Boot Control Profile
Copyright Notice

Copyright © 2008−2010, 2014 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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

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

http://www.dmtf.org/about/policies/disclosures.php.
Contents

34 Foreword ............................................................................................................. 7
35 Introduction ........................................................................................................... 8
36 1 Scope .................................................................................................................... 9
37 2 Normative references .......................................................................................... 9
38 2.1 Approved references ....................................................................................... 9
39 2.2 Other references ............................................................................................. 9
40 3 Terms and definitions ......................................................................................... 9
41 4 Symbols and abbreviated terms ......................................................................... 11
42 5 Synopsis ............................................................................................................. 12
43 6 Description ......................................................................................................... 12
44 6.1 Class diagram .................................................................................................. 14
45 7 Implementation ................................................................................................... 15
46 7.1 CIM_BootService ............................................................................................. 15
47 7.2 CIM_ComputerSystem..................................................................................... 16
48 7.3 Representing boot service capabilities ......................................................... 16
49 7.4 Boot configurations ......................................................................................... 17
50 7.5 Applying the boot configuration .................................................................... 18
51 7.6 Creating a boot configuration ......................................................................... 19
52 7.7 Deleting a boot configuration ......................................................................... 20
53 7.8 Identifying boot sources ................................................................................. 20
54 7.9 Changing the boot order ................................................................................. 24
55 7.10 Representing a set of aggregated boot sources .............................................. 24
56 7.11 Boot order during the boot process ............................................................... 25
57 7.12 Settings to apply during the boot process ..................................................... 26
48 8 Methods .............................................................................................................. 27
49 8.1 CIM_BootService.CreateBootConfigSetting()............................................. 27
50 8.2 CIM_BootService.ApplyBootConfigSetting().............................................. 28
51 8.3 CIM_BootConfigSetting.ChangeBootOrder() ............................................... 30
52 8.4 Profile conventions for operations .................................................................. 30
53 8.5 CIM_BootService ........................................................................................... 31
54 8.6 CIM_BootConfigSetting ............................................................................... 31
55 8.7 CIM_BootSettingData .................................................................................... 32
56 8.8 CIM_BootSourceSetting ............................................................................... 33
57 8.9 CIM_ConcreteComponent .............................................................................. 33
58 8.10 CIM_ConcreteDependency ........................................................................... 33
59 8.11 CIM_ElementCapabilities ............................................................................ 34
60 8.12 CIM_ElementSettingData ............................................................................. 34
61 8.13 CIM_BootServiceCapabilities .................................................................... 35
62 8.14 CIM_HostedService ....................................................................................... 36
63 8.15 CIM_LogicalIdentity ..................................................................................... 36
64 8.16 CIM_OrderedComponent ............................................................................. 36
65 8.17 CIM_ServiceAffectsElement ...................................................................... 37
66 9 Use cases ............................................................................................................ 37
67 9.1 Advertising the profile conformance .............................................................. 37
68 9.2 Object diagram for a monolithic server ......................................................... 37
69 9.3 Object diagram for a monolithic server with service processor ...................... 38
70 9.4 Object diagram for a modular system ............................................................ 39
71 9.5 PXE boot source ............................................................................................ 40
72 9.6 Disk boot source ............................................................................................ 41
73 9.7 Local CDROM and floppy boot sources ....................................................... 42
74 9.8 Representing IPL and Boot Control Vector (BCV) lists .................................. 42
75 9.9 Representing settings and boot settings ......................................................... 44

Version 1.1.0  DMTF Standard  3
Boot Control Profile

9.10 Representing the same boot device ................................................................. 44
9.11 Representing the default boot configuration for a computer system .................. 45
9.12 Representing the next boot configuration for a computer system ....................... 46
9.13 Representing the current boot configuration for a booted computer system ........... 47
9.14 Create a new boot configuration ........................................................................ 48
9.15 Apply an existing boot configuration .................................................................... 49
9.16 Find the boot service for a computer system ....................................................... 50
9.17 Find the boot configuration for a computer system .............................................. 50
9.18 Find the boot service capabilities for a computer system .................................... 50
9.19 Find the current boot configuration for a computer system ............................... 50
9.20 Find the default boot configuration for a computer system ............................... 50
9.21 Find the boot configuration that will be used during the next reboot for a computer system ........................................................................................................... 51
9.22 Make a boot configuration applicable for subsequent reboots ............................ 51
9.23 Make a boot configuration applicable for the next reboot only ............................. 51
9.24 Determine whether the computer system supports PXE boot ............................. 51
9.25 Find the boot order for a computer system for the next reboot ......................... 51
9.26 Change the boot order for a computer system .................................................... 52
9.27 Determine whether BootService.ElementName is modifiable ............................. 52
9.28 Determine whether a new boot configuration can be created ............................ 53
9.29 Determine whether a boot configuration can be applied..................................... 53
10 CIM Elements ........................................................................................................ 53
   10.1 CIM_RegisteredProfile ..................................................................................... 54
   10.2 CIM_BootService ............................................................................................ 54
   10.3 CIM_BootServiceCapabilities ....................................................................... 54
   10.4 CIM_BootConfigSetting .................................................................................. 55
   10.5 CIM_BootSettingData ..................................................................................... 55
   10.6 CIM_BootSourceSetting .................................................................................. 55
   10.7 CIM_ConcreteComponent ................................................................................ 56
   10.8 CIM_ConcreteDependency ............................................................................. 57
   10.9 CIM_ElementCapabilities .............................................................................. 58
   10.10 CIM_ElementSettingData .............................................................................. 58
   10.11 CIM_HostedService ...................................................................................... 58
   10.12 CIM_LogicalIdentity ...................................................................................... 59
   10.13 CIM_OrderedComponent ............................................................................... 59
   10.14 CIM_ServiceAffectsElement ........................................................................ 60
122 ANNEX A (informative) Change log ...................................................................... 61

Figures

125 Figure 1 - Boot Control Profile: Class diagram ................................................. 14
126 Figure 2 - Registered Profile ............................................................................. 37
127 Figure 3 - Monolithic server object diagram ..................................................... 38
128 Figure 4 - Monolithic server with service processor object diagram .................. 38
129 Figure 5 - Modular system object diagram ........................................................ 39
130 Figure 6 - Modular system object diagram ........................................................ 40
131 Figure 7 - PXE boot sources object diagram ..................................................... 41
132 Figure 8 - Booting from disk ............................................................................. 41
133 Figure 9 - Booting from CDROM and floppy ..................................................... 42
134 Figure 10 - Booting from IPL and BCV devices ............................................... 43
135 Figure 11 - Setting data and boot setting data .................................................... 44
136 Figure 12 - Representing the same boot device ................................................. 45
Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Related Profiles</td>
</tr>
<tr>
<td>2</td>
<td>Structured name identifiers</td>
</tr>
<tr>
<td>3</td>
<td>CreateBootConfigSetting() Method: Return code values</td>
</tr>
<tr>
<td>4</td>
<td>CreateBootConfigSetting() Method: Parameters</td>
</tr>
<tr>
<td>5</td>
<td>ApplyBootConfigSetting() Method: Return code values</td>
</tr>
<tr>
<td>6</td>
<td>ApplyBootConfigSetting() Method: Parameters</td>
</tr>
<tr>
<td>7</td>
<td>ChangeBootOrder() Method: Return code values</td>
</tr>
<tr>
<td>8</td>
<td>ChangeBootOrder() Method: Parameters</td>
</tr>
<tr>
<td>9</td>
<td>Operations: CIM_BootService</td>
</tr>
<tr>
<td>10</td>
<td>Operations: CIM_BootConfigSetting</td>
</tr>
<tr>
<td>11</td>
<td>Operations: CIM_BootSettingData</td>
</tr>
<tr>
<td>12</td>
<td>Operations: CIM_BootSourceSetting</td>
</tr>
<tr>
<td>13</td>
<td>Operations: CIM_ConcreteComponent</td>
</tr>
<tr>
<td>14</td>
<td>Operations: CIM_ConcreteDependency</td>
</tr>
<tr>
<td>15</td>
<td>Operations: CIM_ElementCapabilities</td>
</tr>
<tr>
<td>16</td>
<td>Operations: CIM_ElementSettingData</td>
</tr>
<tr>
<td>17</td>
<td>Operations: CIM_BootServiceCapabilities</td>
</tr>
<tr>
<td>18</td>
<td>Operations: CIM_HostedService</td>
</tr>
<tr>
<td>19</td>
<td>Operations: CIM_LogicalIdentity</td>
</tr>
<tr>
<td>20</td>
<td>Operations: CIM_OrderedComponent</td>
</tr>
<tr>
<td>21</td>
<td>Operations: CIM_ServiceAffectsElement</td>
</tr>
<tr>
<td>22</td>
<td>CIM Elements – Boot Control Profile</td>
</tr>
<tr>
<td>23</td>
<td>Class: CIM_RegisteredProfile</td>
</tr>
<tr>
<td>24</td>
<td>Class: CIM_BootService</td>
</tr>
<tr>
<td>25</td>
<td>Class: CIM_BootServiceCapabilities</td>
</tr>
<tr>
<td>26</td>
<td>Class: CIM_BootConfigSetting</td>
</tr>
<tr>
<td>27</td>
<td>Class: CIM_BootSettingData</td>
</tr>
<tr>
<td>28</td>
<td>Class: CIM_BootSourceSetting</td>
</tr>
<tr>
<td>29</td>
<td>Class: CIM_ConcreteComponent – Use 1</td>
</tr>
<tr>
<td>30</td>
<td>Class: CIM_ConcreteComponent – Use 2</td>
</tr>
<tr>
<td>31</td>
<td>Class: CIM_ConcreteComponent – Use 3</td>
</tr>
<tr>
<td>32</td>
<td>Class: CIM_ConcreteComponent – Use 4</td>
</tr>
<tr>
<td>33</td>
<td>Class: CIM_ConcreteDependency</td>
</tr>
<tr>
<td>34</td>
<td>Class: CIM_ElementCapabilities</td>
</tr>
<tr>
<td>35</td>
<td>Class: CIM_ElementSettingData</td>
</tr>
<tr>
<td>36</td>
<td>Class: CIM_HostedService</td>
</tr>
<tr>
<td>37</td>
<td>Class: CIM_LogicalIdentity</td>
</tr>
<tr>
<td>38</td>
<td>Class: CIM_OrderedComponent</td>
</tr>
<tr>
<td>39</td>
<td>Class: CIM_ServiceAffectsElement</td>
</tr>
</tbody>
</table>
Foreword

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

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

Acknowledgments

The authors wish to acknowledge the following people.

Editors:
- John Leung – Intel
- Khachatur Papanyan – Dell
- Hemal Shah – Broadcom
- David Simpson – IBM

Contributors:
- Jon Hass – Dell
- Aaron Merkin – IBM
- Jeff Hilland – Hewlett-Packard Company
- Arvind Kumar – Intel
- Khachatur Papanyan – Dell
- Christina Shaw – Hewlett-Packard Company
- Enoch Suen – Dell
- Perry Vincent – Intel
205 Introduction

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

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

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

2 Normative references

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

2.1 Approved references

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

2.2 Other references


3 Terms and definitions

3.1 can

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

3.2 cannot

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

3.3 conditional

used to indicate requirements strictly to be followed, in order to conform to the document when the specified conditions are met
3.4 mandatory
used to indicate requirements strictly to be followed, in order to conform to the document and from which no deviation is permitted

3.5 may
used to indicate a course of action permissible within the limits of the document

3.6 need not
used to indicate a course of action permissible within the limits of the document

3.7 optional
used to indicate a course of action permissible within the limits of the document

3.8 referencing profile
indicates a profile that owns the definition of a class used, but not defined, in this document and can be included in the “Referenced Profiles” table

3.9 shall
used to indicate requirements strictly to be followed, in order to conform to the document and from which no deviation is permitted

3.10 shall not
used to indicate requirements strictly to be followed, in order to conform to the document and from which no deviation is permitted

3.11 should
used to indicate that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.12 should not
used to indicate that a certain possibility or course of action is deprecated but not prohibited

3.13 unspecified
indicates that this profile does not define any constraints for the referenced CIM element or operation

3.14 Boot Configurable System
an instance of CIM_ComputerSystem whose boot configurations are being managed

3.15 Boot Configuration
a collection of settings that are applied to a boot configurable system during the boot process
3.16 **Boot Configuration Representation**
the CIM representation of a boot configuration, which consists of an instance of class CIM_BootConfigSetting and, optionally, all of the instances of classes CIM_BootSourceSetting, CIM_BootSettingData and CIM_SettingData that it directly or indirectly aggregates.

3.17 **Current Boot Configuration**
the instance of CIM_BootConfigSetting that was used the last time the managed system was successfully booted.

3.18 **Default Boot Configuration**
the instance of CIM_BootConfigSetting that the computer system manufacturer or a client has informatively tagged as its default boot configuration.

3.19 **Next Boot Configuration**
the instance of CIM_BootConfigSetting that will be used during the next boot of the Boot Configurable System.

3.20 **Next Single Use Boot Configuration**
the instance of CIM_BootConfigSetting that will only be used during the next boot of the Boot Configurable System and then not used again.

3.21 **Not Next Boot Configuration**
an instance of CIM_BootConfigSetting that will not be used during the next boot.

3.22 **Template Boot Configuration**
an existing instance of CIM_BootConfigSetting that is to be used as the template for creating a new boot configuration.

4 **Symbols and abbreviated terms**

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

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

4.3 **PXE**
Preboot Execution Environment. See the [BIOS Boot Specification](#) for additional information.
5 Synopsis

Profile Name: Boot Control

Version: 1.1.0

Organization: DMTF

CIM Schema Version: 2.19

Central Class: CIM_BootService

Scoping Class: CIM_ComputerSystem

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

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

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

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

Table 1 – Related Profiles

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Organization</th>
<th>Version</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Registration</td>
<td>DMTF</td>
<td>1.0</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

6 Description

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

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

- A computer system can have one or more boot configurations.

- A computer system can contain a boot configuration that is used during each boot.

- A computer system can contain a single-use boot configuration that is used only during the next boot and then not used again.

- A computer system can contain a current boot configuration that represents the boot configuration successfully used in the last boot.

- A computer system can contain a default boot configuration that is set by the computer system manufacturer or a client.

- A computer system can create new boot configurations.

- A computer system can apply a boot configuration to an active or inactive computer system.
A typical boot configuration could have the following characteristics:

- A boot configuration can contain a boot order that specifies the order in which boot devices are accessed. The boot devices include, but are not limited to, floppy device, CD device, hard disks, network controllers (using the PXE protocol), and BCV devices composed of additional boot sources.

- A boot configuration can contain data that can affect various computer system components during the boot process.

- A boot configuration can contain data that can be passed to the booted image (for example, second-stage boot loader or bootblock) in the form of a boot string.

- Boot devices can be local to the computer system or remote to the computer system.

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

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

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

A computer system can have multiple boot configurations. Each boot configuration is modeled by a Boot Configuration Representation, which consists of an instance of CIM_BootConfigSetting class and, optionally, all of the instances of classes CIM_BootSourceSetting, CIM_BootSettingData and CIM_SettingData that the instance of CIM_BootConfigSetting aggregates.
The usage of each Boot Configuration Representation during the boot process is determined by the IsNext property of the CIM_ElementSettingData association between the Boot Configuration Representation and Boot Configurable System whose boot configuration is being managed.

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

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

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

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

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

7 Implementation

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

7.1 CIM_BootService

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

7.1.1 CIM_BootService.ElementName

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

7.1.2 Modifying ElementName is supported

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

Conditional Requirement:

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

2) The CIM_BootServiceCapabilities.ElementNameEditSupport property has the value of TRUE.

3) The CIM_BootServiceCapabilities.MaxElementNameLen property has a non-zero value.

The implementation shall allow the CIM_BootService.ModifyInstance intrinsic operation to change the value of the ElementName property. The ModifyInstance operation shall enforce the length restriction specified in the MaxElementNameLen property.
7.1.3 Modifying ElementName is not supported

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

Conditional Requirement 1:

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

Conditional Requirement 2:

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

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

7.2 CIM_ComputerSystem

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

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

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

7.3 Representing boot service capabilities

Subclause 7.3 describes optional behavior.

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

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

7.3.1 Representing implementation specific boot service capabilities

Subclause 7.3.1 describes optional behavior.

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

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

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

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

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

7.4.1 CIM_ElementSettingData

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

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

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

7.4.2 Default boot configuration

Subclause 7.4.2 describes optional behavior.

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

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

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

7.4.3 Current boot configuration

Subclause 7.4.3 describes optional behavior.

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

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

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

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

Subclause 7.4.4 describes optional behavior.

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

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

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

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

7.4.5 Next single use boot configuration

Subclause 7.4.5 describes optional behavior.

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

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

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

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

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

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

7.4.6 Not next boot configuration

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

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

7.5 Applying the boot configuration

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

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

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

Conditional Requirement:

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

2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 3 (Applies Boot Configuration).

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

7.5.2 Apply boot configuration is not supported

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

Conditional Requirement 1:

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

2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array does not contain a value of 3 (Applies Boot Configuration).

Conditional Requirement 2:

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

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

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

7.6 Creating a boot configuration

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

7.6.1 Creating boot configuration is supported

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

Conditional Requirement:

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

2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 2 (Creates Boot Configuration).

The implementation shall support the CreateBootConfigSetting() method.
7.6.2 Creating boot configuration is not supported

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

Conditional Requirement 1:

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

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

Conditional Requirement 2:

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

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

7.7 Deleting a boot configuration

Subclause 7.7 describes conditional behavior.

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

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

7.8 Identifying boot sources

Subclause 7.8 describes optional behavior.

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

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

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

7.8.1 CIM_BootServiceCapabilities

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

7.8.1.1 CIM_BootServiceCapabilities.BootStringsSupported

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

The presence of a value in the property array means that the specified boot string in each instance of CIM_BootSourceSettings which are associated to an instance of CIM_BootConfigSetting, which in turn is associated to the CIM_BootService, shall not be NULL.
The CIM_BootSourceSetting.ElementName property shall be a character string of variable length (pattern ".*.").

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

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

The CIM_BootSourceSetting.BootString property shall contain a character string.

The CIM_BootSourceSetting.BIOSBootString property may be NULL.

An implementation may support the CIM_BootSourceSetting.BIOSBootString property.

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

Conditional Requirement:

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

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

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

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

Conditional Requirement 1:

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

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

Conditional Requirement 2:

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

The CIM_BootSourceSetting.BOOTString property may be NULL.

An implementation may support the CIM_BootSourceSetting.BIOSBootString property.
7.8.4.1  **CIM_BootSourceSetting.BIOSBootString property is supported**

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

Conditional Requirement:
1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance through an instance of CIM_ElementCapabilities.
2) The CIM_BootServiceCapabilities.BootStringsSupported property array contains a value of 3 (BIOSBootString).

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

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

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

7.8.4.2  **CIM_BootSourceSetting.BIOSBootString property is not supported**

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

Conditional Requirement 1:
1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance through an instance of CIM_ElementCapabilities.
2) The CIM_BootServiceCapabilities.BootStringsSupported property array does not contain a value of 3 (BIOSBootString).

Conditional Requirement 2:
1) An instance of CIM_BootServiceCapabilities is not associated with the CIM_BootService instance through an instance of CIM_ElementCapabilities.

The CIM_BootSourceSetting.BIOSBootString property may be NULL.

7.8.5  **CIM_BootSourceSetting.StructuredBootString property**

An implementation should support the CIM_BootSourceSetting.StructuredBootString property.

7.8.5.1  **CIM_BootSourceSetting.StructuredBootString property is supported**

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

Conditional Requirement:
1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance through an instance of CIM_ElementCapabilities.
2) The CIM_BootServiceCapabilities.BootStringsSupported property array contains a value of 4 (StructuredBootString).

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

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

The value of the CIM_BootSourceSetting.StructuredBootString for instances of CIM_BootSourceSetting that associate to the same CIM_LogicalDevice should have the same OrgId,Identifier, and Index.

The value of the CIM_BootSourceSetting.StructuredBootString should be unique so that this string can be used as a reference.

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

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Unknown&quot;</td>
<td>The boot device type is unknown</td>
</tr>
<tr>
<td>&quot;Floppy&quot;</td>
<td>Boot from a floppy device</td>
</tr>
<tr>
<td>&quot;Hard-Disk&quot;</td>
<td>Boot from a hard drive device</td>
</tr>
<tr>
<td>&quot;CD/DVD&quot;</td>
<td>Boot from a CD or DVD device</td>
</tr>
<tr>
<td>&quot;Network&quot;</td>
<td>Boot from a network device</td>
</tr>
<tr>
<td>&quot;PCMCIA&quot;</td>
<td>Boot from a PCMCIA device</td>
</tr>
<tr>
<td>&quot;BEV&quot;</td>
<td>Boot from a Boot Entry Vector device</td>
</tr>
<tr>
<td>&quot;USB&quot;</td>
<td>Boot from a USB device</td>
</tr>
</tbody>
</table>

### 7.8.5.2 CIM_BootSourceSetting.StructuredBootString property is not supported

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

**Conditional Requirement 1:**

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

2) The CIM_BootServiceCapabilities.BootStringsSupported property array does not contain a value of 4 (StructuredBootString).

**Conditional Requirement 2:**

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

The CIM_BootSourceSetting.StructuredBootString property may be NULL.

### 7.8.6 CIM_ConcreteDependency association

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

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

When the association is used in this manner, its Antecedent property shall reference the instance of a concrete subclass of CIM_LogicalDevice and its Dependent property shall reference the CIM_BootSourceSetting instance.
7.9 Changing the boot order

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

7.9.1 Changing boot order is supported

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

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

Conditional Requirement 2:
1) An instance of CIM_BootServiceCapabilities is not associated with the CIM_BootService instance through an instance of CIM_ElementCapabilities.

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

7.9.2 Changing boot order is not supported

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

Conditional Requirement:
1) An instance of CIM_BootServiceCapabilities is associated with the CIM_BootService instance through an instance of CIM_ElementCapabilities.
2) The CIM_BootServiceCapabilities.BootConfigCapabilities property array contains a value of 6 (Change Boot Order Not Supported).

The implementation shall not support the ChangeBootOrder() method.

7.10 Representing a set of aggregated boot sources

Subclause 7.10 describes optional behavior.

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

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

7.10.1 Aggregated boot sources

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

7.10.2 Aggregated boot configuration

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

7.10.3 Logical identity relationship

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

7.10.3.1 CIM_LogicalIdentity.SystemElement

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

7.10.3.2 CIM_LogicalIdentity.SameElement

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

7.11 Boot order during the boot process

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

7.11.1 CIM_OrderedComponent association

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

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

7.11.1.1 CIM_OrderedComponent.AssignedSequence property

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

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

The boot order shall be interpreted as follows:

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

7.11.2 CIM_BootSourceSetting.FailThroughSupported

The FailThroughSupported property shall describe the behavior of the boot process when the attempt to boot from a boot device represented by an instance of CIM_BootSourceSetting is not successful.
When the FailThroughSupported property has a value of 1 (Is Supported), an unsuccessful boot attempt shall result in continuing through the ordered list for boot sources from which to attempt to boot.

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

### 7.12 Settings to apply during the boot process

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

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

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

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

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

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

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

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

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

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

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

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

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

When the CIM_ConcreteComponent association is used in this manner, its GroupComponent property shall reference the instance of a concrete subclass of CIM_SettingData.
8 Methods

8.1 CIM_BootService.CreateBootConfigSetting()

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

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

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

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

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

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

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

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

- A new instance of CIM_BootConfigSetting shall exist and be referenced by the output NewBootConfig parameter. The new CIM_BootConfigSetting.InstanceID property shall be set to a unique value.
- A new instance of CIM_ElementSettingData shall exist that associates the new CIM_BootConfigSetting to the instance of the Boot Configurable System, which is specified by the ScopingComputerSystem parameter when it is non-NULL or implied by the StartingBootConfig parameter when the ScopingComputerSystem parameter is NULL.
• New instances of CIM_BootSourceSetting shall exist, along with instances of CIM_OrderedComponent, when they are present in the boot configuration represented by the Template Boot Configuration. The new instances shall be duplicates of those found in the boot configuration represented by the Template Boot Configuration, except for the key property value.

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

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

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

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

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

No standard messages are defined.

### Table 3 – CreateBootConfigSetting() Method: Return code values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Completed with no error</td>
</tr>
<tr>
<td>1</td>
<td>Not supported</td>
</tr>
<tr>
<td>2</td>
<td>Error occurred</td>
</tr>
<tr>
<td>4096</td>
<td>Job started</td>
</tr>
</tbody>
</table>

### Table 4 – CreateBootConfigSetting() Method: Parameters

<table>
<thead>
<tr>
<th>Qualifiers</th>
<th>Name</th>
<th>Type</th>
<th>Description/Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>ScopingComputerSystem</td>
<td>CIM_ComputerSystem REF</td>
<td>Reference to an existing CIM_ComputerSystem instance</td>
</tr>
<tr>
<td>IN, REQ</td>
<td>StartingBootConfig</td>
<td>CIM_BootConfigSetting REF</td>
<td>Reference to an existing CIM_BootConfigSetting instance</td>
</tr>
<tr>
<td>OUT</td>
<td>NewBootConfig</td>
<td>CIM_BootConfigSetting REF</td>
<td>Reference to the newly created CIM_BootConfigSetting</td>
</tr>
<tr>
<td>OUT</td>
<td>Job</td>
<td>CIM_ConcreteJob REF</td>
<td>Reference to a CIM_ConcreteJob returned if job started</td>
</tr>
</tbody>
</table>

### 8.2 CIM_BootService.ApplyBootConfigSetting()

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

NOTE Successful execution of the ApplyBootConfigSetting() method can independently apply a boot configuration to a Boot Configurable System regardless of the Next Boot Configuration. The requirements in subclause 7.4.4 shall not apply when a Boot Configurable System is booted using the ApplyBootConfigSetting() method.
The ApplyBootConfigSetting() method shall start the boot process on a specified Boot Configurable System, using the specified boot configuration of the Boot Configurable System. The boot process may be started from a pause in the boot flow or from a reboot of the Boot Configurable System. The method has two input parameters, ScopingComputerSystem and ApplyBootConfig.

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

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

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

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

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

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

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

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

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

No standard messages are defined.

Table 5 – ApplyBootConfigSetting() Method: Return code values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Completed with no error</td>
</tr>
<tr>
<td>1</td>
<td>Not supported</td>
</tr>
<tr>
<td>2</td>
<td>Error occurred</td>
</tr>
<tr>
<td>4096</td>
<td>Job started</td>
</tr>
</tbody>
</table>

Table 6 – ApplyBootConfigSetting() Method: Parameters

<table>
<thead>
<tr>
<th>Qualifiers</th>
<th>Name</th>
<th>Type</th>
<th>Description/Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN, REQ</td>
<td>ScopingComputerSystem</td>
<td>CIM_ComputerSystem REF</td>
<td>Reference to an existing CIM_ComputerSystem instance</td>
</tr>
<tr>
<td>IN, REQ</td>
<td>ApplyBootConfig</td>
<td>CIM_BootConfigSetting REF</td>
<td>Reference to an existing CIM_BootConfigSetting instance</td>
</tr>
<tr>
<td>OUT</td>
<td>Job</td>
<td>CIM_ConcreteJob REF</td>
<td>Reference to a CIM_ConcreteJob returned if job started</td>
</tr>
</tbody>
</table>
8.3 **CIM_BootConfigSetting.ChangeBootOrder()**

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

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

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

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

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

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

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

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

No standard messages are defined.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Completed with No Error</td>
</tr>
<tr>
<td>1</td>
<td>Not Supported</td>
</tr>
<tr>
<td>2</td>
<td>Error Occurred</td>
</tr>
<tr>
<td>4096</td>
<td>Job Started</td>
</tr>
</tbody>
</table>

### Table 8 – ChangeBootOrder() Method: Parameters

<table>
<thead>
<tr>
<th>Qualifiers</th>
<th>Name</th>
<th>Type</th>
<th>Description/Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN, REQ</td>
<td>Source[]</td>
<td>CIM_BootSourceSetting REF</td>
<td>An ordered array of references to CIM_BootSourceSetting instances</td>
</tr>
<tr>
<td>OUT</td>
<td>Job</td>
<td>CIM_ConcreteJob REF</td>
<td>Reference to a CIM_ConcreteJob Returned if job started.</td>
</tr>
</tbody>
</table>

8.4 **Profile conventions for operations**

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


8.5 CIM_BootService

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

Table 9 – Operations: CIM_BootService

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>ModifyInstance</td>
<td>Optional</td>
<td>None</td>
</tr>
<tr>
<td>Associators</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>AssociatorNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>References</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>ReferenceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>

8.5.1 CIM_BootService — ModifyInstance operation

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

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

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

8.6 CIM_BootConfigSetting

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

Table 10 – Operations: CIM_BootConfigSetting

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteInstance</td>
<td>Conditional</td>
<td>None</td>
</tr>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>Associators</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>AssociatorNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>References</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>ReferenceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>
8.6.1 CIM_BootConfigSetting – DeleteInstance

Subclause 8.6.1 describes conditional behavior.

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

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

- The target instance of CIM_BootConfigSetting shall no longer exist.
- The instance of CIM_ElementSettingData that associated the target CIM_BootConfigSetting to the instance of CIM_ComputerSystem shall no longer exist.
- The instances of CIM_ConcreteComponent, which associate the target instance of CIM_BootConfigSetting to instances of a concrete subclass of CIM_SettingData, shall no longer exist.
- The instances of CIM_ConcreteComponent, which associate the target instance of CIM_BootConfigSetting to instances of a concrete subclass of CIM_BootSettingData, shall no longer exist. The instances of the associated concrete subclass of CIM_BootSettingData shall no longer exist.
- The instances of CIM_OrderedComponent, which associate the target instance of CIM_BootConfigSetting to instances of CIM_BootSourceSetting, shall no longer exist. The instances of the associated CIM_BootSourceSetting shall no longer exist.
- The instances of CIM_ConcreteComponent, which associate instances of a concrete subclass of CIM_SettingData to instances of CIM_BootSourceSetting, which in turn are associated to the target instance of CIM_BootConfigSetting, shall no longer exist.
- The instances of CIM_ConcreteComponent, which associate instances of a concrete subclass of CIM_BootSettingData to instances of CIM_BootSourceSetting, which in turn are associated to the target instance of CIM_BootConfigSetting, shall no longer exist. The instances of the associated concrete subclass of CIM_BootSettingData shall no longer exist.
- The instances of CIM_ConcreteDependency, which associate instances of a concrete subclass of CIM_LogicalDevice to instances of CIM_BootSourceSetting, shall no longer exist.
- The instance of CIM_LogicalIdentity, which associates a deleted instance of CIM_BootSourceSetting to an instance of CIM_BootConfigSetting, shall no longer exist. The associated instance of CIM_BootConfigSetting shall no longer exist. The requirements in this subclause shall be applied recursively to the deleted CIM_BootConfigSetting instance.

8.7 CIM_BootSettingData

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

Table 11 – Operations: CIM_BootSettingData

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>Associators</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>AssociatorNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>
8.8 CIM_BootSourceSetting

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

Table 12 – Operations: CIM_BootSourceSetting

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>Associators</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>AssociatorNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>References</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>ReferenceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>

8.9 CIM_ConcreteComponent

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

Table 13 – Operations: CIM_ConcreteComponent

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>

8.10 CIM_ConcreteDependency

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

Table 14 – Operations: CIM_ConcreteDependency

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>
8.11 CIM_ElementCapabilities

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 15 – Operations: CIM_ElementCapabilities

8.12 CIM_ElementSettingData

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>ModifyInstance</td>
<td>Optional</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 16 – Operations: CIM_ElementSettingData

8.12.1 CIM_ElementSettingData – ModifyInstance operation

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

8.12.1.1 CIM_ElementSettingData.IsDefault property

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

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

- Search for an instance of CIM_ElementSettingData that associates an instance of CIM_BootConfigSetting with the instance of CIM_ComputerSystem, which is referenced by the target instance of CIM_ElementSettingData where the IsDefault property has a value of 1 (Is Default).
- If such an instance of CIM_ElementSettingData is found, the ModifyInstance operation shall set the value of the IsDefault property to 2 (Is Not Default).
- For the target instance of CIM_ElementSettingData, when the IsDefault property already has a value of 1 (Is Default), the ModifyInstance operation shall complete successfully.
- For the target instance of CIM_ElementSettingData, set the value of the IsDefault property to 1 (Is Default).
8.12.1.2 CIM_ElementSettingData.IsNext property

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

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

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

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

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

- For the target instance of CIM_ElementSettingData, when the IsNext property has a value of 1 (Is Next), the ModifyInstance operation shall fail.
- Search for an instance of CIM_ElementSettingData that associates an instance of CIM_BootConfigSetting with the instance of CIM_ComputerSystem referenced by the target instance of CIM_ElementSettingData where the IsNext property has a value of 3 (Is Next For Single Use).
- If such an instance of CIM_ElementSettingData is found, the ModifyInstance operation shall set the value of the IsNext property to 2 (Is Not Next).
- For the target instance of CIM_ElementSettingData, when the IsNext property already has a value of 3 (Is Next For Single Use), the ModifyInstance operation shall complete successfully.
- For the target instance of CIM_ElementSettingData, when the IsNext property has a value neither 1 (Is Next) nor 3 (Is Next For Single Use), set the value of the IsNext property to 3 (Is Next For Single Use).

8.12.1.3 CIM_ElementSettingData.IsCurrent property

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

8.13 CIM_BootServiceCapabilities

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>Associators</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>AssociatorNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>References</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>ReferenceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>ReferenceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>

8.14 CIM_HostedService

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

Table 18 – Operations: CIM_HostedService

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>

8.15 CIM_LogicalIdentity

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

Table 19 – Operations: CIM_LogicalIdentity

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>

8.16 CIM_OrderedComponent

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

Table 20 – Operations: CIM_OrderedComponent

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>
8.17 CIM_ServiceAffectsElement

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInstance</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstances</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>EnumerateInstanceNames</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 21 – Operations: CIM_ServiceAffectsElement

9  Use cases

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

9.1 Advertising the profile conformance

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

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

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

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

9.2 Object diagram for a monolithic server

Figure 3 shows the CIM instances required to control the boot configuration for a single, monolithic server, system1. System1 hosts the boot service, bootsvc1, which is used to control the boot configuration, bootcfgsetting1, for system1. System1 is also identified as the Boot Configurable System through the CIM_ServiceAffectsElement association. The capabilities of bootsvc1 are defined by bootsvc1cap1.
The boot configuration, `bootcfgsetting1`, has one boot source, `bootsrcsetting1`.

**Figure 3 – Monolithic server object diagram**

### 9.3 Object diagram for a monolithic server with service processor

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

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

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

Each boot configuration (`bootcfgsetting1`, `bootcfgsetting2`) has one boot source (`bootsrcsetting1`, `bootsrcsetting2`), respectively.

**Figure 4 – Monolithic server with service processor object diagram**
9.4 Object diagram for a modular system

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

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

The capabilities of *bootsvc1* and *bootsrv2* are defined by *bootsvccap1* and *bootsvccap2* respectively.

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

Alternative representation of modular blade system boot sources is shown in the Figure 6. The blade computer systems, in this example, have the same boot sources, and thus, the representation of boot sources can be optimized by instantiating only one CIM_BootSourceSetting shared between the respective CIM_BootConfigSetting instances. This optimization is especially useful when modular system contains many blade computer systems with the similar boot sources. Each blade boot configuration (*bootcfgsetting1*, *bootcfgsetting2*) has one boot source (*bootsrccsetting1*).
Figure 6 – Modular system object diagram

9.5 PXE boot source

Figure 7 shows an instance diagram for a boot configuration, `bootscfgsetting1`, which has two boot sources associated to it, `bootsrcsetting1` and `bootsrcsetting2`, which are both network ports.

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

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

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

Figure 8 shows an instance diagram for a boot configuration, `bootcfgsetting1`, which has a single boot source associated to it, `bootsrcsetting1`, which is a disk device. The CIM_BootSourceSetting.ElementName property identifies "Hard Disk 1" as the boot source, which matches the BIOSBootString property. The BootString property contains the string "C:\", which could be interpreted by the boot process to assign the hard drive the letter "C". The CIM_ConcreteDependency association relates `bootsrcsetting1` to a CIM_LogicalDisk (`disk1`). Because there is only one boot source, the value of the CIM_BootSourceSetting.FailThroughSupported is not meaningful.
9.7 Local CDROM and floppy boot sources

Figure 9 shows an instance diagram for a boot configuration, bootcfgsetting1, which has two boot sources associated to it, bootsrcsetting1 and bootsrcsetting2. BootsRCsetting1 is a CD-ROM device; bootsrcsetting2 is a floppy drive.

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

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

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

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

9.8 Representing IPL and Boot Control Vector (BCV) lists

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

To represent the IPL list, bootcfgsetting1 has three boot sources associated to it, bootsrcsetting1, bootsrcsetting2, and bootsrcsetting3. BootsRCsetting1 is a CD-ROM device. BootsRCsetting2 is a floppy drive. BootsRCsetting3 is a BCV device (boot control vector).

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

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

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

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

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

In total, this use case describes a source boot order that proceeds from bootsrcsetting1 to bootsrcsetting21. Bootsourcing2 will never be used because of its AssignedSequence value of 0 and bootsrcsetting22 will never be used because of the FailThroughSupported value on bootsrcsetting21.

---

**Figure 10 – Booting from IPL and BCV devices**
9.9 Representing settings and boot settings

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

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

![Instance diagram for boot configuration](image)

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

9.10 Representing the same boot device

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

The `bootcfgsetting1` instance represents the next one time boot configuration and is associated to an instance of CIM_BootSourceSetting, `bootsrcsetting1`. The instance `bootsrcsetting1` is associated to the boot device, `netport1`.

There are two different CIM_BootSourceSetting instances `bootsrccsetting1` and `bootsrccsetting2` are associated to the same boot device, `netport1`.

Since both instances of CIM_BootSourceSetting are associated to the same boot device, the StructuredBootString property in each instance is set to the same string ("CIM:Network:1"), namely, both strings have the same OrgId, Identifier, and Index.
9.11 Representing the default boot configuration for a computer system

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

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

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

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

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

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

Bootcfgsetting2 is associated with two instances of CIM_BootSourceSetting (bootsrcsetting1 and bootsrcsetting2), through instances of CIM_OrderedComponent. The respective CIM_OrderedComponent.AssignedSequence properties designate the order in which the boot process should use the boot sources (bootsrscsetting1 followed by bootsrcsetting2).

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

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

When the system represented by system1 is an enabled, but not booted, state. The BootService.ApplyBootConfigSetting() method can be invoked referencing system1 as the BootConfigurableSystem parameter.
9.13 Representing the current boot configuration for a booted computer system

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

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.
9.14 Create a new boot configuration

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

1) From the Boot Configurable System, system1, find the instance of CIM_BootService that manages the boot configurable system by traversing the CIM_ServiceAffectsElement association.

2) Verify that the CreateBootConfigSetting() method is supported (see subclause 9.28). If not, a new boot configuration cannot be created.

3) Find an existing instance of CIM_BootConfigSetting to use as the template. For this use case, bootcfgsetting2 is the only template configuration available.

4) Create the new boot configuration, bootcfgsetting4, by invoking the CIM_BootService.CreateBootConfigSetting() method. The ScopingComputerSystem parameter is set to system1 and the StartingBootConfig parameter is set to bootcfgsetting2.

Figure 15 shows the instance diagram after the CreateBootConfigSetting() method has been invoked and successfully completed on the computer system, system1, shown in Figure 15. The new boot configuration, bootcfgsetting4, is associated to system1 through a new instance of CIM_ElementSettingData.

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

1322 Bootcfgsetting4 is associated through instances of CIM_OrderedComponent to two instances of CIM_BootSourceSetting (bootsrsetting3 and bootsrsetting4), which are copies of bootsrsetting1 and bootsrsetting2, respectively.

1325 The instance of CIM_NetworkPort is not copied. CIM_NetworkPort is a concrete subclass of CIM_LogicalDevice, which is not part of the Boot Control Profile. However, an instance of CIM_ConcreteDependency has been created that associates the instance of CIM_NetworkPort to the new instance of CIM_BootSourceSetting (bootsrsetting4).

1329 CIM_LogicalDisk has been elided from the object diagram to make the diagram less cluttered, but the instance of CIM_LogicalDisk is also not copied. An instance of CIM_ConcreteDependency is created that associates the existing instance of CIM_LogicalDisk to the new instance of CIM_BootSourceSetting (bootsrsetting3).

1333 orderedComponent

OrderedComponent

AssignSequence: 1

AssignSequence: 2

Bootconfigsetting2: BootConfigSetting

Bootconfigsetting4: BootConfigSetting

Bootsrsetting1: BootSourceSetting

ElementName: "Hard Disk 1"
BootString: "Hard Drive C:"

BIOSBootString: "Hard Disk 1"
StructuredBootString: "CIM:Hard-Disk:1"

FailThroughSupported: 1 (Is Supported)

Bootsrsetting3: BootSourceSetting

ElementName: "Hard Disk 1"
BootString: "Hard Drive C:"

BIOSBootString: "Hard Disk 1"
StructuredBootString: "CIM:Hard-Disk:1"

FailThroughSupported: 1 (Is Supported)

ConcreteDependency

netport1: NetworkPort

ConcreteDependency

1334

Figure 16 – System with new CIM_BootConfigSetting

9.15 Apply an existing boot configuration

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

1337 1) Find the instance of CIM_BootService for the boot configurable system as outlined in subclause 9.16.

1339 2) Verify that the ApplyBootConfigSetting() method is supported (see subclause 9.29). If not, a boot configuration cannot be applied.
3) Find the existing instances of CIM_BootConfigSetting for system1 (see subclause 9.19). In this example, this results in bootcfgsetting2. Pick one of them to use as the boot configuration to apply.

4) Apply the selected boot configuration, bootcfgsetting2, by invoking the CIM_BootService.ApplyBootConfigSetting() method. The ScopingComputerSystem parameter is set to system1 and the BootConfigSetting parameter is set to bootcfgsetting2.

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

9.16 Find the boot service for a computer system

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

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

9.17 Find the boot configuration for a computer system

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

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

9.18 Find the boot service capabilities for a computer system

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

1) Find the boot service for the computer system as specified in subclause 9.16.

2) Select the instance of CIM_BootServiceCapabilities through the CIM_ElementCapabilities association.

9.19 Find the current boot configuration for a computer system

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

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

2) Find the instance of CIM_ElementSettingData whose IsCurrent property is set to 1 (Is Current).

3) The CIM_BootConfigSetting instance referenced by this association instance represents the current boot configuration.

9.20 Find the default boot configuration for a computer system

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

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

2) Find the instance of CIM_ElementSettingData whose IsDefault property is set to 1 (Is Default).

3) The CIM_BootConfigSetting instance referenced by this association instance represents the default boot configuration.
9.21 Find the boot configuration that will be used during the next reboot for a computer system

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

1) For the instance of CIM_ComputerSystem, enumerate the CIM_ElementSettingData associations with CIM_BootConfigSetting as the SettingData reference.

2) Find the CIM_ElementSettingData instances for the instance whose IsNext property is set to 3 (Is Next For Single Use). The CIM_BootConfigSetting instance referenced by this association instance represents the next boot configuration.

3) If no instance is found, find the instance of CIM_ElementSettingData whose IsNext property is set to 1 (Is Next). The CIM_BootConfigSetting instance referenced by this association instance represents the next boot configuration.

9.22 Make a boot configuration applicable for subsequent reboots

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

1) From the instance of CIM_ComputerSystem, find the CIM_BootConfigSetting of interest as outlined in subclauses 9.9 through 9.13.

2) On the instance of the CIM_ElementSettingData association that associates the instance of CIM_ComputerSystem to the instance of CIM_BootConfigSetting, use the intrinsic ModifyInstance() to change the IsNext property to 1 (Is Next).

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

9.23 Make a boot configuration applicable for the next reboot only

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

1) From the instance of CIM_ComputerSystem, find the CIM_BootConfigSetting of interest as outlined in subclauses 9.9 through 9.13.

2) On the instance of the CIM_ElementSettingData association that associates the instance of CIM_ComputerSystem to the instance of CIM_BootConfigSetting, use the intrinsic ModifyInstance() to change the IsNext property to 3 (Is Next For Single Use).

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

9.24 Determine whether the computer system supports PXE boot

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

1) For the instance of CIM_ComputerSystem enumerate its instances of CIM_BootConfigSetting as outline in subclause 9.19.

2) For each instance of CIM_BootConfigSetting, enumerate the instances of CIM_BootSourceSetting.

3) For each CIM_BootSourceSetting, inspect the BootString, BIOSBootString, or StructuredBootString property to determine if PXE is supported.

9.25 Find the boot order for a computer system for the next reboot

This use case references the object diagram in Figure 10, which represents a complex boot order.
A client can find the boot order for the next reboot of a computer system as follows:

1) From the instance of CIM_ComputerSystem, system1, find the CIM_BootConfigSetting that will be used during the next reboot, bootcfgsetting1 (see subclause 3).

2) Determine the boot order for bootcfgsetting1 by enumerating the CIM_OrderedComponent associations with bootcfgsetting1 as the GroupComponent reference. The results in this example would be bootsrcsetting1, bootsrcsetting2 and bootsrcsetting3.

3) Use the CIM_OrderedComponent.AssignedSequence property to determine the boot order. The boot order in this example will be bootsrcsetting1 followed by bootsrcsetting3.

4) For each boot source, determine whether any it contains additional boot sources by checking for a CIM_LogicalIdentity association to an instance of CIM_BootConfigSetting; in this example, bootcfgsetting20, and repeat steps in this subclause recursively to find the boot order of the associated boot sources.

9.26 Change the boot order for a computer system

This use case references the object diagram in Figure 14.

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

1) Find the boot configuration of interest from the set of boot configurations for the computer system as outlined in subclause 9.19.

2) Find the set of boot sources for the boot configuration by following the OrderedComponent associations from the selected boot configuration representation (bootcfgsetting2) to all instances of CIM_BootSourceSetting. In this example, this results in bootsrcsetting1 and bootsrcsetting2.

3) Determine the desired boot order.

4) Create an array of CIM_BootSourceSetting references. Assign the existing boot sources to the array in the new order. For instance, if one wanted to reverse the boot order in this example, the array would contain bootsrcsetting2 at index 0 and bootsrcsetting1 at index 1.

5) Invoke the ChangeBootOrder() method on the selected instance of CIM_BootConfigSetting. The Source parameter is set to the array created above.

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

9.27 Determine whether BootService.ElementName is modifiable

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

1) Find the CIM_BootServiceCapabilities instance associated with the CIM_BootService instance through the CIM_ElementCapabilities association.

2) If a CIM_BootConfigCapabilities instance cannot be found, then the CIM_BootService.ElementName property cannot be modified.

3) Query the value of the CIM_BootServiceCapabilities.ElementNameEditSupported.

4) If the value is TRUE, the CIM_BootService.ElementName property can be modified.

5) If the value of ElementNameEditSupported has a value of FALSE, then the CIM_BootService.ElementName property cannot be modified.
9.28 Determine whether a new boot configuration can be created

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

1) Find the CIM_BootServiceCapabilities instance that is associated with the CIM_BootService instance through the CIM_ElementCapabilities association.

2) Query the value of the CIM_BootServiceCapabilities.BootConfigCapabilities property array. If the array contains the value 2 (Creates Boot Configuration), the client's ability to create a boot configuration is supported.

3) If the BootConfigCapabilities property array does not contain the value 2 (Creates Boot Configuration), or there is not an instance of CIM_BootServiceCapabilities associated with the CIM_BootService instance, a boot configuration cannot be created.

9.29 Determine whether a boot configuration can be applied

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

1) Find the CIM_BootServiceCapabilities instance that is associated with the CIM_BootService instance through the CIM_ElementCapabilities association.

2) Query the value of the CIM_BootServiceCapabilities.BootConfigCapabilities property array. If the array contains the value 3 (Applies Boot Configuration), the client's ability to manually apply a boot configuration is supported.

3) If the BootConfigCapabilities property array does not contain the value 3 (Applies Boot Configuration), or there is not an instance of CIM_BootServiceCapabilities associated with the CIM_BootService instance, a boot configuration cannot be manually applied.

10 CIM Elements

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

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

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

Table 23 – Class: CIM_RegisteredProfile

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegisteredName</td>
<td>Mandatory</td>
<td>This property shall have a value of &quot;Boot Control&quot;.</td>
</tr>
<tr>
<td>RegisteredVersion</td>
<td>Mandatory</td>
<td>This property shall have a value of &quot;1.0.1&quot;.</td>
</tr>
<tr>
<td>RegisteredOrganization</td>
<td>Mandatory</td>
<td>This property shall have a value of 2 (DMTF).</td>
</tr>
</tbody>
</table>

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

10.2 CIM_BootService

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

Table 24 – Class: CIM_BootService

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>Name</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>SystemName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>ElementName</td>
<td>Mandatory</td>
<td>See subclause 7.1.1.</td>
</tr>
<tr>
<td>CreateBootConfigSetting()</td>
<td>Conditional</td>
<td>See subclause 8.1.</td>
</tr>
<tr>
<td>ApplyBootConfigSetting()</td>
<td>Conditional</td>
<td>See subclause 8.2.</td>
</tr>
</tbody>
</table>

10.3 CIM_BootServiceCapabilities

Support of the CIM_BootServiceCapabilities class is optional.

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

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceID</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>ElementName</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>ElementNameEditSupported</td>
<td>Mandatory</td>
<td>See subclause 7.1.2</td>
</tr>
<tr>
<td>BootConfigCapabilities</td>
<td>Mandatory</td>
<td>See subclauses 7.5, 7.6, and 7.7.</td>
</tr>
<tr>
<td>OtherBootConfigCapabilities</td>
<td>Conditional</td>
<td>See subclause 7.3.1.</td>
</tr>
<tr>
<td>BootStringsSupported</td>
<td>Optional</td>
<td>See subclause 7.8.</td>
</tr>
</tbody>
</table>

10.4 CIM_BootConfigSetting

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

Table 26 – Class: CIM_BootConfigSetting

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceID</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>ElementName</td>
<td>Mandatory</td>
<td>This property shall be a character string of variable length (pattern &quot;.*&quot;).</td>
</tr>
<tr>
<td>ChangeBootOrder()</td>
<td>Conditional</td>
<td>See subclause 7.9 and 8.3.</td>
</tr>
</tbody>
</table>

10.5 CIM_BootSettingData

Support of the CIM_BootSettingData class is optional.

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

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

Table 27 – Class: CIM_BootSettingData

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceID</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>ElementName</td>
<td>Mandatory</td>
<td>This property shall be a character string of variable length (pattern &quot;.*&quot;).</td>
</tr>
<tr>
<td>OwningEntity</td>
<td>Mandatory</td>
<td>None</td>
</tr>
</tbody>
</table>

10.6 CIM_BootSourceSetting

Support of the CIM_BootSourceSetting class is mandatory.

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

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceID</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>ElementName</td>
<td>Mandatory</td>
<td>See subclause 7.8.2.</td>
</tr>
<tr>
<td>BootString</td>
<td>Conditional</td>
<td>See subclause 7.8.3.</td>
</tr>
<tr>
<td>BIOSBootString</td>
<td>Conditional</td>
<td>See subclause 7.8.4.</td>
</tr>
<tr>
<td>StructuredBootString</td>
<td>Conditional</td>
<td>See subclause 7.8.5.</td>
</tr>
<tr>
<td>FailThroughSupported</td>
<td>Mandatory</td>
<td>See subclause 7.11.2.</td>
</tr>
</tbody>
</table>

10.7 CIM_ConcreteComponent

Subclause 10.7 describes optional behavior.

10.7.1 Relating CIM_BootConfigSetting to a concrete subclass of CIM_SettingData

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

Table 29 – Class: CIM_ConcreteComponent – Use 1

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupComponent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootConfigSetting class. See subclause 7.12. Cardinality is &quot;*&quot;.</td>
</tr>
<tr>
<td>PartComponent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of a concrete subclass of the CIM_SettingData class. See subclause 7.12. Cardinality is &quot;*&quot;.</td>
</tr>
</tbody>
</table>

10.7.2 Relating CIM_BootConfigSetting to a concrete subclass of CIM_BootSettingData

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

Table 30 – Class: CIM_ConcreteComponent – Use 2

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupComponent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootConfigSetting class. See subclause 7.12. Cardinality is &quot;0..1&quot;.</td>
</tr>
<tr>
<td>PartComponent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of a concrete subclass of the CIM_BootSettingData class. See subclause 7.12. Cardinality is &quot;*&quot;.</td>
</tr>
</tbody>
</table>
10.7.3 Relating CIM_BootSourceSetting to a concrete subclass of CIM_SettingData

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

Table 31 – Class: CIM_ConcreteComponent – Use 3

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupComponent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootSourceSetting class. See subclause 7.12. Cardinality is &quot;***&quot;.</td>
</tr>
<tr>
<td>PartComponent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of a concrete subclass of the CIM_SettingData class. See subclause 7.12. Cardinality is &quot;***&quot;.</td>
</tr>
</tbody>
</table>

10.7.4 Relating CIM_BootSourceSetting to a concrete subclass of CIM_BootSettingData

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

Table 32 – Class: CIM_ConcreteComponent – Use 4

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupComponent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootSourceSetting class. See subclause 7.12. Cardinality is &quot;0..1&quot;.</td>
</tr>
<tr>
<td>PartComponent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of a concrete subclass of the CIM_BootSettingData class. See subclause 7.12. Cardinality is &quot;***&quot;.</td>
</tr>
</tbody>
</table>

10.8 CIM_ConcreteDependency

Subclause 10.8 describes optional behavior.

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

Table 33 – Class: CIM_ConcreteDependency

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>Mandatory</td>
<td>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 &quot;***&quot;.</td>
</tr>
<tr>
<td>Dependent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootSourceSetting. See subclause 7.8.5.2. Cardinality is &quot;***&quot;.</td>
</tr>
</tbody>
</table>
10.9 CIM_ElementCapabilities

Subclause 10.9 describes optional behavior. When supported, the CIM_ElementCapabilities association is used to relate an instance of CIM_BootServiceCapabilities with an instance of CIM_BootService. Table 34 contains the requirements for elements of this class.

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedElement</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of CIM_BootService. See subclause 7.1. Cardinality is &quot;1..*&quot;.</td>
</tr>
<tr>
<td>Capabilities</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of CIM_BootServiceCapabilities. See subclause 7.1. Cardinality is &quot;0..1&quot;.</td>
</tr>
</tbody>
</table>

10.10 CIM_ElementSettingData

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

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedElement</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_ComputerSystem class. See subclause 7.4.1. Cardinality is &quot;0..1&quot;.</td>
</tr>
<tr>
<td>SettingData</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootConfigSetting class. See subclause 7.4.1. Cardinality is &quot;***&quot;.</td>
</tr>
<tr>
<td>IsDefault</td>
<td>Mandatory</td>
<td>See subclause 7.4.</td>
</tr>
<tr>
<td>IsCurrent</td>
<td>Mandatory</td>
<td>See subclause 7.4.</td>
</tr>
<tr>
<td>IsNext</td>
<td>Mandatory</td>
<td>See subclause 7.4.</td>
</tr>
</tbody>
</table>

10.11 CIM_HostedService

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

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>Mandatory</td>
<td>This property shall be a reference to the scoping instance of the CIM_ComputerSystem class. See subclause 5. Cardinality is &quot;1&quot;.</td>
</tr>
<tr>
<td>Dependent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootService. See subclause 5. Cardinality is &quot;***&quot;.</td>
</tr>
</tbody>
</table>
10.12 CIM_LogicalIdentity

Support of the CIM_LogicalIdentity association is conditional.

Conditional Requirement: The support is required if instances of CIM_BootSourceSetting are used to represent aggregated boot sources; see subclause 7.10.

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

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemElement</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootSourceSetting class. See subclause 7.10.3. Cardinality is &quot;0..1&quot;</td>
</tr>
<tr>
<td>SameElement</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootConfigSetting class. See subclause 7.10.3. Cardinality is &quot;0..1&quot;</td>
</tr>
</tbody>
</table>

10.13 CIM_OrderedComponent

Support of the CIM_OrderedComponent association is mandatory.

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

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupComponent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootConfigSetting class. See subclause 7.11.1. Cardinality is &quot;1..*&quot;.</td>
</tr>
<tr>
<td>PartComponent</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootSourceSetting class. See subclause 7.11.1. Cardinality is &quot;1..*&quot;.</td>
</tr>
<tr>
<td>AssignedSequence</td>
<td>Mandatory</td>
<td>See subclause 7.11.1.1.</td>
</tr>
</tbody>
</table>
10.14 CIM_ServiceAffectsElement

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

Table 39 – Class: CIM_ServiceAffectsElement

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AffectingElement</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_BootService class. See subclause 7.2. Cardinality is &quot;0..1&quot;.</td>
</tr>
<tr>
<td>AffectedElement</td>
<td>Mandatory</td>
<td>This property shall be a reference to an instance of the CIM_ComputerSystem class. See subclause 7.2. Cardinality is &quot;1..***&quot;.</td>
</tr>
<tr>
<td>ElementEffects</td>
<td>Mandatory</td>
<td>Matches 5 (Manages)</td>
</tr>
</tbody>
</table>
ANNEX A
(informative)

Change log

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