5 Command Line Protocol Service Profile SM CLP
6 Command Mapping Specification

10

CONTENTS

Tables

80

81   Foreword

82   The *Command Line Protocol Service Profile SM CLP Command Mapping Specification* (DSP0801) was
83   prepared by the Server Management Working Group and the Physical Profiles Working Group of the
84   DMTF.

85   **Conventions**

86   The pseudo-code conventions utilized in this document are the Recipe Conventions as defined in SNIA
87   SMI-S 1.1.0, section 7.6.

88   **Acknowledgements**

89   The authors wish to acknowledge the following participants from the DMTF Server Management Working
90   Group:

91   •   Christina Shaw – HP

92   •   Aaron Merkin – IBM

93   •   Jon Hass – Dell

94   •   Khachatur Papanyan – Dell

95   •   Jeff Hilland – HP

96   •   Perry Vincent – Intel

97   •   John Leung – Intel

98

99   Introduction

100   This document defines the SM CLP mapping for CIM elements described in the *CLP Service Profile*. The
101   information in this specification, combined with the *SM CLP-to-CIM Common Mapping Specification 1.0*,
102   is intended to be sufficient to implement SM CLP commands relevant to the classes, properties, and
103   methods described in the *CLP Service Profile* using CIM operations.

104   The target audience for this specification is implementers of the SM CLP support for the *CLP Service*
105   *Profile*.

Command Line Protocol Service 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 *CLP Service 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 DSP1005, *CLP Service Profile 1.0*, http://www.dmtf.org/standards/published_documents/DSP1005_1.0.pdf

SNIA, *Storage Management Initiative Specification (SMI-S) 1.1.0*, November 2005, 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

137   **3.4**
138   **mandatory**
139   indicates requirements to be followed strictly in order to conform to the document and from which no
140   deviation is permitted

141   **3.5**
142   **may**
143   indicates a course of action permissible within the limits of the document

144   **3.6**
145   **need not**
146   indicates a course of action permissible within the limits of the document

147   **3.7**
148   **optional**
149   indicates a course of action permissible within the limits of the document

150   **3.8**
151   **shall**
152   indicates requirements to be followed strictly in order to conform to the document and from which no
153   deviation is permitted

154   **3.9**
155   **shall not**
156   indicates requirements to be followed strictly in order to conform to the document and from which no
157   deviation is permitted

158   **3.10**
159   **should**
160   indicates that among several possibilities, one is recommended as particularly suitable, without
161   mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

162   **3.11**
163   **should not**
164   indicates that a certain possibility or course of action is deprecated but not prohibited

165   **4   Symbols and Abbreviated Terms**

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

167   **4.1**
168   **CIM**
169   Common Information Model

170   **4.2**
171   **CLP**
172   Command Line Protocol

173   **4.3**
174   **DMTF**
175   Distributed Management Task Force

176 **4.4**
177 **IETF**
178 Internet Engineering Task Force

179 **4.5**
180 **SM**
181 Server Management

182 **4.6**
183 **SMI-S**
184 Storage Management Initiative Specification

185 **4.7**
186 **SNIA**
187 Storage Networking Industry Association

188 **4.8**
189 **UFsT**
190 User Friendly selection Tag

191 # 5   Recipes

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

194 • smStartRSC()

195 • smStopRSC()

196 • smResetRSC()

197 • smShowInstance()

198 • smShowInstances()

199 • smSetInstance()

200 • smShowAssociationInstances()

201 • smShowAssociationInstance()

202 • smDeleteInstance

203 • smMakeCommandStatus

204 • smNewInstance

205 # 6   Mappings

206 The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in
207 the *CLP Service Profile*. Requirements specified here related to support for a CLP verb for a particular
208 class are solely within the context of this profile.

209 ## 6.1   CIM_BindsTo

210 The `cd` and `help` verbs shall be supported as described in DSP0216.

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

216   **Table 1 – Command Verb Requirements for CIM_BindsTo**

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 6.1.2.
start	Not supported	
stop	Not supported	

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

6.1.1   Ordering of Results

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

222   • Results for CIM_BindsTo are unordered; therefore, no algorithm is defined.

6.1.2   Show

224 This section describes how to implement the `show` verb when applied to an instance of CIM_BindsTo.
225 Implementations shall support the use of the `show` verb with CIM_BindsTo.

226 The `show` command is used to display information about the CIM_BindsTo instance or instances.

6.1.2.1   Show a Single Instances – CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint

228 This command form is for the `show` verb applied to multiple instances. This command form corresponds
229 to the `show` command issued against CIM_BindsTo where only one reference is specified and the
230 reference is to an instance of CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint.

6.1.2.1.1   Command Form

232 `show <CIM_BindsTo multiple instances>`

6.1.2.1.2   CIM Requirements

234 See CIM_BondsTo in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory
235 properties.

236 **6.1.2.1.3   Behavior Requirements**

237 **6.1.2.1.3.1   Preconditions**

238 $instance contains the instance of CIM_SSHProtcolEndpoint or
239   CIM_TelnetProtocolEndpoingwhich is referenced by CIM_BindsTo

240 **6.1.2.1.3.2   Pseudo Code**

241 &smShowAssociationInstances ("CIM_BindsTo", $instance.getObjectPath());
242   smEnd;

243 **6.1.2.2   Show a Single Instance – CIM_CLPProtocolEndpoint Reference**

244 This command form is for the show verb applied to a single instance. This command form corresponds to
245 the show command issued against CIM_BindsTo where the reference specified is to an instance of
246 CIM_CLPProtocolEndpoint. A single instance of CIM_SSHProtocolEndpoint or
247 CIM_TelnetProtocolEndpoint is associated with each instance of CIM_CLPProtocolEndpoint. Therefore, a
248 single instance will be returned.

249 **6.1.2.2.1   Command Form**

250 **show <CIM_BindsTo *single instance*>**

251 **6.1.2.2.2   CIM Requirements**

252 See the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

253 **6.1.2.2.3   Behavior Requirements**

254 **6.1.2.2.3.1   Preconditions**

255 $instance contains the instance of CIM_CLPProtocolEndpoint which is referenced by
256   CIM_BindsTo

257 **6.1.2.2.3.2   Pseudo Code**

258 &smShowAssociationInstances ("CIM_BindsTo", $instance.getObjectPath());
259   smEnd;

260 **6.1.2.3   Show a Single Instance – Both References**

261 This command form is for the show verb applied to a single instance. This command form corresponds to
262 the show command issued against CIM_BindsTo where a reference to CIM_CLPProtocolEndpoint and a
263 reference to CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint are specified and therefore the
264 desired instance is unambiguously identified.

265 **6.1.2.3.1   Command Form**

266 **show <CIM_BindsTo *single instance*>**

267 **6.1.2.3.2   CIM Requirements**

268 See the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

269 **6.1.2.3.3   Behavior Requirements**

270 **6.1.2.3.3.1   Preconditions**

271 1) $instanceA contains the instance of CIM_CLPProtocolEndpoint which is referenced by

```
272    CIM_BindsTo
273 2) $instanceB contains the instance of CIM_ProtocolEndpoint which is referenced by
274    CIM_BindsTo.
```

275 **6.1.2.3.3.2   Pseudo Code**

```
276 &smShowAssociationInstance ( "CIM_BindsTo",
277    $instanceA.getObjectPath(),$instanceB.getObjectPath() );
278    smEnd;
```

279 **6.2   CIM_CLPCapabilities**

280 The `cd` and `help` verbs shall be supported as described in DSP0216.

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

286   **Table 2 – Command Verb Requirements for CIM_CLPCapabilities**

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	

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

289 **6.2.1   Ordering of Results**

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

292   • Results for CIM_CLPCapabilities are unordered; therefore, no algorithm is defined.

293 **6.2.2   Show**

294 This section describes how to implement the `show` verb when applied to an instance of
295 CIM_CLPCapabilities. Implementations shall support the use of the `show` verb with CIM_CLPCapabilities.

296 The `show` verb is used to display information about an instance or instances of the CIM_CLPCapabilities
297 class.

298 **6.2.2.1   Show a Single Instance**

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

300 **6.2.2.1.1   Command Form**

301 `show <CIM_CLPCapabilities single instance>`

302 **6.2.2.1.2   CIM Requirements**

303 See CIM_CLPCapabilities in the "CIM Elements" section of the *CLP Service Profile* for the list of
304 mandatory properties.

305 **6.2.2.1.3   Behavior Requirements**

306 **6.2.2.1.3.1   Preconditions**

307 `#all is true if the all option was specified with the command; otherwise, #all is`
308 `false`

309 **6.2.2.1.3.2   Pseudo Code**

310 `$instance=<CIM_CLPCapabilities Single Instance>`
311 `#propertylist[] = null;`
312 `if ( false == #all) {`
313 `    #propertylist[] = {//all mandatory non-key properties }`
314 `}`
315 `&smShowInstance($instance.getObjectPath(), #propertylist[]);`
316 `&smEnd;`

317 ## 6.3   CIM_CLPProtocolEndpoint

318 The `cd` and `help` verbs shall be supported as described in DSP0216.

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

324   **Table 3 – Command Verb Requirements for CIM_CLPProtocolEndpoint**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.3.2.
show	Shall	See 6.3.3.
start	Not supported	
stop	May	See 6.3.4.

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

6.3.1   Ordering of Results

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

330 • Results for CIM_CLPProtocolEndpoint are unordered; therefore, no algorithm is defined.

6.3.2   Set

332 This section describes how to implement the `set` verb when it is applied to an instance of
333 CIM_CLPProtocolEndpoint. Implementations may support the use of the `set` verb with
334 CIM_CLPProtocolEndpoint.

335 The `set` verb is used to modify descriptive properties of the CIM_CLPProtocolEndpoint instance.

6.3.2.1   General Usage of Set for a Single Property

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

339 The requirement for supporting modification of a property using this command form shall be equivalent to
340 the requirement for supporting modification of the property using the ModifyInstance operation as defined
341 in the *CLP Service Profile*.

6.3.2.1.1   Command Form

343 `set <CIM_CLPProtocolEndpoint single instance> <propertyname>=<propertyvalue>`

6.3.2.1.2   CIM Requirements

345 See CIM_CLPProtocolEndpoint in the "CIM Elements" section of the *CLP Service Profile* for the list of
346 modifiable properties.

6.3.2.1.3   Behavior Requirements

```
348   $instance = <CIM_CLPProtocolEndpoint Single Instance>
349   #propertyNames[] = {<propertyname>};
350   #propertyValues[] = {<propertyvalue>};
351   &smSetInstance($instance, #propertyNames[], #propertyValues[]);
352   &smEnd;
```

6.3.2.2   General Usage of Set for Multiple Properties

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

357 The requirement for supporting modification of a property using this command form shall be equivalent to
358 the requirement for supporting modification of the property using the ModifyInstance operation as defined
359 in the *CLP Service Profile*.

6.3.2.2.1   Command Form

361 `set <CIM_CLPProtocolEndpoint single instance> <propertyname1>=<propertyvalue1>`
362 `<propertynamen>=<propertyvaluen>`

6.3.2.2.2   CIM Requirements

See CIM_CLPProtocolEndpoint in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.3.2.2.3   Behavior Requirements

```
$instance = <CIM_CLPProtocolEndpoint Single Instance>
#propertyNames[] = {<propertyname>};

for #i < n
{
    #propertyNames[#i] = <propertname#i>
    #propertyValues[#i] = <propertyvalue#i>
}

&smSetInstance($instance, #propertyNames[], #propertyValues[]);
&smEnd;
```

6.3.3   Show

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

The show verb is used to display information about a CLP session.

6.3.3.1   Show a Single Instance

This command form is for the show verb applied to a single instance of CIM_CLPProtocolEndpoint.

6.3.3.1.1   Command Form

```
show <CIM_CLPProtocolEndpoint single instance>
```

6.3.3.1.2   CIM Requirements

See CIM_CLPProtocolEndpoint in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.3.3.1.3   Behavior Requirements

6.3.3.1.3.1   Preconditions

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

6.3.3.1.3.2   Pseudo Code

```
$instance=<CIM_CLPProtocolEndpoint Single Instance>
#propertylist[] = null;
if ( false == #all) {
    #propertylist[] = {//all mandatory non-key properties }
}
&smShowInstance($instance.getObjectPath(), #propertylist[]);
```

401 **6.3.3.2   Show Multiple Instances Scoped by System**

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

404 **6.3.3.2.1   Command Form**

405 `show <CIM_CLPProtocolEndpoint multiple instances>`

406 **6.3.3.2.2   CIM Requirements**

407 See CIM_CLPProtocolEndpoint in the "CIM Elements" section of the *CLP Service Profile* for the list of
408 mandatory properties.

409 **6.3.3.2.3   Behavior Requirements**

410 **6.3.3.2.3.1   Preconditions**

```
411 1)  $containerInstance contains the instance of CIM_ComputerSystem for which we are
412     displaying scoped endpoints (CIM_CLPProtocolEndpoint instances). The CLP Service
413     Profile requires that the CIM_CLPProtocolEndpoint instance be associated with its
414     scoping system via an instance of the CIM_HostedAccessPoint association.
415 2)  #all is true if the all option was specified with the command; otherwise, #all is
416     false
```

417 **6.3.3.2.3.2   Pseudo Code**

```
418 #propertylist[] = null;
419 if (false == #all)  {
420     #propertylist[] = {//all mandatory non-key properties }
421 }
422
423 &smShowInstances ( "CIM_CLPProtocolEndpoint", "CIM_HostedAccessPoint",
424 $containerInstance.getObjectPath(), #propertylist[] );
425 &smEnd;
```

426 **6.3.4   Stop**

427 This section describes how to implement the `stop` verb when applied to an instance of
428 CIM_CLPProtocolEndpoint. Implementations may support the use of the `stop` verb with
429 CIM_CLPProtocolEndpoint.

430 The `stop` verb is used to terminate a CLP session.

431 **6.3.4.1   Stop a Single Instance**

432 This command form is for the `stop` verb applied to a single instance of CIM_CLPProtocolEndpoint. The
433 lifecycle of a CLP session corresponds to the lifecycle of the CIM_CLPProtocolEndpoint which represents
434 it. Therefore, stopping a CLP service corresponds to a deletion of the underlying instance.

435 **6.3.4.1.1   Command Form**

436 `stop <CIM_CLPProtocolEndpoint single instance>`

437 **6.3.4.1.2   CIM Requirements**

438 See CIM_CLPProtocolEndpoint in the "CIM Elements" section of the *CLP Service Profile* for the list of
439 mandatory properties.

440 **6.3.4.1.3 Behavior Requirements**

```
441 $instance=<CIM_CLPProtocolEndpoint Single Instance>
442 &smDeleteInstance($instance.getObjectPath());
443 &smEnd;
```

444 6.4 CIM_CLPSettingData

445 The `cd` and `help` verbs shall be supported as described in DSP0216.

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

451 **Table 4 – Command Verb Requirements for CIM_CLPSettingData**

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

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

453 6.4.1 Ordering of Results

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

456 • Results for CIM_CLPSettingData are unordered; therefore, no algorithm is defined.

457 6.4.2 Set

458 This section describes how to implement the `set` verb when it is applied to an instance of
459 CIM_CLPSettingData. Implementations may support the use of the `set` verb with CIM_CLPSettingData.

460 The `set` verb is used to modify configuration represented by an instance of CIM_CLPSettingData.

461 6.4.2.1 General Usage of Set for a Single Property

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

464 The requirement for supporting modification of a property using this command form shall be equivalent to
465 the requirement for supporting modification of the property using the ModifyInstance operation as defined
466 in the *CLP Service Profile*.

6.4.2.1.1   Command Form

468 `set <CIM_CLPSettingData single instance> <propertyname>=<propertyvalue>`

6.4.2.1.2   CIM Requirements

470 See CIM_CLPSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of
471 modifiable properties.

6.4.2.1.3   Behavior Requirements

```
473 $instance = <CIM_CLPSettingData Single Instance>
474 #propertyNames[] = {<propertyname>};
475 #propertyValues[] = {<propertyvalue>};
476 &smSetInstance($instance, #propertyNames[], #propertyValues[]);
477 &smEnd;
```

6.4.2.2   General Usage of Set for Multiple Properties

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

482 The requirement for supporting modification of a property using this command form shall be equivalent to
483 the requirement for supporting modification of the property using the ModifyInstance operation as defined
484 in the *CLP Service Profile*.

6.4.2.2.1   Command Form

486 `set <CIM_CLPSettingData single instance> <propertyname1>=<propertyvalue1>`
487 `<propertynamen>=<propertyvaluen>`

6.4.2.2.2   CIM Requirements

489 See CIM_CLPSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of
490 mandatory properties.

6.4.2.2.3   Behavior Requirements

```
492 $instance = <CIM_CLPSettingData Single Instance>
493 #propertyNames[] = {<propertyname>};
494 for #i < n
495 {
496     #propertyNames[#i] = <propertname#i>
497     #propertyValues[#i] = <propertyvalue#i>
498 }
499 &smSetInstance($instance, #propertyNames[], #propertyValues[]);
500 &smEnd;
```

6.4.3   Show

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

The `show` verb is used to display information about the CIM_CLPSettingData instance.

6.4.3.1   Show a Single Instance

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

6.4.3.1.1   Command Form

```
show <CIM_CLPSettingData single instance>
```

6.4.3.1.2   CIM Requirements

See CIM_CLPSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.4.3.1.3   Behavior Requirements

6.4.3.1.3.1   Preconditions

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

6.4.3.1.3.2   Pseudo Code

```
$instance=<CIM_CLPSettingData Single Instance>
&lShowPEndpoint($instance, #all);
&smEnd;
```

6.4.3.2   Show Multiple Instances Scoped by ConcreteCollection

This command form is for the `show` verb applied to multiple instances of CIM_CLPSettingData. This command form corresponds to UFsT-based selection within an instance of CIM_ConcreteCollection.

6.4.3.2.1   Command Form

```
show <CIM_CLPSettingData multiple instances>
```

6.4.3.2.2   CIM Requirements

See CIM_CLPSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.4.3.2.3   Behavior Requirements

6.4.3.2.3.1   Preconditions

```
1) $containerInstance contains the instance of CIM_ConcreteCollection for which we are
   displaying contained CIM_CLPSettingData instances. The SMASH Collections Profile
   requires that the CIM_CLPSettingData instances be aggregated into an addressing
   collection via CIM_MemberOfCollection.
2) #all is true if the all option was specified with the command; otherwise, #all is
   false
```

537 **6.4.3.2.3.2   Pseudo Code**

```
538   #propertylist[] = null;
539   //this property list will match the property list in lShowPEndpoint()
540   if (false == #all)  {
541       #propertylist[] = {//all mandatory non-key properties }
542   }
543
544   &smShowInstances ( "CIM_CLPSettingData", "CIM_MemberOfCollection",
545   $containerInstance.getObjectPath(), #propertylist[] );
546   &smEnd;
```

547 **6.5   CIM_ConcreteJob**

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

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

554   **Table 5 – Command Verb Requirements for CIM_ConcreteJob**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.5.2.
show	Shall	See 6.5.3.
start	Not supported	
stop	May	See 6.5.4.

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

557 **6.5.1   Ordering of Results**

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

560   • Results for CIM_ConcreteJob are unordered; therefore, no algorithm is defined.

561 **6.5.2   Set**

562 This section describes how to implement the `set` verb when it is applied to an instance of
563 CIM_ConcreteJob. Implementations may support the use of the `set` verb with CIM_ConcreteJob.

564 The `set` verb is used to modify descriptive properties of the CIM_ConcreteJob instance.

565 **6.5.2.1   General Usage of Set for a Single Property**

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

568 The requirement for supporting modification of a property using this command form shall be equivalent to
569 the requirement for supporting modification of the property using the ModifyInstance operation as defined
570 in the *CLP Service Profile*.

571 **6.5.2.1.1   Command Form**

572 `set <CIM_ConcreteJob single instance> <propertyname>=<propertyvalue>`

573 **6.5.2.1.2   CIM Requirements**

574 See CIM_ConcreteJob in the "CIM Elements" section of the *CLP Service Profile* for the list of modifiable
575 properties.

576 **6.5.2.1.3   Behavior Requirements**

```
577 $instance = <CIM_ConcreteJob Single Instance>
578 #propertyNames[] = {<propertyname>};
579 #propertyValues[] = {<propertyvalue>};
580 &smSetInstance($instance, #propertyNames[], #propertyValues[]);
581 &smEnd;
```

582 **6.5.2.2   General Usage of Set for Multiple Properties**

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

586 The requirement for supporting modification of a property using this command form shall be equivalent to
587 the requirement for supporting modification of the property using the ModifyInstance operation as defined
588 in the *CLP Service Profile*.

589 **6.5.2.2.1   Command Form**

590 `set <CIM_ConcreteJob single instance> <propertyname1>=<propertyvalue1>`
591 `<propertynamen>=<propertyvaluen>`

592 **6.5.2.2.2   CIM Requirements**

593 See CIM_ConcreteJob in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory
594 properties.

595 **6.5.2.2.3   Behavior Requirements**

```
596 $instance = <CIM_ConcreteJob Single Instance>
597 #propertyNames[] = {<propertyname>};
598
599 for #i < n
600 {
601     #propertyNames[#i] = <propertname#i>
602     #propertyValues[#i] = <propertyvalue#i>
603 }
604
```

```
605  &smSetInstance($instance, #propertyNames[], #propertyValues[]);
606  &smEnd;
```

6.5.3   Show

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

The `show` verb is used to display information about the CLP Service's Job Queue.

6.5.3.1   Show a Single Instance

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

6.5.3.1.1   Command Form

```
show <CIM_ConcreteJob single instance>
```

6.5.3.1.2   CIM Requirements

See CIM_ConcreteJob in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory
properties.

6.5.3.1.3   Behavior Requirements

6.5.3.1.3.1   Preconditions

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

6.5.3.1.3.2   Pseudo Code

```
$instance=<CIM_ConcreteJob Single Instance>
#propertylist[] = null;
if ( false == #all) {
    #propertylist[] = {//all mandatory non-key properties }
}
&smShowInstance($instance.getObjectPath(), #propertylist[]);
```

6.5.3.2   Show Multiple Instances Scoped by JobQueue

This command form is for the `show` verb applied to multiple instances of CIM_ConcreteJob. This
command form corresponds to UFsT-based selection within a scoping Job Queue.

6.5.3.2.1   Command Form

```
show <CIM_ConcreteJob multiple instances>
```

6.5.3.2.2   CIM Requirements

See CIM_ConcreteJob in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory
properties.

6.5.3.2.3   Behavior Requirements

6.5.3.2.3.1   Preconditions

```
1) $containerInstance contains the instance of CIM_JobQueue for which we are
   displaying scoped jobs (CIM_ConcreteJob instances). The CLP Service Profile
```

```
641       requires that the CIM_ConcreteJob instance be associated with its scoping job queue
642       via an instance of the CIM_JobDestinationJobs association.
643   2) #all is true if the all option was specified with the command; otherwise, #all is
644       false
```

6.5.3.2.3.2   Pseudo Code

```
646   #propertylist[] = null;
647   if (false == #all)  {
648       #propertylist[] = {//all mandatory non-key properties }
649   }
650
651   &smShowInstances ( "CIM_ConcreteJob", "CIM_JobDestinationJobs",
652       $containerInstance.getObjectPath(), #propertylist[] );
653       &smEnd;
```

6.5.4   Stop

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

The stop verb is used to terminate or kill a CLP operation.

6.5.4.1   Stop a Single Instance

This command form is for the stop verb applied to a single instance of CIM_ConcreteJob.

6.5.4.1.1   Command Form

stop [-f] <CIM_ProtocolService *single instance*>

6.5.4.1.2   CIM Requirements

```
663   uint16 EnabledState;
664   uint16 RequestedState;
665   uint32 EnabledLogicalElement.RequestStateChange (
666       [IN] uint16 RequestedState,
667       [OUT] REF CIM_ConcreteJob Job,
668       [IN] datetime TimeoutPeriod );
```

6.5.4.1.3   Behavior Requirements

6.5.4.1.3.1   Preconditions

```
671   #force is true if force option was specified; otherwise, #force is false
672   if (#force)  {
673       #requestedState = 5;//kill
674   }
675   else {
676       #requestedstate = 4;//terminate
677   }
678       $instance=<CIM_ProtocolService Single Instance>
679   &smRequestStateChange ( $instance.getObjectPath(), #requestedState );
680   smEnd;
```

6.6   CIM_ElementCapabilities

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

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

Table 6 – 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 6.6.2.
start	Not supported	
stop	Not supported	

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

6.6.1   Ordering of Results

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

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

6.6.2   Show

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

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

6.6.2.1   Show Multiple Instances – CIM_CLPCapabilities Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_ElementCapabilities where the reference specified is to an instance of CIM_CLPCapabilities. Multiple instances of CIM_ProtocolService can be associated with each instance of a CIM_CLPCapabilities.

6.6.2.1.1   Command Form

`show <CIM_ElementCapabilities multiple instances>`

708 **6.6.2.1.2 CIM Requirements**

709 See CIM_ElementCapabilities in the "CIM Elements" section of the *CLP Service Profile* for the list of
710 mandatory properties.

711 **6.6.2.1.3 Behavior Requirements**

712 **6.6.2.1.3.1 Preconditions**

```
713 $instance contains the instance of CIM_CLPCapabilities which is referenced by
714 CIM_ElementCapabilities
```

715 **6.6.2.1.3.2 Pseudo Code**

```
716 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );
717     smEnd;
```

718 **6.6.2.2 Show a Single Instance – CIM_ProtocolService Reference**

719 This command form is for the show verb applied to a single instance. This command form corresponds to
720 the show command issued against CIM_ElementCapabilities where the reference specified is to an
721 instance of CIM_ProtocolService. A single instance of CIM_CLPCapabilities is associated with each
722 instance of CIM_ProtocolService. Therefore, a single instance will be returned.

723 **6.6.2.2.1 Command Form**

724 **show <CIM_ElementCapabilities *single instance*>**

725 **6.6.2.2.2 CIM Requirements**

726 See CIM_ElementCapabilities in the "CIM Elements" section of the *CLP Service Profile* for the list of
727 mandatory properties.

728 **6.6.2.2.3 Behavior Requirements**

729 **6.6.2.2.3.1 Preconditions**

730 $instance contains the instance of CIM_ProtocolService which is referenced by
731 CIM_ElementCapabilities

732 **6.6.2.2.3.2 Pseudo Code**

```
733 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );
734     smEnd;
```

735 **6.6.2.3 Show a Single Instance – Both References**

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

739 **6.6.2.3.1 Command Form**

740 **show <CIM_ElementCapabilities *single instance*>**

741 **6.6.2.3.2 CIM Requirements**

742 See CIM_ElementCapabilities in the "CIM Elements" section of the *CLP Service Profile* for the list of
743 mandatory properties.

744   **6.6.2.3.3   Behavior Requirements**

745   **6.6.2.3.3.1   Preconditions**

```
746   1) $instanceA contains the instance of CIM_CLPCapabilities  which is referenced by
747      CIM_ElementCapabilities
748   2) $instanceB contains the instance of CIM_ProtocolService which is  referenced by
749      CIM_ElementCapabilities.
```

750   **6.6.2.3.3.2   Pseudo Code**

```
751   &smShowAssociationInstance ( "CIM_ElementCapabilities",
752      $instanceA.getObjectPath(),$instanceB.getObjectPath() );
753      smEnd;
```

754   ## 6.7   CIM_ElementSettingData

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

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

761   **Table 7 – Command Verb Requirements for CIM_ElementSettingData**

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

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

764   **6.7.1   Ordering of Results**

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

767   • Results for CIM_ElementSettingData are unordered; therefore, no algorithm is defined.

768   **6.7.2   Set**

769   This section describes how to implement the `set` verb when it is applied to an instance of
770   CIM_ElementSettingData. Implementations may support the use of the `set` verb with
771   CIM_ElementSettingData.

772   The set verb is used to modify properties of the CIM_ElementSettingData instance.

6.7.2.1   Set of IsNext

774   The IsNext property is the only property of CIM_ElementSettingData which can be modified directly via
775   the set verb.

6.7.2.1.1   Command Form

777   **set <CIM_ElementSettingData single instance> IsNext=<propertyvalue>**

6.7.2.1.2   CIM Requirements

779   See CIM_ElementSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of
780   mandatory properties.

6.7.2.1.3   Behavior Requirements

782   $instance = <CIM_ElementSettingData Single Instance>
783   #propertyNames[] = {"IsNext"};
784   #propertyValues[] = {<propertyvalue>};
785   &smSetInstance($instance, #propertyNames[], #propertyValues[]);
786   &smEnd;

6.7.3   Show

788   This section describes how to implement the show verb when applied to an instance of
789   CIM_ElementSettingData. Implementations shall support the use of the show verb with
790   CIM_ElementSettingData.

791   The show command is used to display information about the CIM_ElementSettingData instance or
792   instances.

6.7.3.1   Show Multiple Instances – CIM_CLPSettingData and CIM_CLPProtocolEndpoint

794   This command form corresponds to the show command issued against CIM_ElementSettingData where
795   the reference specified is to an instance of CIM_CLPSettingData. Note that when an instance of
796   CIM_CLPSettingData is associated with an instance of CIM_CLPProtocolEndpoint, the IsCurrent property
797   is the mandatory property.

6.7.3.1.1   Command Form

799   **show <CIM_ElementSettingData *multiple instances*>**

6.7.3.1.2   CIM Requirements

801   See CIM_ElementSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of
802   mandatory properties.

6.7.3.1.3   Behavior Requirements

6.7.3.1.3.1   Preconditions

805   1) $instance contains the instance of CIM_CLPSettingData which is referenced by
806   CIM_ElementSettingData
807   2) #all is true if the all option was specified

808   6.7.3.1.3.2   Pseudo Code

```
809   #propertylist = NULL;
810   if (false == #all) {
811       #propertylist = { "IsCurrent" };
812   }
813       &smShowAssociationInstances ( "CIM_ElementSettingData", $instance.getObjectPath(),
814       #propertylist[] );
815       smEnd;
```

816   6.7.3.2   Show Multiple Instances – CIM_CLPProtocolEndpoint Reference

817   This command form corresponds to the show command issued against CIM_ElementSettingData where
818   the reference specified is to an instance of CIM_CLPProtocolEndpoint. Note that when an instance of
819   CIM_CLPSettingData is associated with an instance of CIM_CLPProtocolEndpoint, the IsCurrent property
820   is the mandatory property.

821   6.7.3.2.1   Command Form

822   **show <CIM_ElementSettingData *multiple instances*>**

823   6.7.3.2.2   CIM Requirements

824   See CIM_ElementSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of
825   mandatory properties.

826   6.7.3.2.3   Behavior Requirements

827   6.7.3.2.3.1   Preconditions

```
828   1) $instance contains the instance of CIM_CLPProtocolEndpoint which is referenced by
829      CIM_ElementSettingData
830   2) #all is true if the all option was specified
831
```

832   6.7.3.2.3.2   Pseude Code

```
833   #propertylist = NULL;
834   if (false == #all) {
835       #propertylist = { "IsCurrent" };
836   }
837   &smShowAssociationInstances ( "CIM_ElementSettingData", $instance.getObjectPath(),
838   #propertylist[] );
839   smEnd;
```

840   6.7.3.3   Show a Single Instance – CIM_CLPSettingData and CIM_CLPProtocolEndpoint

841   This command form is for the show verb applied to a single instance. This command form corresponds to
842   the show command issued against CIM_ElementSettingData where both references are specified and
843   therefore the desired instance is unambiguously identified.

844   6.7.3.3.1   Command Form

845   **show <CIM_ElementSettingData *single instance*>**

846 **6.7.3.3.2   CIM Requirements**

847 See CIM_ElementSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of
848 mandatory properties.

849 **6.7.3.3.3   Behavior Requirements**

850 **6.7.3.3.3.1   Preconditions**

851 1) $instanceA contains the instance of CIM_CLPSettingData which is referenced by
852   CIM_ElementSettingData
853 2) $instanceB contains the instance of CIM_CLPProtocolEndpoint which is referenced by
854   CIM_ElementSettingData.
855 3) #all is true if the all option was specified

856 **6.7.3.3.3.2   Pseudo Code**

```
857 #propertylist = NULL;
858 if (false == #all)  {
859     #propertylist = { "IsCurrent" };
860 }
861 &smShowAssociationInstance ( "CIM_ElementSettingData",
862     $instanceA.getObjectPath(),$instanceB.getObjectPath(),
863 #propertylist[] );
864 smEnd;
```

865 **6.7.3.4   Show Multiple Instances – CIM_CLPSettingData and CIM_ProtocolService**

866 This command form corresponds to the show command issued against CIM_ElementSettingData where
867 the reference specified is to an instance of CIM_CLPSettingData. Note that when an instance of
868 CIM_CLPSettingData is associated with an instance of CIM_ProtocolService, the IsNext and IsDefault
869 properties are mandatory.

870 **6.7.3.4.1   Command Form**

871 **show <CIM_ElementSettingData *multiple instances*>**

872 **6.7.3.4.2   CIM Requirements**

873 See CIM_ElementSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of
874 mandatory properties.

875 **6.7.3.4.3   Behavior Requirements**

876 **6.7.3.4.3.1   Preconditions**

877 1) $instance contains the instance of CIM_CLPSettingData which is referenced by
878   CIM_ElementSettingData
879 2) #all is true if the all option was specified

880 **6.7.3.4.3.2   Pseudo Code**

```
881 #propertylist[] = NULL;
882 if (false == #all)  {
883     #propertylist = { "IsNext", "IsDefault" };
884 }
885     &smShowAssociationInstances ( "CIM_ElementSettingData", $instance.getObjectPath(),
```

886   ```
       #propertylist[]);
887   smEnd;
       ```

### 6.7.3.5    Show Multiple Instances – CIM_ProtocolService Reference

889    This command form corresponds to the show command issued against CIM_ElementSettingData where
890    the reference specified is to an instance of CIM_ProtocolService. Note that when an instance of
891    CIM_CLPSettingData is associated with an instance of CIM_ProtocolService, the IsNext and IsDefault
892    properties are mandatory.

#### 6.7.3.5.1    Command Form

894    ```
       show <CIM_ElementSettingData multiple instances>
       ```

6.7.3.5.2   CIM Requirements

896   See CIM_ElementSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of
897   mandatory properties.

6.7.3.5.3   Behavior Requirements

6.7.3.5.3.1   Preconditions

900   ```
901   1) $instance contains the instance of CIM_ProtocolService which is referenced by
           CIM_ElementSettingData
902   2) #all is true if the all option was specified
       ```

##### 6.7.3.5.3.2    Pseudo Code

904    ```
       #propertylist[] = NULL;
905    if (false == #all)  {
906        #propertylist = { "IsNext", "IsDefault" };
907    }
908    &smShowAssociationInstances ( "CIM_ElementSettingData", $instance.getObjectPath(),
909    #propertylist[] );
910    smEnd;
       ```

6.7.3.6   Show a Single Instance – CIM_CLPSettingData and CIM_ProtocolService

912   This command form is for the show verb applied to a single instance. This command form corresponds to
913   the show command issued against CIM_ElementSettingData where both references are specified and
914   therefore the desired instance is unambiguously identified.

6.7.3.6.1   Command Form

916   ```
       show <CIM_ElementSettingData single instance>
       ```

#### 6.7.3.6.2    CIM Requirements

918    See CIM_ElementSettingData in the "CIM Elements" section of the *CLP Service Profile* for the list of
919    mandatory properties.

#### 6.7.3.6.3    Behavior Requirements

##### 6.7.3.6.3.1    Preconditions

922    ```
923    1) $instanceA contains the instance of CIM_CLPSettingData  which is referenced by
           CIM_ElementSettingData
       ```

924   2) $instanceB contains the instance of CIM_ProtocolService which is referenced by
925   CIM_ElementSettingData.
926   3) #all is true if the all option was specified

927 **6.7.3.6.3.2   Pseudo Code**

```
928  #propertylist[] = NULL;
929  if (false == #all)  {
930      #propertylist = { "IsNext", "IsDefault" };
931  }
932  &smShowAssociationInstance ( "CIM_ElementSettingData",
933      $instanceA.getObjectPath(),$instanceB.getObjectPath(), #propertylist[] );
934  smEnd;
```

935 ## 6.8   CIM_Error

936 CIM_Error is utilized as a template for embedded instances. Instances of the class are not exposed by
937 the underlying profile. Therefore, the usage of any CLP verbs shall not be supported for instances of
938 CIM_Error.

939 ## 6.9   CIM_HostedAccessPoint

940 The `cd` and `help` verbs shall be supported as described in **DSP0216**.

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

946 **Table 8 – 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 6.9.2.
start	Not supported	
stop	Not supported	

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

949 **6.9.1   Ordering of Results**

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

952   • Results for CIM_HostedAccessPoint are unordered; therefore, no algorithm is defined.

953 6.9.2 Show

954 This section describes how to implement the `show` verb when applied to an instance of
955 CIM_HostedAccessPoint. Implementations shall support the use of the `show` verb with
956 CIM_HostedAccessPoint.

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

959 6.9.2.1 Show Multiple Instances

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

963 6.9.2.2 Command Form

964 `show <CIM_HostedAccessPoint multiple instances>`

965 6.9.2.2.1 CIM Requirements

966 See CIM_HostedAccessPoint in the "CIM Elements" section of the *CLP Service Profile* for the list of
967 mandatory properties.

968 6.9.2.2.2 Behavior Requirements

969 6.9.2.2.2.1 Preconditions

970 1) $instance contains the instance of CIM_ComputerSystem which is referenced by
971   CIM_HostedAccessPoint

972 6.9.2.2.2.2 Pseudo Code

973 `&smShowAssociationInstances ( "CIM_HostedAccessPoint", $instance.getObjectPath() );`
974 `smEnd;`

975 6.9.2.3 Show a Single Instance – CIM_CLPProtocolEndpoint Reference

976 This command form is for the `show` verb applied to a single instance. This command form corresponds to
977 the `show` command issued against CIM_HostedAccessPoint where the reference specified is to an
978 instance of CIM_CLPProtocolEndpoint. A single instance will be returned.

979 6.9.2.3.1 Command Form

980 `show <CIM_HostedAccessPoint single instance>`

981 6.9.2.3.2 CIM Requirements

982 See CIM_HostedAccessPoint in the "CIM Elements" section of the *CLP Service Profile* for the list of
983 mandatory properties.

984 6.9.2.3.3 Behavior Requirements

985 6.9.2.3.3.1 Preconditions

986 1) $instance contains the instance of CIM_ProtocolEndpoint that is referenced by
987   CIM_HostedAccessPoint

988 **6.9.2.3.3.2 Pseudo Code**

```
989  &smShowAssociationInstances ( "CIM_HostedAccessPoint", $instance.getObjectPath() );
990  smEnd;
```

991 **6.9.2.4 Show a Single Instance – Both References**

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

995 **6.9.2.4.1 Command Form**

996 **show <CIM_HostedAccessPoint *single instance*>**

997 **6.9.2.4.2 CIM Requirements**

998 See CIM_HostedAccessPoint in the "CIM Elements" section of the *CLP Service Profile* for the list of
999 mandatory properties.

1000 **6.9.2.4.3 Behavior Requirements**

1001 **6.9.2.4.3.1 Preconditions**

```
1002  1) $instanceA contains the instance of CIM_ComputerSystem which is referenced by
1003     CIM_HostedAccessPoint
1004  2) $instanceB contains the instance of CIM_CLPProtocolEndpoint that is referenced by
1005     CIM_HostedAccessPoint
```

1006 **6.9.2.4.3.2 Pseudo Code**

```
1007  &smShowAssociationInstance ( "CIM_HostedAccessPoint",
1008  $instanceA.getObjectPath(),$instanceB.getObjectPath() );
1009  smEnd;
```

1010 ## 6.10 CIM_HostedJobDestination

1011 The cd and help verbs shall be supported as described in DSP0216.

1012 Table 9 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of
1013 the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
1014 verb and target. Table 9 is for informational purposes only; in case of a conflict between Table 9 and
1015 requirements detailed in the following sections, the text detailed in the following sections supersedes the
1016 information in Table 9.

1017 **Table 9 – Command Verb Requirements for CIM_HostedJobDestination**

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.

Command Verb	Requirement	Comments
start	Not supported	
stop	Not supported	

1018 No mapping is defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,
1019 `reset`, `set`, `start`, and `stop`.

6.10.1   Ordering of Results

1021 When results are returned for multiple instances of CIM_HostedJobDestination, implementations shall
1022 utilize the following algorithm to produce the natural (that is, default) ordering:

1023 • Results for CIM_HostedJobDestination are unordered; therefore, no algorithm is defined.

6.10.2   Show

1025 This section describes how to implement the `show` verb when applied to an instance of
1026 CIM_HostedJobDestination. Implementations shall support the use of the `show` verb with
1027 CIM_HostedJobDestination.

1028 The `show` command is used to display information about the CIM_HostedJobDestination instance or
1029 instances.

6.10.2.1   Show a Single Instance – CIM_JobQueue Reference

1031 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1032 the `show` command issued against CIM_HostedJobDestination where the reference specified is to an
1033 instance of CIM_JobQueue. An instance of CIM_JobQueue is referenced by exactly one instance of
1034 CIM_HostedJobDestination. Therefore, a single instance will be returned.

6.10.2.1.1   Command Form

1036 `show <CIM_HostedJobDestination single instance>`

6.10.2.1.2   CIM Requirements

1038 See CIM_HostedJobDestination in the "CIM Elements" section of the *CLP Service Profile* for the list of
1039 mandatory properties.

6.10.2.1.3   Behavior Requirements

6.10.2.1.3.1   Preconditions

```
1042  1) $instance contains the instance of CIM_JobQueue
1043     which is referenced by CIM_HostedJobDestination
1044     &smShowAssociationInstances ( "CIM_HostedJobDestination",
1045     $instance.getObjectPath() );
1046     smEnd;
```

6.10.2.2   Show a Single Instance – CIM_ComputerSystem Reference

1048 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1049 the `show` command issued against CIM_HostedJobDestination where the reference specified is to an
1050 instance of CIM_ComputerSystem. An instance of CIM_ComputerSystem is referenced by exactly one
1051 instance of CIM_HostedJobDestination. Therefore, a single instance will be returned.

6.10.2.2.1   Command Form

```
show <CIM_HostedJobDestination single instance>
```

6.10.2.2.2   CIM Requirements

See CIM_HostedJobDestination in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.10.2.2.3   Behavior Requirements

6.10.2.2.3.1   Preconditions

```
1) $instance contains the instance of CIM_ComputerSystem which is referenced by
   CIM_HostedJobDestination
```

6.10.2.2.3.2   Pseudo Code

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

6.10.2.3   Show a Single Instance – Both References

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

6.10.2.3.1   Command Form

```
show <CIM_HostedJobDestination single instance>
```

6.10.2.3.2   CIM Requirements

See CIM_HostedJobDestination in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.10.2.3.3   Behavior Requirements

6.10.2.3.3.1   Preconditions

```
1) $instanceA contains the instance of CIM_ComputerSystem which is referenced by
   CIM_HostedJobDestination
2) $instanceB contains the instance of CIM_JobQueue which is referenced by
   CIM_HostedJobDestination.
```

6.10.2.3.3.2   Pseudo Code

```
&smShowAssociationInstance ( "CIM_HostedJobDestination",
    $instanceA.getObjectPath(),$instanceB.getObjectPath() );
smEnd;
```

6.11   CIM_HostedService

The cd and help verbs shall be supported as described in DSP0216.

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

1090 **Table 10 – 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 6.11.2.
start	Not supported	
stop	Not supported	

1091 No mapping is defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,
1092 `reset`, `set`, `start`, and `stop`.

6.11.1 Ordering of Results

1094 When results are returned for multiple instances of CIM_HostedService, implementations shall utilize the
1095 following algorithm to produce the natural (that is, default) ordering:

1096 • Results for CIM_HostedService are unordered; therefore, no algorithm is defined.

6.11.2 Show

1098 This section describes how to implement the `show` verb when applied to an instance of
1099 CIM_HostedService. Implementations shall support the use of the `show` verb with CIM_HostedService.

1100 The `show` command is used to display information about the CIM_HostedService instance or instances.

6.11.2.1 Show Multiple Instances

1102 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1103 to a `show` command issued against CIM_HostedService where only one reference is specified and the
1104 reference is to an instance of CIM_ComputerSystem.

6.11.2.1.1 Command Form

1106 `show <CIM_HostedService multiple instances>`

6.11.2.1.2 CIM Requirements

1108 See CIM_HostedService in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory
1109 properties.

6.11.2.1.3 Behavior Requirements

6.11.2.1.3.1 Preconditions

1112 `1)  $instance contains the instance of CIM_ComputerSystem which is referenced by`
1113 `    CIM_HostedService`

6.11.2.1.3.2   Pseudo Code

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

6.11.2.2   Show a Single Instance – CIM_ProtocolService Reference

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

6.11.2.2.1   Command Form

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

6.11.2.2.2   CIM Requirements

See CIM_HostedService in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.11.2.2.3   Behavior Requirements

6.11.2.2.3.1   Preconditions

```
1) $instance contains the instance of CIM_ProtocolService which is referenced by
   CIM_HostedService
```

6.11.2.2.3.2   Pseudo Code

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

6.11.2.3   Show a Single Instance – Both References

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

6.11.2.3.1   Command Form

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

6.11.2.3.2   CIM Requirements

See CIM_HostedService in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.11.2.3.3   Behavior Requirements

6.11.2.3.3.1   Preconditions

```
1) $instanceA contains the instance of CIM_ComputerSystem which is referenced by
   CIM_HostedService
2) $instanceB contains the instance of CIM_ProtocolService which is referenced by
   CIM_HostedService.
```

1149 **6.11.2.3.3.2   Pseudo Code**

```
1150  &smShowAssociationInstance ( "CIM_HostedService",
1151      $instanceA.getObjectPath(),$instanceB.getObjectPath() );
1152  smEnd;
```

1153 6.12   CIM_JobDestinationJobs

1154 The cd and help verbs shall be supported as described in DSP0216.

1155 Table 11 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
1156 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
1157 verb and target. Table 11 is for informational purposes only; in case of a conflict between Table 11 and
1158 requirements detailed in the following sections, the text detailed in the following sections supersedes the
1159 information in Table 11.

1160   **Table 11 – Command Verb Requirements for CIM_JobDestinationJobs**

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	

1161 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
1162 reset, set, start, and stop.

1163 6.12.1   Ordering of Results

1164 When results are returned for multiple instances of CIM_JobDestinationJobs, implementations shall utilize
1165 the following algorithm to produce the natural (that is, default) ordering:

1166   • Results for CIM_JobDestinationJobs are unordered; therefore, no algorithm is defined.

1167 6.12.2   Show

1168 This section describes how to implement the show verb when applied to an instance of
1169 CIM_JobDestinationJobs. Implementations shall support the use of the show verb with
1170 CIM_JobDestinationJobs.

1171 The show command is used to display information about the CIM_JobDestinationJobs instance or
1172 instances.

6.12.2.1   Show a Single Instance – CIM_ConcreteJob Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_JobDestinationJobs where the reference specified is to an instance of CIM_ConcreteJob. A single instance of CIM_JobQueue is associated with each instance of a CIM_ConcreteJob Therefore, a single instance will be returned.

6.12.2.1.1   Command Form

```
show <CIM_JobDestinationJobs single instance>
```

6.12.2.1.2   CIM Requirements

See CIM_JobDestinationJobs in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.12.2.1.3   Behavior Requirements

6.12.2.1.3.1   Preconditions

```
$instance contains the instance of CIM_ConcreteJob which is referenced by
    CIM_JobDestinationJobs
```

6.12.2.1.3.2   Pseudo Code

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

6.12.2.2   Show Multiple Instances – CIM_JobQueue Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_JobDestinationJobs where the reference specified is to an instance of CIM_JobQueue. A single instance of CIM_JobQueue is associated with multiple instances of CIM_ConcreteJob. Therefore, multiple instances may be returned.

6.12.2.2.1   Command Form

```
show <CIM_JobDestinationJobs multiple instances>
```

6.12.2.2.2   CIM Requirements

See CIM_JobDestinationJobs in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.12.2.2.3   Behavior Requirements

6.12.2.2.3.1   Preconditions

```
$instance contains the instance of CIM_JobQueue which is referenced by
    CIM_JobDestinationJobs
```

6.12.2.2.3.2   Pseudo Code

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

1207 **6.12.2.3   Show a Single Instance – Both References**

1208 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1209 the `show` command issued against CIM_JobDestinationJobs where both references are specified and
1210 therefore the desired instance is unambiguously identified.

1211 **6.12.2.3.1   Command Form**

1212 `show <CIM_JobDestinationJobs single instance>`

1213 **6.12.2.3.2   CIM Requirements**

1214 See CIM_JobDestinationJobs in the "CIM Elements" section of the *CLP Service Profile* for the list of
1215 mandatory properties.

1216 **6.12.2.3.3   Behavior Requirements**

1217 **6.12.2.3.3.1   Preconditions**

```
1218  1)  $instanceA contains the instance of CIM_ConcreteJob which is referenced by
1219      CIM_JobDestinationJobs
1220  2)  $instanceB contains the instance of CIM_JobQueue which is referenced by
1221      CIM_JobDestinationJobs.
```

1222 **6.12.2.3.3.2   Pseudo Code**

```
1223  &smShowAssociationInstance ( "CIM_JobDestinationJobs",
1224      $instanceA.getObjectPath(),$instanceB.getObjectPath() );
1225  smEnd;
```

1226 ## 6.13   CIM_JobQueue

1227 The `cd` and `help` verbs shall be supported as described in DSP0216.

1228 Table 12 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
1229 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
1230 verb and target. Table 12 is for informational purposes only; in case of a conflict between Table 12 and
1231 requirements detailed in the following sections, the text detailed in the following sections supersedes the
1232 information in Table 12.

1233   **Table 12 – Command Verb Requirements for CIM_JobQueue**

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 6.13.2.
start	Not supported	
stop	Not supported	

1234 No mapping is defined for the following verbs for the specified target: `dump` and `load`.

6.13.1   Ordering of Results

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

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

6.13.2   Show

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

The `show` verb is used to display information about the CLP Service's Job Queue.

6.13.2.1   Show a Single Instance

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

6.13.2.1.1   Command Form

```
show <CIM_JobQueue single instance>
```

6.13.2.1.2   CIM Requirements

See CIM_JobQueue in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.13.2.1.3   Behavior Requirements

6.13.2.1.3.1   Preconditions

```
#all is true if the all option was specified with the command; otherwise, #all is
false
$instance=<CIM_JobQueue Single Instance>
#propertylist[] = null;
if ( false == #all) {
    #propertylist[] = {//all mandatory non-key properties }
}
&smShowInstance($instance.getObjectPath(), #propertylist[]);
```

6.13.2.2   Show Multiple Instances

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

6.13.2.2.1   Command Form

```
show <CIM_JobQueue single instance>
```

6.13.2.2.2   CIM Requirements

See CIM_JobQueue in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.13.2.2.3   Behavior Requirements

6.13.2.2.3.1   Preconditions

```
1) $containerInstance contains the instance of CIM_ComputerSystem for which we are
   displaying scoped CIM_JobQueue instances.
```

```
1271   2) #all is true if the all option was specified with the command; otherwise, #all is
1272      false
```

1273   **6.13.2.2.3.2   Pseudo Code**

```
1274   #propertylist[] = null;
1275   if (false == #all)  {
1276       #propertylist[] = {//all mandatory non-key properties }
1277   }
1278   &smShowInstances ( "CIM_JobQueue", "CIM_HostedJobDestination",
1279       $containerInstance.getObjectPath(), #propertylist[] );
1280       &smEnd;
```

1281   **6.14   CIM_OwningJobElement**

1282   The cd and help verbs shall be supported as described in DSP0216.

1283   Table 13 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
1284   of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
1285   verb and target. Table 13 is for informational purposes only; in case of a conflict between Table 13 and
1286   requirements detailed in the following sections, the text detailed in the following sections supersedes the
1287   information in Table 13.

1288   **Table 13 – Command Verb Requirements for CIM_OwningJobElement**

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 6.14.2.
start	Not supported	
stop	Not supported	

1289   No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
1290   reset, set, start, and stop.

1291   **6.14.1   Ordering of Results**

1292   When results are returned for multiple instances of CIM_OwningJobElement, implementations shall utilize
1293   the following algorithm to produce the natural (that is, default) ordering:

1294   •   Results for CIM_OwningJobElement are unordered; therefore, no algorithm is defined.

1295   **6.14.2   Show**

1296   This section describes how to implement the show verb when applied to an instance of
1297   CIM_OwningJobElement. Implementations shall support the use of the show verb with
1298   CIM_OwningJobElement.

1299 The `show` command is used to display information about the CIM_OwningJobElement instance or
1300 instances.

6.14.2.1 Show a Single Instance – CIM_ConcreteJob Reference

1302 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1303 the `show` command issued against CIM_OwningJobElement where the reference specified is to an
1304 instance of CIM_ConcreteJob. A single instance of CIM_ProtocolService is associated with each instance
1305 of a CIM_ConcreteJob Therefore, a single instance will be returned.

6.14.2.1.1 Command Form

1307 `show <CIM_OwningJobElement single instance>`

6.14.2.1.2 CIM Requirements

1309 See CIM_OwningJobElement in the "CIM Elements" section of the *CLP Service Profile* for the list of
1310 mandatory properties.

6.14.2.1.3 Behavior Requirements

6.14.2.1.3.1 Preconditions

1313 `1) $instance contains the instance of CIM_ConcreteJob which is referenced by`
1314 `   CIM_OwningJobElement`

6.14.2.1.3.2 Pseudo Code

1316 `&smShowAssociationInstances ( "CIM_OwningJobElement", $instance.getObjectPath() );`
1317 `smEnd;`

6.14.2.2 Show Multiple Instances – CIM_ProtocolService Reference

1319 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1320 the `show` command issued against CIM_OwningJobElement where the reference specified is to an
1321 instance of CIM_ProtocolService. A single instance of CIM_ProtocolService is associated with multiple
1322 instances of CIM_ConcreteJob. Therefore, multiple instances may be returned.

6.14.2.2.1 Command Form

1324 `show <CIM_OwningJobElement multiple instances>`

6.14.2.2.2 CIM Requirements

1326 See CIM_OwningJobElement in the "CIM Elements" section of the *CLP Service Profile* for the list of
1327 mandatory properties.

6.14.2.2.3 Behavior Requirements

6.14.2.2.3.1 Preconditions

1330 `1) $instance contains the instance of CIM_ProtocolService which is referenced by`
1331 `   CIM_OwningJobElement`

6.14.2.2.3.2 Pseudo Code

1333 `&smShowAssociationInstances ( "CIM_OwningJobElement", $instance.getObjectPath() );`
1334 `smEnd;`

6.14.2.3   Show a Single Instance – Both References

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

6.14.2.3.1   Command Form

```
show <CIM_OwningJobElement single instance>
```

6.14.2.3.2   CIM Requirements

See CIM_OwningJobElement in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.14.2.3.3   Behavior Requirements

6.14.2.3.3.1   Preconditions

```
1)  $instanceA contains the instance of CIM_ConcreteJob which is referenced by
    CIM_OwningJobElement
2)  $instanceB contains the instance of CIM_ProtocolService which is referenced by
    CIM_OwningJobElement.
```

6.14.2.3.3.2   Pseudo Code

```
&smShowAssociationInstance ( "CIM_OwningJobElement",
    $instanceA.getObjectPath(),$instanceB.getObjectPath() );
smEnd;
```

6.15   CIM_ProtocolService

The cd and help verbs shall be supported as described in DSP0216.

Table 14 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 14 is for informational purposes only; in case of a conflict between Table 14 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 14.

Table 14 – Command Verb Requirements for CIM_ProtocolService

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	May	See 6.15.2.
set	May	See 6.15.3.
show	Shall	See 6.15.4.
start	May	See 6.15.5.
stop	May	See 6.15.6.

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

6.15.1   Ordering of Results

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

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

6.15.2   Reset

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

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

6.15.2.1   Reset a Single Instance

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

6.15.2.1.1   Command Form

```
reset <CIM_ProtocolService single instance>
```

6.15.2.1.2   CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
    [IN] uint16 RequestedState,
    [OUT] REF CIM_ConcreteJob Job,
    [IN] datetime TimeoutPeriod );
```

6.15.2.1.3   Behavior Requirements

```
$instance=<CIM_ProtocolService Single Instance>
smResetRSC($instance.getObjectPath());
&smEnd;
```

6.15.3   Set

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

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

6.15.3.1   General Usage of Set for a Single Property

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

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *CLP Service Profile*.

6.15.3.1.1   Command Form

```
set <CIM_ProtocolService Single Instance> <propertyname>=<propertyvalue>
```

6.15.3.1.2   CIM Requirements

See CIM_ProtocolService in the "CIM Elements" section of the *CLP Service Profile* for the list of modifiable properties.

6.15.3.1.3   Behavior Requirements

```
$instance = <CIM_ProtocolService Single Instance>
#propertyNames[] = {<propertyname>};
#propertyValues[] = {<propertyvalue>};
&smSetInstance($instance, #propertyNames[], #propertyValues[]);
&smEnd;
```

6.15.3.2   General Usage of Set for Multiple Properties

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

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *CLP Service Profile*.

6.15.3.2.1   Command Form

```
set <CIM_ProtocolService Single Instance> <propertyname1>=<propertyvalue1>
<propertynamen>=<propertyvaluen>
```

6.15.3.2.2   CIM Requirements

See CIM_ProtocolService in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.15.3.2.3   Behavior Requirements

```
$instance = <CIM_ProtocolService Single Instance>
#propertyNames[] = {<propertyname>};

for #i < n
{
    #propertyNames[#i] = <propertname#i>
    #propertyValues[#i] = <propertyvalue#i>
}

&smSetInstance($instance, #propertyNames[], #propertyValues[]);
&smEnd;
```

6.15.4   Show

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

The show verb is used to display information about the CIM_ProtocolService.

6.15.4.1   Show a Single Instance

This command form is for the show verb applied to a single instance of CIM_ProtocolService.

1441 **6.15.4.1.1 Command Form**
1442 `show <CIM_ProtocolService single instance>`

1443 **6.15.4.1.2 CIM Requirements**

1444 See CIM_ProtocolService in the "CIM Elements" section of the *CLP Service Profile* for the list of
1445 mandatory properties.

1446 **6.15.4.1.3 Behavior Requirements**

1447 **6.15.4.1.3.1 Preconditions**
1448 `#all is true if the all option was specified with the command; otherwise, #all is`
1449 `false`
1450 `$instance=<CIM_ProtocolService Single Instance>`

1451 **6.15.4.1.3.2 Pseudo Code**
1452 `#propertylist[] = null;`
1453 `if (false == #all) {`
1454 `#propertylist[] = { //all mandatory non-key properties };`
1455 `}`
1456
1457 `&smShowInstance($instance.getObjectPath(), #propertylist[]);`
1458 `&smEnd;`

1459 **6.15.4.2 Show Multiple Instances**

1460 This command form is for the `show` verb applied to multiple instances of CIM_ProtocolService. This
1461 command form corresponds to UFsT-based selection within a scoping system.

1462 **6.15.4.2.1 Command Form**
1463 `show <CIM_ProtocolService multiple instances>`

1464 **6.15.4.2.2 CIM Requirements**

1465 See CIM_ProtocolService in the "CIM Elements" section of the *CLP Service Profile* for the list of
1466 mandatory properties.

1467 **6.15.4.2.3 Behavior Requirements**

1468 **6.15.4.2.3.1 Preconditions**
1469 `1) $containerInstance contains the instance of CIM_ComputerSystem for which we are`
1470 `   displaying scoped instances of the CIM_ProtocolService. The CLP Service Profile`
1471 `   requires that the CIM_ProtocolService instance be associated with its scoping`
1472 `   system via an instance of the CIM_HostedService association.`
1473 `2) #all is true if the all option was specified with the command; otherwise, #all is`
1474 `   false`

1475 **6.15.4.2.3.2 Pseudo Code**
1476 `#propertylist[] = null;`
1477 `if (false == #all) {`
1478 `    #propertylist[] = { //all mandatory non-key properties };`
1479 `}`
1480
1481 `&smShowInstances ( "CIM_ProtocolService", "CIM_HostedService",`
1482 `$containerInstance.getObjectPath(), #propertylist[] );`
1483 `&smEnd;`

6.15.5   Start

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

The `start` verb is used to enable the CIM_ProtocolService.

6.15.5.1   Start a Single Instance

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

6.15.5.1.1   Command Form

```
start <CIM_ProtocolService single instance>
```

6.15.5.1.2   CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
    [IN] uint16 RequestedState,
    [OUT] REF CIM_ConcreteJob Job,
    [IN] datetime TimeoutPeriod );
```

6.15.5.1.3   Behavior Requirements

```
$instance=<CIM_ProtocolService Single Instance>
smStartRSC($instance.getObjectPath());
&smEnd;
```

6.15.6   Stop

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

The `stop` verb is used to disable the CIM_ProtocolService.

6.15.6.1   Stop a Single Instance

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

6.15.6.1.1   Command Form

```
stop <CIM_ProtocolService single instance>
```

6.15.6.1.2   CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
    [IN] uint16 RequestedState,
    [OUT] REF CIM_ConcreteJob Job,
    [IN] datetime TimeoutPeriod );
```

6.15.6.1.3   Behavior Requirements

```
$instance=<CIM_ProtocolService Single Instance>
smStopRSC($instance.getObjectPath());
&smEnd;
```

6.16 CIM_ProvidesEndpoint

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

Table 15 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 15 is for informational purposes only; in case of a conflict between Table 15 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 15.

Table 15 – Command Verb Requirements for CIM_ProvidesEndpoint

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 6.16.2.
start	Not supported	
stop	Not supported	

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

6.16.1 Ordering of Results

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

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

6.16.2 Show

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

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

6.16.2.1 Show a Single Instance – CIM_CLPProtocolEndpoint Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_ProvidesEndpoint where the reference specified is to an instance of CIM_CLPProtocolEndpoint. A single instance of CIM_ProtocolService is associated with each instance of a CIM_CLPProtocolEndpoint. Therefore, a single instance will be returned.

6.16.2.1.1 Command Form

`show <CIM_ProvidesEndpoint single instance>`

6.16.2.1.2   CIM Requirements

See CIM_ProvidesEndpoint in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.16.2.1.3   Behavior Requirements

6.16.2.1.3.1   Preconditions

```
1) $instance contains the instance of CIM_CLPProtocolEndpoint which is referenced by
   CIM_ProvidesEndpoint
```

6.16.2.1.3.2   Pseudo Code

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

6.16.2.2   Show Multiple Instances – CIM_ProtocolService Reference

This command form is for the show verb applied to a single instance. This command form corresponds to the show command issued against CIM_ProvidesEndpoint where the reference specified is to an instance of CIM_ProtocolService. A single instance of CIM_ProtocolService is associated with multiple instances of a CIM_CLPProtocolEndpoint. Therefore, multiple instances may be returned.

6.16.2.2.1   Command Form

```
show <CIM_ProvidesEndpoint multiple instances>
```

6.16.2.2.2   CIM Requirements

See CIM_ProvidesEndpoint in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.16.2.2.3   Behavior Requirements

6.16.2.2.3.1   Preconditions

```
1) $instance contains the instance of CIM_ProtocolService which is referenced by
   CIM_ProvidesEndpoint
```

6.16.2.2.3.2   Pseudo Code

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

6.16.2.3   Show a Single Instance – Both References

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

6.16.2.3.1   Command Form

```
show <CIM_ProvidesEndpoint single instance>
```

6.16.2.3.2   CIM Requirements

See CIM_ProvidesEndpoint in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

1586 **6.16.2.3.3   Behavior Requirements**

1587 **6.16.2.3.3.1   Preconditions**

```
1588  1) $instanceA contains the instance of CIM_CLPProtocolEndpoint which is referenced by
1589     CIM_ProvidesEndpoint
1590  2) $instanceB contains the instance of CIM_ProtocolService which is referenced by
1591     CIM_ProvidesEndpoint.
```

1592 **6.16.2.3.3.2   Pseudo Code**

```
1593  &smShowAssociationInstance ( "CIM_ProvidesEndpoint",
1594      $instanceA.getObjectPath(),$instanceB.getObjectPath() );
1595  smEnd;
```

1596 ## 6.17   CIM_ServiceAffectsElement

1597 The `cd` and `help` verbs shall be supported as described in DSP0216.

1598 Table 16 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
1599 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
1600 verb and target. Table 16 is for informational purposes only; in case of a conflict between Table 16 and
1601 requirements detailed in the following sections, the text detailed in the following sections supersedes the
1602 information in Table 16.

1603   **Table 16 – 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 6.17.2.
start	Not supported	
stop	Not supported	

1604 No mapping is defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,
1605 `reset`, `set`, `start`, and `stop`.

1606 **6.17.1   Ordering of Results**

1607 When results are returned for multiple instances of CIM_ServiceAffectsElement, implementations shall
1608 utilize the following algorithm to produce the natural (that is, default) ordering:

1609   • Results for CIM_ServiceAffectsElement are unordered; therefore, no algorithm is defined.

1610 **6.17.2   Show**

1611 This section describes how to implement the `show` verb when applied to an instance of
1612 CIM_ServiceAffectsElement. Implementations shall support the use of the `show` verb with
1613 CIM_ServiceAffectsElement.

1614 The show command is used to display information about the CIM_ServiceAffectsElement instance or
1615 instances.

6.17.2.1   Show Multiple Instances – CIM_ProtocolService Reference

1617 This command form is for the show verb applied to a single instance. This command form corresponds to
1618 the show command issued against CIM_ServiceAffectsElement where the reference specified is to an
1619 instance of CIM_ProtocolService. Multiple instances of CIM_ServiceAffectsElement reference an instance
1620 of CIM_ProtocolService.

6.17.2.1.1   Command Form

1622 `show < CIM_ServiceAffectsElement `*`multiple instances`*`>`

6.17.2.1.2   CIM Requirements

1624 See CIM_ServiceAffectsElement in the "CIM Elements" section of the *CLP Service Profile* for the list of
1625 mandatory properties.

6.17.2.1.3   Behavior Requirements

6.17.2.1.3.1   Preconditions

1628 `1) $instance contains the instance of CIM_ProtocolService which is referenced by`
1629 `    CIM_ServiceAffectsElement`

6.17.2.1.3.2   Pseudo Code

1631 `&smShowAssociationInstances ( "CIM_ServiceAffectsElement",`
1632 `    $instance.getObjectPath() );`
1633 `smEnd;`

6.17.2.2   Show a Single Instance – CIM_JobQueue Reference

1635 This command form is for the show verb applied to a single instance. This command form corresponds to
1636 the show command issued against CIM_ServiceAffectsElement where the reference specified is to an
1637 instance of CIM_JobQueue. A single instance of CIM_ProtocolService is associated with each instance of
1638 a CIM_JobQueue. Therefore, a single instance will be returned.

6.17.2.2.1   Command Form

1640 `show < CIM_ServiceAffectsElement `*`single instance`*`>`

6.17.2.2.1.1   CIM Requirements

1642 See CIM_ServiceAffectsElement in the "CIM Elements" section of the *CLP Service Profile* for the list of
1643 mandatory properties.

6.17.2.2.2   Behavior Requirements

6.17.2.2.2.1   Preconditions

1646 `1) $instance contains the instance of CIM_JobQueue which is referenced by`
1647 `    CIM_ServiceAffectsElement`

6.17.2.2.2.2   Pseudo Code

1649 `&smShowAssociationInstances ( "CIM_ServiceAffectsElement",`
1650 `    $instance.getObjectPath() );`
1651 `smEnd;`

6.17.2.3   Show a Single Instance – CIM_AdminDomain Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_ServiceAffectsElement where the reference specified is to an instance of CIM_AdminDomain. A single instance of CIM_ProtocolService is associated with each instance of a CIM_AdminDomain. Therefore, a single instance will be returned.

6.17.2.3.1   Command Form

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

6.17.2.3.2   CIM Requirements

See CIM_AffectsElement in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.17.2.3.3   Behavior Requirements

6.17.2.3.3.1   Preconditions

```
$instance contains the instance of CIM_AdminDomain which is referenced by
   CIM_ServiceAffectsElement
```

6.17.2.3.3.2   Pseudo Code

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

6.17.2.4   Show a Single Instance – Both References

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

6.17.2.4.1   Command Form

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

6.17.2.4.2   CIM Requirements

See CIM_AffectsElement in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.17.2.4.3   Behavior Requirements

6.17.2.4.3.1   Preconditions

```
1) $instanceA contains the instance of CIM_JobQueue or CIM_AdminDomain that is
   referenced by CIM_ServiceAffectsElement
2) $instanceB contains the instance of CIM_ProtocolService which is referenced by
   CIM_ServiceAffectsElement.
```

6.17.2.4.3.2   Pseudo Code

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

1690
ANNEX A
1691
(informative)

1692

1693

1694
Change Log

Version	Date	Author	Description
1.0.0	7/29/09		DMTF Standard Release

1695