



1

2

3

4

**Document Number: DSP1019**

**Date: 2009-06-10**

**Version: 1.0.0**

5 **Device Tray Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

9

## 10 Copyright Notice

11 Copyright © 2006, 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

32

# CONTENTS

33 Foreword ..... 5

34 Introduction ..... 6

35 1 Scope ..... 7

36 2 Normative References..... 7

37 2.1 Approved References ..... 7

38 2.2 Other References..... 7

39 3 Terms and Definitions..... 7

40 4 Symbols and Abbreviated Terms..... 9

41 5 Synopsis ..... 9

42 6 Description ..... 9

43 7 Implementation..... 10

44 7.1 CIM\_LogicalModule ..... 10

45 7.2 Managing the Device Tray's State ..... 10

46 7.3 State Management Supported..... 11

47 7.4 ElementName Is Modifiable ..... 11

48 7.5 Physical Asset Profile ..... 12

49 7.6 Shared Device Management Profile ..... 12

50 7.7 Aggregated Devices Modeled..... 12

51 8 Methods..... 13

52 8.1 Method: CIM\_LogicalModule.RequestStateChange() ..... 13

53 8.2 Profile Conventions for Operations ..... 14

54 8.3 CIM\_ConcreteComponent ..... 14

55 8.4 CIM\_ElementCapabilities ..... 15

56 8.5 CIM\_EnabledLogicalElementCapabilities..... 15

57 8.6 CIM\_LogicalModule ..... 15

58 8.7 CIM\_SystemDevice ..... 16

59 9 Use Cases ..... 16

60 9.1 Object Diagrams ..... 16

61 9.2 Determine Aggregated Devices..... 18

62 9.3 Find FRU Information..... 18

63 10 CIM Elements ..... 19

64 10.1 CIM\_ConcreteComponent ..... 19

65 10.2 CIM\_ElementCapabilities ..... 19

66 10.3 CIM\_EnabledLogicalElementCapabilities..... 20

67 10.4 CIM\_LogicalModule ..... 20

68 10.5 CIM\_RegisteredProfile..... 20

69 10.6 CIM\_SystemDevice ..... 21

70 ANNEX A (informative) Change Log ..... 22

71

## 72 Figures

73 Figure 1 – Device Tray Profile: Class Diagram..... 10

74 Figure 2 – Device Tray Object Diagram..... 17

75 Figure 3 – Registered Profile ..... 18

76

## 77 Tables

|    |   |    |
|----|---|----|
| 78 | Table 1 – Referenced Profiles .....   | 9  |
| 79 | Table 2 – CIM_LogicalModule.RequestStateChange() Method: Return Code Values ..... | 13 |
| 80 | Table 3 – CIM_LogicalModule.RequestStateChange() Method: Parameters .....         | 14 |
| 81 | Table 4 – Operations: CIM_ConcreteComponent .....                                 | 14 |
| 82 | Table 5 – Operations: CIM_ElementCapabilities .....                               | 15 |
| 83 | Table 6 – Operations: CIM_EnabledLogicalElementCapabilities .....                 | 15 |
| 84 | Table 7 – Operations: CIM_LogicalModule.....                                      | 15 |
| 85 | Table 8 – Operations: CIM_SystemDevice.....                                       | 16 |
| 86 | Table 9 – CIM Elements: Device Tray Profile .....                                 | 19 |
| 87 | Table 10 – Class: CIM_ConcreteComponent .....                                     | 19 |
| 88 | Table 11 – Class: CIM_ElementCapabilities.....                                    | 19 |
| 89 | Table 12 – Class: CIM_EnabledLogicalElementCapabilities.....                      | 20 |
| 90 | Table 13 – Class: CIM_LogicalModule .....   | 20 |
| 91 | Table 14 – Class: CIM_RegisteredProfile.....                                      | 21 |
| 92 | Table 15 – Class: CIM_SystemDevice .....  | 21 |
| 93 |   |    |

94

## Foreword

95 The *Device Tray Profile* (DSP1019) was prepared by the Physical Platform Profiles Working Group and  
96 the Server Management Working Group of the DMTF.

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

## 99 **Acknowledgements**

100 The authors wish to acknowledge the following people.

101 Editor:

- 102 • Aaron Merkin – IBM

103 Participants from the DMTF Server Management Working Group:

- 104 • Jon Hass – Dell
- 105 • Khachatur Papanyan – Dell
- 106 • Enoch Suen – Dell
- 107 • Jeff Hilland – HP
- 108 • Christina Shaw – HP
- 109 • Aaron Merkin – IBM
- 110 • Jeff Lynch – IBM
- 111 • Perry Vincent – Intel
- 112 • John Leung – Intel

113

114

## Introduction

115 The information in this specification should be sufficient for a provider or consumer of this data to identify  
116 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to  
117 represent and manage a device tray modeled using the DMTF Common Information Model (CIM) core  
118 and extended model definitions (see [DSP0004](#)). The target audience for this specification is implementers  
119 who are writing CIM based providers or consumers of management interfaces representing the  
120 component described in this document.

121

# Device Tray Profile

## 122 1 Scope

123 The *Device Tray Profile* is a component profile for modeling a device tray of a modular system.

## 124 2 Normative References

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

### 128 2.1 Approved References

129 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,  
130 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.5.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.5.0.pdf)

131 DMTF DSP0200, *CIM Operations over HTTP 1.2*,  
132 [http://www.dmtf.org/standards/published\\_documents/DSP200.pdf](http://www.dmtf.org/standards/published_documents/DSP200.pdf)

133 DMTF DSP0207, *WBEM URI Mapping Specification 1.0*,  
134 [http://www.dmtf.org/standards/published\\_documents/DSP0207.pdf](http://www.dmtf.org/standards/published_documents/DSP0207.pdf)

135 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,  
136 [http://www.dmtf.org/standards/published\\_documents/DSP1001.pdf](http://www.dmtf.org/standards/published_documents/DSP1001.pdf)

137 DMTF DSP1011, *Physical Asset Profile 1.0*,  
138 [http://www.dmtf.org/standards/published\\_documents/DSP1011\\_1.0.1.pdf](http://www.dmtf.org/standards/published_documents/DSP1011_1.0.1.pdf)

139 DMTF DSP1021, *Shared Device Management Profile 1.0*,  
140 [http://www.dmtf.org/standards/published\\_documents/DSP1021\\_1.0.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1021_1.0.0.pdf)

141 DMTF DSP1033, *Profile Registration Profile 1.0*,  
142 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.0.pdf)

### 143 2.2 Other References

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

## 146 3 Terms and Definitions

147 For the purposes of this document, the following terms and definitions apply.

### 148 3.1

#### 149 **can**

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

### 151 3.2

#### 152 **cannot**

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

- 154 **3.3**  
155 **conditional**  
156 indicates requirements to be followed strictly in order to conform to the document when the specified  
157 conditions are met
- 158 **3.4**  
159 **mandatory**  
160 indicates requirements to be followed strictly in order to conform to the document and from which no  
161 deviation is permitted
- 162 **3.5**  
163 **may**  
164 indicates a course of action permissible within the limits of the document
- 165 **3.6**  
166 **need not**  
167 indicates a course of action permissible within the limits of the document
- 168 **3.7**  
169 **optional**  
170 indicates a course of action permissible within the limits of the document
- 171 **3.8**  
172 **referencing profile**  
173 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
174 "Related Profiles" table
- 175 **3.9**  
176 **shall**  
177 indicates requirements to be followed strictly in order to conform to the document and from which no  
178 deviation is permitted
- 179 **3.10**  
180 **shall not**  
181 indicates requirements to be followed strictly in order to conform to the document and from which no  
182 deviation is permitted
- 183 **3.11**  
184 **should**  
185 indicates that among several possibilities, one is recommended as particularly suitable, without  
186 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 187 **3.12**  
188 **should not**  
189 indicates that a certain possibility or course of action is deprecated but not prohibited
- 190 **3.13**  
191 **unspecified**  
192 indicates that this profile does not define any constraints for the referenced CIM element or operation



## 193 4 Symbols and Abbreviated Terms

194 The following symbols and abbreviations are used in this document.

### 195 4.1

#### 196 CIM

197 Common Information Model

## 198 5 Synopsis

199 **Profile Name:** Device Tray

200 **Version:** 1.0.0

201 **Organization:** DMTF

202 **CIM Schema Version:** 2.22

203 **Central Class:** CIM\_Processor

204 **Scoping Class:** CIM\_ComputerSystem

205 The *Device Tray Profile* defines the management and modeling of a device tray.

206

**Table 1 – Referenced Profiles**

| Profile Name                             | Organization | Version | Description        |
|--|--------------|---------|--------------------|
| <a href="#">Profile Registration</a>     | DMTF         | 1.0     | Mandatory          |
| <a href="#">Physical Asset</a>           | DMTF         | 1.0     | Optional. See 7.5. |
| <a href="#">Shared Device Management</a> | DMTF         | 1.0     | Optional. See 7.6. |

### 207 5.1.1 Central Instance

208 CIM\_LogicalModule shall be the central class of this profile. The instance of CIM\_LogicalModule shall be  
209 the central instance of this profile.

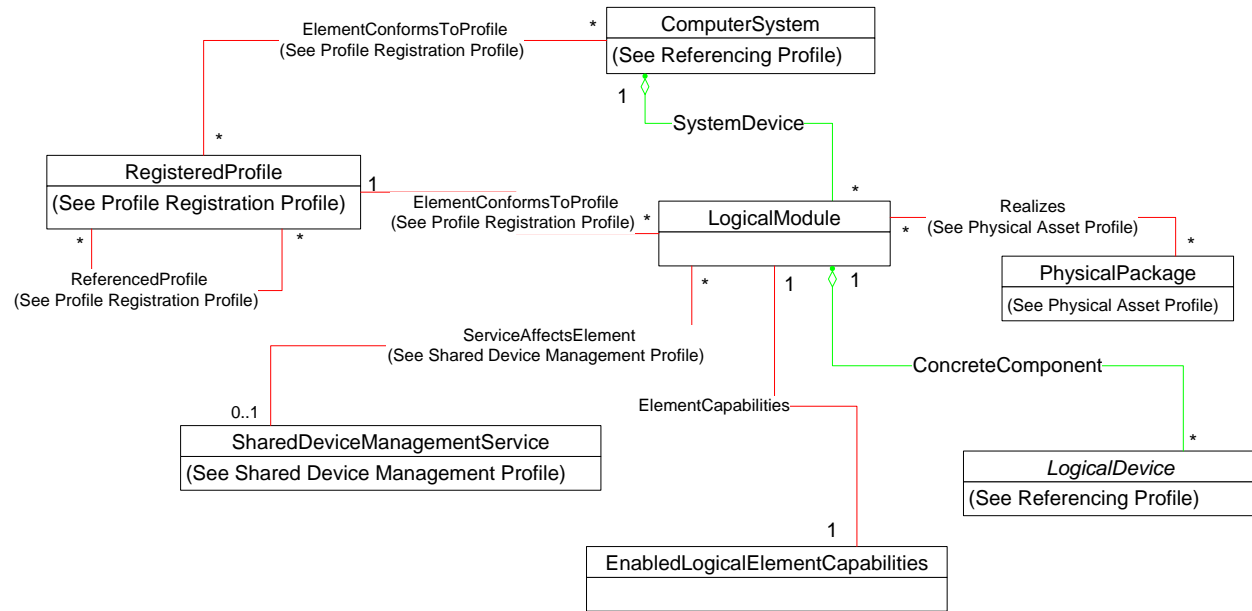
### 210 5.1.2 Scoping Instance

211 CIM\_ComputerSystem shall be the central class of this profile. The instance of CIM\_ComputerSystem  
212 with which the central instance is associated via an instance of CIM\_SystemDevice shall be the central  
213 instance of this profile.

## 214 6 Description

215 The *Device Tray Profile* describes a device tray. A device tray is a device which provides aggregation of  
216 other devices for the purposes of management. It is commonly used as an aggregation point for media  
217 devices in a modular system or rack configuration.

218 Figure 1 represents the class schema for the *Device Tray Profile*. For simplicity, the prefix CIM\_ has been  
219 removed from the name of the classes.



220

221

**Figure 1 – Device Tray Profile: Class Diagram**

222 The device tray is modeled with an instance of CIM\_LogicalModule. It is scoped to its owning system via  
 223 an instance of the CIM\_SystemDevice association. The physical aspects of the device tray can be  
 224 optionally modeled using CIM\_PhysicalPackage. Conformance with this profile is advertised using the  
 225 CIM\_RegisteredProfile class.

## 226 7 Implementation

227 The list of all required methods can be found in 8 (“Methods”) and properties in 10 (“CIM Elements”).

### 228 7.1 CIM\_LogicalModule

229 A device tray aggregates one or more logical devices which are then managed as a group. There shall be  
 230 an instance of CIM\_LogicalModule to represent the device tray.

### 231 7.2 Managing the Device Tray's State

232 This section describes the usage of the RequestedState and EnabledState properties to represent the  
 233 state of an instance of CIM\_LogicalModule.

#### 234 7.2.1 Indicating Support for State Management

235 There shall be exactly one instance of CIM\_EnabledLogicalElementCapabilities to indicate support for  
 236 managing state of the Device Tray.

#### 237 7.2.2 CIM\_LogicalModule.EnabledState

238 When the RequestedState parameter has a value of Enabled or Disabled, upon successful completion of  
 239 the CIM\_LogicalModule.RequestStateChange() method, the value of the EnabledState property shall  
 240 equal the value of the RequestedState property. If the method does not complete successfully, the value  
 241 of the EnabledState property is indeterminate. The EnabledState property shall have the value 2  
 242 (Enabled) or 3 (Disabled).

## 243 7.3 State Management Supported

244 Support for managing the state of the device tray is conditional behavior. This section describes the CIM  
245 elements and behaviors that shall be implemented when this behavior is supported.

246 **Conditional Determination:** A client can determine whether state management is supported as follows:

- 247 1) Find the CIM\_EnabledLogicalElementCapabilities instance associated with the  
248 CIM\_LogicalModule instance.
- 249 2) Query the value of the RequestedStatesSupported property. If at least one value is specified,  
250 state management is supported.

### 251 7.3.1 CIM\_LogicalModule.RequestStateChange() Supported

252 When the CIM\_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least  
253 one value, the CIM\_LogicalModule.RequestStateChange() method shall be implemented and supported.  
254 The CIM\_LogicalModule.RequestStateChange() method shall not return a value of 1 (Not Supported).

### 255 7.3.2 CIM\_LogicalModule.RequestedState

256 When state management is supported, the RequestedState property shall be supported. When state  
257 management is not supported, the RequestedState property may be supported. If the RequestedState  
258 property is supported and state management is not supported, the RequestedState property shall have  
259 the value 12 (Not Applicable).

260 The RequestedState property shall have one of the following values: 2 (Enabled), 3 (Disabled), 11  
261 (Reset), 5 (No Change), or 12 (Not Applicable). The initial value of the  
262 CIM\_LogicalModule.RequestedState property shall be 5 (No Change).

263 Upon successful invocation of the CIM\_LogicalModule.RequestStateChange() method, the value of the  
264 RequestedState property shall be the value of the RequestedState parameter. If the method is not  
265 successfully invoked, the value of the RequestedState property is indeterminate.

### 266 7.3.3 CIM\_EnabledLogicalElementCapabilities

267 When state management is supported, the RequestedStatesSupported property of the  
268 CIM\_EnabledLogicalElementCapabilities shall contain at least one value. The  
269 RequestedStatesSupported property may have zero or more of the following values: 2 (Enabled),  
270 3 (Disabled), or 11 (Reset).

## 271 7.4 ElementName Is Modifiable

272 Implementations may allow the CIM\_LogicalModule.ElementName to be modifiable by a client. This is  
273 conditional behavior. This section describes the CIM elements and behavior requirements when an  
274 implementation supports client modification of the CIM\_LogicalModule.ElementName property.

275 **Client Determination:** A client can determine whether the ElementName is modifiable as follows:

- 276 1) Find the CIM\_EnabledLogicalElementCapabilities instance associated with the  
277 CIM\_LogicalModule instance.
- 278 2) Query the value of the ElementNameEditSupported property of the instance. If the value is  
279 TRUE, the CIM\_LogicalModule.ElementName property is modifiable by a client.

### 280 7.4.1 CIM\_EnabledLogicalElementCapabilities.ElementNameEditSupported

281 This property shall have a value of TRUE when the implementation supports client modification of the  
282 CIM\_LogicalModule.ElementName property.

#### 283 **7.4.2 CIM\_EnabledLogicalElement.MaxElementNameLen**

284 The MaxElementNameLen property shall be implemented when the ElementNameEditSupported  
285 property has a value of TRUE.

#### 286 **7.4.3 CIM\_LogicalModule — ModifyInstance**

287 When the ElementNameEditSupported property of the CIM\_EnabledLogicalElementCapabilities has a  
288 value of true, the implementation shall allow the ModifyInstance operation to change the value of the  
289 ElementName property of the CIM\_LogicalModule instance. The ModifyInstance operation shall enforce  
290 the length restriction specified in the MaxElementNameLen property of the  
291 CIM\_EnabledLogicalElementCapabilities.

292 When the ElementNameEditSupported property of the CIM\_EnabledLogicalElementCapabilities has a  
293 value of false, the implementation shall not allow the ModifyInstance operation to change the value of the  
294 ElementName property of the CIM\_LogicalModule instance.

### 295 **7.5 Physical Asset Profile**

296 When an implementation instruments one or more instances of CIM\_PhysicalElement to represent the  
297 physical aspects of the device tray, these instances may be conformant with the [Physical Asset Profile](#).

298 **Condition Determination:** This profile places no restrictions on identifying conformance with the [Physical](#)  
299 [Asset Profile](#) beyond those specified in the [Physical Asset Profile](#) itself.

### 300 **7.6 Shared Device Management Profile**

301 The CIM\_LogicalModule is the focal point for management of the aggregated logical devices. In general,  
302 services which would directly manage the device if it was not aggregated will instead manage the  
303 CIM\_LogicalModule instance. The instrumentation requirements in the following paragraph reflect these  
304 guidelines.

305 When the [Shared Device Management Profile](#) is instrumented for providing management of a  
306 CIM\_LogicalModule which is conformant with this profile, there shall be an instance of the  
307 CIM\_ServiceAffectsElement association which references the CIM\_LogicalModule and the  
308 CIM\_SharedDeviceManagementService. There shall not be an instance of the  
309 CIM\_ServiceAffectsElement association which references the CIM\_SharedDeviceManagementService  
310 and references an instance of CIM\_LogicalDevice which is associated with the CIM\_LogicalModule  
311 instance via an instance of the CIM\_ConcreteComponent association.

312 **Conditional Determination:** This profile places no restrictions on advertising conformance with the  
313 [Shared Device Management Profile](#) beyond those specified in the [Shared Device Management Profile](#).

### 314 **7.7 Aggregated Devices Modeled**

315 Support for modeling devices aggregated into the device tray is conditional behavior.

316 **Client Determination:** A client can determine whether aggregated devices are being modeled as follows:

- 317 1) Query for instances of CIM\_ConcreteComponent where a reference to the CIM\_LogicalModule  
318 instance is the value of the GroupComponent property and a reference to a CIM\_LogicalDevice  
319 instance is the value of the PartComponent.

320 **7.7.1 Relationship between Device Tray and Components**

321 For each aggregated CIM\_LogicalDevice instance, there shall be exactly one instance of  
 322 CIM\_ConcreteComponent which associates the CIM\_LogicalDevice instance with a CIM\_LogicalModule  
 323 instance.

324 **7.7.2 CIM\_ConcreteComponent.GroupComponent**

325 The instance of CIM\_LogicalModule which represents the device tray shall be the value of the  
 326 GroupComponent.

327 **7.7.3 CIM\_ConcreteComponent.PartComponent**

328 An instance of CIM\_LogicalDevice which represents an aggregated device shall be the value of the  
 329 PartComponent.

330 **8 Methods**

331 **8.1 Method: CIM\_LogicalModule.RequestStateChange()**

332 CIM\_LogicalModule.RequestStateChange() method invocation will change the element’s state to the  
 333 value specified in the RequestedState parameter. The “Enabled”/“Disabled” values of the RequestedState  
 334 parameter will correspond to enabling or disabling the module represented by the instance of  
 335 CIM\_LogicalModule on/off accordingly. The value 11 (Reset) shall correspond to initiating a reset of the  
 336 device tray.

337 See 7.3.2 for information about the effect of this method on the RequestedState property.

338 The method shall be considered successful if the availability of the module upon completion of the  
 339 method corresponds to the desired availability indicated by the RequestedState parameter. It is not  
 340 necessary that an actual change in state occur for the method to be considered successful. It is sufficient  
 341 that the resultant state be equal to the requested state. Upon successful completion of the method, the  
 342 Return Value shall be zero.

343 See 7.2.2 for information about the effect of this method on the EnabledState property.

344 RequestStateChange() method’s detailed requirements are specified in Table 2 and Table 3.

345 **Table 2 – CIM\_LogicalModule.RequestStateChange() Method: Return Code Values**

| Value  | Description   |
|--------|---|
| 0      | Request was successfully executed.                    |
| 2      | Error occurred.                                       |
| 3      | Request timed out.                                    |
| 4      | Failed  |
| 0x1000 | Job started: REF returned to started CIM_ConcreteJob. |

346 No standard messages are defined.

347

**Table 3 – CIM\_LogicalModule.RequestStateChange() Method: Parameters**

| Qualifiers | Name           | Type                | Description/Values   |
|------------|----------------|---------------------|--|
| IN, REQ    | RequestedState | uint16              | Valid state values :<br>2 (Enabled)<br>3 (Disabled)<br>11 (Reset)  |
| OUT        | Job            | CIM_ConcreteJob REF | Returned if job started.   |
| IN, REQ    | TimeoutPeriod  | datetime            | Client specified maximum amount of time the transition to a new state is supposed to take.<br>0 or NULL – No time requirements<br><interval> – Maximum time allowed. |

348 Invoking the CIM\_LogicalModule.RequestStateChange() method multiple times could result in earlier  
349 requests being overwritten/lost.

## 350 8.2 Profile Conventions for Operations

351 For each profile class (including associations), the implementation requirements for operations, including  
352 those in the following default list, are specified in class-specific subclauses of this clause.

353 The default list of operations is as follows:

- 354 • GetInstance
- 355 • Associators
- 356 • AssociatorNames
- 357 • References
- 358 • ReferenceNames
- 359 • EnumerateInstances
- 360 • EnumerateInstanceNames

## 361 8.3 CIM\_ConcreteComponent

362 Table 4 lists implementation requirements for operations. If implemented, these operations shall be  
363 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 4, all operations in  
364 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

365 NOTE: Related profiles may define additional requirements on operations for the profile class.

366

**Table 4 – Operations: CIM\_ConcreteComponent**

| Operation       | Requirement   | Messages |
|-----------------|---------------|----------|
| ModifyInstance  | Not Supported | None     |
| Associators     | Not Supported | None     |
| AssociatorNames | Not Supported | None     |
| References      | Not Supported | None     |
| ReferenceNames  | Not Supported | None     |

367 **8.4 CIM\_ElementCapabilities**

368 Table 5 lists implementation requirements for operations. If implemented, these operations shall be  
 369 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 5, all operations in  
 370 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

371 NOTE: Related profiles may define additional requirements on operations for the profile class.

372 **Table 5 – Operations: CIM\_ElementCapabilities**

| Operation       | Requirement   | Messages |
|-----------------|---------------|----------|
| ModifyInstance  | Not Supported | None     |
| Associators     | Not Supported | None     |
| AssociatorNames | Not Supported | None     |
| References      | Not Supported | None     |
| ReferenceNames  | Not Supported | None     |

373 **8.5 CIM\_EnabledLogicalElementCapabilities**

374 Table 6 lists implementation requirements for operations. If implemented, these operations shall be  
 375 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 6, all operations in  
 376 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

377 NOTE: Related profiles may define additional requirements on operations for the profile class.

378 **Table 6 – Operations: CIM\_EnabledLogicalElementCapabilities**

| Operation      | Requirement   | Messages |
|----------------|---------------|----------|
| ModifyInstance | Not Supported | None     |

379 **8.6 CIM\_LogicalModule**

380 Table 7 lists implementation requirements for operations. If implemented, these operations shall be  
 381 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 7, all operations in  
 382 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

383 NOTE: Related profiles may define additional requirements on operations for the profile class.

384 **Table 7 – Operations: CIM\_LogicalModule**

| Operation              | Requirement          | Messages |
|------------------------|----------------------|----------|
| GetInstance            | Mandatory            | None     |
| ModifyInstance         | Optional (see 8.6.1) | None     |
| Associators            | Mandatory            | None     |
| AssociatorNames        | Mandatory            | None     |
| References             | Mandatory            | None     |
| ReferenceNames         | Mandatory            | None     |
| EnumerateInstances     | Mandatory            | None     |
| EnumerateInstanceNames | Mandatory            | None     |

### 385 **8.6.1 ModifyInstance**

386 There is conditional behavior which affects the requirements for implementing the ModifyInstance  
387 operation for CIM\_LogicalModule (see 7.4.3).

## 388 **8.7 CIM\_SystemDevice**

389 Table 8 lists implementation requirements for operations. If implemented, these operations shall be  
390 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 8, all operations in  
391 the default list in 8.2 shall be implemented as defined in [DSP0200](#).

392 NOTE: Related profiles may define additional requirements on operations for the profile class.

393 **Table 8 – Operations: CIM\_SystemDevice**

| Operation       | Requirement   | Messages |
|-----------------|---------------|----------|
| ModifyInstance  | Not Supported | None     |
| Associators     | Not Supported | None     |
| AssociatorNames | Not Supported | None     |
| References      | Not Supported | None     |
| ReferenceNames  | Not Supported | None     |

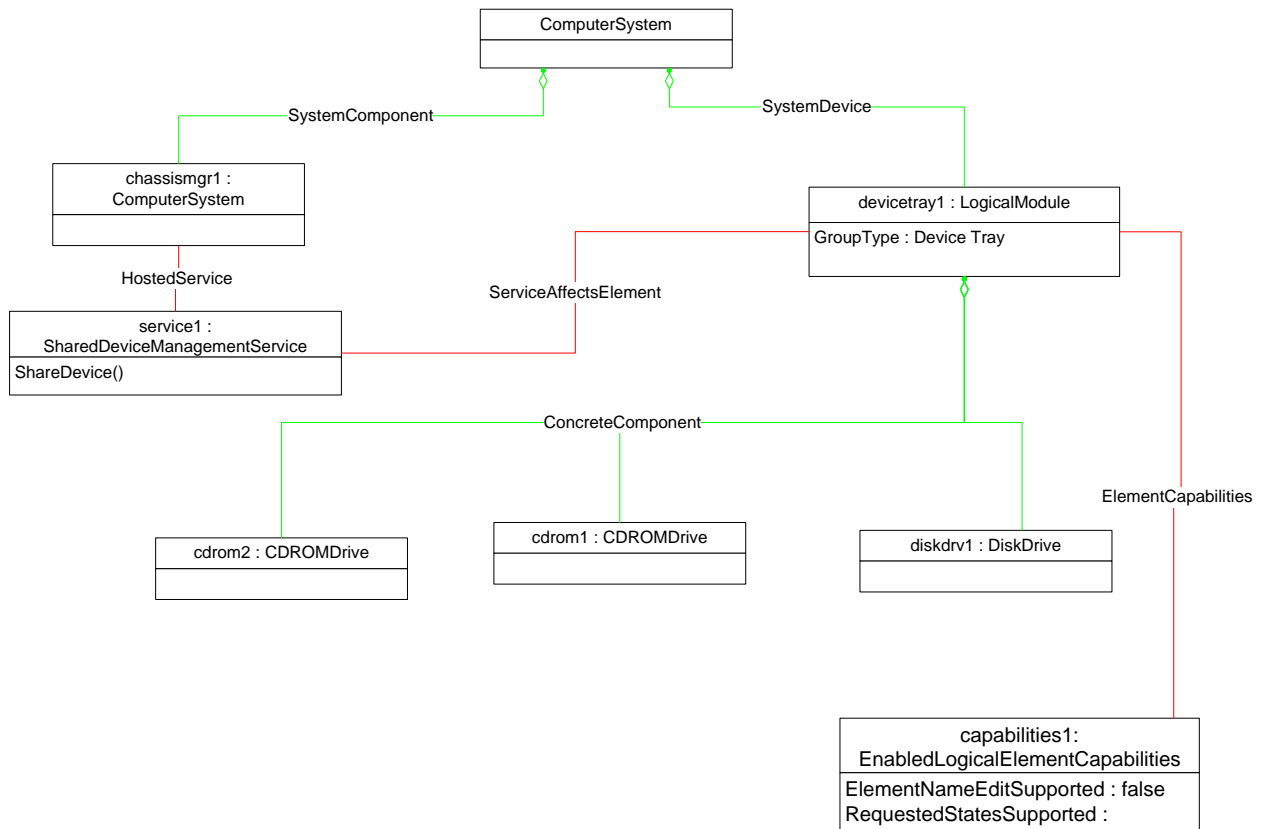
## 394 **9 Use Cases**

395 This section outlines the use cases of a device tray. Use cases are informative and not intended to define  
396 the requirements for conformance.

### 397 **9.1 Object Diagrams**

398 Figure 2 illustrates a device or media tray which serves as the focal point for management of the shared  
399 CD-Rom and disk drives. Ownership of, or access to, the CIM\_LogicalModule instance imparts access to  
400 the associated shared components. The CIM\_SharedDeviceManagementService instance is associated  
401 with the CIM\_LogicalModule instance via the ServiceAffectsElement association because ownership of  
402 the CIM\_LogicalModule instance is managed, rather than ownership of the individual shared devices. The  
403 properties on the associated CIM\_EnabledLogicalElementCapabilities instance indicate that changing  
404 states on the device tray is not supported. Changing the ElementName property is not supported either.





405

406

**Figure 2 – Device Tray Object Diagram**

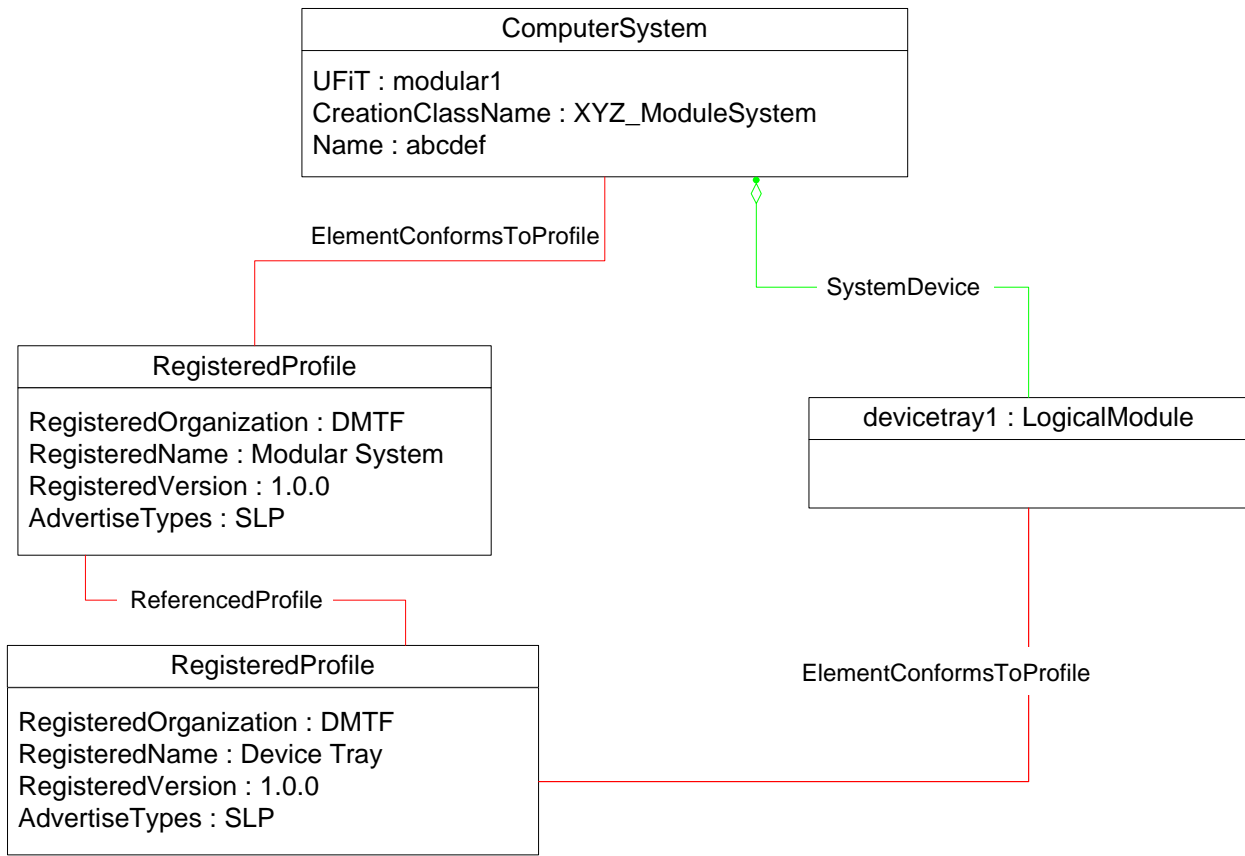
407

408

409

410

Figure 3 is a class diagram indicating how an implementation would advertise conformance with the *Device Tray Profile*. Note that the instance of *CIM\_RegisteredProfile* would be created in the Interop namespace while the *CIM\_ComputerSystem* instance would be created in a namespace of the instrumentation's choosing.



411

412

Figure 3 – Registered Profile

## 413 9.2 Determine Aggregated Devices

414 A client can determine which logical devices are aggregated into a device tray as follows:

- 415 1) Find all instances of `CIM_LogicalDevice` associated with the `CIM_LogicalModule` instance via  
 416 an instance of `CIM_ConcreteComponent` where the `GroupComponent` property is a reference to  
 417 the `CIM_LogicalModule` instance.

## 418 9.3 Find FRU Information

419 FRU information can be provided by for a device tray. A client can determine the FRU information if  
 420 provided as follows:

- 421 1) Find an instance of `CIM_PhysicalPackage` associated with the `CIM_LogicalModule` instance via  
 422 an instance of the `CIM_Realizes` association.
- 423 2) View the FRU information as indicated by the [Physical Asset Profile](#).

424 **10 CIM Elements**

425 Table 9 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be  
 426 implemented as described in Table 9. Sections 7 (“Implementation”) and 8 (“Methods”) may impose  
 427 additional requirements on these elements.

428 **Table 9 – CIM Elements: Device Tray Profile**

| Element Name                          | Requirement | Description       |
|---------------------------------------|-------------|-------------------|
| <b>Classes</b>                        |             |                   |
| CIM_ConcreteComponent                 | Conditional | See 10.1 and 7.7. |
| CIM_ElementCapabilities               | Mandatory   | See 10.2.         |
| CIM_EnabledLogicalElementCapabilities | Mandatory   | See 10.3.         |
| CIM_LogicalModule                     | Mandatory   | See 10.4.         |
| CIM_RegisteredProfile                 | Mandatory   | See 10.5.         |
| CIM_SystemDevice                      | Mandatory   | See 10.6.         |
| <b>Indications</b>                    |             |                   |
| None defined in this profile          |             |                   |

429 **10.1 CIM\_ConcreteComponent**

430 CIM\_ConcreteComponent is used to associate an instance of CIM\_LogicalModule with a  
 431 CIM\_LogicalDevice which is aggregated into the device tray. Table 10 contains the requirements for  
 432 elements of this class.

433 **Table 10 – Class: CIM\_ConcreteComponent**

| Properties     | Requirement | Description |
|----------------|-------------|-------------|
| GroupComponent | Mandatory   | See 7.7.2.  |
| PartComponent  | Mandatory   | See 7.7.3.  |

434 **10.2 CIM\_ElementCapabilities**

435 CIM\_ElementCapabilities is used to associate an instance of CIM\_EnabledLogicalElementCapabilities  
 436 with the CIM\_LogicalModule. Table 11 contains the requirements for elements of this class.

437 **Table 11 – Class: CIM\_ElementCapabilities**

| Properties     | Requirement | Notes   |
|----------------|-------------|---|
| ManagedElement | Mandatory   | This shall be a reference to the CIM_LogicalModule instance.                        |
| Capabilities   | Mandatory   | This shall be a reference to the instance of CIM_EnabledLogicalElementCapabilities. |

### 438 10.3 CIM\_EnabledLogicalElementCapabilities

439 CIM\_EnabledLogicalElementCapabilities is used to indicate support for managing the Device Tray.  
440 Table 12 contains the requirements for elements of this class.

441 **Table 12 – Class: CIM\_EnabledLogicalElementCapabilities**

| Properties               | Requirement | Notes      |
|--------------------------|-------------|------------|
| InstanceID               | Mandatory   | <b>Key</b> |
| RequestedStatesSupported | Mandatory   | See 7.3.3. |
| ElementNameEditSupported | Mandatory   | See 7.4.1. |
| MaxElementNameLen        | Conditional | See 7.4.2. |

### 442 10.4 CIM\_LogicalModule

443 CIM\_LogicalModule is used to represent components such as a device or media tray which serve as an  
444 aggregation point for the management of shared devices. Table 13 contains the requirements for  
445 elements of this class.

446 **Table 13 – Class: CIM\_LogicalModule**

| Properties              | Requirement | Description   |
|-------------------------|-------------|---|
| ModuleNumber            | Mandatory   |   |
| SystemCreationClassName | Mandatory   | <b>Key</b>  |
| SystemName              | Mandatory   | <b>Key</b>  |
| CreationClassName       | Mandatory   | <b>Key</b>  |
| DeviceID                | Mandatory   | <b>Key</b>  |
| EnabledState            | Mandatory   | See 7.2.2.  |
| RequestedState          | Conditional | See 7.3.2.  |
| EnabledDefault          | Mandatory   | Matches 3 (Disabled) or 2 (Enabled)   |
| OperationalStatus       | Mandatory   |   |
| StatusDescriptions      | Conditional | If the OperationalStatus property has a value of 0 (Other), this property shall be supported. |
| LogicalModuleType       | Mandatory   | This property shall have a value of 2 ("Device Tray").  |
| RequestStateChange()    | Conditional | See 7.3.  |

### 447 10.5 CIM\_RegisteredProfile

448 CIM\_RegisteredProfile identifies the *Device Tray Profile* in order for a client to determine whether an  
449 instance of CIM\_LogicalModule is conformant with this profile. The CIM\_RegisteredProfile class is  
450 defined by the [Profile Registration Profile](#). With the exception of the mandatory values specified for the  
451 properties below, the behavior of the CIM\_RegisteredProfile instance is per the [Profile Registration](#)  
452 [Profile](#). Table 14 contains the requirements for elements of this class.

453

**Table 14 – Class: CIM\_RegisteredProfile**

| Properties             | Requirement | Description  |
|------------------------|-------------|--|
| RegisteredName         | Mandatory   | This property shall have a value of "Device Tray". |
| RegisteredVersion      | Mandatory   | This property shall have a value of "1.0.0".       |
| RegisteredOrganization | Mandatory   | This property shall have a value of "DMTF".        |

454 **10.6 CIM\_SystemDevice**

455 CIM\_SystemDevice is used to associate an instance of CIM\_LogicalModule with an instance of  
 456 CIM\_ComputerSystem representing a modular enclosure. Table 15 contains the requirements for  
 457 elements of this class.

458

**Table 15 – Class: CIM\_SystemDevice**

| Properties     | Requirement | Description  |
|----------------|-------------|--|
| GroupComponent | Mandatory   | Scoping system defined outside of this specification.                    |
| PartComponent  | Mandatory   | This property shall be a reference to the instance of CIM_LogicalModule. |

459

460  
461  
462  
463  
464

## **ANNEX A** (informative)

### **Change Log**

| <b>Version</b> | <b>Date</b> | <b>Author</b> | <b>Description</b>    |
|----------------|-------------|---------------|-----------------------|
| 1.0.0          | 06-10-2009  |               | DMTF Standard Release |

465