



1
2
3
4

Document Number: DSP0835

Date: 2009-06-04

Version: 1.0.0

5 **Indicator LED Profile SM CLP Command**
6 **Mapping Specification**

7 **Document Type: Specification**
8 **Document Status: DMTF Standard**
9 **Document Language: E**

10

11 Copyright notice

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

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

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

33

CONTENTS

34	Foreword	5
35	Introduction	6
36	1 Scope	7
37	2 Normative References.....	7
38	2.1 Approved References	7
39	2.2 Other References.....	7
40	3 Terms and Definitions.....	7
41	4 Symbols and Abbreviated Terms.....	8
42	5 Recipes.....	9
43	6 Mappings.....	9
44	6.1 CIM_AssociatedIndicatorLED	9
45	6.2 CIM_ElementCapabilities	12
46	6.3 CIM_IndicatorLEDCapabilities.....	14
47	6.4 CIM_IndicatorLED.....	16
48	6.5 CIM_SystemDevice	19
49	ANNEX A (informative) Change Log	22
50		

51 **Tables**

52	Table 1 – Command Verb Requirements for CIM_AssociatedIndicatorLED.....	9
53	Table 2 – Command Verb Requirements for CIM_ElementCapabilities	12
54	Table 3 – Command Verb Requirements for CIM_IndicatorLEDCapabilities.....	14
55	Table 4 – Command Verb Requirements for CIM_IndicatorLED.....	16
56	Table 5 – Command Verb Requirements for CIM_SystemDevice	19
57		

59

Foreword

60 The *Indicator LED Profile SM CLP Command Mapping Specification* (DSP0835) was prepared by the
61 DMTF Server Management Working Group.

62 **Conventions**

63 The pseudo-code conventions utilized in this document are the Recipe Conventions as defined in SNIA
64 [SMI-S 1.1.0](#), section 7.6.

65 **Acknowledgements**

66 The authors wish to acknowledge the following participants from the DMTF Server Management Working
67 Group:

- 68 • Aaron Merkin – IBM
- 69 • Khachatur Papanyan – Dell
- 70 • Jon Hass – Dell
- 71 • Jeff Hilland – HP
- 72 • Christina Shaw – HP
- 73 • John Leung – Intel

74

75

Introduction

76 This document defines the SM CLP mapping for CIM elements described in the [Indicator LED Profile](#).
77 The information in this specification, combined with the *SM CLP-to-CIM Common Mapping*
78 *Specification 1.0* ([DSP0216](#)), is intended to be sufficient to implement SM CLP commands relevant to the
79 classes, properties, and methods described in the [Indicator LED Profile](#) using CIM operations.

80 The target audience for this specification is implementers of the SM CLP support for the [Indicator LED](#)
81 [Profile](#).

82
83

Indicator LED Profile SM CLP Command Mapping Specification

1 Scope

This specification contains the requirements for an implementation of the SM CLP to provide access to, and implement the behaviors of, the [Indicator LED Profile](#).

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 DSP1074, *Indicator LED Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1074_1.0.pdf

DMTF DSP0216, *SM CLP-to-CIM Common Mapping Specification 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0216_1.0.pdf

SNIA, *Storage Management Initiative Specification (SMI-S) 1.1.0*,
http://www.snia.org/tech_activities/standards/curr_standards/smi

2.2 Other References

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

3 Terms and Definitions

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

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

indicates requirements to be followed strictly in order to conform to the document when the specified conditions are met

- 113 **3.4**
114 **mandatory**
115 indicates requirements to be followed strictly in order to conform to the document and from which no
116 deviation is permitted
- 117 **3.5**
118 **may**
119 indicates a course of action permissible within the limits of the document
- 120 **3.6**
121 **need not**
122 indicates a course of action permissible within the limits of the document
- 123 **3.7**
124 **optional**
125 indicates a course of action permissible within the limits of the document
- 126 **3.8**
127 **shall**
128 indicates requirements to be followed strictly in order to conform to the document and from which no
129 deviation is permitted
- 130 **3.9**
131 **shall not**
132 indicates requirements to be followed strictly in order to conform to the document and from which no
133 deviation is permitted
- 134 **3.10**
135 **should**
136 indicates that among several possibilities, one is recommended as particularly suitable, without
137 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 138 **3.11**
139 **should not**
140 indicates that a certain possibility or course of action is deprecated but not prohibited

141 **4 Symbols and Abbreviated Terms**

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

- 143 **4.1**
144 **CIM**
145 Common Information Model
- 146 **4.2**
147 **CLP**
148 Command Line Protocol
- 149 **4.3**
150 **DMTF**
151 Distributed Management Task Force

152 **4.4**
 153 **SM**
 154 Server Management

155 **4.5**
 156 **SMI-S**
 157 Storage Management Initiative Specification

158 **4.6**
 159 **SNIA**
 160 Storage Networking Industry Association

161 **4.7**
 162 **UFsT**
 163 User Friendly selection Tag

164 **5 Recipes**

165 The following is a list of the common recipes used by the mappings in this specification. For a definition of
 166 each recipe, see *SM CLP-to-CIM Common Mapping Specification 1.0* ([DSP0216](#)).

- 167 • smShowInstance
- 168 • smShowInstances
- 169 • smShowAssociationInstance
- 170 • smShowAssociationInstances
- 171 • smSetInstance

172 **6 Mappings**

173 The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in
 174 the [Indicator LED Profile](#). Requirements specified here related to the support for a CLP verb for a
 175 particular class are solely within the context of this profile.

176 **6.1 CIM_AssociatedIndicatorLED**

177 The `cd`, `exit`, `help`, and `version` verbs shall be supported as described in [DSP0216](#).

178 Table 1 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
 179 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and
 180 target. Table 1 is for informational purposes only; in case of a conflict between Table 1 and requirements
 181 detailed in the following sections, the text detailed in the following sections supersedes the information in
 182 Table 1.

183 **Table 1 – Command Verb Requirements for CIM_AssociatedIndicatorLED**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	

Command Verb	Requirement	Comments
reset	Not supported	
set	Not supported	
show	Shall	See 6.1.2.
start	Not supported	
stop	Not supported	

184 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
185 reset, set, start, and stop.

186 6.1.1 Ordering of Results

187 When results are returned for multiple instances of CIM_AssociatedIndicatorLED, implementations shall
188 utilize the following algorithm to produce the natural (that is, default) ordering:

- 189 • Results for CIM_AssociatedIndicatorLED are unordered; therefore, no algorithm is defined.

190 6.1.2 Show

191 This section describes how to implement the `show` verb when applied to an instance of
192 CIM_AssociatedIndicatorLED. Implementations shall support the use of the `show` verb with
193 CIM_AssociatedIndicatorLED.

194 6.1.2.1 Show Command Form for Multiple Instances Target – CIM_ManagedSystemElement 195 Reference

196 This command form is used to show many instances of CIM_AssociatedIndicatorLED. This command
197 form corresponds to a `show` command issued against the instance of CIM_AssociatedIndicatorLED
198 where only one reference is specified and the reference is to the instance of
199 CIM_ManagedSystemElement.

200 6.1.2.1.1 Command Form

```
201 show <CIM_AssociatedIndicatorLED multiple instances>
```

202 6.1.2.1.2 CIM Requirements

203 See CIM_AssociatedIndicatorLED in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of
204 mandatory properties.

205 6.1.2.1.3 Behavior Requirements

206 6.1.2.1.3.1 Preconditions

207 `$instance` represents the instance of a CIM_ManagedSystemElement, which is referenced by
208 CIM_AssociatedIndicatorLED.

209 6.1.2.1.3.2 Pseudo Code

```
210 &smShowAssociationInstances ( "CIM_AssociatedIndicatorLED",  
211     $instance.getObjectPath() );  
212 &smEnd;
```

213 **6.1.2.2 Show Command Form for Multiple Instance Target – CIM_IndicatorLED Reference**

214 This command form is used to show multiple instance of CIM_AssociatedIndicatorLED. This command
215 form corresponds to a `show` command issued against multiple instances of CIM_AssociatedIndicatorLED,
216 where only one reference is specified and the reference is to the instance of CIM_IndicatorLED.

217 **6.1.2.2.1 Command Form**

```
218 show <CIM_AssociatedIndicatorLED multiple instances>
```

219 **6.1.2.2.2 CIM Requirements**

220 See CIM_AssociatedIndicatorLED in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of
221 mandatory properties.

222 **6.1.2.2.3 Behavior Requirements**

223 **6.1.2.2.3.1 Preconditions**

224 `$instance` represents the instance of CIM_IndicatorLED which is referenced by
225 CIM_AssociatedIndicatorLED.

226 **6.1.2.2.3.2 Pseudo Code**

```
227 &smShowAssociationInstances ( "CIM_AssociatedIndicatorLED",  
228     $instance.getObjectPath() );  
229 &smEnd;
```

230 **6.1.2.3 Show Command Form for a Single Instance Target – Both References**

231 This command form is for the `show` verb applied to a single instance. This command form corresponds to
232 the `show` command issued against CIM_AssociatedIndicatorLED where both references are specified;
233 therefore, the desired instance is unambiguously identified.

234 **6.1.2.3.1 Command Form**

```
235 show <CIM_AssociatedIndicatorLED single instance>
```

236 **6.1.2.3.2 CIM Requirements**

237 See CIM_AssociatedIndicatorLED in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of
238 mandatory properties.

239 **6.1.2.3.3 Behavior Requirements**

240 **6.1.2.3.3.1 Preconditions**

241 `$instanceA` represents the referenced instance of CIM_IndicatorLED through
242 CIM_AssociatedIndicatorLED association.

243 `$instanceB` represents the instance of CIM_ManagedSystemElement which is referenced by
244 CIM_AssociatedIndicatorLED.

245 **6.1.2.3.3.2 Pseudo Code**

```
246 &smShowAssociationInstance ( "CIM_AssociatedIndicatorLED", $instanceA.getObjectPath(),  
247     $instanceB.getObjectPath() );  
248 &smEnd;
```

249 6.2 CIM_ElementCapabilities

250 The `cd`, `exit`, `help`, and `version` verbs shall be supported as described in [DSP0216](#).

251 Table 2 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
 252 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and
 253 target. Table 2 is for informational purposes only; in case of a conflict between Table 2 and requirements
 254 detailed in the following sections, the text detailed in the following sections supersedes the information in
 255 Table 2.

256 **Table 2 – Command Verb Requirements for CIM_ElementCapabilities**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.2.2.
start	Not supported	
stop	Not supported	

257 No mapping is defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,
 258 `reset`, `set`, `start`, and `stop`.

259 6.2.1 Ordering of Results

260 When results are returned for multiple instances of `CIM_ElementCapabilities`, implementations shall
 261 utilize the following algorithm to produce the natural (that is, default) ordering:

- 262 • Results for `CIM_ElementCapabilities` are unordered; therefore, no algorithm is defined.

263 6.2.2 Show

264 This section describes how to implement the `show` verb when applied to an instance of
 265 `CIM_ElementCapabilities`. Implementations shall support the use of the `show` verb with
 266 `CIM_ElementCapabilities`.

267 6.2.2.1 Show Command Form for Multiple Instances Target – CIM_IndicatorLEDCapabilities 268 Reference

269 This command form is used to show many instances of `CIM_ElementCapabilities`. This command form
 270 corresponds to a `show` command issued against instances of `CIM_ElementCapabilities` where only one
 271 reference is specified and the reference is to an instance of `CIM_IndicatorLEDCapabilities`.

272 6.2.2.1.1 Command Form

273 `show <CIM_ElementCapabilities multiple instances>`

274 6.2.2.1.2 CIM Requirements

275 See CIM_ElementCapabilities in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of
276 mandatory properties.

277 6.2.2.1.3 Behavior Requirements

278 6.2.2.1.3.1 Preconditions

279 \$instance represents the instance of CIM_IndicatorLEDCapabilities which is referenced by
280 CIM_ElementCapabilities.

281 6.2.2.1.3.2 Pseudo Code

```
282 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );  
283 &smEnd;
```

284 6.2.2.2 Show Command Form for a Single Instance – CIM_IndicatorLED Reference

285 This command form is used to show a single instance of CIM_ElementCapabilities. This command form
286 corresponds to a `show` command issued against a single instance of CIM_ElementCapabilities where
287 only one reference is specified and the reference is to the instance of CIM_IndicatorLED.

288 6.2.2.2.1 Command Form

```
289 show <CIM_ElementCapabilities single instance>
```

290 6.2.2.2.2 CIM Requirements

291 See CIM_ElementCapabilities in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of
292 mandatory properties.

293 6.2.2.2.3 Behavior Requirements

294 6.2.2.2.3.1 Preconditions

295 \$instance represents the instance of CIM_IndicatorLED which is referenced by
296 CIM_ElementCapabilities.

297 6.2.2.2.3.2 Pseudo Code

```
298 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );  
299 &smEnd;
```

300 6.2.2.3 Show Command Form for a Single Instance Target – Both References

301 This command form is for the `show` verb applied to a single instance. This command form corresponds to
302 the `show` command issued against CIM_ElementCapabilities where both references are specified and
303 therefore the desired instance is unambiguously identified.

304 6.2.2.3.1 Command Form

```
305 show <CIM_ElementCapabilities single instance>
```

306 6.2.2.3.2 CIM Requirements

307 See CIM_ElementCapabilities in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of
308 mandatory properties.

309 **6.2.2.3.3 Behavior Requirements**310 **6.2.2.3.3.1 Preconditions**

311 \$instanceA represents the referenced instance of CIM_IndicatorLED through CIM_ElementCapabilities
312 association.

313 \$instanceB represents the instance of CIM_IndicatorLEDCapabilities which is referenced by
314 CIM_ElementCapabilities.

315 **6.2.2.3.3.2 Pseudo Code**

```
316 &smShowAssociationInstance ( "CIM_ElementCapabilities", $instanceA.getObjectPath(),
317     $instanceB.getObjectPath() );
318 &smEnd;
```

319 **6.3 CIM_IndicatorLEDCapabilities**

320 The `cd`, `exit`, `help`, and `version` verbs shall be supported as described in [DSP0216](#).

321 Table 3 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
322 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and
323 target. Table 3 is for informational purposes only; in case of a conflict between Table 3 and requirements
324 detailed in the following sections, the text detailed in the following sections supersedes the information in
325 Table 3.

326 **Table 3 – Command Verb Requirements for CIM_IndicatorLEDCapabilities**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.3.2.
start	Not supported	
stop	Not supported	

327 No mapping is defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,
328 `reset`, `set`, `start`, and `stop`.

329 **6.3.1 Ordering of Results**

330 When results are returned for multiple instances of CIM_IndicatorLEDCapabilities, implementations shall
331 utilize the following algorithm to produce the natural (that is, default) ordering:

- 332
- Results for CIM_IndicatorLEDCapabilities are unordered; therefore, no algorithm is defined.

333 **6.3.2 Show**

334 This section describes how to implement the `show` verb when applied to an instance of
 335 `CIM_IndicatorLEDCapabilities`. Implementations shall support the use of the `show` verb with
 336 `CIM_IndicatorLEDCapabilities`.

337 **6.3.2.1 Show Command Form for Multiple Instances Target**

338 This command form is used to show many instances of `CIM_IndicatorLEDCapabilities`.

339 **6.3.2.1.1 Command Form**

```
340 show <CIM_IndicatorLEDCapabilities multiple instances>
```

341 **6.3.2.1.2 CIM Requirements**

342 See `CIM_IndicatorLEDCapabilities` in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of
 343 mandatory properties.

344 **6.3.2.1.3 Behavior Requirements**

345 **6.3.2.1.3.1 Preconditions**

346 `$containerInstance` represents the instance of `CIM_ConcreteCollection` with `ElementName` property
 347 that contains “Capabilities” and is associated to the targeted instances of `CIM_IndicatorLEDCapabilities`
 348 through the `CIM_MemberOfCollection` association.

349 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

350 **6.3.2.1.3.2 Pseudo Code**

```
351 #propertylist[] = NULL;
352 if ( false == #all )
353     {
354         #propertylist[] = <array of mandatory non-key property names (see CIM
355             Requirements)>;
356     }
357 &smShowInstances ( "CIM_IndicatorLEDCapabilities", "CIM_MemberOfCollection",
358     $containerInstance.getObjectPath(), #propertylist[] );
359 &smEnd;
```

360 **6.3.2.2 Show Command Form for a Single Instance Target**

361 This command form is used to show a single instance of `CIM_IndicatorLEDCapabilities`.

362 **6.3.2.2.1 Command Form**

```
363 show <CIM_IndicatorLEDCapabilities single instance>
```

364 **6.3.2.2.2 CIM Requirements**

365 See `CIM_IndicatorLEDCapabilities` in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of
 366 mandatory properties.

367 **6.3.2.2.3 Behavior Requirements**368 **6.3.2.2.3.1 Preconditions**

369 \$instance represents the targeted instance of CIM_IndicatorLEDCapabilities.

370 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

371 **6.3.2.2.3.2 Pseudo Code**

```

372 $instance=<CIM_IndicatorLEDCapabilities single instance>;
373 #propertylist[] = NULL;
374 if ( false == #all )
375 {
376     #propertylist[] = <array of mandatory non-key property names (see CIM
377         Requirements)>;
378 }
379 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
380 &smEnd;

```

381 **6.4 CIM_IndicatorLED**382 The cd, exit, help, and version verbs shall be supported as described in [DSP0216](#).

383 Table 4 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
 384 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and
 385 target. Table 4 is for informational purposes only; in case of a conflict between Table 4 and requirements
 386 detailed in the following sections, the text detailed in the following sections supersedes the information in
 387 Table 4.

388 **Table 4 – Command Verb Requirements for CIM_IndicatorLED**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.4.2.
show	Shall	See 6.4.3.
start	Not supported	
stop	Not supported	

389 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
 390 reset, start, and stop.

391 **6.4.1 Ordering of Results**

392 When results are returned for multiple instances of CIM_IndicatorLED, implementations shall utilize the
 393 following algorithm to produce the natural (that is, default) ordering:

- 394
- Results for CIM_IndicatorLED are unordered; therefore, no algorithm is defined.

395 **6.4.2 Set**

396 This section describes how to implement the `set` verb when it is applied to an instance of
397 `CIM_IndicatorLED`. Implementations may support the use of the `set` verb with `CIM_IndicatorLED`.

398 The `set` verb is used to modify descriptive properties of the `CIM_IndicatorLED` instance.

399 **6.4.2.1 General Usage of Set for a Single Property**

400 This command form corresponds to the general usage of the `set` verb to modify a single property of a
401 target instance. This is the most common case.

402 The requirement for supporting modification of a property using this command form shall be equivalent to
403 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined
404 in the [Indicator LED Profile](#).

405 **6.4.2.1.1 Command Form**

```
406 set <CIM_IndicatorLED single object> <propertyname>=<propertyvalue>
```

407 **6.4.2.1.2 CIM Requirements**

408 See `CIM_IndicatorLED` in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of mandatory
409 properties.

410 **6.4.2.1.3 Behavior Requirements**

```
411 $instance=<CIM_IndicatorLED single object>
412 #propertyName[] = {<propertyname>};
413 #propertyValues[] = {<propertyvalue>};
414 &smSetInstance ( $instance, #propertyName[], #propertyValues[] );
415 &smEnd;
```

416 **6.4.2.2 General Usage of Set for Multiple Properties**

417 This command form corresponds to the general usage of the `set` verb to modify multiple properties of a
418 target instance where there is not an explicit relationship between the properties. This is the most
419 common case.

420 The requirement for supporting modification of a property using this command form shall be equivalent to
421 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined
422 in the [Indicator LED Profile](#).

423 **6.4.2.2.1 Command Form**

```
424 set <CIM_IndicatorLED multiple objects> <propertyname1>=<propertyvalue1>
425 <propertynamen>=<propertyvaluen>
```

426 **6.4.2.2.2 CIM Requirements**

427 See `CIM_IndicatorLED` in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of mandatory
428 properties.

429 6.4.2.2.3 Behavior Requirements

```

430 $instance=<CIM_IndicatorLED multiple objects>
431 #propertyNames[] = {<propertyname>};
432 for #i < n
433     {
434         #propertyNames[#i] = <propertyname#i>
435         #propertyValues[#i] = <propertyvalue#i>
436     }
437 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
438 &smEnd;
```

439 6.4.3 Show

440 This section describes how to implement the `show` verb when applied to an instance of
 441 CIM_IndicatorLED. Implementations shall support the use of the `show` verb with CIM_IndicatorLED.

442 6.4.3.1 Show Command Form for Multiple Instances Target

443 This command form is used to show many instances of CIM_IndicatorLED.

444 6.4.3.1.1 Command Form

```

445 show <CIM_IndicatorLED multiple instances>
```

446 6.4.3.1.2 CIM Requirements

447 See CIM_IndicatorLED in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of mandatory
 448 properties.

449 6.4.3.1.3 Behavior Requirements

450 6.4.3.1.3.1 Preconditions

451 `$containerInstance` represents the instance of CIM_ComputerSystem which represents the
 452 container system and is associated to the targeted instances of CIM_IndicatorLED through the
 453 CIM_SystemDevice association.

454 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

455 6.4.3.1.3.2 Pseudo Code

```

456 #propertylist[] = NULL;
457 if ( false == #all )
458     {
459         #propertylist[] = <array of mandatory non-key property names (see CIM
460             Requirements)>;
461     }
462 &smShowInstances ( "CIM_IndicatorLED", "CIM_SystemDevice",
463     $containerInstance.getObjectPath(), #propertylist[] );
464 &smEnd;
```

465 6.4.3.2 Show Command Form for a Single Instance Target

466 This command form is used to show a single instance of CIM_IndicatorLED.

467 **6.4.3.2.1 Command Form**

468 `show <CIM_IndicatorLED single instance>`

469 **6.4.3.2.2 CIM Requirements**

470 See CIM_IndicatorLED in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of mandatory
471 properties.

472 **6.4.3.2.3 Behavior Requirements**

473 **6.4.3.2.3.1 Preconditions**

474 \$instance represents the targeted instance of CIM_IndicatorLED.

475 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

476 **6.4.3.2.3.2 Pseudo Code**

```
477 $instance=<CIM_IndicatorLED single instance>;
478 #propertylist[] = NULL;
479 if ( false == #all )
480 {
481     #propertylist[] = <array of mandatory non-key property names (see CIM
482     Requirements)>;
483 }
484 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
485 &smEnd;
```

486 **6.5 CIM_SystemDevice**

487 The cd, exit, help, and version verbs shall be supported as described in [DSP0216](#).

488 Table 5 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
489 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and
490 target. Table 5 is for informational purposes only; in case of a conflict between Table 5 and requirements
491 detailed in the following sections, the text detailed in the following sections supersedes the information in
492 Table 5.

493 **Table 5 – Command Verb Requirements for CIM_SystemDevice**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.5.2.
start	Not supported	
stop	Not supported	

494 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
495 reset, set, start, and stop.

496 6.5.1 Ordering of Results

497 When results are returned for multiple instances of CIM_SystemDevice, implementations shall utilize the
498 following algorithm to produce the natural (that is, default) ordering:

- 499 • Results for CIM_SystemDevice are unordered; therefore, no algorithm is defined.

500 6.5.2 Show

501 This section describes how to implement the `show` verb when applied to an instance of
502 CIM_SystemDevice. Implementations shall support the use of the `show` verb with CIM_SystemDevice.

503 6.5.2.1 Show Command Form for Multiple Instances Target – CIM_ComputerSystem Reference

504 This command form is used to show many instances of CIM_SystemDevice. This command form
505 corresponds to a `show` command issued against the instance of CIM_SystemDevice where only one
506 reference is specified and the reference is to the scoping instance of CIM_ComputerSystem.

507 6.5.2.1.1 Command Form

```
508 show <CIM_SystemDevice multiple instances>
```

509 6.5.2.1.2 CIM Requirements

510 See CIM_SystemDevice in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of
511 mandatory properties.

512 6.5.2.1.3 Behavior Requirements

513 6.5.2.1.3.1 Preconditions

514 `$instance` represents the instance of a CIM_ComputerSystem, which is referenced by
515 CIM_SystemDevice.

516 6.5.2.1.3.2 Pseudo Code

```
517 &smShowAssociationInstances ( "CIM_SystemDevice", $instance.getObjectPath() );  
518 &smEnd;
```

519 6.5.2.2 Show Command Form for a Single Instance Target – CIM_IndicatorLED Reference

520 This command form is used to show a single instance of CIM_SystemDevice. This command form
521 corresponds to a `show` command issued against a single instance of CIM_SystemDevice, where only one
522 reference is specified and the reference is to the instance of CIM_IndicatorLED.

523 6.5.2.2.1 Command Form

```
524 show <CIM_SystemDevice single instance>
```

525 6.5.2.2.2 CIM Requirements

526 See CIM_IndicatorLED in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of mandatory
527 properties.

528 6.5.2.2.3 Behavior Requirements

529 6.5.2.2.3.1 Preconditions

530 \$instance represents the instance of CIM_IndicatorLED which is referenced by the CIM_SystemDevice
531 association.

532 6.5.2.2.3.2 Pseudo Code

```
533 &smShowAssociationInstances ( "CIM_SystemDevice", $instance.getObjectPath() );  
534 &smEnd;
```

535 6.5.2.3 Show Command Form for a Single Instance Target – Both References

536 This command form is for the *show* verb applied to a single instance. This command form corresponds to
537 the *show* command issued against CIM_SystemDevice where both references are specified; therefore,
538 the desired instance is unambiguously identified.

539 6.5.2.3.1 Command Form

```
540 show <CIM_SystemDevice single instance>
```

541 6.5.2.3.2 CIM Requirements

542 See CIM_IndicatorLED in the “CIM Elements” section of the [Indicator LED Profile](#) for the list of mandatory
543 properties.

544 6.5.2.3.3 Behavior Requirements

545 6.5.2.3.3.1 Preconditions

546 \$instanceA represents the referenced instance of CIM_IndicatorLED through CIM_SystemDevice
547 association.

548 \$instanceB represents the instance of CIM_ComputerSystem which is referenced by
549 CIM_SystemDevice.

550 6.5.2.3.3.2 Pseudo Code

```
551 &smShowAssociationInstance ( "CIM_SystemDevice", $instanceA.getObjectPath(),  
552     $instanceB.getObjectPath() );  
553 &smEnd;
```

554

555
556
557
558
559

ANNEX A (informative)

Change Log

Version	Date	Author	Description
1.0.0	2009-06-04		DMTF Standard Release

560