



1
2
3
4

Document Number: DSP0816

Date: 2009-07-14

Version: 1.0.0

5 **Host LAN Network Port Profile SM CLP**
6 **Command 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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

34

CONTENTS

35 Foreword 5

36 Introduction 6

37 1 Scope 7

38 2 Normative References..... 7

39 2.1 Approved References 7

40 2.2 Other References..... 7

41 3 Terms and Definitions..... 7

42 4 Symbols and Abbreviated Terms..... 8

43 5 Recipes..... 9

44 5.1 IShowEndpoint 10

45 6 Mappings..... 10

46 6.1 CIM_ControlledBy..... 10

47 6.2 CIM_ElementCapabilities 12

48 6.3 CIM_EnabledLogicalElementCapabilities..... 15

49 6.4 CIM_HostedAccessPoint 17

50 6.5 CIM_HostedService 19

51 6.6 CIM_LANEndpoint 21

52 6.7 CIM_NetworkPort..... 30

53 6.8 CIM_NetworkPortConfigurationService 35

54 6.9 CIM_PortController 37

55 6.10 CIM_DeviceSAPImplementation 42

56 6.11 CIM_ServiceAffectsElement 44

57 6.12 CIM_SystemDevice 46

58 ANNEX A (informative) Change Log 50

59

60 Tables

61 Table 1 – Local Recipes..... 9

62 Table 2 – Command Verb Requirements for CIM_ControlledBy..... 10

63 Table 3 – Command Verb Requirements for CIM_ElementCapabilities 13

64 Table 4 – Command Verb Requirements for CIM_EnabledLogicalElementCapabilities..... 15

65 Table 5 – Command Verb Requirements for CIM_HostedAccessPoint 17

66 Table 6 – Command Verb Requirements for CIM_HostedService 19

67 Table 7 – Command Verb Requirements for CIM_LANEndpoint 22

68 Table 8 – Command Verb Requirements for CIM_NetworkPort..... 31

69 Table 9 – Command Verb Requirements for CIM_NetworkPortConfigurationService 35

70 Table 10 – Command Verb Requirements for CIM_PortController 37

71 Table 11 – Command Verb Requirements for CIM_DeviceSAPImplementation 42

72 Table 12 – Command Verb Requirements for CIM_ServiceAffectsElement 44

73 Table 13 – Command Verb Requirements for CIM_SystemDevice 47

74

76

Foreword

77 The *Host LAN Network Port Profile SM CLP Command Mapping Specification* (DSP0816) was prepared
78 by the Server Management Working Group.

79 **Conventions**

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

82 **Acknowledgements**

83 The authors wish to acknowledge the following participants from the DTMF Server Management Working
84 Group:

- 85 • Aaron Merkin – IBM
- 86 • Jon Hass – Dell
- 87 • Khachatur Papanyan – Dell
- 88 • Jeff Hilland – HP
- 89 • Christina Shaw – HP
- 90 • Perry Vincent – Intel
- 91 • John Leung – Intel

92

93

Introduction

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

98 The target audience for this specification is implementers of the SM CLP support for the [Host LAN Network Port Profile](#).

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

DMTF DSP1035, *Host LAN Network Port Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1035_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

- 131 **3.4**
132 **mandatory**
133 indicates requirements to be followed strictly in order to conform to the document and from which no
134 deviation is permitted
- 135 **3.5**
136 **may**
137 indicates a course of action permissible within the limits of the document
- 138 **3.6**
139 **need not**
140 indicates a course of action permissible within the limits of the document
- 141 **3.7**
142 **optional**
143 indicates a course of action permissible within the limits of the document
- 144 **3.8**
145 **shall**
146 indicates requirements to be followed strictly in order to conform to the document and from which no
147 deviation is permitted
- 148 **3.9**
149 **shall not**
150 indicates requirements to be followed strictly in order to conform to the document and from which no
151 deviation is permitted
- 152 **3.10**
153 **should**
154 indicates that among several possibilities, one is recommended as particularly suitable, without
155 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 156 **3.11**
157 **should not**
158 indicates that a certain possibility or course of action is deprecated but not prohibited

159 **4 Symbols and Abbreviated Terms**

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

- 161 **4.1**
162 **CIM**
163 Common Information Model
- 164 **4.2**
165 **CLP**
166 Command Line Protocol
- 167 **4.3**
168 **DMTF**
169 Distributed Management Task Force

- 170 **4.4**
- 171 **IETF**
- 172 Internet Engineering Task Force
- 173 **4.5**
- 174 **SM**
- 175 Server Management
- 176 **4.6**
- 177 **SMI-S**
- 178 Storage Management Initiative Specification
- 179 **4.7**
- 180 **SNIA**
- 181 Storage Networking Industry Association
- 182 **4.8**
- 183 **UFsT**
- 184 User Friendly selection Tag

185 **5 Recipes**

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

- 188 • smStartRSC()
- 189 • smStopRSC()
- 190 • smResetRSC()
- 191 • smShowInstance()
- 192 • smShowInstances()
- 193 • smSetInstance()
- 194 • smShowAssociationInstances()
- 195 • smShowAssociationInstance()
- 196 • smMakeCommandStatus()
- 197 • smNewInstance()

198 For convenience, Table 1 lists each recipe defined in this mapping which is used for more than one verb
 199 or class mapping.

200 **Table 1 – Local Recipes**

Recipe Name	Description	Definition
IShowEndpoint	Show an instance of CIM_LANEndpoint	See 5.1.

201 The following sections detail Local Recipes defined for use in this mapping.

202 5.1 IShowEndpoint

203 5.1.1 Description

204 IShowEndpoint is a reusable recipe for displaying an instance of CIM_LANEndpoint. A recipe is defined
205 for re-use by the `show` and `create` verbs applied to CIM_LANEndpoint.

206 5.1.2 Preconditions

207 `$endpoint` contains the instance of CIM_LANEndpoint to display.

208 `#all` indicates whether the “-all” option was specified.

209 5.1.3 Pseudo Code

```
210 sub lShowEndpoint($endpoint, #all)
211 {
212 #propertylist[] = NULL;
213 //if we're not displaying all of the properties, provide a list
214 if (false == #all)
215     {
216     #propertylist[] = { "ProtocolIFType", "MACAddress", "RequestedState",
217     "EnabledState", "ElementName" };
218     }
219 &smShowInstance ( $endpoint.GetObjectPath(), #propertyList[] );
220 &smEnd;
221 } //lShowEndpoint()
```

222 6 Mappings

223 The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in
224 the [Host LAN Network Port Profile](#). Requirements specified here related to support for a CLP verb for a
225 particular class are solely within the context of this profile.

226 6.1 CIM_ControlledBy

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

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

233 **Table 2 – Command Verb Requirements for CIM_ControlledBy**

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 6.2.2.
start	Not supported	
stop	Not supported	

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

236 6.1.1 Ordering of Results

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

- 239 • Results for CIM_ControlledBy are unordered; therefore, no algorithm is defined.

240 6.1.2 Show

241 This section describes how to implement the show verb when applied to an instance of
242 CIM_ControlledBy. Implementations shall support the use of the show verb with CIM_ControlledBy.

243 The show command is used to display information about the CIM_ControlledBy instance or instances.

244 6.1.2.1 Show Multiple Instances – CIM_PortController Reference

245 This command form is for the show verb applied to multiple instances. This command form corresponds
246 to a show command issued against CIM_ControlledBy where only one reference is specified and the
247 reference is to an instance of CIM_PortController.

248 6.1.2.1.1 Command Form

```
249 show <CIM_ControlledBy multiple instances>
```

250 6.1.2.1.2 CIM Requirements

251 See CIM_ControlledBy in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
252 mandatory properties.

253 6.1.2.1.3 Behavior Requirements

254 6.1.2.1.3.1 Preconditions

255 \$instance contains the instance of CIM_PortController which is referenced by CIM_ControlledBy.

256 6.1.2.1.3.2 Pseudo Code

```
257 &smShowAssociationInstances ( "CIM_ControlledBy", $instance.getObjectPath() );  
258 &smEnd;
```

259 6.1.2.2 Show a Single Instance – CIM_NetworkPort Reference

260 This command form is for the show verb applied to a single instance. This command form corresponds to
261 a show command issued against CIM_ControlledBy where the reference specified is to an instance of
262 CIM_NetworkPort.

263 **6.1.2.2.1 Command Form**264

```
show <CIM_ControlledBy single instance>
```

265 **6.1.2.2.2 CIM Requirements**266 See CIM_ControlledBy in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
267 mandatory properties.268 **6.1.2.2.3 Behavior Requirements**269 **6.1.2.2.3.1 Preconditions**

270 \$instance contains the instance of CIM_NetworkPort which is referenced by CIM_ControlledBy.

271 **6.1.2.2.3.2 Pseudo Code**272

```
&smShowAssociationInstances ( "CIM_ControlledBy", $instance.getObjectPath() );  
273 &smEnd;
```

274 **6.1.2.3 Show a Single Instance – Both References**275 This command form is for the show verb applied to a single instance. This command form corresponds to
276 a show command issued against CIM_ControlledBy where both references are specified and therefore
277 the desired instance is unambiguously identified.278 **6.1.2.3.1 Command Form**279

```
show <CIM_ControlledBy single instance>
```

280 **6.1.2.3.2 CIM Requirements**281 See CIM_ControlledBy in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
282 mandatory properties.283 **6.1.2.3.3 Behavior Requirements**284 **6.1.2.3.3.1 Preconditions**

285 \$instanceA contains the instance of CIM_NetworkPort which is referenced by CIM_ControlledBy.

286 \$instanceB contains the instance of CIM_PortController which is referenced by CIM_ControlledBy.

287 **6.1.2.3.3.2 Pseudo Code**288

```
&smShowAssociationInstance ( "CIM_ControlledBy", $instanceA.getObjectPath(),  
289 $instanceB.getObjectPath() );  
290 &smEnd;
```

291 **6.2 CIM_ElementCapabilities**292 The cd and help verbs shall be supported as described in [DSP0216](#).293 Table 3 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of
294 the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
295 verb and target. Table 3 is for informational purposes only; in case of a conflict between Table 3 and
296 requirements detailed in the following sections, the text detailed in the following sections supersedes the
297 information in Table 3.

298

Table 3 – 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	

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

301 **6.2.1 Ordering of Results**

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

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

305 **6.2.2 Show**

306 This section describes how to implement the `show` verb when applied to an instance of
 307 `CIM_ElementCapabilities`. Implementations shall support the use of the `show` verb with
 308 `CIM_ElementCapabilities`.

309 The `show` command is used to display information about the `CIM_ElementCapabilities` instance or
 310 instances.

311 **6.2.2.1 Show Multiple Instances – CIM_EnabledLogicalElementCapabilities Reference**

312 This command form is for the `show` verb applied to multiple instances. This command form corresponds
 313 to a `show` command issued against `CIM_ElementCapabilities` where only one reference is specified and
 314 the reference is to an instance of `CIM_EnabledLogicalElementCapabilities`.

315 **6.2.2.1.1 Command Form**

316 `show <CIM_ElementCapabilities multiple instances>`

317 **6.2.2.1.2 CIM Requirements**

318 See `CIM_ElementCapabilities` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the
 319 list of mandatory properties.

320 **6.2.2.1.3 Behavior Requirements**

321 **6.2.2.1.3.1 Preconditions**

322 `$instance` contains the instance of `CIM_EnabledLogicalElementCapabilities` which is referenced by
 323 `CIM_ElementCapabilities`.

324 6.2.2.1.3.2 Pseudo Code

```
325 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );  
326 &smEnd;
```

327 6.2.2.2 Show a Single Instance – CIM_EnabledLogicalElement Reference

328 This command form is for the `show` verb applied to a single instance. This command form corresponds to
329 a `show` command issued against `CIM_ElementCapabilities` where the reference specified is to an
330 instance of `CIM_EnabledLogicalElement`. The [Host LAN Network Port Profile](#) specifies the usage of
331 `CIM_EnabledLogicalElementCapabilities` with multiple subclasses of `CIM_EnabledLogicalElement`. A
332 single instance of `CIM_EnabledLogicalElementCapabilities` can be associated with each instance of a
333 `CIM_EnabledLogicalElement` subclass. Therefore, a single instance will be returned.

334 6.2.2.2.1 Command Form

```
335 show <CIM_ElementCapabilities single instance>
```

336 6.2.2.2.2 CIM Requirements

337 See `CIM_ElementCapabilities` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the
338 list of mandatory properties.

339 6.2.2.2.3 Behavior Requirements**340 6.2.2.2.3.1 Preconditions**

341 `$instance` contains the instance of `CIM_PortController`, `CIM_NetworkPort`, or `CIM_LANEndpoint` which
342 is referenced by `CIM_ElementCapabilities`.

343 6.2.2.2.3.2 Pseudo Code

```
344 &smShowAssociationInstances ( "CIM_ElementCapabilities",  
345     $instance.getObjectPath() );  
346 &smEnd;
```

347 6.2.2.3 Show a Single Instance – Both References

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

351 6.2.2.3.1 Command Form

```
352 show <CIM_ElementCapabilities single instance>
```

353 6.2.2.3.2 CIM Requirements

354 See `CIM_ElementCapabilities` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the
355 list of mandatory properties.

356 6.2.2.3.3 Behavior Requirements**357 6.2.2.3.3.1 Preconditions**

358 `$instanceA` contains the instance of `CIM_EnabledLogicalElementCapabilities` which is referenced by
359 `CIM_ElementCapabilities`.

360 \$instanceB contains the instance of CIM_LANEndpoint, CIM_NetworkPort or CIM_PortController which
 361 is referenced by CIM_ElementCapabilities.

362 6.2.2.3.2 Pseudo Code

```
363 &smShowAssociationInstance ( "CIM_ElementCapabilities", $instanceA.getObjectPath(),
364     $instanceB.getObjectPath() );
365 &smEnd;
```

366 6.3 CIM_EnabledLogicalElementCapabilities

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

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

373 **Table 4 – Command Verb Requirements for CIM_EnabledLogicalElementCapabilities**

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	

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

376 6.3.1 Ordering of Results

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

- 379 • Results for CIM_EnabledLogicalElementCapabilities are unordered; therefore, no algorithm is
 380 defined.

381 6.3.2 Show

382 This section describes how to implement the `show` verb when applied to an instance of
 383 CIM_EnabledLogicalElementCapabilities. Implementations shall support the use of the `show` verb with
 384 CIM_EnabledLogicalElementCapabilities.

385 The `show` verb is used to display information about an instance or instances of the
 386 CIM_EnabledLogicalElementCapabilities class.

387 6.3.2.1 Show a Single Instance

388 This command form is for the `show` verb applied to a single instance of
389 `CIM_EnabledLogicalElementCapabilities`.

390 6.3.2.1.1 Command Form

```
391 show <CIM_EnabledLogicalElementCapabilities single instance>
```

392 6.3.2.1.2 CIM Requirements

393 See `CIM_EnabledLogicalElementCapabilities` in the “CIM Elements” section of the [Host LAN Network](#)
394 [Port Profile](#) for the list of mandatory properties.

395 6.3.2.1.3 Behavior Requirements

396 6.3.2.1.3.1 Preconditions

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

398 6.3.2.1.3.2 Pseudo Code

```
399 $instance=<CIM_EnabledLogicalElementCapabilities single instance>  
400 #propertylist[] = NULL;  
401 if ( false == #all)  
402 {  
403     #propertylist[] = {“RequestedStatesSupported”, “ElementNameEditSupported”,  
404         “MaxElementNameLen”}  
405 }  
406 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );  
407 &smEnd;
```

408 6.3.2.2 Show Multiple Instances

409 This command form is for the `show` verb applied to multiple instances of
410 `CIM_EnabledLogicalElementCapabilities`. This command form corresponds to UfsT-based selection
411 within a capabilities collection.

412 6.3.2.2.1 Command Form

```
413 show <CIM_EnabledLogicalElementCapabilities multiple instances>
```

414 6.3.2.2.2 CIM Requirements

415 See `CIM_EnabledLogicalElementCapabilities` in the “CIM Elements” section of the [Host LAN Network](#)
416 [Port Profile](#) for the list of mandatory properties.

417 6.3.2.2.3 Behavior Requirements

418 6.3.2.2.3.1 Preconditions

419 `$containerInstance` contains the instance of `CIM_ConcreteCollection` for which contained
420 `CIM_Capabilities` instances are displayed. `CIM_Capabilities` instances are addressed via an aggregating
421 instance of `CIM_ConcreteCollection`.

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

423 6.3.2.2.3.2 Pseudo Code

```

424 #propertylist[] = NULL;
425 if ( false == #all)
426     {
427         #propertylist[] = {"RequestedStatesSupported", "ElementNameEditSupported",
428             "MaxElementNameLen"}
429     }
430 &smShowInstances ( "CIM_EnabledLogicalElementCapabilities", "CIM_MemberOfCollection",
431     $containerInstance.getObjectPath(), #propertylist[] );
432 &smEnd;

```

433 6.4 CIM_HostedAccessPoint

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

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

440 **Table 5 – Command Verb Requirements for CIM_HostedAccessPoint**

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	

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

443 6.4.1 Ordering of Results

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

- 446 • Results for `CIM_HostedAccessPoint` are unordered; therefore, no algorithm is defined.

447 6.4.2 Show

448 This section describes how to implement the `show` verb when applied to an instance of
 449 `CIM_HostedAccessPoint`. Implementations shall support the use of the `show` verb with
 450 `CIM_HostedAccessPoint`.

451 The `show` command is used to display information about the `CIM_HostedAccessPoint` instance or
452 instances.

453 **6.4.2.1 Show Multiple Instances – CIM_ComputerSystem Reference**

454 This command form is for the `show` verb applied to multiple instances. This command form corresponds
455 to a `show` command issued against `CIM_HostedAccessPoint` where only one reference is specified and
456 the reference is to an instance of `CIM_ComputerSystem`.

457 **6.4.2.1.1 Command Form**

```
458 show <CIM_HostedAccessPoint multiple instances>
```

459 **6.4.2.1.2 CIM Requirements**

460 See `CIM_HostedAccessPoint` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the
461 list of mandatory properties.

462 **6.4.2.1.3 Behavior Requirements**

463 **6.4.2.1.3.1 Preconditions**

464 `$instance` contains the instance of `CIM_ComputerSystem` which is referenced by
465 `CIM_HostedAccessPoint`.

466 **6.4.2.1.3.2 Pseudo Code**

```
467 &smShowAssociationInstances ( "CIM_HostedAccessPoint", $instance.getObjectPath() );  
468 &smEnd;
```

469 **6.4.2.2 Show a Single Instance – CIM_LANEndpoint Reference**

470 This command form is for the `show` verb applied to a single instance. This command form corresponds to
471 a `show` command issued against `CIM_HostedAccessPoint` where the reference specified is to an
472 instance of `CIM_LANEndpoint`. An instance of `CIM_LANEndpoint` is referenced by exactly one instance of
473 `CIM_HostedAccessPoint`. Therefore, a single instance will be returned.

474 **6.4.2.2.1 Command Form**

```
475 show <CIM_HostedAccessPoint single instance>
```

476 **6.4.2.2.2 CIM Requirements**

477 See `CIM_HostedAccessPoint` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the
478 list of mandatory properties.

479 **6.4.2.2.3 Behavior Requirements**

480 **6.4.2.2.3.1 Preconditions**

481 `$instance` contains the instance of `CIM_LANEndpoint` which is referenced by `CIM_HostedAccessPoint`.

482 **6.4.2.2.3.2 Pseudo Code**

```
483 &smShowAssociationInstances ( "CIM_HostedAccessPoint", $instance.getObjectPath() );  
484 &smEnd;
```

485 6.4.2.3 Show a Single Instance – Both References

486 This command form is for the `show` verb applied to a single instance. This command form corresponds to
 487 a `show` command issued against `CIM_HostedAccessPoint` where both references are specified and
 488 therefore the desired instance is unambiguously identified.

489 6.4.2.3.1 Command Form

```
490 show <CIM_HostedAccessPoint single instance>
```

491 6.4.2.3.2 CIM Requirements

492 See `CIM_HostedAccessPoint` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the
 493 list of mandatory properties.

494 6.4.2.3.3 Behavior Requirements

495 6.4.2.3.3.1 Preconditions

496 `$instanceA` contains the instance of `CIM_ComputerSystem` which is referenced by
 497 `CIM_HostedAccessPoint`.

498 `$instanceB` contains the instance of `CIM_LANEndpoint` which is referenced by
 499 `CIM_HostedAccessPoint`.

500 6.4.2.3.3.2 Pseudo Code

```
501 &smShowAssociationInstance ( "CIM_HostedAccessPoint", $instanceA.getObjectPath(),  
502     $instanceB.getObjectPath() );  
503 &smEnd;
```

504 6.5 CIM_HostedService

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

506 Table 6 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of
 507 the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
 508 verb and target. Table 6 is for informational purposes only; in case of a conflict between Table 6 and
 509 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 510 information in Table 6.

511 **Table 6 – Command Verb Requirements for `CIM_HostedService`**

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	

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

514 6.5.1 Ordering of Results

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

- 517 • Results for CIM_HostedService are unordered; therefore, no algorithm is defined.

518 6.5.2 Show

519 This section describes how to implement the `show` verb when applied to an instance of
520 CIM_HostedService. Implementations shall support the use of the `show` verb with CIM_HostedService.

521 The `show` command is used to display information about the CIM_HostedService instance or instances.

522 6.5.2.1 Show Multiple Instances – CIM_ComputerSystem Reference

523 This command form is for the `show` verb applied to multiple instances. This command form corresponds
524 to a `show` command issued against CIM_HostedService where only one reference is specified and the
525 reference is to an instance of CIM_ComputerSystem.

526 6.5.2.1.1 Command Form

```
527 show <CIM_HostedService multiple instances>
```

528 6.5.2.1.2 CIM Requirements

529 See CIM_HostedService in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
530 mandatory properties.

531 6.5.2.1.3 Behavior Requirements

532 6.5.2.1.3.1 Preconditions

533 `$instance` contains the instance of CIM_ComputerSystem which is referenced by CIM_HostedService.

534 6.5.2.1.3.2 Pseudo Code

```
535 &smShowAssociationInstances ( "CIM_HostedService", $instance.getObjectPath() );  
536 &smEnd;
```

537 6.5.2.2 Show a Single Instance – CIM_NetworkPortConfigurationService Reference

538 This command form is for the `show` verb applied to a single instance. This command form corresponds to
539 a `show` command issued against CIM_HostedService where the reference specified is to an instance of
540 CIM_NetworkPortConfigurationService. An instance of CIM_NetworkPortConfigurationService is
541 referenced by exactly one instance of CIM_HostedService. Therefore, a single instance will be returned.

542 6.5.2.2.1 Command Form

```
543 show <CIM_HostedService single instance>
```

544 6.5.2.2.2 CIM Requirements

545 See CIM_HostedService in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
546 mandatory properties.

547 **6.5.2.2.3 Behavior Requirements**

548 **6.5.2.2.3.1 Preconditions**

549 `$instance` contains the instance of `CIM_NetworkPortConfigurationService` which is referenced by
550 `CIM_HostedService`.

551 **6.5.2.2.3.2 Pseudo Code**

```
552 &smShowAssociationInstances ( "CIM_HostedService", $instance.getObjectPath() );
553 &smEnd;
```

554 **6.5.2.3 Show a Single Instance – Both References**

555 This command form is for the `show` verb applied to a single instance. This command form corresponds to
556 a `show` command issued against `CIM_HostedService` where both references are specified and therefore
557 the desired instance is unambiguously identified.

558 **6.5.2.3.1 Command Form**

```
559 show <CIM_HostedService single instance>
```

560 **6.5.2.3.2 CIM Requirements**

561 See `CIM_HostedService` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
562 mandatory properties.

563 **6.5.2.3.3 Behavior Requirements**

564 **6.5.2.3.3.1 Preconditions**

565 `$instanceA` contains the instance of `CIM_ComputerSystem` which is referenced by
566 `CIM_HostedService`.

567 `$instanceB` contains the instance of `CIM_NetworkPortConfigurationService` which is referenced by
568 `CIM_HostedService`.

569 **6.5.2.3.3.2 Pseudo Code**

```
570 &smShowAssociationInstance ( "CIM_HostedService", $instanceA.getObjectPath(),
571     $instanceB.getObjectPath() );
572 &smEnd;
```

573 **6.6 CIM_LANEndpoint**

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

575 Table 7 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of
576 the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
577 verb and target. Table 7 is for informational purposes only; in case of a conflict between Table 7 and
578 requirements detailed in the following sections, the text detailed in the following sections supersedes the
579 information in Table 7.

580

Table 7 – Command Verb Requirements for CIM_LANEndpoint

Command Verb	Requirement	Comments
create	May	See 6.6.2.
delete	May	See 6.6.3.
dump	Not supported	
load	Not supported	
reset	May	See 6.6.4.
set	May	See 6.6.5.
show	Shall	See 6.6.6.
start	May	See 6.6.7.
stop	May	See 6.6.8.

581 No mapping is defined for the following verbs for the specified target: dump and load.

582 6.6.1 Ordering of Results

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

- 585 • Results for CIM_LANEndpoint are unordered; therefore, no algorithm is defined.

586 6.6.2 Create

587 This section describes how to implement the `create` verb when applied to an instance of
588 CIM_LANEndpoint. Implementations may support the use of the `create` verb with CIM_LANEndpoint.

589 The `create` verb is used to create an additional CIM_LANEndpoint instance bound to a
590 CIM_NetworkPort instance.

591 6.6.2.1 Create Specifying Required/Optional Parameters

592 In order to create an instance of CIM_LANEndpoint, a client is required to supply the desired network
593 address. The client can optionally supply one or more group addresses and alias addresses.

594 6.6.2.1.1 Command Form

```
595 create CIM_LANEndpoint address=<desiredaddress>
596     groupaddresses=<desiregroupaddresses>]
597     alias addresses=<desiredaliasaddresses>]
598     lanid=<desiredlanid>]
```

599 6.6.2.1.2 CIM Requirements

```
600 CIM_NetworkPort
601 uint32 CIM_NetworkPortConfigurationService.AddLANEndpoint(
602     [IN] uint16 Port,
603     [OUT] REF CIM_LANEndpoint Endpoint,
604     [IN] string Address,
605     [IN] string LANID,
606     [IN] string AliasAddresses,
607     [IN] string GroupAddresses);
608 CIM_LANEndpoint
```

609 **6.6.2.1.3 Behavior Requirements**610 **6.6.2.1.3.1 Preconditions**

611 \$port contains the CIM_NetworkPort instance for which a new endpoint is created. When using SM ME
 612 Addressing, this instance would be the container instance specified in the Resultant Address.

613 **6.6.2.1.3.2 Pseudo Code**

```

614 //the desired address is required, if it is not specified, fail
615 if (NULL == <desiredaddress>) {
616     &smMakeError();
617 }
618 #address = <desiredaddress>;
619 if (NULL != <desiredgroupaddresses>) {
620     //convert from command line array assignment to array variable
621     #groupaddresses[] = <desiredgroupaddresses>;
622 }
623 else {
624     #groupaddresses[] = NULL;
625 }
626 if (NULL != <desiredaliasaddresses>) {
627     //convert from command line array assignment to array variable
628     #aliasaddresses[] = <desiredaliasaddresses>;
629 }
630 else {
631     #aliasaddresses[] = NULL;
632 }
633 if (NULL != <desiredlanid>) {
634     //convert from command line parameter to variable
635     #LANID = <desiredlanid>;
636 }
637 else {
638     #LANID = NULL;
639 }
640 //find the NetworkPortConfigurationService for the target CIM_NetworkPort
641 $Services[] = smOpAssociators(
642     $instancePath->,
643     "CIM_ServiceAffectsElement",
644     "CIM_NetworkPortConfigurationService",
645     NULL,
646     NULL);
647 //there should be one, if not, make an error
648 if (NULL = $Services[0]) {
649     //unsupported
650     $OperationError = smNewInstance("CIM_Error");
651     //CIM_ERR_NOT_SUPPORTED
652     $OperationError.CIMStatusCode = 7;
653     //Other
654     $OperationError.ErrorType = 1;
655     //Low
  
```

```
656     $OperationError.PerceivedSeverity = 2;
657     $OperationError.OwningEntity = DMTF:SMCLP;
658     $OperationError.MessageID = 0x00000001;
659     $OperationError.Message = "Operation is not supported.";
660     &smAddError($job, $OperationError);
661     &smMakeCommandStatus($job);
662     &smEnd;
663 }
664 //build the parameter lists and invoke the method
665 %InArguments[] = {newArgument("Port", $Port.GetObjectPath()),
666     newArgument ("LANID", #LANID),
667     newArgument ("AliasAddresses", #aliasaddresses[]),
668     newArgument ("GroupAddresses", #groupaddresses[]),
669     %OutArguments[] = { newArgument("Endpoint",
670         $Endpoint.GetObjectPath()) };
671 //invoke method
672 #returnStatus = smOpInvokeMethod ($Service->,
673     "AddLANEndpoint",
674     %InArguments[],
675     %OutArguments[]);
676 // process return code to CLP Command Status
677 if (0 != #Error.code) {
678     //method invocation failed
679     if ( (NULL != #Error.$error) && (NULL != #Error.$error[0]) ) {
680         //if the method invocation contains an embedded error
681         //use it for the Error for the overall job
682         &smAddError($job, #Error.$error[0]);
683         &smMakeCommandStatus($job);
684         &smEnd;
685     }
686     else {
687         //operation failed, but no detailed error instance, need to make one up
688         //make an Error instance and associate with job for Operation
689         $OperationError = smNewInstance("CIM_Error");
690         //CIM_ERR_FAILED
691         $OperationError.CIMStatusCode = 1;
692         //Software Error
693         $OperationError.ErrorType = 4;
694         //Unknown
695         $OperationError.PerceivedSeverity = 0;
696         $OperationError.OwningEntity = DMTF:SMCLP;
697         $OperationError.MessageID = 0x00000009;
698         $OperationError.Message = "An internal software error has occurred.";
699         &smAddError($job, $OperationError);
700         &smMakeCommandStatus($job);
701         &smEnd;
702     }
703 } //if CIM op failed
704 else if (0 == #returnStatus) {
```



```

705 //completed successfully
706 &lShowEndpoint($endpoint, "false");
707 &smEnd;
708 }
709 else if (4 == #returnStatus) {
710 //generic failure
711 $OperationError = smNewInstance("CIM_Error");
712 //CIM_ERR_FAILED
713 $OperationError.CIMStatusCode = 1;
714 //Other
715 $OperationError.ErrorType = 1;
716 //Low
717 $OperationError.PerceivedSeverity = 2;
718 $OperationError.OwningEntity = DMTF:SMCLP;
719 $OperationError.MessageID = 0x00000002;
720 $OperationError.Message = "Failed. No further information is available.";
721 &smAddError($job, $OperationError);
722 &smMakeCommandStatus($job);
723 }
724 else {
725 //invalid parameter
726 $OperationError = smNewInstance("CIM_Error");
727 //CIM_ERR_FAILED
728 $OperationError.CIMStatusCode = 1;
729 //Other
730 $OperationError.ErrorType = 1;
731 //Low
732 $OperationError.PerceivedSeverity = 2;
733 $OperationError.OwningEntity = DMTF:SMCLP;
734 $OperationError.MessageID = 0x00000004;
735 $OperationError.Message = "One or more parameters specified are invalid.";
736 &smAddError($job, $OperationError);
737 &smMakeCommandStatus($job);
738 &smEnd;
739 }

```

740 6.6.3 Delete

741 This section describes how to implement the `delete` verb when applied to an instance of
742 `CIM_LANEndpoint`. Implementations may support the use of the `delete` verb with `CIM_LANEndpoint`.

743 The `delete` command is used to remove an instance of `CIM_LANEndpoint` which represents a virtual
744 MAC.

745 6.6.3.1 Delete a Single Instance

746 Delete a single instance of `CIM_LANEndpoint`.

747 6.6.3.1.1 Command Form

```
748 delete <CIM_LANEndpoint single instance>
```

749 **6.6.3.1.2 CIM Requirements**

750 See CIM_LANEndpoint in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
751 mandatory properties.

752 **6.6.3.1.3 Behavior Requirements**

```
753 $instance=<CIM_LANEndpoint single instance>
754 &smOpDeleteInstance ( $instance.GetObjectPath() );
```

755 **6.6.4 Reset**

756 This section describes how to implement the `reset` verb when applied to an instance of
757 CIM_LANEndpoint. Implementations may support the use of the `reset` verb with CIM_LANEndpoint.

758 The `reset` verb is used to initiate a reset of the CIM_LANEndpoint.

759 **6.6.4.1 Reset a Single Instance**

760 This command form is for the initiation of a reset action against a network endpoint. The mapping is
761 implemented as an invocation of the `RequestStateChange()` method on the instance.

762 **6.6.4.1.1 Command Form**

```
763 reset <CIM_LANEndpoint single instance>
```

764 **6.6.4.1.2 CIM Requirements**

```
765 uint16 EnabledState;
766 uint16 RequestedState;
767 uint32 EnabledLogicalElement.RequestStateChange (
768     [IN] uint16 RequestedState = "reset",
769     [OUT] REF CIM_ConcreteJob Job,
770     [IN] datetime TimeoutPeriod );
```

771 **6.6.4.1.3 Behavior Requirements**

```
772 $instance=<CIM_LANEndpoint single instance>
773 smResetRSC ( $instance.getObjectPath() );
774 &smEnd;
```

775 **6.6.5 Set**

776 This section describes how to implement the `set` verb when it is applied to an instance of
777 CIM_LANEndpoint. Implementations may support the use of the `set` verb with CIM_LANEndpoint.

778 The `set` verb is used to modify descriptive properties of the CIM_LANEndpoint instance.

779 **6.6.5.1 General Usage of Set for a Single Property**

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

782 The requirement for supporting modification of a property using this command form shall be equivalent to
783 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined
784 in the [Host LAN Network Port Profile](#).

785 6.6.5.1.1 Command Form

```
786 set <CIM_LANEndpoint single instance> <propertyname>=<propertyvalue>
```

787 6.6.5.1.2 CIM Requirements

788 See CIM_LANEndpoint in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
789 mandatory properties.

790 6.6.5.1.3 Behavior Requirements

```
791 $instance=<CIM_LANEndpoint single instance>
792 #propertyNames[] = {<propertyname>;
793 #propertyValues[] = {<propertyvalue>;
794 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
795 &smEnd;
```

796 6.6.5.2 General Usage of Set for Multiple Properties

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

800 The requirement for supporting modification of a property using this command form shall be equivalent to
801 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined
802 in the [Host LAN Network Port Profile](#).

803 6.6.5.2.1 Command Form

```
804 set <CIM_LANEndpoint single instance> <propertyname1>=<propertyvalue1>
805 <propertynamen>=<propertyvaluen>
```

806 6.6.5.2.2 CIM Requirements

807 See CIM_LANEndpoint in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
808 mandatory properties.

809 6.6.5.2.3 Behavior Requirements

```
810 $instance=<CIM_LANEndpoint single instance>
811 #propertyNames[] = {<propertyname>;
812 for #i < n
813 {
814     #propertyNames[#i] = <propertyname#i>
815     #propertyValues[#i] = <propertyvalue#i>
816 }
817 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
818 &smEnd;
```

819 6.6.6 Show

820 This section describes how to implement the `show` verb when applied to an instance of
821 CIM_LANEndpoint. Implementations shall support the use of the `show` verb with CIM_LANEndpoint.

822 The `show` verb is used to display information about a network endpoint.

823 6.6.6.1 Show a Single Instance

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

825 6.6.6.1.1 Command Form

```
826 show <CIM_LANEndpoint single instance>
```

827 6.6.6.1.2 CIM Requirements

828 See `CIM_LANEndpoint` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
829 mandatory properties.

830 6.6.6.1.3 Behavior Requirements

831 6.6.6.1.3.1 Preconditions

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

833 6.6.6.1.3.2 Pseudo Code

```
834 $instance=<CIM_LANEndpoint single instance>  
835 &lShowEndpoint ( $instance, #all );  
836 &smEnd;
```

837 6.6.6.2 Show Multiple Instances Scoped by a System

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

840 6.6.6.2.1 Command Form

```
841 show <CIM_LANEndpoint multiple instances>
```

842 6.6.6.2.2 CIM Requirements

843 See `CIM_LANEndpoint` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
844 mandatory properties.

845 6.6.6.2.3 Behavior Requirements

846 6.6.6.2.3.1 Preconditions

847 `$containerInstance` contains the instance of `CIM_ComputerSystem` for which scoped endpoints
848 (`CIM_LANEndpoint` instances) are displayed. The [Host LAN Network Port Profile](#) requires that the
849 `CIM_LANEndpoint` instance be associated with its scoping system via an instance of the
850 `CIM_SystemDevice` association.

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

852 6.6.6.2.3.2 Pseudo Code

```
853 #propertylist[] = NULL;  
854 //this property list will match the property list in lShowEndpoint()  
855 if (false == #all)  
856 {  
857     #propertylist[] = { "ProtocolIFType", "MACAddress", "RequestedState",  
858         "EnabledState", "ElementName" };  
859 }
```

```

860 &smShowInstances ( "CIM_LANEndpoint", "CIM_SystemDevice",
861     $containerInstance.getObjectPath(), #propertylist[] );
862 &smEnd;

```

863 6.6.6.3 Show Multiple Instances Scoped by a Network Port

864 This command form is for the `show` verb applied to multiple instances of `CIM_LANEndpoint`. This
 865 command form corresponds to UFT-based selection within a scoping network port.

866 6.6.6.3.1 Command Form

```

867 show <CIM_LANEndpoint multiple instances>

```

868 6.6.6.3.2 CIM Requirements

869 See `CIM_LANEndpoint` in the "CIM Elements" section of the [Host LAN Network Port Profile](#) for the list of
 870 mandatory properties.

871 6.6.6.3.3 Behavior Requirements

872 6.6.6.3.3.1 Preconditions

873 `$containerInstance` contains the instance of `CIM_NetworkPort` for which scoped endpoints
 874 (`CIM_LANEndpoint` instances) are displayed. The [Host LAN Network Port Profile](#) requires that the
 875 `CIM_LANEndpoint` instance be associated with a scoping network port via an instance of the
 876 `CIM_DeviceSAPImplementation` association.

877 `#all` is true if the "-all" option was specified with the command; otherwise, `#all` is false.

878 6.6.6.3.3.2 Pseudo Code

```

879 #propertylist[] = NULL;
880 //this property list will match the property list in lShowEndpoint()
881 if (false == #all)
882     {
883     #propertylist[] = { "ProtocolIFType", "MACAddress", "RequestedState",
884     "EnabledState", "ElementName" };
885     }
886 &smShowInstances ( "CIM_LANEndpoint", "CIM_DeviceSAPImplementation",
887     $containerInstance.getObjectPath(), #propertylist[] );
888 &smEnd;

```

889 6.6.7 Start

890 This section describes how to implement the `start` verb when applied to an instance of
 891 `CIM_LANEndpoint`. Implementations may support the use of the `start` verb with `CIM_LANEndpoint`.

892 The `start` verb is used to enable an endpoint.

893 6.6.7.1 Start a Single Instance

894 This command form is for the `start` verb applied to a single instance of `CIM_LANEndpoint`.

895 6.6.7.1.1 Command Form

```

896 start <CIM_LANEndpoint single instance>

```

897 6.6.7.1.2 CIM Requirements

```
898 uint16 EnabledState;
899 uint16 RequestedState;
900 uint32 EnabledLogicalElement.RequestStateChange (
901     [IN] uint16 RequestedState = "enabled",
902     [OUT] REF CIM_ConcreteJob Job,
903     [IN] datetime TimeoutPeriod );
```

904 6.6.7.1.3 Behavior Requirements

```
905 $instance=<CIM_LANEndpoint single instance>
906 smStartRSC ( $instance.GetObjectPath() );
907 &smEnd;
```

908 6.6.8 Stop

909 This section describes how to implement the `stop` verb when applied to an instance of
910 `CIM_LANEndpoint`. Implementations may support the use of the `stop` verb with `CIM_LANEndpoint`.

911 The `stop` verb is used to disable an endpoint.

912 6.6.8.1 Stop a Single Instance

913 This command form is for the `stop` verb applied to a single instance of `CIM_LANEndpoint`.

914 6.6.8.1.1 Command Form

```
915 stop <CIM_LANEndpoint single instance>
```

916 6.6.8.1.2 CIM Requirements

```
917 uint16 EnabledState;
918 uint16 RequestedState;
919 uint32 EnabledLogicalElement.RequestStateChange (
920     [IN] uint16 RequestedState = "disabled",
921     [OUT] REF CIM_ConcreteJob Job,
922     [IN] datetime TimeoutPeriod );
```

923 6.6.8.1.3 Behavior Requirements

```
924 $instance=<CIM_LANEndpoint single instance>
925 smStopRSC ( $instance.GetObjectPath() );
926 &smEnd;
```

927 6.7 CIM_NetworkPort

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

929 Table 8 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of
930 the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
931 verb and target. Table 8 is for informational purposes only; in case of a conflict between Table 8 and
932 requirements detailed in the following sections, the text detailed in the following sections supersedes the
933 information in Table 8.

934

Table 8 – Command Verb Requirements for CIM_NetworkPort

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	May	See 6.7.2.
set	May	See 6.7.3.
show	Shall	See 6.7.4.
start	May	See 6.7.5.
stop	May	See 6.7.6.

935 No mapping is defined for the following verbs for the specified target: create, delete, dump, and load.

936 **6.7.1 Ordering of Results**

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

- 939 • Results for CIM_NetworkPort are unordered; therefore, no algorithm is defined.

940 **6.7.2 Reset**

941 This section describes how to implement the `reset` verb when applied to an instance of
 942 CIM_NetworkPort. Implementations may support the use of the `reset` verb with CIM_NetworkPort.

943 The `reset` verb is used to initiate a reset of the CIM_NetworkPort.

944 **6.7.2.1 Reset a Single Instance**

945 This command form is for the initiation of a reset action against a single endpoint. The mapping is
 946 implemented as an invocation of the `RequestStateChange()` method on the instance.

947 **6.7.2.1.1 Command Form**

948 `reset <CIM_NetworkPort single instance>`

949 **6.7.2.1.2 CIM Requirements**

```

950 uint16 EnabledState;
951 uint16 RequestedState;
952 uint32 EnabledLogicalElement.RequestStateChange (
953     [IN] uint16 RequestedState = "enabled",
954     [OUT] REF CIM_ConcreteJob Job,
955     [IN] datetime TimeoutPeriod );
    
```

956 **6.7.2.1.3 Behavior Requirements**

```

957 $instance=<CIM_NetworkPort single instance>
958 smResetRSC ( $instance.GetObjectPath() );
959 &smEnd;
    
```

960 6.7.3 Set

961 This section describes how to implement the `set` verb when it is applied to an instance of
962 `CIM_NetworkPort`. Implementations may support the use of the `set` verb with `CIM_NetworkPort`.

963 The `set` verb is used to modify descriptive properties of the `CIM_NetworkPort` instance.

964 6.7.3.1 General Usage of Set for a Single Property

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

967 The requirement for supporting modification of a property using this command form shall be equivalent to
968 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined
969 in the [Host LAN Network Port Profile](#).

970 6.7.3.1.1 Command Form

```
971 set <CIM_NetworkPort single instance> <propertyname>=<propertyvalue>
```

972 6.7.3.1.2 CIM Requirements

973 See `CIM_NetworkPoint` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
974 modifiable properties.

975 6.7.3.1.3 Behavior Requirements

```
976 $instance=<CIM_NetworkPort single instance>
977 #propertyNames[] = {<propertyname>};
978 #propertyValues[] = {<propertyvalue>};
979 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
980 &smEnd;
```

981 6.7.3.2 General Usage of Set for Multiple Properties

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

985 The requirement for supporting modification of a property using this command form shall be equivalent to
986 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined
987 in the [Host LAN Network Port Profile](#).

988 6.7.3.2.1 Command Form

```
989 set <CIM_NetworkPort Single Instance> <propertyname1>=<propertyvalue1>
990 <propertynamen>=<propertyvaluen>
```

991 6.7.3.2.2 CIM Requirements

992 See `CIM_NetworkPoint` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
993 supported properties.

994 6.7.3.2.3 Behavior Requirements

```
995 $instance=<CIM_NetworkPort single instance>
996 #propertyNames[] = {<propertyname>};
997 for #i < n
```



```

998 {
999   #propertyNames[#i] = <propertname#i>
1000   #propertyValues[#i] = <propertyvalue#i>
1001 }
1002 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
1003 &smEnd;

```

1004 **6.7.4 Show**

1005 This section describes how to implement the `show` verb when applied to an instance of
 1006 `CIM_NetworkPort`. Implementations shall support the use of the `show` verb with `CIM_NetworkPort`.

1007 The `show` verb is used to display information about the network port.

1008 **6.7.4.1 Show a Single Instance**

1009 This command form is for the `show` verb applied to a single instance of `CIM_NetworkPort`.

1010 **6.7.4.1.1 Command Form**

```
1011 show <CIM_NetworkPort single instance>
```

1012 **6.7.4.1.2 CIM Requirements**

1013 See `CIM_NetworkPoint` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
 1014 mandatory properties.

1015 **6.7.4.1.3 Behavior Requirements**

1016 **6.7.4.1.3.1 Preconditions**

1017 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

1018 **6.7.4.1.3.2 Pseudo Code**

```

1019 $instance=<CIM_NetworkPort single instance>
1020 #propertylist[] = NULL;
1021 if (false == #all)
1022 {
1023   #propertylist[] = { "LinkTechnology", "PermanentAddress", "DeviceID",
1024     "ElementName" };
1025 }
1026 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
1027 &smEnd;

```

1028 **6.7.4.2 Show Multiple Instances**

1029 This command form is for the `show` verb applied to multiple instances of `CIM_NetworkPort`. This
 1030 command form corresponds to UFT-based selection within a scoping system.

1031 **6.7.4.2.1 Command Form**

```
1032 show <CIM_NetworkPort multiple instances>
```

1033 **6.7.4.2.2 CIM Requirements**

1034 See CIM_NetworkPort in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
1035 mandatory properties.

1036 **6.7.4.2.3 Behavior Requirements**1037 **6.7.4.2.3.1 Preconditions**

1038 \$containerInstance contains the instance of CIM_ComputerSystem for which scoped network ports
1039 (CIM_NetworkPort instances) are displayed. The [Host LAN Network Port Profile](#) requires that the
1040 CIM_NetworkPort instance be associated with its scoping system via an instance of the
1041 CIM_SystemDevice association.

1042 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1043 **6.7.4.2.3.2 Pseudo Code**

```
1044 #propertylist[] = NULL;
1045 if (false == #all)
1046     {
1047         #propertylist[] = { "LinkTechnology", "PermanentAddress", "DeviceID",
1048             "ElementName" };
1049     }
1050 &smShowInstances ( "CIM_NetworkPort", "CIM_SystemDevice",
1051     $containerInstance.getObjectPath(), #propertylist[] );
1052 &smEnd;
```

1053 **6.7.5 Start**

1054 This section describes how to implement the `start` verb when applied to an instance of
1055 CIM_NetworkPort. Implementations may support the use of the `start` verb with CIM_NetworkPort.

1056 The `start` verb is used to enable a network port.

1057 **6.7.5.1 Start a Single Instance**

1058 This command form is for the `start` verb applied to a single instance of CIM_NetworkPort.

1059 **6.7.5.1.1 Command Form**

```
1060 start <CIM_NetworkPort single instance>
```

1061 **6.7.5.1.2 CIM Requirements**

```
1062 uint16 EnabledState;
1063 uint16 RequestedState;
1064 uint32 EnabledLogicalElement.RequestStateChange (
1065     [IN] uint16 RequestedState = "enabled",
1066     [OUT] REF CIM_ConcreteJob Job,
1067     [IN] datetime TimeoutPeriod );
```

1068 **6.7.5.1.3 Behavior Requirements**

```
1069 $instance=<CIM_NetworkPort single instance>
1070 smStartRSC ( $instance.getObjectPath() );
1071 &smEnd;
```

1072 **6.7.6 Stop**

1073 This section describes how to implement the `stop` verb when applied to an instance of
 1074 `CIM_NetworkPort`. Implementations may support the use of the `stop` verb with `CIM_NetworkPort`.

1075 The `stop` verb is used to disable a network port.

1076 **6.7.6.1 Stop a Single Instance**

1077 This command form is for the `stop` verb applied to a single instance of `CIM_NetworkPort`.

1078 **6.7.6.1.1 Command Form**

1079 `stop <CIM_NetworkPort single instance>`

1080 **6.7.6.1.2 CIM Requirements**

```
1081 uint16 EnabledState;
1082 uint16 RequestedState;
1083 uint32 EnabledLogicalElement.RequestStateChange (
1084     [IN] uint16 RequestedState = "Disabled",
1085     [OUT] REF CIM_ConcreteJob Job,
1086     [IN] datetime TimeoutPeriod );
```

1087 **6.7.6.1.3 Behavior Requirements**

```
1088 $instance=<CIM_NetworkPort single instance>
1089 smStopRSC ( $instance.getObjectPath() );
1090 &smEnd;
```

1091 **6.8 CIM_NetworkPortConfigurationService**

1092 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1093 Table 9 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of
 1094 the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
 1095 verb and target. Table 9 is for informational purposes only; in case of a conflict between Table 9 and
 1096 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1097 information in Table 9.

1098 **Table 9 – Command Verb Requirements for CIM_NetworkPortConfigurationService**

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 6.8.2.
start	Not supported	
stop	Not supported	

1099 No mapping is defined for the following verbs for the specified target: create, delete, dump, and load.

1100 6.8.1 Ordering of Results

1101 When results are returned for multiple instances of CIM_NetworkPortConfigurationService,
1102 implementations shall utilize the following algorithm to produce the natural (that is, default) ordering:

- 1103 • Results for CIM_NetworkPortConfigurationService are unordered; therefore, no algorithm is
1104 defined.

1105 6.8.2 Show

1106 This section describes how to implement the `show` verb when applied to an instance of
1107 CIM_NetworkPortConfigurationService. Implementations shall support the use of the `show` verb with
1108 CIM_NetworkPortConfigurationService.

1109 The `show` verb is used to display information about the CIM_NetworkPortConfigurationService.

1110 6.8.2.1 Show a Single Instance

1111 This command form is for the `show` verb applied to a single instance of
1112 CIM_NetworkPortConfigurationService.

1113 6.8.2.1.1 Command Form

```
1114 show <CIM_NetworkPortConfigurationService single instance>
```

1115 6.8.2.1.2 CIM Requirements

1116 See CIM_NetworkPortConfigurationService in the “CIM Elements” section of the [Host LAN Network Port](#)
1117 [Profile](#) for the list of mandatory properties.

1118 6.8.2.1.3 Behavior Requirements

1119 6.8.2.1.3.1 Preconditions

1120 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1121 6.8.2.1.3.2 Pseudo Code

```
1122 $instance=<CIM_NetworkPortConfigurationService single instance>  
1123 #propertylist[] = NULL;  
1124 if (false == #all)  
1125 {  
1126     #propertylist[] = { "ElementName" };  
1127 }  
1128 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );  
1129 &smEnd;
```

1130 6.8.2.2 Show Multiple Instances

1131 This command form is for the `show` verb applied to multiple instances of
1132 CIM_NetworkPortConfigurationService. This command form corresponds to UfsT-based selection within a
1133 scoping system.

1134 6.8.2.2.1 Command Form

```
1135 show <CIM_NetworkPortConfigurationService multiple instances>
```

1136 **6.8.2.2.2 CIM Requirements**

1137 See CIM_NetworkPortConfigurationService in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of mandatory properties.

1139 **6.8.2.2.3 Behavior Requirements**

1140 **6.8.2.2.3.1 Preconditions**

1141 \$containerInstance contains the instance of CIM_ComputerSystem for which scoped instances of
 1142 the CIM_NetworkPortConfigurationService are displayed. The [Host LAN Network Port Profile](#) requires
 1143 that the CIM_NetworkPortConfigurationService instance be associated with its scoping system via an
 1144 instance of the CIM_SystemDevice association.

1145 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1146 **6.8.2.2.3.2 Pseudo Code**

```

1147 #propertylist[] = NULL;
1148 if (false == #all)
1149     {
1150         #propertylist[] = { "ElementName" };
1151     }
1152 &smShowInstances ( "CIM_NetworkPortConfigurationService", "CIM_SystemDevice",
1153     $containerInstance.getObjectPath(), #propertylist[] );
1154 &smEnd;
    
```

1155 **6.9 CIM_PortController**

1156 The cd and help verbs shall be supported as described in [DSP0216](#).

1157 Table 10 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
 1158 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
 1159 verb and target. Table 10 is for informational purposes only; in case of a conflict between Table 10 and
 1160 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1161 information in Table 10.

1162 **Table 10 – Command Verb Requirements for CIM_PortController**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	May	See 6.9.2.
set	May	See 6.9.3.
show	Shall	See 6.9.5.
start	May	See 6.9.6.
stop	May	See 6.9.7.

1163 No mapping is defined for the following verbs for the specified target: create, delete, dump, and load.

1164 6.9.1 Ordering of Results

1165 When results are returned for multiple instances of CIM_PortController, implementations shall utilize the
1166 following algorithm to produce the natural (that is, default) ordering:

- 1167 • Results for CIM_PortController are unordered; therefore, no algorithm is defined.

1168 6.9.2 Reset

1169 This section describes how to implement the `reset` verb when applied to an instance of
1170 CIM_PortController. Implementations may support the use of the `reset` verb with CIM_PortController.

1171 The `reset` verb is used to initiate a reset of the CIM_PortController.

1172 6.9.2.1 Reset a Single Instance

1173 This command form is for the initiation of a reset action against a port controller. The mapping is
1174 implemented as an invocation of the `RequestStateChange()` method on the instance.

1175 6.9.2.1.1 Command Form

```
1176 reset <CIM_PortController single instance>
```

1177 6.9.2.1.2 CIM Requirements

```
1178 uint16 EnabledState;
1179 uint16 RequestedState;
1180 uint32 EnabledLogicalElement.RequestStateChange (
1181     [IN] uint16 RequestedState = "Reset",
1182     [OUT] REF CIM_ConcreteJob Job,
1183     [IN] datetime TimeoutPeriod );
```

1184 6.9.2.1.3 Behavior Requirements

```
1185 $instance=<CIM_PortController single instance>
1186 smResetRSC ( $instance.getObjectPath() );
1187 &smEnd;
```

1188 6.9.3 Set

1189 This section describes how to implement the `set` verb when it is applied to an instance of
1190 CIM_PortController. Implementations may support the use of the `set` verb with CIM_PortController.

1191 The `set` verb is used to modify descriptive properties of the CIM_PortController instance.

1192 6.9.3.1 General Usage of Set for a Single Property

1193 This command form corresponds to the general usage of the `set` verb to modify a single property of a
1194 target instance. This is the most common case.

1195 The requirement for supporting modification of a property using this command form shall be equivalent to
1196 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined
1197 in the [Host LAN Network Port Profile](#).

1198 6.9.4 Command Form

```
1199 set <CIM_PortController single instance> <propertyname>=<propertyvalue>
```

1200 6.9.4.1.1 CIM Requirements

1201 See CIM_PortController in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
1202 modifiable properties.

1203 6.9.4.1.2 Behavior Requirements

```
1204 $instance=<CIM_PortController single instance>
1205 #propertyNames[] = {<propertyname>};
1206 #propertyValues[] = {<propertyvalue>};
1207 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
1208 &smEnd;
```

1209 6.9.4.2 General Usage of Set for Multiple Properties

1210 This command form corresponds to the general usage of the `set` verb to modify multiple properties of a
1211 target instance where there is not an explicit relationship between the properties. This is the most
1212 common case.

1213 The requirement for supporting modification of a property using this command form shall be equivalent to
1214 the requirement for supporting modification of the property using the ModifyInstance operation as defined
1215 in the [Host LAN Network Port Profile](#).

1216 6.9.4.2.1 Command Form

```
1217 set <CIM_PortController single instance> <propertyname1>=<propertyvalue1>
1218 <propertynamen>=<propertyvaluen>
```

1219 6.9.4.2.2 CIM Requirements

1220 See CIM_PortController in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
1221 supported properties.

1222 6.9.4.2.3 Behavior Requirements

```
1223 $instance=<CIM_PortController single instance>
1224 #propertyNames[] = {<propertyname>};
1225 for #i < n
1226 {
1227     #propertyNames[#i] = <propertyname#i>
1228     #propertyValues[#i] = <propertyvalue#i>
1229 }
1230 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
1231 &smEnd;
```

1232 6.9.5 Show

1233 This section describes how to implement the `show` verb when applied to an instance of
1234 CIM_PortController. Implementations shall support the use of the `show` verb with CIM_PortController.

1235 The `show` verb is used to display information about the port controller(s).

1236 6.9.5.1 Show a Single Instance

1237 This command form is for the `show` verb applied to a single instance of CIM_PortController.

1238 **6.9.5.1.1 Command Form**1239 `show <CIM_PortController single instance>`1240 **6.9.5.1.2 CIM Requirements**1241 See CIM_PortController in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
1242 mandatory properties.1243 **6.9.5.1.3 Behavior Requirements**1244 **6.9.5.1.3.1 Preconditions**

1245 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1246 **6.9.5.1.3.2 Pseudo Code**

```

1247 $instance=<CIM_PortController single instance>
1248 #propertylist[] = NULL;
1249 if (false == #all)
1250 {
1251     #propertylist[] = { "ProtocolType", "MaxUnitsSupported", "ControllerType",
1252         "ElementName" };
1253 }
1254 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
1255 &smEnd;

```

1256 **6.9.5.2 Show Multiple Instances**1257 This command form is for the show verb applied to multiple instances of CIM_PortController. This
1258 command form corresponds to UFsT-based selection within a scoping system.1259 **6.9.5.2.1 Command Form**1260 `show <CIM_PortController multiple instances>`1261 **6.9.5.2.2 CIM Requirements**1262 See CIM_PortController in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
1263 mandatory properties.1264 **6.9.5.2.3 Behavior Requirements**1265 **6.9.5.2.3.1 Preconditions**1266 \$containerInstance contains the instance of CIM_ComputerSystem for which scoped
1267 CIM_PortController instances are displayed. The [Host LAN Network Port Profile](#) requires that the
1268 CIM_PortController instance be associated with its scoping system via an instance of the
1269 CIM_SystemDevice association.

1270 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1271 **6.9.5.2.3.2 Pseudo Code**

```

1272 #propertylist[] = NULL;
1273 if (false == #all)
1274 {
1275     #propertylist[] = { "ProtocolType", "MaxUnitsSupported", "ControllerType",

```



```

1276     "ElementName" };
1277     }
1278 &smShowInstances ( "CIM_PortController", "CIM_SystemDevice",
1279     $containerInstance.getObjectPath(), #propertylist[] );
1280 &smEnd;

```

1281 6.9.6 Start

1282 This section describes how to implement the `start` verb when applied to an instance of
 1283 `CIM_PortController`. Implementations may support the use of the `start` verb with `CIM_PortController`.

1284 The `start` verb is used to enable a port controller.

1285 6.9.6.1 Start a Single Instance

1286 This command form is for the `start` verb applied to a single instance of `CIM_PortController`.

1287 6.9.6.1.1 Command Form

```

1288 start <CIM_PortController single instance>

```

1289 6.9.6.1.2 CIM Requirements

```

1290 uint16 EnabledState;
1291 uint16 RequestedState;
1292 uint32 EnabledLogicalElement.RequestStateChange (
1293     [IN] uint16 RequestedState = "Enabled",
1294     [OUT] REF CIM_ConcreteJob Job,
1295     [IN] datetime TimeoutPeriod );

```

1296 6.9.6.1.3 Behavior Requirements

```

1297 $instance=<CIM_PortController single instance>
1298 smStartRSC ( $instance.getObjectPath() );
1299 &smEnd;

```

1300 6.9.7 Stop

1301 This section describes how to implement the `stop` verb when applied to an instance of
 1302 `CIM_PortController`. Implementations may support the use of the `stop` verb with `CIM_PortController`.

1303 The `stop` verb is used to disable a port controller.

1304 6.9.7.1 Stop a Single Instance

1305 This command form is for the `stop` verb applied to a single instance of `CIM_PortController`.

1306 6.9.7.1.1 Command Form

```

1307 stop <CIM_PortController single instance>

```

1308 6.9.7.1.2 CIM Requirements

```

1309 uint16 EnabledState;
1310 uint16 RequestedState;
1311 uint32 EnabledLogicalElement.RequestStateChange (
1312     [IN] uint16 RequestedState = "Disabled",

```

```
1313 [OUT] REF CIM_ConcreteJob Job,
1314 [IN] datetime TimeoutPeriod );
```

1315 6.9.7.1.3 Behavior Requirements

```
1316 $instance=<CIM_PortController single instance>
1317 smStopRSC ( $instance.getObjectPath() );
1318 &smEnd;
```

1319 6.10 CIM_DeviceSAPImplementation

1320 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1321 Table 11 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
 1322 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
 1323 verb and target. Table 11 is for informational purposes only; in case of a conflict between Table 11 and
 1324 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1325 information in Table 11.

1326 **Table 11 – Command Verb Requirements for CIM_DeviceSAPImplementation**

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 6.10.2.
start	Not supported	
stop	Not supported	

1327 The following verbs shall not be supported for the specified target: `create`, `delete`, `dump`, `load`,
 1328 `reset`, `set`, `start`, and `stop`.

1329 6.10.1 Ordering of Results

1330 When results are returned for multiple instances of `CIM_DeviceSAPImplementation`, implementations
 1331 shall utilize the following algorithm to produce the natural (that is, default) ordering:

- 1332 • Results for `CIM_DeviceSAPImplementation` are unordered; therefore, no algorithm is defined.

1333 6.10.2 Show

1334 This section describes how to implement the `show` verb when applied to an instance of
 1335 `CIM_DeviceSAPImplementation`. Implementations shall support the use of the `show` verb with
 1336 `CIM_DeviceSAPImplementation`.

1337 The `show` command is used to display information about the `CIM_DeviceSAPImplementation` instance or
 1338 instances.

1339 **6.10.2.1 Show Multiple Instances**

1340 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1341 to a `show` command issued against `CIM_DeviceSAPImplementation` where only one reference is
1342 specified and the reference is to an instance of `CIM_PortController`.

1343 **6.10.2.1.1 Command Form**

```
1344 show <CIM_DeviceSAPImplementation multiple instances>
```

1345 **6.10.2.1.2 CIM Requirements**

1346 See `CIM_DeviceSAPImplementation` in the “CIM Elements” section of the [Host LAN Network Port Profile](#)
1347 for the list of mandatory properties.

1348 **6.10.2.1.3 Behavior Requirements**

1349 **6.10.2.1.3.1 Preconditions**

1350 `$instance` contains the instance of `CIM_PortController` which is referenced by
1351 `CIM_DeviceSAPImplementation`.

1352 **6.10.2.1.3.2 Pseudo Code**

```
1353 &smShowAssociationInstances ( "CIM_DeviceSAPImplementation",  
1354     $instance.getObjectPath() );  
1355 &smEnd;
```

1356 **6.10.2.2 Show a Single Instance – CIM_NetworkPort Reference**

1357 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1358 a `show` command issued against `CIM_DeviceSAPImplementation` where the reference specified is to an
1359 instance of `CIM_NetworkPort`. An instance of `CIM_NetworkPort` is referenced by exactly one instance of
1360 `CIM_DeviceSAPImplementation`. Therefore, a single instance will be returned.

1361 **6.10.2.2.1 Command Form**

```
1362 show <CIM_DeviceSAPImplementation single instance>
```

1363 **6.10.2.2.2 CIM Requirements**

1364 See `CIM_DeviceSAPImplementation` in the “CIM Elements” section of the [Host LAN Network Port Profile](#)
1365 for the list of mandatory properties.

1366 **6.10.2.2.3 Behavior Requirements**

1367 **6.10.2.2.3.1 Preconditions**

1368 `$instance` contains the instance of `CIM_NetworkPort` which is referenced by
1369 `CIM_DeviceSAPImplementation`.

1370 **6.10.2.2.3.2 Pseudo Code**

```
1371 &smShowAssociationInstances ( "CIM_DeviceSAPImplementation",  
1372     $instance.getObjectPath() );  
1373 &smEnd;
```

1374 **6.10.2.3 Show a Single Instance – Both References**

1375 This command form is for the `show` verb applied to a single instance. This command form corresponds to
 1376 a `show` command issued against `CIM_DeviceSAPImplementation` where both references are specified
 1377 and therefore the desired instance is unambiguously identified.

1378 **6.10.2.3.1 Command Form**

1379 `show <CIM_DeviceSAPImplementation single instance>`

1380 **6.10.2.3.2 CIM Requirements**

1381 See `CIM_DeviceSAPImplementation` in the “CIM Elements” section of the [Host LAN Network Port Profile](#)
 1382 for the list of mandatory properties.

1383 **6.10.2.3.3 Behavior Requirements**1384 **6.10.2.3.3.1 Preconditions**

1385 `$instanceA` contains the instance of `CIM_PortController` which is referenced by
 1386 `CIM_DeviceSAPImplementation`.

1387 `$instanceB` contains the instance of `CIM_NetworkPort` which is referenced by
 1388 `CIM_DeviceSAPImplementation`.

1389 **6.10.2.3.3.2 Pseudo Code**

```
1390 &smShowAssociationInstance ( "CIM_DeviceSAPImplementation",
1391     $instanceA.getObjectPath(), $instanceB.getObjectPath() );
1392 &smEnd;
```

1393 **6.11 CIM_ServiceAffectsElement**

1394 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1395 Table 12 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
 1396 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
 1397 verb and target. Table 12 is for informational purposes only; in case of a conflict between Table 12 and
 1398 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1399 information in Table 12.

1400 **Table 12 – Command Verb Requirements for `CIM_ServiceAffectsElement`**

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 6.11.2.
start	Not supported	
stop	Not supported	

1401 The following verbs shall not be supported for the specified target: create, delete, dump, load,
1402 reset, set, start, and stop.

1403 **6.11.1 Ordering of Results**

1404 When results are returned for multiple instances of CIM_ServiceAffectsElement, implementations shall
1405 utilize the following algorithm to produce the natural (that is, default) ordering:

- 1406 • Results for CIM_ServiceAffectsElement are unordered; therefore, no algorithm is defined.

1407 **6.11.2 Show**

1408 This section describes how to implement the `show` verb when applied to an instance of
1409 CIM_ServiceAffectsElement. Implementations shall support the use of the `show` verb with
1410 CIM_ServiceAffectsElement.

1411 The `show` command is used to display information about the CIM_ServiceAffectsElement instance or
1412 instances.

1413 **6.11.2.1 Show Multiple Instances**

1414 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1415 to a `show` command issued against CIM_ServiceAffectsElement where only one reference is specified
1416 and the reference is to an instance of CIM_ComputerSystem.

1417 **6.11.2.1.1 Command Form**

```
1418 show <CIM_ServiceAffectsElement multiple instances>
```

1419 **6.11.2.1.2 CIM Requirements**

1420 See CIM_ServiceAffectsElement in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for
1421 the list of mandatory properties.

1422 **6.11.2.1.3 Behavior Requirements**

1423 **6.11.2.1.3.1 Preconditions**

1424 `$instance` contains the instance of CIM_ComputerSystem which is referenced by
1425 CIM_ServiceAffectsElement.

1426 **6.11.2.1.3.2 Pseudo Code**

```
1427 &smShowAssociationInstances ( "CIM_ServiceAffectsElement",  
1428     $instance.getObjectPath() );  
1429 &smEnd;
```

1430 **6.11.2.2 Show a Single Instance – CIM_NetworkPort**

1431 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1432 a `show` command issued against CIM_ServiceAffectsElement where the reference specified is to an
1433 instance of CIM_NetworkPort. An instance of CIM_NetworkPort is referenced by exactly one instance of
1434 CIM_ServiceAffectsElement. Therefore, a single instance will be returned.

1435 **6.11.2.2.1 Command Form**

```
1436 show <CIM_ServiceAffectsElement single instance>
```

1437 6.11.2.2.2 CIM Requirements

1438 See CIM_ServiceAffectsElement in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for
1439 the list of mandatory properties.

1440 6.11.2.2.3 Behavior Requirements

1441 6.11.2.2.3.1 Preconditions

1442 \$instance contains the instance of CIM_NetworkPort which is referenced by
1443 CIM_ServiceAffectsElement.

1444 6.11.2.2.3.2 Pseudo Code

```
1445 &smShowAssociationInstances ( "CIM_ServiceAffectsElement",  
1446     $instance.getObjectPath() );  
1447 &smEnd;
```

1448 6.11.2.3 Show a Single Instance – Both References

1449 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1450 a `show` command issued against CIM_ServiceAffectsElement where both references are specified and
1451 therefore the desired instance is unambiguously identified.

1452 6.11.2.3.1 Command Form

```
1453 show <CIM_ServiceAffectsElement single instance>
```

1454 6.11.2.3.2 CIM Requirements

1455 See CIM_ServiceAffectsElement in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for
1456 the list of mandatory properties.

1457 6.11.2.3.3 Behavior Requirements

1458 6.11.2.3.3.1 Preconditions

1459 \$instanceA contains the instance of CIM_ComputerSystem which is referenced by
1460 CIM_ServiceAffectsElement.

1461 \$instanceB contains the instance of CIM_NetworkPort or CIM_PortController which is referenced by
1462 CIM_ServiceAffectsElement.

1463 6.11.2.3.3.2 Pseudo Code

```
1464 &smShowAssociationInstance ( "CIM_ServiceAffectsElement", $instanceA.getObjectPath(),  
1465     $instanceB.getObjectPath() );  
1466 &smEnd;
```

1467 6.12 CIM_SystemDevice

1468 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1469 Table 13 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
1470 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
1471 verb and target. Table 13 is for informational purposes only; in case of a conflict between Table 13 and
1472 requirements detailed in the following sections, the text detailed in the following sections supersedes the
1473 information in Table 13.

1474

Table 13 – 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 6.12.2.
start	Not supported	
stop	Not supported	

1475 The following verbs shall not be supported for the specified target: `create`, `delete`, `dump`, `load`,
 1476 `reset`, `set`, `start`, and `stop`.

1477 6.12.1 Ordering of Results

1478 When results are returned for multiple instances of `CIM_SystemDevice` implementations shall utilize the
 1479 following algorithm to produce the natural (that is, default) ordering:

- 1480 • Results for `CIM_SystemDevice` are unordered; therefore, no algorithm is defined.

1481 6.12.2 Show

1482 This section describes how to implement the `show` verb when applied to an instance of
 1483 `CIM_SystemDevice`. Implementations shall support the use of the `show` verb with `CIM_SystemDevice`.

1484 The `show` command is used to display information about the `CIM_SystemDevice` instance or instances.

1485 6.12.2.1 Show Multiple Instances

1486 This command form is for the `show` verb applied to multiple instances. This command form corresponds
 1487 to a `show` command issued against `CIM_SystemDevice` where only one reference is specified and the
 1488 reference is to an instance of `CIM_ComputerSystem`.

1489 6.12.2.1.1 Command Form

```
1490 show <CIM_SystemDevice multiple instances>
```

1491 6.12.2.1.2 CIM Requirements

1492 See `CIM_SystemDevice` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
 1493 mandatory properties.

1494 6.12.2.1.3 Behavior Requirements

1495 6.12.2.1.3.1 Preconditions

1496 `instance` contains the instance of `CIM_ComputerSystem` which is referenced by `CIM_SystemDevice`.

1497 6.12.2.1.3.2 Pseudo Code

```
1498 &smShowAssociationInstances ( "CIM_SystemDevice", $instance.getObjectPath() );  
1499 &smEnd;
```

1500 6.12.2.2 Show a Single Instance – CIM_NetworkPort or CIM_PortController Reference

1501 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1502 a `show` command issued against `CIM_SystemDevice` where the reference specified is to an instance of
1503 `CIM_NetworkPort` or `CIM_PortController`. An instance is referenced by exactly one instance of
1504 `CIM_SystemDevice`. Therefore, a single instance will be returned.

1505 6.12.2.2.1 Command Form

```
1506 show <CIM_SystemDevice single instance>
```

1507 6.12.2.2.2 CIM Requirements

1508 See `CIM_SystemDevice` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
1509 mandatory properties.

1510 6.12.2.2.3 Behavior Requirements**1511 6.12.2.2.3.1 Preconditions**

1512 `$instance` contains the instance of `CIM_NetworkPort` or `CIM_PortController` which is referenced by
1513 `CIM_SystemDevice`.

1514 6.12.2.2.3.2 Pseudo Code

```
1515 &smShowAssociationInstances ( "CIM_SystemDevice", $instance.getObjectPath() );  
1516 &smEnd;
```

1517 6.12.2.3 Show a Single Instance – Both References

1518 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1519 a `show` command issued against `CIM_SystemDevice` where both references are specified and therefore
1520 the desired instance is unambiguously identified.

1521 6.12.2.3.1 Command Form

```
1522 show <CIM_SystemDevice single instance>
```

1523 6.12.2.3.2 CIM Requirements

1524 See `CIM_SystemDevice` in the “CIM Elements” section of the [Host LAN Network Port Profile](#) for the list of
1525 mandatory properties.

1526 6.12.2.3.3 Behavior Requirements**1527 6.12.2.3.3.1 Preconditions**

1528 `$instanceA` contains the instance of `CIM_ComputerSystem` which is referenced by `CIM_SystemDevice`.

1529 `$instanceB` contains the instance of `CIM_NetworkPort` or `CIM_PortController` which is referenced by
1530 `CIM_SystemDevice`.

1531 **6.12.2.3.3.2 Pseudo Code**

```
1532 &smShowAssociationInstance ( "CIM_SystemDevice", $instanceA.getObjectPath(),  
1533     $instanceB.getObjectPath() );  
1534 &smEnd;
```

1535

ANNEX A
(informative)**Change Log**

Version	Date	Author	Description
1.0.0	2009-07-14		DMTF Standard Release

1536
1537
1538
1539
1540

1541