



1
2
3
4

Document Number: DSP0803

Date: 2009-06-04

Version: 1.0.0

5 **SM CLP Admin Domain Profile SM CLP**
6 **Command Mapping**

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

34

CONTENTS

35	Foreword	5
36	Introduction	6
37	1 Scope	7
38	2 Normative References.....	7
39	2.1 Approved 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_AdminDomain.....	9
45	6.2 CIM_ConcreteCollection.....	11
46	6.3 CIM_MemberOfCollection	12
47	6.4 CIM_OwningCollectionElement.....	15
48	6.5 CIM_SystemComponent.....	17
49	ANNEX A (informative) Change Log	20
50		

51 Tables

52	Table 1 – Command Verb Requirements for CIM_AdminDomain.....	9
53	Table 2 – Command Verb Requirements for CIM_ConcreteCollection.....	11
54	Table 3 – Command Verb Requirements for CIM_MemberOfCollection	13
55	Table 4 – Command Verb Requirements for CIM_OwningCollectionElement.....	15
56	Table 5 – Command Verb Requirements for CIM_SystemComponent.....	17
57		

59

Foreword

60 The *SM CLP Admin Domain Profile SM CLP Command Mapping* (DSP0803) was prepared by the Server
61 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 DTMF Server Management Working
67 Group:

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

75

76

Introduction

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

81 The target audience for this specification is implementers of the SM CLP support for the [SM CLP Admin](#)
82 [Domain Profile](#).

83 SM CLP Admin Domain Profile SM CLP Command Mapping

84 1 Scope

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

87 2 Normative References

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

91 2.1 Approved References

92 DMTF DSP1007, *SM CLP Admin Domain Profile 1.0*,
93 http://www.dmtf.org/standards/published_documents/DSP1007_1.0.pdf

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

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

98 Other References

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

101 3 Terms and Definitions

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

103 3.1

104 **can**

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

106 3.2

107 **cannot**

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

109 3.3

110 **conditional**

111 indicates requirements to be followed strictly in order to conform to the document when the specified
112 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 **IETF**
154 Internet Engineering Task Force

155 **4.5**
 156 **SM**
 157 Server Management

158 **4.6**
 159 **SMI-S**
 160 Storage Management Initiative Specification

161 **4.7**
 162 **SNIA**
 163 Storage Networking Industry Association

164 **4.8**
 165 **UFsT**
 166 User Friendly selection Tag

167 **5 Recipes**

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

- 170 • smShowInstance()
- 171 • smShowInstances()
- 172 • smShowAssociationInstances()
- 173 • smShowAssociationInstance()

174 This mapping does not define any recipes for local reuse.

175 **6 Mappings**

176 The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in
 177 the [SM CLP Admin Domain Profile](#).

178 **6.1 CIM_AdminDomain**

179 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

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

185 **Table 1 – Command Verb Requirements for CIM_AdminDomain**

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	

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

188 6.1.1 Ordering of Results

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

- 191 • Results for `CIM_AdminDomain` are unordered; therefore, no algorithm is defined.

192 6.1.2 Show

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

195 The `show` verb is used to display information about the management domain.

196 6.1.2.1 Show a Single Instance

197 This command form is for the `show` verb applied to a single instance of `CIM_AdminDomain`.

198 6.1.2.1.1 Command Form

```
199 show <CIM_AdminDomain single instance>
```

200 6.1.2.1.2 CIM Requirements

201 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

202 6.1.2.1.3 Behavior Requirements

203 6.1.2.1.3.1 Preconditions

204 The Admin Domain is a special case where we never show anything but `ElementName`.

205 6.1.2.1.3.2 Pseudo Code

```
206 $instance=<CIM_AdminDomain single instance>
207 if (#all == true) {
208     #propertylist[] = { "ElementName" };
209 }
210 else {
211     #propertylist[] = {};
212 }
213 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
214 &smEnd;
```

215 6.2 CIM_ConcreteCollection

216 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

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

222 **Table 2 – Command Verb Requirements for CIM_ConcreteCollection**

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	

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

225 6.2.1 Ordering of Results

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

- 228 • Results for `CIM_ConcreteCollection` are unordered; therefore, no algorithm is defined.

229 6.2.2 Show

230 This section describes how to implement the `show` verb when applied to an instance of
 231 `CIM_ConcreteCollection`. Implementations shall support the use of the `show` verb with
 232 `CIM_ConcreteCollection`.

233 The `show` verb is used to display information about the collection.

234 6.2.2.1 Show a Single Instance

235 This command form is for the `show` verb applied to a single instance of `CIM_ConcreteCollection`.

236 6.2.2.1.1 Command Form

237 `show <CIM_ConcreteCollection single instance>`

238 6.2.2.1.2 CIM Requirements

239 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

240 6.2.2.1.3 Behavior Requirements

241 6.2.2.1.3.1 Preconditions

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

243 6.2.2.1.3.2 Pseudo Code

```
244 $instance=<CIM_ConcreteCollection single instance>
245 #propertylist[] = NULL;
246 if (false == #all) {
247     #propertylist[] = { //all non-key mandatory properties }
248 }
249 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
250 &smEnd;
```

251 6.2.2.2 Show Multiple Instances

252 This command form is for the `show` verb applied to multiple instances of `CIM_ConcreteCollection`. This
253 command form corresponds to UFsT-based selection within a scoping system.

254 6.2.2.2.1 Command Form

```
255 show <CIM_ConcreteCollection multiple instances>
```

256 6.2.2.2.2 CIM Requirements

257 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

258 6.2.2.2.3 Behavior Requirements

259 6.2.2.2.3.1 Preconditions

260 \$containerInstance contains the instance of `CIM_AdminDomain` for which we are displaying scoped
261 `CIM_ConcreteCollection` instances. The CLP Admin Domain Profile requires that the
262 `CIM_ConcreteCollection` instance be associated with its scoping `AdminDomain` instance via an instance
263 of the `CIM_OwningCollectionElement` association.

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

265 6.2.2.2.3.2 Pseudo Code

```
266 #propertylist[] = NULL;
267 if (false == #all) {
268     #propertylist[] = { //all non-key mandatory properties }
269 }
270 &smShowInstances ( "CIM_ConcreteCollection", "CIM_OwningCollectionElement",
271     $containerInstance.getObjectPath(), #propertylist[] );
272 &smEnd;
```

273 6.3 CIM_MemberOfCollection

274 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

275 Table 3 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of
276 the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
277 verb and target. Table 3 is for informational purposes only; in case of a conflict between Table 3 and

278 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 279 information in Table 3.

280 **Table 3 – Command Verb Requirements for CIM_MemberOfCollection**

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	

281 No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,
 282 `reset`, `set`, `start`, and `stop`.

283 6.3.1 Ordering of Results

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

- 286 • Results for `CIM_MemberOfCollection` are unordered; therefore, no algorithm is defined.

287 6.3.2 Show

288 This section describes how to implement the `show` verb when applied to an instance of
 289 `CIM_MemberOfCollection`. Implementations shall support the use of the `show` verb with
 290 `CIM_MemberOfCollection`.

291 The `show` command is used to display information about the `CIM_MemberOfCollection` instance or
 292 instances.

293 6.3.2.1 Show Multiple Instances – CIM_ConcreteCollection Reference

294 This command form is for the `show` verb applied to multiple instances. This command form corresponds
 295 to a `show` command issued against `CIM_MemberOfCollection` where only one reference is specified and
 296 the reference is to an instance of `CIM_ConcreteCollection`.

297 6.3.2.1.1 Command Form

```
298 show <CIM_MemberOfCollection multiple instances>
```

299 6.3.2.1.2 CIM Requirements

300 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

301 6.3.2.1.3 Behavior Requirements

302 6.3.2.1.3.1 Preconditions

303 `instance` contains the instance of `CIM_ConcreteCollection` which is referenced by
 304 `CIM_MemberOfCollection`.

305 6.3.2.1.3.2 Pseudo Code

```
306 &smShowAssociationInstances ( "CIM_MemberOfCollection", $instance.getObjectPath() );  
307 &smEnd;
```

308 6.3.2.2 Show a Single Instance – CIM_PhysicalElement Reference

309 This command form is for the `show` verb applied to a single instance. This command form corresponds to
310 the `show` command issued against `CIM_MemberOfCollection` where the reference specified is to an
311 instance of `CIM_PhysicalElement`. An instance of `CIM_PhysicalElement` is referenced by exactly one
312 instance of `CIM_MemberOfCollection`. Therefore, a single instance will be returned.

313 6.3.2.2.1 Command Form

```
314 show <CIM_MemberOfCollection single instance>
```

315 6.3.2.2.2 CIM Requirements

316 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

317 6.3.2.2.3 Behavior Requirements**318 6.3.2.2.3.1 Preconditions**

319 `$instance` contains the instance of `CIM_PhysicalElement` which is referenced by
320 `CIM_MemberOfCollection`.

321 6.3.2.2.3.2 Pseudo Code

```
322 &smShowAssociationInstances ( "CIM_MemberOfCollection", $instance.getObjectPath() );  
323 &smEnd;
```

324 6.3.2.3 Show a Single Instance – Both References

325 This command form is for the `show` verb applied to a single instance. This command form corresponds to
326 the `show` command issued against `CIM_MemberOfCollection` where both references are specified and
327 therefore the desired instance is unambiguously identified.

328 6.3.2.3.1 Command Form

```
329 show <CIM_MemberOfCollection single instance>
```

330 6.3.2.3.2 CIM Requirements

331 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

332 6.3.2.3.3 Behavior Requirements**333 6.3.2.3.3.1 Preconditions**

334 `$instanceA` contains the instance of `CIM_ConcreteCollection` which is referenced by
335 `CIM_MemberOfCollection`.

336 `$instanceB` contains the instance of `CIM_PhysicalElement` which is referenced by
337 `CIM_MemberOfCollection`.

338 **6.3.2.3.2 Pseudo Code**

```

339 &smShowAssociationInstance ( "CIM_MemberOfCollection", $instanceA.getObjectPath(),
340     $instanceB.getObjectPath() );
341 &smEnd;

```

342 **6.4 CIM_OwningCollectionElement**

343 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

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

349 **Table 4 – Command Verb Requirements for CIM_OwningCollectionElement**

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 6.4.2.
start	Not supported	
stop	Not supported	

350 No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,
 351 `reset`, `set`, `start`, and `stop`.

352 **6.4.1 Ordering of Results**

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

- 355 • Results for `CIM_OwningCollectionElement` are unordered; therefore, no algorithm is defined.

356 **6.4.2 Show**

357 This section describes how to implement the `show` verb when applied to an instance of
 358 `CIM_OwningCollectionElement`. Implementations shall support the use of the `show` verb with
 359 `CIM_OwningCollectionElement`.

360 The `show` command is used to display information about the `CIM_OwningCollectionElement` instance or
 361 instances.

362 **6.4.2.1 Show a Single Instance – CIM_AdminDomain Reference**

363 This command form is for the `show` verb applied to a single instance. This command form corresponds to
 364 a `show` command issued against `CIM_OwningCollectionElement` where only one reference is specified
 365 and the reference is to an instance of `CIM_AdminDomain`.

366 6.4.2.1.1 Command Form

```
367 show <CIM_OwningCollectionElement single instance>
```

368 6.4.2.1.2 CIM Requirements

369 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

370 6.4.2.1.3 Behavior Requirements

371 6.4.2.1.3.1 Preconditions

372 \$instance contains the instance of CIM_AdminDomain which is referenced by
373 CIM_OwningCollectionElement.

374 6.4.2.1.3.2 Pseudo Code

```
375 &smShowAssociationInstances ( "CIM_OwningCollectionElement",  
376     $instance.getObjectPath() );  
377 &smEnd;
```

378 6.4.2.2 Show a Single Instance – CIM_ConcreteCollection Reference

379 This command form is for the `show` verb applied to a single instance. This command form corresponds to
380 a `show` command issued against `CIM_OwningCollectionElement` where the reference specified is to an
381 instance of `CIM_ConcreteCollection`. An instance of `CIM_ConcreteCollection` is referenced by exactly one
382 instance of `CIM_OwningCollectionElement`. Therefore, a single instance will be returned.

383 6.4.2.2.1 Command Form

```
384 show <CIM_OwningCollectionElement single instance>
```

385 6.4.2.2.2 CIM Requirements

386 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

387 6.4.2.2.3 Behavior Requirements

388 6.4.2.2.3.1 Preconditions

389 \$instance contains the instance of `CIM_ConcreteCollection` which is referenced by
390 `CIM_OwningCollectionElement`.

391 6.4.2.2.3.2 Pseudo Code

```
392 &smShowAssociationInstances ( "CIM_OwningCollectionElement",  
393     $instance.getObjectPath() );  
394 &smEnd;
```

395 6.4.2.3 Show Single Instance – Both References

396 This command form is for the `show` verb applied to a single instance. This command form corresponds to
397 the `show` command issued against `CIM_OwningCollectionElement` where both references are specified
398 and therefore the desired instance is unambiguously identified.

399 6.4.2.3.1 Command Form

```
400 show <CIM_OwningCollectionElement single instance>
```


401 **6.4.2.3.2 CIM Requirements**

402 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

403 **6.4.2.3.3 Behavior Requirements**

404 **6.4.2.3.3.1 Preconditions**

405 \$instanceA contains the instance of CIM_AdminDomain which is referenced by
406 CIM_OwningCollectionElement.

407 \$instanceB contains the instance of CIM_ConcreteCollection which is referenced by
408 CIM_OwningCollectionElement.

409 **6.4.2.3.3.2 Pseudo Code**

```
410 &smShowAssociationInstance ( "CIM_OwningCollectionElement",
411     $instanceA.getObjectPath(), $instanceB.getObjectPath() );
412 &smEnd;
```

413 **6.5 CIM_SystemComponent**

414 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

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

420 **Table 5 – Command Verb Requirements for CIM_SystemComponent**

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	

421 No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,
422 `reset`, `set`, `start`, and `stop`.

423 **6.5.1 Ordering of Results**

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

- 426 • Results for CIM_SystemComponent are unordered; therefore, no algorithm is defined.

427 6.5.2 Show

428 This section describes how to implement the `show` verb when applied to an instance of
429 `CIM_SystemComponent`. Implementations shall support the use of the `show` verb with
430 `CIM_SystemComponent`.

431 The `show` command is used to display information about the `CIM_SystemComponent` instance or
432 instances.

433 6.5.2.1 Show Multiple Instances – CIM_AdminDomain Reference

434 This command form is for the `show` verb applied to multiple instances. This command form corresponds
435 to a `show` command issued against `CIM_SystemComponent` where only one reference is specified and
436 the reference is to an instance of `CIM_AdminDomain`.

437 6.5.2.1.1 Command Form

```
438 show <CIM_SystemComponent multiple instances>
```

439 6.5.2.1.2 CIM Requirements

440 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

441 6.5.2.1.3 Behavior Requirements

442 6.5.2.1.3.1 Preconditions

443 `$instance` contains the instance of `CIM_AdminDomain` which is referenced by
444 `CIM_SystemComponent`.

445 6.5.2.1.3.2 Pseudo Code

```
446 &smShowAssociationInstances ( "CIM_SystemComponent", $instance.getObjectPath() );  
447 &smEnd;
```

448 6.5.2.2 Show a Single Instance – CIM_ComputerSystem Reference

449 This command form is for the `show` verb applied to a single instance. This command form corresponds to
450 a `show` command issued against `CIM_SystemComponent` where the reference specified is to an instance
451 of `CIM_ComputerSystem`. An instance of `CIM_ComputerSystem` is referenced by exactly one instance of
452 `CIM_SystemComponent` in the [SM CLP Admin Domain Profile](#). Therefore, a single instance will be
453 returned for this mapping.

454 6.5.2.2.1 Command Form

```
455 show <CIM_SystemComponent single instance>
```

456 6.5.2.2.2 CIM Requirements

457 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

458 6.5.2.2.3 Behavior Requirements

459 6.5.2.2.3.1 Preconditions

460 `$instance` contains the instance of `CIM_AdminDomain` which is referenced by
461 `CIM_SystemComponent`.

462 6.5.2.2.3.2 Pseudo Code

```
463 &smShowAssociationInstances ( "CIM_SystemComponent", $instance.getObjectPath() );  
464 &smEnd;
```

465 6.5.2.3 Show a Single Instance – Both References

466 This command form is for the `show` verb applied to a single instance. This command form corresponds to
467 a `show` command issued against `CIM_SystemComponent` where both references are specified and
468 therefore the desired instance is unambiguously identified.

469 6.5.2.3.1 Command Form

```
470 show <CIM_SystemComponent single instance>
```

471 6.5.2.3.2 CIM Requirements

472 See the “CIM Elements” section in the [SM CLP Admin Domain Profile](#).

473 6.5.2.3.3 Behavior Requirements**474 6.5.2.3.3.1 Preconditions**

475 `$instanceA` contains the instance of `CIM_ComputerSystem` which is referenced by
476 `CIM_SystemComponent`.

477 `$instanceB` contains the instance of `CIM_AdminDomain` which is referenced by
478 `CIM_SystemComponent`.

479 6.5.2.3.3.2 Pseudo Code

```
480 &smShowAssociationInstance ( "CIM_SystemComponent", $instanceA.getObjectPath(),  
481     $instanceB.getObjectPath() );  
482 &smEnd;
```

483
484
485
486
487

ANNEX A
(informative)

Change Log

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

488