



1
2
3
4

Document Number: DSP0828

Date: 2009-07-14

Version: 1.0.0

5 **Text Console Redirection Profile SM CLP**
6 **Mapping Specification**

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

10

11 Copyright Notice

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

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

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

33

34

CONTENTS

35 Foreword 5

36 Introduction 6

37 1 Scope 7

38 2 Normative References..... 7

39 2.1 Approved References 7

40 2.2 Other References..... 7

41 3 Terms and Definitions..... 7

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

43 5 Recipes..... 9

44 6 Mappings..... 9

45 6.1 CIM_BindsTo 9

46 6.2 CIM_ElementCapabilities 12

47 6.3 CIM_EnabledLogicalElementCapabilities..... 14

48 6.4 CIM_HostedService 16

49 6.5 CIM_HostedAccessPoint 19

50 6.6 CIM_SAPAvailableForElement..... 21

51 6.7 CIM_ServiceAccessBySAP 23

52 6.8 CIM_ServiceAffectsElement 26

53 6.9 CIM_TextRedirectionSAP 29

54 6.10 CIM_TextRedirectionService 33

55 ANNEX A (informative) Change Log 38

56

57 Tables

58 Table 1 – Command Verb Requirements for CIM_BindsTo 10

59 Table 2 – Command Verb Requirements for CIM_ElementCapabilities 12

60 Table 3 – Command Verb Requirements for CIM_EnabledLogicalElementCapabilities..... 14

61 Table 4 – Command Verb Requirements for CIM_HostedService 16

62 Table 5 – Command Verb Requirements for CIM_HostedAccessPoint 19

63 Table 6 – Command Verb Requirements for CIM_SAPAvailableForElement..... 21

64 Table 7 – Command Verb Requirements for CIM_ServiceAccessBySAP 24

65 Table 8 – Command Verb Requirements for CIM_ServiceAffectsElement 26

66 Table 9 – Command Verb Requirements for CIM_TextRedirectionSAP 29

67 Table 10 – Command Verb Requirements for CIM_TextRedirectionService 34

68

70

Foreword

71 The *Text Console Redirection Profile SM CLP Mapping Specification* (DSP0828) was prepared by the
72 Server Management Working Group.

73 **Conventions**

74 The pseudo code conventions utilized in this document are the Recipe Conventions as defined in SNIA
75 [SMI-S V1.1.0](#), section 7.6.

76 **Acknowledgements**

77 The authors wish to acknowledge the following participants from the DMTF Server Management Working
78 Group:

- 79 • John Leung – Intel
- 80 • Aaron Merkin – IBM
- 81 • Christina Shaw – HP
- 82 • Enoch Suen – Dell
- 83 • Jon Hass – Dell
- 84 • Jeff Hilland – HP
- 85 • Khachatur Papanyan – Dell
- 86 • Arvind Kumar – Intel
- 87 • Perry Vincent – Intel

88

89

Introduction

90 This document defines the SM CLP mapping for CIM elements described in the [Text Console Redirection](#)
91 [Profile](#). The information in this specification, combined with the *SM CLP-to-CIM Common Mapping*
92 *Specification 1.0* ([DSP0216](#)), is intended to be sufficient to implement SM CLP commands relevant to the
93 classes, properties, and methods described in the [Text Console Redirection Profile](#) using CIM operations.

94 The target audience for this specification is implementers of the SM CLP support for the [Text Console](#)
95 [Redirection Profile](#).

96
97

Text Console Redirection Profile SM CLP Mapping Specification

1 Scope

99 This specification contains the requirements for an implementation of the SM CLP to provide access to,
100 and implement the behaviors of, the [Text Console Redirection Profile](#).

2 Normative References

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

2.1 Approved References

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

108 DMTF DSP1024, *Text Console Redirection Profile 1.0*,
109 http://www.dmtf.org/standards/published_documents/DSP1024_1.0.pdf

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

2.2 Other References

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

3 Terms and Definitions

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

3.1

can

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

3.2

cannot

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

3.3

conditional

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

- 127 **3.4**
128 **mandatory**
129 indicates requirements to be followed strictly in order to conform to the document and from which no
130 deviation is permitted
- 131 **3.5**
132 **may**
133 indicates a course of action permissible within the limits of the document
- 134 **3.6**
135 **need not**
136 indicates a course of action permissible within the limits of the document
- 137 **3.7**
138 **optional**
139 indicates a course of action permissible within the limits of the document
- 140 **3.8**
141 **shall**
142 indicates requirements to be followed strictly in order to conform to the document and from which no
143 deviation is permitted
- 144 **3.9**
145 **shall not**
146 indicates requirements to be followed strictly in order to conform to the document and from which no
147 deviation is permitted
- 148 **3.10**
149 **should**
150 indicates that among several possibilities, one is recommended as particularly suitable, without
151 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 152 **3.11**
153 **should not**
154 indicates that a certain possibility or course of action is deprecated but not prohibited

155 **4 Symbols and Abbreviated Terms**

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

- 157 **4.1**
158 **CIM**
159 Common Information Model
- 160 **4.2**
161 **CLP**
162 Command Line Protocol
- 163 **4.3**
164 **DMTF**
165 Distributed Management Task Force

166 **4.4**
167 **IETF**
168 Internet Engineering Task Force

169 **4.5**
170 **SM**
171 Server Management

172 **4.6**
173 **SMI**
174 Storage Management Initiative

175 **4.7**
176 **SNIA**
177 Storage Networking Industry Association

178 **5 Recipes**

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

- 181 • smOpDeleteInstance()
- 182 • smShowInstance()
- 183 • smShowInstances()
- 184 • smSetInstance()
- 185 • smShowAssociationInstances()

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

187 **6 Mappings**

188 The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in
189 the [Text Console Redirection Profile](#). Requirements specified here related to support for a CLP verb for a
190 particular class are solely within the context of this profile.

191 **6.1 CIM_BindsTo**

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

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

198

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	

199 No mappings are defined for the following verbs for the specified target: create, delete, dump, exit,
200 load, reset, set, start, and stop.

201 6.1.1 Ordering of Results

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

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

205 6.1.2 Show

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

208 6.1.2.1 Show a Single Instance – CIM_TextRedirectionSAP Reference

209 This command form is used when the show verb applies to a single instance. This command form
210 corresponds to a show command issued against instances of CIM_BindsTo where only one reference is
211 specified and the reference is to the instance of CIM_TextRedirectionSAP.

212 The [Text Console Redirection Profile](#) imposes a cardinality of 1 on CIM_BindsTo.Antecedent, which
213 references an instance of CIM_ProtocolEndpoint. Therefore, for a given instance of a concrete subclass
214 of CIM_TextRedirectionSAP, a single instance of CIM_BindsTo is found.

215 6.1.2.1.1 Command Form

```
216 show <CIM_BindsTo single objects>
```

217 6.1.2.1.2 CIM Requirements

218 See CIM_BindsTo in the “CIM Elements” section of the [Text Console Redirection Profile](#) for the list of
219 mandatory properties.

220 6.1.2.1.3 Behavior Requirements

221 6.1.2.1.3.1 Preconditions

222 \$instance represents the instance of CIM_TextRedirectionSAP which is referenced by CIM_BindsTo.

223 6.1.2.1.3.2 Pseudo Code

```
224 &smOpReferences ( $instance.getObjectPath(), "CIM_BindsTo", "Dependent", NULL, NULL,
225   $reference[] );
226 &smShowInstance ( $reference->[0]), NULL );
227 &smEnd;
```

228 6.1.2.2 Show a Single Instance – CIM_ProtocolEndpoint Reference

229 This command form is used when the `show` verb applies to instances of `CIM_BindsTo` where only one
230 reference is specified and the reference is to the instance of a concrete subclass of
231 `CIM_ProtocolEndpoint`.

232 The [Text Console Redirection Profile](#) imposes a cardinality of 1 on `CIM_BindsTo.Dependent`, which
233 references an instance of `CIM_TextRedirectionSAP`. Therefore, for a given instance of
234 `CIM_ProtocolEndpoint`, a single instance of `CIM_BindsTo` is found.

235 6.1.2.2.1 Command Form

```
236 show <CIM_BindsTo single object>
```

237 6.1.2.2.2 CIM Requirements

238 See `CIM_BindsTo` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for the list of
239 mandatory properties.

240 6.1.2.2.3 Behavior Requirements

241 6.1.2.2.3.1 Preconditions

242 `$instance` represents the instance of `CIM_ProtocolEndpoint` which is referenced by `CIM_BindsTo`.

243 6.1.2.2.3.2 Pseudo Code

```
244 &smOpReferences ( $instance.getObjectPath(), "CIM_BindsTo", "Antecedent", NULL, NULL,
245   $reference[] );
246 &smShowInstance ( $reference->[0]), NULL );
247 &smEnd;
```

248 6.1.2.3 Show a Single Instance – Both References

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

252 6.1.2.3.1 Command Form

```
253 show <CIM_BindsTo single object>
```

254 6.1.2.3.2 CIM Requirements

255 See `CIM_BindsTo` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for the list of
256 mandatory properties.

257 **6.1.2.3.3 Behavior Requirements**258 **6.1.2.3.3.1 Preconditions**

259 \$instanceA represents the instance of CIM_TextRedirectionSAP which is referenced by CIM_BindsTo.

260 \$instanceB represents the instance of CIM_ProtocolEndpoint which is referenced by CIM_BindsTo.

261 **6.1.2.3.3.2 Pseudo Code**

```
262 &smShowAssociationInstance ( "CIM_BindsTo", $instanceA.getObjectPath(),
263     $instanceB.getObjectPath() );
264 &smEnd;
```

265 **6.2 CIM_ElementCapabilities**266 The `cd`, `help`, `version`, and `exit` verbs shall be supported as described in [DSP0216](#).

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

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

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	Not supported	
Show	Shall	See 6.2.2.
Start	Not supported	
Stop	Not supported	

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

275 **6.2.1 Ordering of Results**

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

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

279 **6.2.2 Show**

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

283 **6.2.2.1 Show a Single Instance Target – CIM_TextRedirectionService Reference**

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

287 The [Text Console Redirection Profile](#) imposes a cardinality of 0..1 on the
288 CIM_ElementCapabilities.Capabilities property, which references an instance of
289 CIM_EnabledLogicalElementCapabilities. Therefore, for a given instance of CIM_TextRedirectionService,
290 at most a single instance of CIM_ElementCapabilities is found.

291 **6.2.2.1.1 Command Form**

```
292 show <CIM_ElementCapabilities multiple instances>
```

293 **6.2.2.1.2 CIM Requirements**

294 See CIM_ElementCapabilities in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
295 the list of mandatory properties.

296 **6.2.2.1.3 Behavior Requirements**

297 **6.2.2.1.3.1 Preconditions**

298 `$instance` represents the instance of a CIM_TextRedirectionService, which is referenced by
299 CIM_ElementCapabilities.

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

301 **6.2.2.1.3.2 Pseudo Code**

```
302 &smOpReferences ( $instance.getObjectPath(), "CIM_ElementCapabilities",  
303     "ManagedElement", NULL, NULL, $references[] );  
304 &smShowInstance ( $references->[0]), NULL );  
305 &smEnd;
```

306 **6.2.2.2 Show Multiple Instances – CIM_EnabledLogicalElementCapabilities Reference**

307 This command form is used to show a single instance of CIM_ElementCapabilities. This command form
308 corresponds to a `show` command issued against a single instance of CIM_ElementCapabilities where
309 only one reference is specified and the reference is to the instance of
310 CIM_EnabledLogicalElementCapabilities.

311 **6.2.2.2.1 Command Form**

```
312 show <CIM_ElementCapabilities multiple instances>
```

313 **6.2.2.2.2 CIM Requirements**

314 See CIM_ElementCapabilities in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
315 the list of mandatory properties.

316 **6.2.2.2.3 Behavior Requirements**

317 **6.2.2.2.3.1 Preconditions**

318 `$instance` represents the instance of a CIM_EnabledLogicalElementCapabilities, which is referenced by
319 CIM_ElementCapabilities.

320 **6.2.2.2.3.2 Pseudo Code**

```
321 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath(),
322     NULL );
323 &smEnd;
```

324 **6.2.2.3 Show a Single Instance Target – Both References**

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

328 **6.2.2.3.1 Command Form**

```
329 show <CIM_ElementCapabilities single instance>
```

330 **6.2.2.3.2 CIM Requirements**

331 See `CIM_ElementCapabilities` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
 332 the list of mandatory properties.

333 **6.2.2.3.3 Behavior Requirements**334 **6.2.2.3.3.1 Preconditions**

335 `$instanceA` represents the instance of a `CIM_TextRedirectionService` which is referenced by the
 336 `CIM_ElementCapabilities`.

337 `$instanceB` represents the instance of a `CIM_EnableLogicalElementCapabilities` which is referenced by
 338 the `CIM_ElementCapabilities`.

339 **6.2.2.3.3.2 Pseudo Code**

```
340 &smShowAssociationInstance ( "CIM_ElementCapabilities", $instanceA.getObjectPath(),
341     $instanceB.getObjectPath(), NULL );
342 &smEnd;
```

343 **6.3 CIM_EnabledLogicalElementCapabilities**

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

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

350 **Table 3 – Command Verb Requirements for CIM_EnabledLogicalElementCapabilities**

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

Command Verb	Requirement	Comments
set	Not supported	
show	Shall	See 6.3.2.
start	Not supported	
stop	Not supported	

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

353 6.3.1 Ordering of Results

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

- 356 • Results for CIM_EnabledLogicalElementCapabilities are unordered; therefore, no algorithm is
357 defined.

358 6.3.2 Show

359 The `show` verb is used to display information about instances of
360 CIM_EnabledLogicalElementCapabilities. Implementations shall support the use of the `show` verb with
361 CIM_EnabledLogicalElementCapabilities.

362 6.3.2.1 Show a Single Instance

363 This command form is used to display the information about a single instance of
364 CIM_EnabledLogicalElementCapabilities.

365 6.3.2.1.1 Command Form

```
366 show <CIM_EnabledLogicalElementCapabilities single object>
```

367 6.3.2.1.2 CIM Requirements

368 See CIM_EnabledLogicalElementCapabilities in the “CIM Elements” section of the [Text Console](#)
369 [Redirection Profile](#) for the list of mandatory properties.

370 6.3.2.1.3 Behavior Requirements

371 6.3.2.1.3.1 Preconditions

372 `$instance` represents the instance of CIM_EnabledLogicalElementCapabilities.

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

374 `#propertylist[]` is an array of mandatory non-key property names.

375 6.3.2.1.3.2 Pseudo Code

```
376 if (false != #all) { #propertylist[] = NULL; }
377 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
378 &smEnd;
```

379 6.3.2.2 Show Multiple Instances

380 This command form is used to display the information about multiple instances of
381 CIM_EnabledLogicalElementCapabilities. This command form corresponds to UFT-based selection
382 within a scoping system.

383 6.3.2.2.1 Command Form

```
384 show <CIM_EnabledLogicalElementCapabilities multiple objects>
```

385 6.3.2.2.2 CIM Requirements

386 See CIM_EnabledLogicalElementCapabilities in the “CIM Elements” section of the [Text Console](#)
387 [Redirection Profile](#) for the list of mandatory properties.

388 6.3.2.2.3 Behavior Requirements

389 6.3.2.2.3.1 Preconditions

390 \$containerInstance represents the instance of CIM_ConcreteCollection with ElementName property
391 that contains “Capabilities” and is associated to the targeted instances of
392 CIM_EnabledLogicalElementCapabilities through the CIM_MemberOfCollection association.

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

394 #propertylist[] is an array of mandatory non-key property names.

395 6.3.2.2.3.2 Pseudo Code

```
396 if (false != #all) { #propertylist[] = NULL; }
397 &smShowInstances ( "CIM_EnabledLogicalElementCapabilities", "CIM_ConcreteCollection",
398     $containerInstance.getObjectPath(), #propertylist[] );
399 &smEnd;
```

400 6.4 CIM_HostedService

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

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

407 **Table 4 – 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	

Command Verb	Requirement	Comments
Set	Not supported	
Show	Shall	See 6.4.2.
Start	Not supported	
Stop	Not supported	

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

410 **6.4.1 Ordering of Results**

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

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

414 **6.4.2 Show**

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

417 **6.4.2.1 Show Multiple Instances – CIM_ComputerSystem Reference**

418 This command form applies the `show` verb to multiple instances. This command form corresponds to a
 419 `show` command issued against instances of `CIM_HostedService` where only one reference is specified
 420 and the reference is to an instance of `CIM_ComputerSystem`.

421 **6.4.2.1.1 Command Form**

```
422 show <CIM_HostedService multiple objects>
```

423 **6.4.2.1.2 CIM Requirements**

424 See `CIM_HostedService` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for the list
 425 of mandatory properties.

426 **6.4.2.1.3 Behavior Requirements**

427 **6.4.2.1.3.1 Preconditions**

428 `$instance` represents the instance of `CIM_ComputerSystem`, which is referenced by
 429 `CIM_HostedService`.

430 **6.4.2.1.3.2 Pseudo Code**

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

433 **6.4.2.2 Show a Single Instance – CIM_TextRedirectionService Reference**

434 This command form is used when the `show` verb applies to instances of `CIM_HostedService` where only
 435 one reference is specified and the reference is to an instance of `CIM_TextRedirectionService`.

436 The [Text Console Redirection Profile](#) imposes a cardinality of 1 on CIM_HostedService.Dependent, which
437 references an instance of CIM_ComputerSystem. Therefore, for a given instance of
438 CIM_TextRedirectionService, a single instance of CIM_HostedService is found.

439 6.4.2.2.1 Command Form

```
440 show <CIM_HostedService single object>
```

441 6.4.2.2.2 CIM Requirements

442 See CIM_HostedService in the “CIM Elements” section of the [Text Console Redirection Profile](#) for the list
443 of mandatory properties.

444 6.4.2.2.3 Behavior Requirements

445 6.4.2.2.3.1 Preconditions

446 \$instance represents the instance of CIM_TextRedirectionService, which is referenced by
447 CIM_HostedService.

448 6.4.2.2.3.2 Pseudo Code

```
449 &smOpReferences ( $instance.getObjectPath(), "CIM_HostedService", "Antecedent", NULL,  
450 NULL, $references[] );  
451 &smShowInstance ( $references->[0]), NULL );  
452 &smEnd;
```

453 6.4.2.3 Show a Single Instance – Both References

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

457 6.4.2.3.1 Command Form

```
458 show <CIM_HostedService single object>
```

459 6.4.2.3.2 CIM Requirements

460 See CIM_HostedService in the “CIM Elements” section of the [Text Console Redirection Profile](#) for the list
461 of mandatory properties.

462 6.4.2.3.3 Behavior Requirements

463 6.4.2.3.3.1 Preconditions

464 \$instanceA represents the referenced instance of CIM_ComputerSystem through CIM_HostedService
465 association.

466 \$instanceB represents the other instance of CIM_TextRedirectionService which is referenced by
467 CIM_HostedService.

468 6.4.2.3.3.2 Pseudo Code

```
469 &smShowAssociationInstance ("CIM_HostedService", $instanceA.getObjectPath(),  
470 $instanceB.getObjectPath() );  
471 &smEnd;
```

472 6.5 CIM_HostedAccessPoint

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

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

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

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	Not supported	
Show	Shall	See 6.5.2.
Start	Not supported	
Stop	Not supported	

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

482 6.5.1 Ordering of Results

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

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

486 6.5.2 Show

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

490 6.5.2.1 Show Multiple Instances – CIM_ComputerSystem Reference

491 This command form applies the `show` verb to multiple instances. This command form corresponds to a
 492 `show` command issued against instances of `CIM_HostedAccessPoint` where only one reference is
 493 specified and the reference is to an instance of `CIM_ComputerSystem`.

494 6.5.2.1.1 Command Form

```
495 show <CIM_HostedAccessPoint multiple objects>
```

496 6.5.2.1.2 CIM Requirements

497 See CIM_HostedAccessPoint in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
498 the list of mandatory properties.

499 6.5.2.1.3 Behavior Requirements

500 6.5.2.1.3.1 Preconditions

501 \$instance represents the instance of CIM_ComputerSystem, which is referenced by
502 CIM_HostedAccessPoint.

503 6.5.2.1.3.2 Pseudo Code

```
504 &smShowAssociationInstances ( "CIM_HostedAccessPoint", $instance.getObjectPath() );  
505 &smEnd;
```

506 6.5.2.2 Show a Single Instance – CIM_TextRedirectionSAP Reference

507 This command form is used when the `show` verb applies to instances of CIM_HostedAccessPoint where
508 only one reference is specified and the reference is to an instance of CIM_TextRedirectionSAP.

509 The [Text Console Redirection Profile](#) imposes a cardinality of 1 on CIM_HostedAccessPoint.Dependent,
510 which references an instance of CIM_ComputerSystem. Therefore; for a given instance of
511 CIM_TextRedirectionSAP, a single instance of CIM_HostedAccessPoint is found.

512 6.5.2.2.1 Command Form

```
513 show <CIM_HostedAccessPoint single object>
```

514 6.5.2.2.2 CIM Requirements

515 See CIM_HostedAccessPoint in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
516 the list of mandatory properties.

517 6.5.2.2.3 Behavior Requirements

518 6.5.2.2.3.1 Preconditions

519 \$instance represents the instance of CIM_TextRedirectionSAP, which is referenced by
520 CIM_HostedAccessPoint.

521 6.5.2.2.3.2 Pseudo Code

```
522 &smOpReferences ( $instance.getObjectPath(), "CIM_HostedAccessPoint", "Antecedent",  
523 NULL, NULL, $references[] );  
524 &smShowInstance ( $references->[0]), NULL );  
525 &smEnd;
```

526 6.5.2.3 Show a Single Instance – Both References

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

530 **6.5.2.3.1 Command Form**

531 `show <CIM_HostedAccessPoint single object>`

532 **6.5.2.3.2 CIM Requirements**

533 See CIM_HostedAccessPoint in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
 534 the list of mandatory properties.

535 **6.5.2.3.3 Behavior Requirements**

536 **6.5.2.3.3.1 Preconditions**

537 \$instanceA represents the referenced instance of CIM_ComputerSystem which is referenced by
 538 CIM_HostedAccessPoint.

539 \$instanceB represents the other instance of CIM_TextRedirectionSAP which is referenced by
 540 CIM_HostedAccessPoint.

541 **6.5.2.3.3.2 Pseudo Code**

```
542 &smShowAssociationInstance ( "CIM_HostedAccessPoint", $instanceA.getObjectPath(),
543     $instanceB.getObjectPath() );
544 &smEnd;
```

545 **6.6 CIM_SAPAvailableForElement**

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

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

552 **Table 6 – Command Verb Requirements for CIM_SAPAvailableForElement**

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	

553 No mappings are defined for the following verbs for the specified target: create, delete, dump, exit,
 554 load, reset, set, start, and stop.

555 6.6.1 Ordering of Results

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

- 558 • Results for CIM_SAPAvailableForElement are unordered; therefore, no algorithm is defined.

559 6.6.2 Show

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

563 6.6.2.1 Show Multiple Instances – CIM_ComputerSystem Reference

564 This command form applies the `show` verb to multiple instances. This command form corresponds to a
565 `show` command issued against instances of CIM_SAPAvailableForElement where only one reference is
566 specified and the reference is to an instance of CIM_ComputerSystem.

567 6.6.2.1.1 Command Form

```
568 show <CIM_SAPAvailableForElement multiple objects>
```

569 6.6.2.1.2 CIM Requirements

570 See CIM_SAPAvailableForElement in the “CIM Elements” section of the [Text Console Redirection Profile](#)
571 for the list of mandatory properties.

572 6.6.2.1.3 Behavior Requirements

573 6.6.2.1.3.1 Preconditions

574 `$instance` represents the instance of CIM_ComputerSystem, which is referenced by
575 CIM_SAPAvailableForElement.

576 6.6.2.1.3.2 Pseudo Code

```
577 &smShowAssociationInstances ( "CIM_SAPAvailableForElement",  
578     $instance.getObjectPath() );  
579 &smEnd;
```

580 6.6.2.2 Show a Single Instance – CIM_TextRedirectionSAP Reference

581 This command form is used when the `show` verb applies to instances of CIM_SAPAvailableForElement
582 where only one reference is specified and the reference is to an instance of CIM_TextRedirectionSAP.

583 The [Text Console Redirection Profile](#) imposes a cardinality of 1 on
584 CIM_SAPAvailableForElement.Dependent, which references an instance of CIM_ComputerSystem.
585 Therefore, for a given instance of CIM_TextRedirectionSAP, a single instance of
586 CIM_SAPAvailableForElement is found.

587 6.6.2.2.1 Command Form

```
588 show <CIM_SAPAvailableForElement single object>
```

589 6.6.2.2.2 CIM Requirements

590 See CIM_SAPAvailableForElement in the “CIM Elements” section of the [Text Console Redirection Profile](#)
591 for the list of mandatory properties.

592 6.6.2.2.3 Behavior Requirements

593 6.6.2.2.3.1 Preconditions

594 \$instance represents the instance of CIM_TextRedirectionSAP, which is referenced by
595 CIM_SAPAvailableForElement.

596 6.6.2.2.3.2 Pseudo Code

```
597 &smOpReferences ( $instance.getObjectPath(), "CIM_SAPAvailableForElement",
598     "Antecedent", NULL, NULL, $references[] );
599 &smShowInstance ( $references->[0]), NULL );
600 &smEnd;
```

601 6.6.2.3 Show a Single Instance – Both References

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

605 6.6.2.3.1 Command Form

```
606 show <CIM_SAPAvailableForElement single object>
```

607 6.6.2.3.2 CIM Requirements

608 See `CIM_SAPAvailableForElement` in the “CIM Elements” section of the [Text Console Redirection Profile](#)
609 for the list of mandatory properties.

610 6.6.2.3.3 Behavior Requirements

611 6.6.2.3.3.1 Preconditions

612 \$instanceA represents the referenced instance of `CIM_ComputerSystem` which is referenced by
613 `CIM_SAPAvailableForElement`.

614 \$instanceB represents the other instance of `CIM_TextRedirectionSAP` which is referenced by
615 `CIM_SAPAvailableForElement`.

616 6.6.2.3.3.2 Pseudo Code

```
617 &smShowAssociationInstance ( "CIM_SAPAvailableForElement", $instanceA.getObjectPath(),
618     $instanceB.getObjectPath() );
619 &smEnd;
```

620 6.7 CIM_ServiceAccessBySAP

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

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

627

Table 7 – Command Verb Requirements for CIM_ServiceAccessBySAP

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 6.7.2.
Start	Not supported	
Stop	Not supported	

628 No mappings are defined for the following verbs for the specified target: create, delete, dump, exit,
629 load, reset, set, start, and stop.

630 6.7.1 Ordering of Results

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

- 633 • Results for CIM_ServiceAccessBySAP are unordered; therefore, no algorithm is defined.

634 6.7.2 Show

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

638 6.7.2.1 Show Multiple Instances – CIM_TextRedirectionService Reference

639 This command form applies the show verb to multiple instances. This command form corresponds to a
640 show command issued against instances of CIM_ServiceAccessBySAP where only one reference is
641 specified and the reference is to an instance of CIM_TextRedirectionService.

642 6.7.2.1.1 Command Form

```
643 show <CIM_ServiceAccessBySAP multiple objects>
```

644 6.7.2.1.2 CIM Requirements

645 See CIM_ServiceAccessBySAP in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
646 the list of mandatory properties.

647 6.7.2.1.3 Behavior Requirements

648 6.7.2.1.3.1 Preconditions

649 \$instance represents the instance of CIM_TextRedirectionService, which is referenced by
650 CIM_ServiceAccessBySAP.

651 **6.7.2.1.3.2 Pseudo Code**

```
652 &smShowAssociationInstances ( "CIM_ServiceAccessBySAP", $instance.getObjectPath() );
653 &smEnd;
```

654 **6.7.2.2 Show a Single Instance – CIM_TextRedirectionSAP Reference**

655 This command form is used when the `show` verb applies to instances of `CIM_ServiceAccessBySAP`
 656 where only one reference is specified and the reference is to an instance of `CIM_TextRedirectionSAP`.

657 The [Text Console Redirection Profile](#) imposes a cardinality of 1 on the
 658 `CIM_ServiceAccessBySAP.Antecedent` property, which references an instance of
 659 `CIM_TextRedirectionService`. Therefore, for a given instance of `CIM_TextRedirectionSAP`, a single
 660 instance of `CIM_ServiceAccessBySAP` is found.

661 **6.7.2.2.1 Command Form**

```
662 show <CIM_ServiceAccessBySAP single object>
```

663 **6.7.2.2.2 CIM Requirements**

664 See `CIM_ServiceAccessBySAP` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
 665 the list of mandatory properties.

666 **6.7.2.2.3 Behavior Requirements**

667 **6.7.2.2.3.1 Preconditions**

668 `$instance` represents the instance of `CIM_TextRedirectionSAP`, which is referenced by
 669 `CIM_ServiceAccessBySAP`.

670 **6.7.2.2.3.2 Pseudo Code**

```
671 &smOpReferences ( $instance.getObjectPath(), "CIM_ServiceAccessBySAP", "Dependent",
672 NULL, NULL, $references[] );
673 &smShowInstance ( $references->[0]), NULL );
674 &smEnd;
```

675 **6.7.2.3 Show a Single Instance – Both References**

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

679 **6.7.2.3.1 Command Form**

```
680 show <CIM_ServiceAccessBySAP single object>
```

681 **6.7.2.3.2 CIM Requirements**

682 See `CIM_ServiceAccessBySAP` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
 683 the list of mandatory properties.

684 **6.7.2.3.3 Behavior Requirements**

685 **6.7.2.3.3.1 Preconditions**

686 `$instanceA` represents the referenced instance of `CIM_TextRedirectionService` which is referenced by
 687 `CIM_ServiceAccessBySAP`.

688 \$instanceB represents the other instance of CIM_TextRedirectionSAP which is referenced by
689 CIM_ServiceAccessBySAP.

690 6.7.2.3.2 Pseudo Code

```
691 &smShowAssociationInstance ( "CIM_SAPAvailableForElement", $instanceA.getObjectPath(),
692     $instanceB.getObjectPath() );
693 &smEnd;
```

694 6.8 CIM_ServiceAffectsElement

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

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

701 **Table 8 – 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 6.8.2.
Start	Not supported	
Stop	Not supported	

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

704 6.8.1 Ordering of Results

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

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

708 6.8.2 Show

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

712 **6.8.2.1 Show Multiple Instances – CIM_ComputerSystem Reference**

713 This command form is used when the `show` verb applies to multiple instances. This command form
714 corresponds to a `show` command issued against instances of `CIM_ServiceAffectsElement` where only
715 one reference is specified and the reference is to an instance of `CIM_ComputerSystem`.

716 **6.8.2.1.1 Command Form**

```
717 show <CIM_ServiceAffectsElement multiple instances>
```

718 **6.8.2.1.2 CIM Requirements**

719 See `CIM_ServiceAffectsElement` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
720 the list of mandatory properties.

721 **6.8.2.1.3 Behavior Requirements**

722 **6.8.2.1.3.1 Preconditions**

723 `$instance` represents the instance of `CIM_ComputerSystem`, which is referenced by
724 `CIM_ServiceAffectsElement`.

725 **6.8.2.1.3.2 Pseudo Code**

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

729 **6.8.2.2 Show a Single Instance – CIM_LogicalDevice Reference**

730 This command form is used when the `show` verb applies to multiple instances. This command form
731 corresponds to a `show` command issued against instances of `CIM_ServiceAffectsElement` where only
732 one reference is specified and the reference is to an instance of a concrete subclass of
733 `CIM_LogicalDevice`.

734 The [Text Console Redirection Profile](#) imposes a cardinality of 1 on
735 `CIM_ServiceAffectsElement.Antecedent`, which references an instance of
736 `CIM_TextConsoleRedirectionService`. Therefore, for a given instance of a concrete subclass of
737 `CIM_LogicalDevice`, a single instance of `CIM_TextConsoleRedirectionService` is found.

738 **6.8.2.2.1 Command Form**

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

740 **6.8.2.2.2 CIM Requirements**

741 See `CIM_ServiceAffectsElement` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
742 the list of mandatory properties.

743 **6.8.2.2.3 Behavior Requirements**

744 **6.8.2.2.3.1 Preconditions**

745 `$instance` represents the instance of a concrete subclass of `CIM_LogicalDevice`, which is referenced
746 by `CIM_ServiceAffectsElement`.

747 6.8.2.2.3.2 Pseudo Code

```
748 &smShowAssociationInstance ( "CIM_ServiceAffectsElement", $instance.getObjectPath() );  
749 &smEnd;
```

750 6.8.2.3 Show Multiple Instances – CIM_TextRedirectionService Reference

751 This command form is used when the `show` verb applies to multiple instances. This command form
752 corresponds to a `show` command issued against instances of `CIM_ServiceAffectsElement` where only
753 one reference is specified and the reference is to an instance of `CIM_TextRedirectionService`.

754 6.8.2.3.1 Command Form

```
755 show <CIM_ServiceAffectsElement multiple instances>
```

756 6.8.2.3.2 CIM Requirements

757 See `CIM_ServiceAffectsElement` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
758 the list of mandatory properties.

759 6.8.2.3.3 Behavior Requirements**760 6.8.2.3.3.1 Preconditions**

761 `$instance` represents the instance of `CIM_TextRedirectionService`, which is referenced by
762 `CIM_ServiceAffectsElement`.

763 6.8.2.3.3.2 Pseudo Code

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

767 6.8.2.4 Show a Single Instance – Both References

768 This command form is used when the `show` verb applies to a single instance. This command form
769 corresponds to the `show` verb issued against an instance of `CIM_ServiceAffectsElement` where both
770 references are specified and therefore the desired instance is unambiguously identified.

771 6.8.2.4.1 Command Form

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

773 6.8.2.4.2 CIM Requirements

774 See `CIM_ServiceAffectsElement` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
775 the list of mandatory properties.

776 6.8.2.4.3 Behavior Requirements**777 6.8.2.4.3.1 Preconditions**

778 `$instanceA` represents the instance of `CIM_TextRedirectionService`, which is referenced by
779 `CIM_ServiceAffectsElement`.

780 `$instanceB` represents the instance of `CIM_ComputerSystem` or a concrete subclass of
781 `CIM_LogicalDevice` which is referenced by `CIM_ServiceAffectsElement`.

782 **6.8.2.4.3.2 Pseudo Code**

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

786 **6.9 CIM_TextRedirectionSAP**

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

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

793 **Table 9 – Command Verb Requirements for CIM_TextRedirectionSAP**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.9.2.
show	Shall	See 6.9.3.
start	May	See 6.9.4.
stop	May	See 6.9.5.

794 No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `exit`,
 795 `load`, and `reset`.

796 **6.9.1 Ordering of Results**

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

- 799 • Results for `CIM_TextRedirectionSAP` are unordered; therefore, no algorithm is defined.

800 **6.9.2 Set**

801 The `set` verb is used to set properties on an instance of `CIM_TextRedirectionSAP`. Implementations may
 802 support the use of the `set` verb with `CIM_TextRedirectionSAP`.

803 **6.9.2.1 General Usage of Set for a Single Property**

804 This command form corresponds to the general usage of the `set` verb to modify a single property of a
 805 target instance. The setting of a single property shall be deterministic.

806 The requirements for supporting modification of a property using this command form shall be equivalent
 807 to the requirement for supporting modification of the property using the `ModifyInstance` operation as
 808 defined in the [Text Console Redirection Profile](#).

809 **6.9.2.1.1 Command Form**810

```
set <CIM_TextRedirectionSAP single object> <propertyname>=<propertyvalue>
```

811 **6.9.2.1.2 CIM Requirements**812 See CIM_TextRedirectionSAP in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
813 the list of mandatory properties.814 **6.9.2.1.3 Behavior Requirements**815

```
$instance=<CIM_TextRedirectionSAP single object>  
816 #propertyNames[] = <propertyname>  
817 #propertyValues[] = <propertyvalue>  
818 &smSetInstance ( $instance, #propertyNames, #propertyValues );  
819 &smEnd;
```

820 **6.9.2.2 General Usage of Set for Multiple Properties**821 This command form corresponds to the general usage of the `set` verb to modify multiple properties of a
822 target instance. The setting of multiple properties may be deterministic.823 The requirements for supporting modification of a property using this command form shall be equivalent
824 to the requirement for supporting modification of the property using the `ModifyInstance` operation as
825 defined in the [Text Console Redirection Profile](#).826 **6.9.2.2.1 Command Form**827

```
set <CIM_TextRedirectionSAP single object> <propertyname1>=<propertyvalue1>  
828 <propertynameN>=<propertyvalueN>
```

829 **6.9.2.2.2 CIM Requirements**830 See CIM_TextRedirectionSAP in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
831 the list of mandatory properties.832 **6.9.2.2.3 Behavior Requirements**833 **6.9.2.2.3.1 Preconditions**834 `$instance` represents the instance of CIM_TextRedirectionSAP.835 **6.9.2.2.3.2 Pseudo Code**836

```
for #i < n  
837 {  
838 #propertyNames[#i] = <propertyname#i>  
839 #propertyValues[#i] = <propertyvalue#i>  
840 }  
841 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );  
842 &smEnd;
```

843 **6.9.3 Show**844 The `show` verb is used to display information about instances of CIM_TextRedirectionSAP.
845 Implementations shall support the use of the `show` verb with CIM_TextRedirectionSAP.

846 **6.9.3.1 Show a Single Instance**

847 This command form is used to display information about a single instance of CIM_TextRedirectionSAP.

848 **6.9.3.1.1 Command Form**

```
849 show <CIM_TextRedirectionSAP single object>
```

850 **6.9.3.1.2 CIM Requirements**

851 See CIM_TextRedirectionSAP in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
852 the list of mandatory properties.

853 **6.9.3.1.3 Behavior Requirements**

854 **6.9.3.1.3.1 Preconditions**

855 \$instance represents the instance of CIM_TextRedirectionSAP.

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

857 #propertylist[] is an array of mandatory non-key property names.

858 **6.9.3.1.3.2 Pseudo Code**

```
859 if (false != #all) { #propertylist[] = NULL; }
860 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
861 &smEnd;
```

862 **6.9.3.2 Show Multiple Instances – CIM_HostedAccessPoint**

863 This command form is used to display information about multiple instances of CIM_TextRedirectionSAP.

864 **6.9.3.2.1 Command Form**

```
865 show <CIM_TextRedirectionSAP multiple objects>
```

866 **6.9.3.2.2 CIM Requirements**

867 See CIM_TextRedirectionSAP in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
868 the list of mandatory properties.

869 **6.9.3.2.3 Behavior Requirements**

870 **6.9.3.2.3.1 Preconditions**

871 \$containerInstance represents the instance of CIM_ComputerSystem for which the scoped instance
872 of CIM_TextRedirectionSAP is being displayed. The CIM_TextRedirectionSAP is associated to
873 CIM_ComputerSystem via a CIM_HostedAccessPoint association.

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

875 #propertylist[] is an array of mandatory non-key property names.

876 **6.9.3.2.3.2 Pseudo Code**

```
877 if (false != #all) { #propertylist[] = NULL; }
878 &smShowInstances ( "CIM_TextRedirectionSAP", "CIM_HostedAccessPoint",
879   $containerInstance.getObjectPath(), #propertylist[] );
880 &smEnd;
```

881 6.9.3.3 Show Multiple Instances – CIM_SAPAvailableForElement

882 This command form is used to display information about multiple instances of CIM_TextRedirectionSAP.

883 6.9.3.3.1 Command Form

```
884 show <CIM_TextRedirectionSAP multiple objects>
```

885 6.9.3.3.2 CIM Requirements

886 See CIM_TextRedirectionSAP in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
887 the list of mandatory properties.

888 6.9.3.3.3 Behavior Requirements**889 6.9.3.3.3.1 Preconditions**

890 \$containerInstance represents the instance of CIM_ComputerSystem for which the scoped instance
891 of CIM_TextRedirectionSAP is being displayed. The CIM_TextRedirectionSAP is associated to
892 CIM_ComputerSystem via a CIM_SAPAvailableForElement association.

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

894 #propertylist[] is an array of mandatory non-key property names.

895 6.9.3.3.3.2 Pseudo Code

```
896 if (false != #all) { #propertylist[] = NULL; }
897 &smShowInstances ( "CIM_TextRedirectionSAP", "CIM_SAPAvailableForElement",
898     $containerInstance.getObjectPath(), #propertylist[] );
899 &smEnd;
```

900 6.9.4 Start

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

904 The `start` verb is used to enable a Text Console Redirection Session Destination.

905 6.9.4.1 Start a Single Instance

906 This command form is for the `start` verb applied to a single instance of CIM_TextRedirectionSAP, in
907 which the CIM_TextRedirectionSAP does not use a CLP session for the text console redirection session.

908 6.9.4.1.1 Command Form

```
909 start <CIM_TextRedirectionSAP single object>
```

910 6.9.4.1.2 CIM Requirements

911 See CIM_TextRedirectionSAP in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
912 the list of mandatory properties.

913 6.9.4.1.3 Behavior Requirements

```
914 $instance=<CIM_TextRedirectionSAP single object>
915 &smStartRSC ( $instance.getObjectPath() );
916 &smEnd;
```


917 **6.9.4.2 Start a Single In-line Instance**

918 This command form is for the `start` verb applied to a single instance of `CIM_TextRedirectionSAP`, in
 919 which the `CIM_TextRedirectionSAP` uses the CLP session for the text console redirection session.
 920 Termination of this text console redirection session occurs when the character sequence contained in the
 921 `CIM_TextRedirectionSAP.SessionTerminationSequence` property is entered into the text flow. Upon
 922 termination, the CLP session is restored.

923 **6.9.4.2.1 Command Form**

```
924 start <CIM_TextRedirectionSAP single object>
```

925 **6.9.4.2.2 CIM Requirements**

926 See `CIM_TextRedirectionSAP` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
 927 the list of mandatory properties.

928 **6.9.4.2.3 Behavior Requirements**

```
929 $instance=<CIM_TextRedirectionSAP single object>
930 &smStartRSC ( $instance.GetObjectPath() );
931 //The text redirection session will use the CLP session until it is terminated.
932 &smEnd;
```

933 **6.9.5 Stop**

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

937 The `stop` verb is used to place a Text Console Redirection Session Destination into a 6 (Enabled but
 938 Offline) state.

939 **6.9.5.1 Stop a Single Instance**

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

941 **6.9.5.1.1 Command Form**

```
942 stop <CIM_TextRedirectionSAP single object>
```

943 **6.9.5.1.2 CIM Requirements**

944 See `CIM_TextRedirectionSAP` in the “CIM Elements” section of the [Text Console Redirection Profile](#) for
 945 the list of mandatory properties.

946 **6.9.5.1.3 Behavior Requirements**

```
947 $instance=<CIM_TextRedirectionSAP single object>
948 &smRequestStateChange ( $instance.GetObjectPath(), "Enabled but Offline" );
949 &smEnd;
```

950 **6.10 CIM_TextRedirectionService**

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

952 Table 10 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
 953 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and

954 target. Table 10 is for informational purposes only; in case of a conflict between Table 10 and
 955 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 956 information in Table 10.

957 **Table 10 – Command Verb Requirements for CIM_TextRedirectionService**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.10.2.
show	Shall	See 6.10.3.
start	May	See 6.10.4.
stop	May	See 6.10.5.

958 No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `exit`,
 959 `load`, and `reset`.

960 6.10.1 Ordering of Results

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

- 963 • Results for `CIM_TextRedirectionService` are unordered; therefore, no algorithm is defined.

964 6.10.2 Set

965 The `set` verb is used to set properties on an instance of `CIM_TextRedirectionService`. Implementations
 966 may support the use of the `set` verb with `CIM_TextRedirectionService`.

967 6.10.2.1 General Usage of Set for a Single Property

968 This command form corresponds to the general usage of the `set` verb to modify a single property of a
 969 target instance. The setting of a single property shall be deterministic.

970 The requirements for supporting modification of a property using this command form shall be equivalent
 971 to the requirement for supporting modification of the property using the `ModifyInstance` operation as
 972 defined in the [Text Console Redirection Profile](#).

973 6.10.2.1.1 Command Form

```
974 set <CIM_TextRedirectionService single object> <propertyname>=<propertyvalue>
```

975 6.10.2.1.2 CIM Requirements

976 See `CIM_TextRedirectionService` in the “CIM Elements” section of the [Text Console Redirection Profile](#)
 977 for the list of mandatory properties.

978 6.10.2.1.3 Behavior Requirements

```
979 $instance=<CIM_TextRedirectionService single object>  
980 #propertyName[] = <propertyname>
```

```

981 #propertyValues[] = <propertyvalue>
982 &smSetInstance ( $instance, #propertyNames, #propertyValues );
983 &smEnd;

```

984 6.10.2.2 General Usage of Set for Multiple Properties

985 This command form corresponds to the general usage of the `set` verb to modify multiple properties of a
 986 target instance. The setting of multiple properties may be deterministic.

987 The requirements for supporting modification of a property using this command form shall be equivalent
 988 to the requirement for supporting modification of the property using the `ModifyInstance` operation as
 989 defined in the [Text Console Redirection Profile](#).

990 6.10.2.2.1 Command Form

```

991 set <CIM_TextRedirectionService single object> <propertyname1>=<propertyvalue1>
992 <propertynameN>=<propertyvalueN>

```

993 6.10.2.2.2 CIM Requirements

994 See `CIM_TextRedirectionService` in the “CIM Elements” section of the [Text Console Redirection Profile](#)
 995 for the list of mandatory properties.

996 6.10.2.2.3 Behavior Requirements

997 6.10.2.2.3.1 Preconditions

998 `$instance` represents the instance of `CIM_TextRedirectionService`.

999 6.10.2.2.3.2 Pseudo Code

```

1000 for #i < n
1001 {
1002   #propertyNames[#i] = <propertyname#i>
1003   #propertyValues[#i] = <propertyvalue#i>
1004 }
1005 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
1006 &smEnd;

```

1007 6.10.3 Show

1008 The `show` verb is used to display information about instances of `CIM_TextRedirectionService`.
 1009 Implementations shall support the use of the `show` verb with `CIM_TextRedirectionService`.

1010 6.10.3.1 Show a Single Instance

1011 This command form is used to display information about a single instance of
 1012 `CIM_TextRedirectionService`.

1013 6.10.3.1.1 Command Form

```

1014 show <CIM_TextRedirectionService single object>

```

1015 6.10.3.1.2 CIM Requirements

1016 See `CIM_TextRedirectionService` in the “CIM Elements” section of the [Text Console Redirection Profile](#)
 1017 for the list of mandatory properties.

1018 6.10.3.1.3 Behavior Requirements

1019 6.10.3.1.3.1 Preconditions

1020 \$instance represents the instance of CIM_TextRedirectionService.

1021 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1022 #propertylist[] is an array of mandatory non-key property names.

1023 6.10.3.1.3.2 Pseudo Code

```
1024 if (false != #all) { #propertylist[] = NULL; }
1025 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
1026 &smEnd;
```

1027 6.10.3.2 Show Multiple Instances

1028 This command form is used to display information about multiple instances of
1029 CIM_TextRedirectionService.

1030 6.10.3.2.1 Command Form

```
1031 show <CIM_TextRedirectionService multiple objects>
```

1032 6.10.3.2.2 CIM Requirements

1033 See CIM_TextRedirectionService in the “CIM Elements” section of the [Text Console Redirection Profile](#)
1034 for the list of mandatory properties.

1035 6.10.3.2.3 Behavior Requirements

1036 6.10.3.2.3.1 Preconditions

1037 \$containerInstance represents the instance of CIM_ComputerSystem for which the scoped instance
1038 of CIM_TextRedirectionService is being displayed. The CIM_TextRedirectionService is associated to
1039 CIM_ComputerSystem via a CIM_HostedService association.

1040 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1041 #propertylist[] is an array of mandatory non-key property names.

1042 6.10.3.2.3.2 Pseudo Code

```
1043 if (false != #all) { #propertylist[] = NULL; }
1044 &smShowInstances ( "CIM_TextRedirectionService", "CIM_HostedService",
1045     $containerInstance.getObjectPath(), #propertylist[] );
1046 &smEnd;
```

1047 6.10.4 Start

1048 This section describes how to implement the start verb when applied to an instance of
1049 CIM_TextRedirectionService. Implementations may support the use of the start verb with
1050 CIM_TextRedirectionService.

1051 The start verb is used to enable a Text Console Redirection Session Source.

1052 **6.10.4.1 Start a Single Instance**

1053 This command form is for the `start` verb applied to a single instance of `CIM_TextRedirectionService`.

1054 **6.10.4.1.1 Command Form**

```
1055 start <CIM_TextRedirectionService single object>
```

1056 **6.10.4.1.2 CIM Requirements**

1057 See `CIM_TextRedirectionService` in the “CIM Elements” section of the [Text Console Redirection Profile](#)
1058 for the list of mandatory properties.

1059 **6.10.4.1.3 Behavior Requirements**

```
1060 $instance=<CIM_TextRedirectionService single object>  
1061 &smStartRSC ( $instance.GetObjectPath() );  
1062 &smEnd;
```

1063 **6.10.5 Stop**

1064 This section describes how to implement the `stop` verb when applied to an instance of
1065 `CIM_TextRedirectionService`. Implementations may support the use of the `stop` verb with
1066 `CIM_TextRedirectionService`.

1067 The `stop` verb is used to place a Text Console Redirection Session Source in a 6 (Enabled but Offline)
1068 state.

1069 **6.10.5.1 Stop a Single Instance**

1070 This command form is for the `stop` verb applied to a single instance of `CIM_TextRedirectionService`.

1071 **6.10.5.1.1 Command Form**

```
1072 stop <CIM_TextRedirectionService single object>
```

1073 **6.10.5.1.2 CIM Requirements**

1074 See `CIM_TextRedirectionService` in the “CIM Elements” section of the [Text Console Redirection Profile](#)
1075 for the list of mandatory properties.

1076 **6.10.5.1.3 Behavior Requirements:**

```
1077 $instance=<CIM_TextRedirectionService single object>  
1078 &smRequestStateChange ( $instance.GetObjectPath(), "Disabled" );  
1079 &smEnd;
```

1080

ANNEX A
(informative)

Change Log

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

1081
1082
1083
1084
1085

1086