent Reboots**

1160 A client can make a boot configuration apply to a computer system for subsequent reboots as follows:

- 1161 1. From the instance of CIM_ComputerSystem, find the CIM_BootConfigSetting of interest as outlined in
1162 subclauses 9.9 through 9.12.
- 1163 2. On the instance of the CIM_ElementSettingData association that associates the instance of
1164 CIM_ComputerSystem to the instance of CIM_BootConfigSetting, use the intrinsic ModifyInstance()
1165 to change the IsNext property to 1 (Is Next).

1166 Note that this boot configuration applies for all subsequence reboots, unless it is overridden by a Next
1167 Single Use Boot Configuration that is associated to the CIM_ComputerSystem of interest.

1168 **9.22 Make a Boot Configuration Applicable for the Next Reboot Only**

1169 A client can make a boot configuration apply to a computer system for only the next reboot as follows:

- 1170 3. From the instance of CIM_ComputerSystem, find the CIM_BootConfigSetting of interest as outlined in
1171 subclauses 9.9 through 9.12.
- 1172 4. On the instance of the CIM_ElementSettingData association that associates the instance of
1173 CIM_ComputerSystem to the instance of CIM_BootConfigSetting, use the intrinsic ModifyInstance()
1174 to change the IsNext property to 3 (Is Next For Single Use).

1175 The behavior of this property after the next boot is specified in subclause 7.4.5.

1176 **9.23 Determine If the Computer System Supports PXE Boot**

1177 A client can determine if the computer system supports PXE boot as follows:

- 1178 1. For the instance of CIM_ComputerSystem enumerate its instances of CIM_BootConfigSetting as
1179 outline in subclause 9.18.
- 1180 2. For each instance of CIM_BootConfigSetting, enumerate the instances of CIM_BootSourceSetting.
- 1181 3. For each CIM_BootSourceSetting, inspect the BootString, BIOSBootString, or StructuredBootString
1182 property to determine if PXE is supported.

1183 **9.24 Find the Boot Order for a Computer System for the Next Reboot**

1184 This use case references the object diagram in Figure 9, which represents a complex boot order.

1185 A client can find the boot order for the next reboot of a computer system as follows:

- 1186 1. From the instance of CIM_ComputerSystem, *system1*, find the CIM_BootConfigSetting that will be
1187 used during the next reboot, *bootcfgsetting1* (see subclause 3).
- 1188 2. Determine the boot order for *bootcfgsetting1* by enumerating the CIM_OrderedComponent
1189 associations with *bootcfgsetting1* as the GroupComponent reference. The results in this example
1190 would be *bootsrcsetting1*, *bootsrcsetting2* and *bootsrcsetting3*.
- 1191 3. Use the CIM_OrderedComponent.AssignedSequence property to determine the boot order. The boot
1192 order in this example will be *bootsrcsetting1* followed by *bootsrcsetting3*. The boot source
1193 represented by *bootsrcsetting2* will be ignored because its associated AssignedSequence property
1194 value is 0.
- 1195 4. For each boot source, determine whether any it contains additional boot sources by checking for a
1196 CIM_LogicalIdentity association to an instance of CIM_BootConfigSetting; in this example,
1197 *bootcfgsetting20*, and repeat steps in this subclause recursively to find the boot order of the
1198 associated boot sources.

1199 **9.25 Change the Boot Order for a Computer System**

1200 This use case references the object diagram in Figure 12.

1201 A client can change the boot order for a computer system as follows:

- 1202 1. Find the boot configuration of interest from the set of boot configurations for the computer system as
1203 outlined in subclause 9.18.
- 1204 2. Find the set of boot sources for the boot configuration by following the OrderedComponent
1205 associations from the selected boot configuration representation (*bootcfgsetting2*) to all instances of
1206 CIM_BootSourceSetting. In this example, this results in *bootsrcsetting1* and *bootsrcsetting2*.
- 1207 3. Determine the desired boot order.
- 1208 4. Create an array of CIM_BootSourceSetting references. Assign the existing boot sources to the array
1209 in the new order. For instance, if one wanted to reverse the boot order in this example, the array
1210 would contain *bootsrcsetting2* at index 0 and *bootsrcsetting1* at index 1.
- 1211 5. Invoke the ChangeBootOrder() method on the selected instance of CIM_BootConfigSetting. The
1212 Source parameter is set to the array created above.

1213 Note: The order of each boot configuration must be changed independently. Thus if the computer system
1214 has a complex boot structure, such as that illustrated in Figure 9, changing the boot order for the system
1215 may require changing the boot order for multiple CIM_BootConfigSetting instances.

1216 **9.26 Determine Whether BootService.ElementName Is Modifiable**

1217 A client can determine whether the ElementName can be modified as follows:

- 1218 1. Find the CIM_BootServiceCapabilities instance associated with the CIM_BootService instance
1219 through the CIM_ElementCapabilities association.
- 1220 2. If a CIM_BootConfigCapabilities instance cannot be found, then the CIM_BootService.ElementName
1221 property cannot be modified.
- 1222 3. Query the value of the CIM_BootServiceCapabilities.ElementNameEditSupported.
- 1223 4. If the value is TRUE, the CIM_BootService.ElementName property can be modified
- 1224 5. If the value of ElementNameEditSupported has a value of FALSE, then the
1225 CIM_BootService.ElementName property cannot be modified.

1226 **9.27 Determine Whether a New Boot Configuration Can Be Created**

1227 A client can determine whether a new boot configuration can be created as follows:

- 1228 1. Find the CIM_BootServiceCapabilities instance that is associated with the CIM_BootService instance
1229 through the CIM_ElementCapabilities association.
- 1230 2. Query the value of the CIM_BootServiceCapabilities.BootConfigCapabilities property array. If the
1231 array contains the value 2 (Creates Boot Configuration), the client's ability to create a boot
1232 configuration is supported.
- 1233 3. If the BootConfigCapabilities property array does not contain the value 2 (Creates Boot
1234 Configuration), or there is not an instance of CIM_BootServiceCapabilities associated with the
1235 CIM_BootService instance, a boot configuration cannot be created.

1236 **9.28 Determine Whether a Boot Configuration Can Be Applied**

1237 A client can determine whether a boot configuration can be manually applied to the boot configurable
1238 systems as follows:

- 1239 1. Find the CIM_BootServiceCapabilities instance that is associated with the CIM_BootService instance
1240 through the CIM_ElementCapabilities association.
- 1241 2. Query the value of the CIM_BootServiceCapabilities.BootConfigCapabilities property array. If the
1242 array contains the value 3 (Applies Boot Configuration), the client's ability to manually apply a boot
1243 configuration is supported.
- 1244 3. If the BootConfigCapabilities property array does not contain the value 3 (Applies Boot Configuration),
1245 or there is not an instance of CIM_BootServiceCapabilities associated with the CIM_BootService
1246 instance, a boot configuration cannot be manually applied.

1247 **10 CIM Elements**

1248 Table 22 shows the instances of CIM Elements for this profile. Instances of the following CIM Elements
 1249 shall be implemented as described in Table 22. Clauses 7 (“Implementation”) and 8 (“Methods”) may
 1250 impose additional requirements on these elements.

1251 **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.

1252 **10.1 CIM_RegisteredProfile**

1253 CIM_RegisteredProfile identifies the *Boot Control Profile* in order for a client to determine whether an
 1254 instance of CIM_ComputerSystem is conformant with this profile. The CIM_RegisteredProfile class is
 1255 defined by the *Profile Registration Profile*. With the exception of the mandatory values specified for the
 1256 properties below, the behavior of the CIM_RegisteredProfile instance is per the *Profile Registration*
 1257 *Profile*. Table 23 contains the requirements for elements of this class.

1258 **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 "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

Boot Control Profile

1259 Note: Previous versions of this document included the suffix 'Profile' for the RegisteredName value. If
1260 implementations querying for RegisteredName value find the suffix 'Profile', they should ignore the suffix,
1261 with any surrounding white spaces, before any comparison is done with the value as specified in this
1262 document.

1263 10.2 CIM_BootService

1264 The CIM_BootService class represents the ability to view and control the boot settings of a computer
1265 system. Table 24 contains the requirements for elements of this class.

1266 **Table 24 – Class: CIM_BootService**

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

1267 10.3 CIM_BootServiceCapabilities

1268 Support of the CIM_BootServiceCapabilities class is optional.

1269 When supported, CIM_BootServiceCapabilities is used to indicate the capabilities of the boot service.
1270 Table 25 contains the requirements for elements of this class.

1271 **Table 25 – Class: CIM_BootServiceCapabilities**

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

1272 **10.4 CIM_BootConfigSetting**

1273 The CIM_BootConfigSetting class represents a boot configuration of a computer system. Table 26
 1274 contains the requirements for elements of this class.

1275 **Table 26 – Class: CIM_BootConfigSetting**

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

1276 **10.5 CIM_BootSettingData**

1277 Support of the CIM_BootSettingData class is optional.

1278 The CIM_BootSettingData class represents the settings that apply during booting of a computer system.
 1279 Table 27 contains the requirements for elements of this class.

1280 For each property added in a concrete subclass of CIM_BootSettingData, there shall be a Description
 1281 qualifier that contains a string which describes the setting. When the range of the setting is bounded and
 1282 discrete, the Values and ValueMap qualifiers should contain the values and name of the values,
 1283 respectively, which are applicable for the setting.

1284 **Table 27 – Class: CIM_BootSettingData**

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

1285 **10.6 CIM_BootSourceSetting**

1286 Support of the CIM_BootSourceSetting class is optional.

1287 The CIM_BootSourceSetting class represents a boot source, from which booting is attempted during the
 1288 boot process. Table 28 contains the requirements for elements of this class.

1289 **Table 28 – Class: CIM_BootSourceSetting**

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

1290 **10.7 CIM_ConcreteComponent**

1291 Subclause 10.7 describes optional behavior.

1292 **10.7.1 Relating CIM_BootConfigSetting to a Concrete Subclass of CIM_SettingData**

1293 When the CIM_ConcreteComponent association is used to relate an instance of a concrete subclass of
 1294 CIM_SettingData to a CIM_BootConfigSetting instance, Table 29 contains the requirements for elements
 1295 of this class.

1296 **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 "**".

1297 **10.7.2 Relating CIM_BootConfigSetting to a Concrete Subclass of CIM_BootSettingData**

1298 When the CIM_ConcreteComponent association is used to relate an instance of a concrete subclass of
 1299 CIM_BootSettingData to a CIM_BootConfigSetting instance, Table 30 contains the requirements for
 1300 elements of this class.

1301 **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 "**".

1302 **10.7.3 Relating CIM_BootSourceSetting to a Concrete Subclass of CIM_SettingData**

1303 When the CIM_ConcreteComponent association is used to relate an instance of a concrete subclass of
 1304 CIM_SettingData to a CIM_BootSourceSetting instance. Table 31 contains the requirements for elements
 1305 of this class.

1306 **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 "**".

1307 **10.7.4 Relating CIM_BootSourceSetting to a Concrete Subclass of**
 1308 **CIM_BootSettingData**

1309 When the CIM_ConcreteComponent association is used to relate an instance a concrete subclass of
 1310 CIM_BootSettingData to a CIM_BootSourceSetting instance, Table 32 contains the requirements for
 1311 elements of this class.

1312 **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 "**".

1313 **10.8 CIM_ConcreteDependency**

1314 Subclause 10.8 describes optional behavior.

1315 When the CIM_ConcreteDependency association is used to relate the dependency of a
 1316 CIM_BootSourceSetting instance on an instance of a concrete subclass of CIM_LogicalDevice, Table 33
 1317 contains the requirements for elements of this class.

1318 **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 "**".

1319 **10.9 CIM_ElementCapabilities**

1320 Subclause 10.9 describes optional behavior.

1321 When the CIM_ElementCapabilities association is used to relate an instance of
 1322 CIM_BootServiceCapabilities with an instance of CIM_BootService, Table 34 contains the requirements
 1323 for elements of this class.

1324

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".

1325 **10.10 CIM_ElementSettingData**

1326 The CIM_ElementSettingData association is used to relate the CIM_BootConfigSetting instance to the
1327 CIM_ComputerSystem instance to which it applies. Table 35 contains the requirements for elements of
1328 this class.

1329

Table 35 – Class: CIM_ElementSettingData

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

1330 **10.11 CIM_HostedService**

1331 The CIM_HostedService association is used to relate the CIM_BootService to the CIM_ComputerSystem
1332 on which it is hosted. Table 36 contains the requirements for elements of this class.

1333

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 "**".

1334 **10.12 CIM_LogicalIdentity**

1335 Support of the CIM_LogicalIdentity association is conditional.

1336 Conditional Requirement: The support is required if instances of CIM_BootSourceSetting are used to
1337 represent aggregated boot sources; see subclause 7.9.

1338 When supported, CIM_LogicalIdentity is used to associate an instance of CIM_BootSourceSetting with an
 1339 instance of CIM_BootConfigSetting. Table 37 contains the requirements for elements of this class.

1340 **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"

1341 **10.13 CIM_OrderedComponent**

1342 Support of the CIM_OrderedComponent association is conditional.

1343 Conditional Requirement: The support is required if the CIM_BootSourceSetting instances are used to
 1344 represent boot sources; see 7.11.1.

1345 When supported, the CIM_OrderedComponent association is used to indicate the order in which
 1346 CIM_BootSourceSetting instances should be attempted for a CIM_BootConfigSetting instance. Table 38
 1347 contains the requirements for elements of this class.

1348 **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.

1349 **10.14 CIM_ServiceAffectsElement**

1350 The CIM_ServiceAffectsElement association is used to associate the CIM_BootService instance with a
 1351 CIM_ComputerSystem instance that it affects. Table 39 contains the requirements for elements of this
 1352 class.

1353 **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)

Boot Control Profile

1354

**ANNEX A
(informative)**

Change Log

1355
1356
1357
1358

Version	Date	Description
1.0.0a	10/10/06	Preliminary Standard
1.0.0	11/03/08	Final Standard

1359

ANNEX B (informative)

Acknowledgments

1360
1361
1362
1363

1364 **Editor:**

1365 John Leung – Intel

1366 David Simpson – IBM

1367 **Contributors:**

1368 • Aaron Merkin – IBM

1369 • Jon Hass – Dell

1370 • Khachatur Papanyan – Dell

1371 • Enoch Suen – Dell

1372 • Jeff Hilland – HP

1373 • Hemal Shah - Broadcom

1374 • Christina Shaw – HP

1375 • Perry Vincent – Intel

1376 • Arvind Kumar – Intel

1377