



1
2
3
4

Document Number: DSP0831

Date: 2009-06-04

Version: 1.0.0

5 **Platform Watchdog Profile SM CLP Mapping**
6 **Specification**

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

10

11 Copyright notice

12 Copyright © 2007, 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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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_DeviceServiceImplementation.....	9
46	6.2 CIM_ElementCapabilities	12
47	6.3 CIM_HostedService	14
48	6.4 CIM_ServiceAffectsElement	16
49	6.5 CIM_ServiceAvailableToElement	19
50	6.6 CIM_PlatformWatchdogService.....	21
51	6.7 CIM_PlatformWatchdogServiceCapabilities.....	25
52	ANNEX A (informative) Change Log	28
53		

54 Tables

55	Table 1 – Command Verb Requirements for CIM_DeviceServiceImplementation.....	9
56	Table 2 – Command Verb Requirements for CIM_ElementCapabilities	12
57	Table 3 – Command Verb Requirements for CIM_HostedService	14
58	Table 4 – Command Verb Requirements for CIM_ServiceAffectsElement	16
59	Table 5 – Command Verb Requirements for CIM_ServiceAvailableToElement	19
60	Table 6 – Command Verb Requirements for CIM_PlatformWatchdogService.....	22
61	Table 7 – Command Verb Requirements for CIM_PlatformWatchdogServiceCapabilities.....	25
62		

64

Foreword

65 The *Platform Watchdog Profile SM CLP Mapping Specification* (DSP0831) was prepared by the Server
66 Management Working Group.

67 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
68 management and interoperability.

69 **Conventions**

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

72 **Acknowledgements**

73 The authors wish to acknowledge the following participants from the DMTF Server Management Working
74 Group:

- 75 • John Leung – Intel
- 76 • Aaron Merkin – IBM
- 77 • Christina Shaw – HP
- 78 • Jon Hass – Dell
- 79 • Jeff Hilland – HP
- 80 • Khachatur Papanyan – Dell

81

82

Introduction

83 This document defines the SM CLP mapping for CIM elements described in [Platform Watchdog Profile](#).
84 The information in this specification, combined with *SM CLP-to-CIM Common Mapping Specification 1.0*
85 ([DSP0216](#)), is intended to be sufficient to implement SM CLP commands relevant to the classes,
86 properties, and methods described in [Platform Watchdog Profile](#) using CIM operations.

87 The target audience for this specification is implementers of the SM CLP support for [Platform Watchdog](#)
88 [Profile](#).

89 Platform Watchdog Profile SM CLP Mapping Specification

90 1 Scope

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

93 2 Normative References

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

97 2.1 Approved References

98 DMTF DSP1040, *Platform Watchdog Profile 1.0*,
99 http://www.dmtf.org/standards/published_documents/DSP1040_1.0.pdf

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

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

104 2.2 Other References

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

107 3 Terms and Definitions

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

109 3.1

110 **can**

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

112 3.2

113 **cannot**

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

115 3.3

116 **conditional**

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

119 3.4

120 **mandatory**

121 indicates requirements to be followed strictly in order to conform to the document and from which no
122 deviation is permitted

- 123 **3.5**
124 **may**
125 indicates a course of action permissible within the limits of the document
- 126 **3.6**
127 **need not**
128 indicates a course of action permissible within the limits of the document
- 129 **3.7**
130 **optional**
131 indicates a course of action permissible within the limits of the document
- 132 **3.8**
133 **shall**
134 indicates requirements to be followed strictly in order to conform to the document and from which no
135 deviation is permitted
- 136 **3.9**
137 **shall not**
138 indicates requirements to be followed strictly in order to conform to the document and from which no
139 deviation is permitted
- 140 **3.10**
141 **should**
142 indicates that among several possibilities, one is recommended as particularly suitable, without
143 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 144 **3.11**
145 **should not**
146 indicates that a certain possibility or course of action is deprecated but not prohibited

147 **4 Symbols and Abbreviated Terms**

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

- 149 **4.1**
150 **CIM**
151 Common Information Model
- 152 **4.2**
153 **CLP**
154 Command Line Protocol
- 155 **4.3**
156 **DMTF**
157 Distributed Management Task Force
- 158 **4.4**
159 **IETF**
160 Internet Engineering Task Force

- 161 **4.5**
 162 **SM**
 163 Server Management
- 164 **4.6**
 165 **SMI-S**
 166 Storage Management Initiative Specification
- 167 **4.7**
 168 **SNIA**
 169 Storage Networking Industry Association
- 170 **4.8**
 171 **UFsT**
 172 User Friendly selection Tag

173 **5 Recipes**

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

- 176 • smShowInstance()
- 177 • smShowInstances()
- 178 • smSetInstance()
- 179 • smShowAssociationInstance()
- 180 • smShowAssociationInstances()

181 **6 Mappings**

182 The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in
 183 the [Platform Watchdog Profile](#). Requirements specified here related to the support for a CLP verb for a
 184 particular class are solely within the context of this profile.

185 **6.1 CIM_DeviceServiceImplementation**

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

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

192 **Table 1 – Command Verb Requirements for CIM_DeviceServiceImplementation**

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

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

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

195 6.1.1 Ordering of Results

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

- 198 • Results for CIM_DeviceServiceImplementation are unordered; therefore, no algorithm is
199 defined.

200 6.1.2 Show

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

204 6.1.2.1 Show Command Form for Multiple Instances – CIM_PlatformWatchdogService Reference

205 This command form is used to show many instances of CIM_DeviceServiceImplementation. This
206 command form corresponds to the `show` command issued against instances of
207 CIM_DeviceServiceImplementation where only one reference is specified and the reference is to the
208 instance of CIM_PlatformWatchdogService.

209 6.1.2.1.1 Command Form

```
210 show <CIM_DeviceServiceImplementation multiple instances>
```

211 6.1.2.1.2 CIM Requirements

212 See CIM_DeviceServiceImplementation in the “CIM Elements” section of the [Platform Watchdog Profile](#)
213 for the list of mandatory properties.

214 6.1.2.1.3 Behavior Requirements

215 6.1.2.1.3.1 Preconditions

216 `$instance` represents the instance of CIM_PlatformWatchdogService which is referenced by
217 CIM_DeviceServiceImplementation.

218 6.1.2.1.3.2 Pseudo Code

```
219 &smShowAssociationInstances ( "CIM_DeviceServiceImplementation" ,  
220     $instance.GetObjectPath() );  
221 &smEnd;
```

222 **6.1.2.2 Show Command Form for Multiple Instances – CIM_LogicalDevice Reference**

223 This command form is used to show many instances of CIM_DeviceServiceImplementation. The
 224 command form corresponds to the `show` command issued against the instance of
 225 CIM_DeviceServiceImplementation where only one reference is specified and the reference is to the
 226 instance of a concrete subclass of CIM_LogicalDevice.

227 **6.1.2.2.1 Command Form**

```
228 show <CIM_DeviceServiceImplementation multiple instances>
```

229 **6.1.2.2.2 CIM Requirements**

230 See CIM_DeviceServiceImplementation in the “CIM Elements” section of the [Platform Watchdog Profile](#)
 231 for the list of mandatory properties.

232 **6.1.2.2.3 Behavior Requirements**

233 **6.1.2.2.3.1 Preconditions**

234 `$instance` represents the instance of a concrete subclass of CIM_LogicalDevice which is referenced by
 235 CIM_DeviceServiceImplementation.

236 **6.1.2.2.3.2 Pseudo Code**

```
237 &smShowAssociationInstances ( "CIM_DeviceServiceImplementation",  

  238     $instance.getObjectPath() );  

  239 &smEnd;
```

240 **6.1.2.3 Show Command Form for a Single Instance – Both References**

241 This command form is used to show a single instance of CIM_DeviceServiceImplementation. This
 242 command form corresponds to the `show` command issued against instances of
 243 CIM_DeviceServiceImplementation where both references are specified and therefore the desired
 244 instance is unambiguously identified.

245 **6.1.2.3.1 Command Form**

```
246 show <CIM_DeviceServiceImplementation single instance>
```

247 **6.1.2.3.2 CIM Requirements**

248 See CIM_DeviceServiceImplementation in the “CIM Elements” section of the [Platform Watchdog Profile](#)
 249 for the list of mandatory properties.

250 **6.1.2.3.3 Behavior Requirements**

251 **6.1.2.3.3.1 Preconditions**

252 `$instanceA` represents the instance of CIM_PlatformWatchdogService which is referenced by
 253 CIM_DeviceServiceImplementation.

254 `$instanceB` represents the instance of a concrete subclass of CIM_LogicalDevice which is referenced
 255 by CIM_DeviceServiceImplementation.

256 **6.1.2.3.3.2 Pseudo Code**

```
257 &smShowAssociationInstance ( "CIM_DeviceServiceImplementation",  

  258     $instanceA.getObjectPath(), $instanceB.getObjectPath() );  

  259 &smEnd;
```

260 6.2 CIM_ElementCapabilities

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

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

267 **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	

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

270 6.2.1 Ordering of Results

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

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

274 6.2.2 Show

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

278 6.2.2.1 Show Command Form for a Single Instance – CIM_PlatformWatchdogService Reference

279 This command form is used to show a single instance of `CIM_ElementCapabilities`. This command form
 280 corresponds to the `show` command issued against instances of `CIM_ElementCapabilities` where only one
 281 reference is specified and the reference is to the scoping instance of `CIM_PlatformWatchdogService`.

282 6.2.2.1.1 Command Form

```
283 show <CIM_ElementCapabilities single instance>
```

284 6.2.2.1.2 CIM Requirements

285 See CIM_ElementCapabilities in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the list
286 of mandatory properties.

287 6.2.2.1.3 Behavior Requirements

288 6.2.2.1.3.1 Preconditions

289 \$instance represents the instance of a CIM_PlatformWatchdogService, which is referenced by
290 CIM_ElementCapabilities.

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

292 6.2.2.1.3.2 Pseudo Code

```
293 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );  
294 &smEnd;
```

295 6.2.2.2 Show Command Form for Multiple Instances – CIM_PlatformWatchdogServiceCapabilities 296 Reference

297 This command form is used to show multiple instances of CIM_ElementCapabilities. This command form
298 corresponds to the show command issued against instances of CIM_ElementCapabilities where only one
299 reference is specified and the reference is to the instance of CIM_PlatformWatchdogServiceCapabilities

300 6.2.2.2.1 Command Form

```
301 show <CIM_ElementCapabilities multiple instances>
```

302 6.2.2.2.2 CIM Requirements

303 See CIM_ElementCapabilities in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the list
304 of mandatory properties.

305 6.2.2.2.3 Behavior Requirements

306 6.2.2.2.3.1 Preconditions

307 \$instance represents the instance of a CIM_PlatformWatchdogServiceCapabilities, which is referenced
308 by CIM_ElementCapabilities.

309 6.2.2.2.3.2 Pseudo Code

```
310 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath(),  
311 NULL );  
312 &smEnd;
```

313 6.2.2.3 Show a Single Instance Target – Both References

314 This command form is used to show a single instance of CIM_ElementCapabilities. This command form
315 corresponds to the show command issued against instances of CIM_ElementCapabilities where both
316 references are specified and therefore the desired instance is unambiguously identified.

317 **6.2.2.3.1 Command Form**318 `show <CIM_ElementCapabilities single instance>`319 **6.2.2.3.2 CIM Requirements**320 See CIM_ElementCapabilities in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the list
321 of mandatory properties.322 **6.2.2.3.3 Behavior Requirements**323 **6.2.2.3.3.1 Preconditions**324 \$instanceA represents the instance of a CIM_PlatformWatchdogService which is referenced by the
325 CIM_ElementCapabilities.326 \$instanceB represents the instance of a CIM_PlatformWatchdogServiceCapabilities which is
327 referenced by the CIM_ElementCapabilities.328 **6.2.2.3.3.2 Pseudo Code**329 `&smShowAssociationInstance ("CIM_ElementCapabilities", $instanceA.getObjectPath(),`
330 `$instanceB.getObjectPath());`
331 `&smEnd;`332 **6.3 CIM_HostedService**333 The `cd`, `exit`, `help`, and `version` verbs shall be supported as described in [DSP0216](#).334 Table 3 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
335 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and
336 target. Table 3 is for informational purposes only; in case of a conflict between Table 3 and requirements
337 detailed in the following sections, the text detailed in the following sections supersedes the information in
338 Table 3.

339

Table 3 – Command Verb Requirements for CIM_HostedService

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

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

342 6.3.1 Ordering of Results

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

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

346 6.3.2 Show

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

349 6.3.2.1 Show Command Form for Multiple Instances – CIM_ComputerSystem Reference

350 This command form is used to show multiple instances of CIM_HostedService. This command form
351 corresponds to the `show` command issued against instances of CIM_HostedService where only one
352 reference is specified and the reference is to an instance of CIM_ComputerSystem.

353 6.3.2.1.1 Command Form

```
354 show <CIM_HostedService multiple objects>
```

355 6.3.2.1.2 CIM Requirements

356 See CIM_HostedService in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the list of
357 mandatory properties.

358 6.3.2.1.3 Behavior Requirements

359 6.3.2.1.3.1 Preconditions

360 `$instance` represents the instance of CIM_ComputerSystem, which is referenced by
361 CIM_HostedService.

362 6.3.2.1.3.2 Pseudo Code

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

365 6.3.2.2 Show Command Form for a Single Instance – CIM_PlatformWatchdogService Reference

366 This command form is to show a single instance of CIM_HostedService. The command form corresponds
367 to the `show` verb issued against instances of CIM_HostedService where only one reference is specified
368 and the reference is to an instance of CIM_PlatformWatchdogService.

369 6.3.2.2.1 Command Form

```
370 show <CIM_HostedService single object>
```

371 6.3.2.2.2 CIM Requirements

372 See CIM_HostedService in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the list of
373 mandatory properties.

374 **6.3.2.2.3 Behavior Requirements**375 **6.3.2.2.3.1 Preconditions**

376 `$instance` represents the instance of `CIM_PlatformWatchdogService`, which is referenced by
377 `CIM_HostedService`.

378 **6.3.2.2.3.2 Pseudo Code**

```
379 &smShowAssociationInstances ( "CIM_HostedService", $instance.GetObjectPath() );
380 &smEnd;
```

381 **6.3.2.3 Show Command Form for a Single Instance – Both References**

382 This command form is to show a single instance of `CIM_HostedService`. This command form
383 corresponds to the `show` command issued against `CIM_HostedService` where both references are
384 specified and therefore the desired instance is unambiguously identified.

385 **6.3.2.3.1 Command Form**

```
386 show <CIM_HostedService single object>
```

387 **6.3.2.3.2 CIM Requirements**

388 See `CIM_HostedService` in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the list of
389 mandatory properties.

390 **6.3.2.3.3 Behavior Requirements**391 **6.3.2.3.3.1 Preconditions**

392 `$instanceA` represents the referenced instance of `CIM_ComputerSystem` through `CIM_HostedService`
393 association.

394 `$instanceB` represents the other instance of `CIM_PlatformWatchdogService` which is referenced by
395 `CIM_HostedService`.

396 **6.3.2.3.3.2 Pseudo Code**

```
397 &smShowAssociationInstance ( "CIM_HostedService", instanceA.GetObjectPath(),
398     $instanceB.GetObjectPath() );
399 &smEnd;
```

400 **6.4 CIM_ServiceAffectsElement**

401 The `cd`, `exit`, `help`, and `version` 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_ServiceAffectsElement**

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

Command Verb	Requirement	Comments
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.4.2.
start	Not supported	
stop	Not supported	

408 No mapping is 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_ServiceAffectsElement, implementations shall
412 utilize the following algorithm to produce the natural (that is, default) ordering:

- 413 • Results for CIM_ServiceAffectsElement 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_ServiceAffectsElement. Implementations shall support the use of the `show` verb with
417 CIM_ServiceAffectsElement.

418 6.4.2.1 Show Command Form for Multiple Instances – CIM_ComputerSystem Reference

419 This command form is used to show multiple instances of CIM_ComputerSystem. This command form
420 corresponds to the `show` command issued against instances of CIM_ServiceAffectsElement where only
421 one reference is specified and the reference is to an instance of CIM_ComputerSystem.

422 6.4.2.1.1 Command Form

```
423 show <CIM_ServiceAffectsElement multiple instances>
```

424 6.4.2.1.2 CIM Requirements

425 See CIM_ServiceAffectsElement in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the
426 list of mandatory properties.

427 6.4.2.1.3 Behavior Requirements

428 6.4.2.1.3.1 Preconditions

429 `$instance` represents the instance of CIM_ComputerSystem, which is referenced by
430 CIM_ServiceAffectsElement.

431 6.4.2.1.3.2 Pseudo Code

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

435 6.4.2.2 Show Command Form for a Single Instance – CIM_LogicalElement Reference

436 This command form is used to show a single instance of CIM_ServiceAffectsElement. This command
437 form corresponds to the `show` command issued against instances of CIM_ServiceAffectsElement where

438 only one reference is specified and the reference is to an instance of a concrete subclass of
439 CIM_LogicalElement.

440 6.4.2.2.1 Command Form

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

442 6.4.2.2.2 CIM Requirements

443 See CIM_ServiceAffectsElement in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the
444 list of mandatory properties.

445 6.4.2.2.3 Behavior Requirements

446 6.4.2.2.3.1 Preconditions

447 \$instance represents the instance of a concrete subclass of CIM_LogicalElement, which is referenced
448 by CIM_ServiceAffectsElement.

449 6.4.2.2.3.2 Pseudo Code

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

453 6.4.2.3 Show Command Form for Multiple Instance – CIM_PlatformWatchdogService Reference

454 This command form is used when to show multiple instances of CIM_ServiceAffectsElement. This
455 command form corresponds to the show command issued against instances of
456 CIM_ServiceAffectsElement where only one reference is specified and the reference is to an instance of
457 CIM_PlatformWatchdogService.

458 6.4.2.3.1 Command Form

```
459 show <CIM_ServiceAffectsElement multiple instances>
```

460 6.4.2.3.2 CIM Requirements

461 See CIM_ServiceAffectsElement in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the
462 list of mandatory properties.

463 6.4.2.3.3 Behavior Requirements

464 6.4.2.3.3.1 Preconditions

465 \$instance represents the instance of CIM_PlatformWatchdogService, which is referenced by
466 CIM_ServiceAffectsElement.

467 6.4.2.3.3.2 Pseudo Code

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

471 6.4.2.4 Show Command Form for a Single Instance – Both References

472 This command form is used to show a single instance of CIM_ServiceAffectsElement. This command
473 form corresponds to the show verb issued against instances of CIM_ServiceAffectsElement where both
474 references are specified and therefore the desired instance is unambiguously identified.

475 6.4.2.4.1 Command Form

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

477 6.4.2.4.2 CIM Requirements

478 See CIM_ServiceAffectsElement in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the
479 list of mandatory properties.

480 6.4.2.4.3 Behavior Requirements

481 6.4.2.4.3.1 Preconditions

482 \$instanceA represents the instance of CIM_PlatformWatchdogService, which is referenced by
483 CIM_ServiceAffectsElement.

484 \$instanceB represents the instance of CIM_ComputerSystem or a concrete subclass of
485 CIM_LogicalElement which is referenced by CIM_ServiceAffectsElement.

486 6.4.2.4.3.2 Pseudo Code

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

490 6.5 CIM_ServiceAvailableToElement

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

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

497 **Table 5 – Command Verb Requirements for CIM_ServiceAvailableToElement**

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	

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

500 6.5.1 Ordering of Results

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

- 503 • Results for CIM_ServiceAvailableToElement are unordered; therefore, no algorithm is defined.

504 6.5.2 Show

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

508 6.5.2.1 Show Command Form for a Single Instance – CIM_LogicalElement Reference

509 This command form is used to show a single instance of CIM_ServiceAvailableToElement. This
510 command form corresponds to the `show` command issued against instances of
511 CIM_ServiceAvailableToElement where only one reference is specified and the reference is to an
512 instance of a concrete subclass of CIM_LogicalElement.

513 6.5.2.1.1 Command Form

```
514 show <CIM_ServiceAvailableToElement single instance>
```

515 6.5.2.1.2 CIM Requirements

516 See CIM_ServiceAvailableToElement in the “CIM Elements” section of the [Platform Watchdog Profile](#) for
517 the list of mandatory properties.

518 6.5.2.1.3 Behavior Requirements

519 6.5.2.1.3.1 Preconditions

520 `$instance` represents the instance of a concrete subclass of CIM_LogicalElement, which is referenced
521 by CIM_ServiceAvailableToElement.

522 6.5.2.1.3.2 Pseudo Code

```
523 &smShowAssociationInstances ( "CIM_ServiceAvailableToElement",  
524     $instance.getObjectPath() );  
525 &smEnd;
```

526 6.5.2.2 Show Command Form for a Single Instance – CIM_PlatformWatchdogService Reference

527 This command form is used when to show a single instance of CIM_ServiceAvailableToElement. This
528 command form corresponds to the `show` command issued against instances of
529 CIM_ServiceAvailableToElement where only one reference is specified and the reference is to an
530 instance of CIM_PlatformWatchdogService.

531 6.5.2.2.1 Command Form

```
532 show <CIM_ServiceAvailableToElement single instance>
```

533 6.5.2.2.2 CIM Requirements

534 See CIM_ServiceAvailableToElement in the “CIM Elements” section of the [Platform Watchdog Profile](#) for
535 the list of mandatory properties.

536 6.5.2.2.3 Behavior Requirements

537 6.5.2.2.3.1 Preconditions

538 `$instance` represents the instance of `CIM_PlatformWatchdogService`, which is referenced by
539 `CIM_ServiceAvailableToElement`.

540 6.5.2.2.3.2 Pseudo Code

```
541 &smShowAssociationInstances ( "CIM_ServiceAvailableToElement",
542     $instance.GetObjectPath() );
543 &smEnd;
```

544 6.5.2.3 Show Command Form for a Single Instance – Both References

545 This command form is used to show a single instance of `CIM_ServiceAvailableToElement`. This
546 command form corresponds to the `show` verb issued against instances of
547 `CIM_ServiceAvailableToElement` where both references are specified and therefore the desired instance
548 is unambiguously identified.

549 6.5.2.3.1 Command Form

```
550 show <CIM_ServiceAvailableToElement single instance>
```

551 6.5.2.3.2 CIM Requirements

552 See `CIM_ServiceAvailableToElement` in the “CIM Elements” section of the [Platform Watchdog Profile](#) for
553 the list of mandatory properties.

554 6.5.2.3.3 Behavior Requirements

555 6.5.2.3.3.1 Preconditions

556 `$instanceA` represents the instance of `CIM_PlatformWatchdogService`, which is referenced by
557 `CIM_ServiceAvailableToElement`.

558 `$instanceB` represents the instance of a concrete subclass of `CIM_LogicalElement` which is referenced
559 by `CIM_ServiceAvailableToElement`.

560 6.5.2.3.3.2 Pseudo Code

```
561 &smShowAssociationInstance ( "CIM_ServiceAvailableToElement",
562     $instanceA.GetObjectPath(), $instanceB.GetObjectPath() );
563 &smEnd;
```

564 6.6 CIM_PlatformWatchdogService

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

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

571

Table 6 – Command Verb Requirements for CIM_PlatformWatchdogService

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

572 No mapping is defined for the following verbs for the specified target: `create`, `delete`, `dump`, `exit`,
573 `load`, and `reset`.

574 6.6.1 Ordering of Results

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

- 577 • Results for `CIM_PlatformWatchdogService` are unordered; therefore, no algorithm is defined.

578 6.6.2 Set

579 This section describes how to implement the `set` verb when it is applied to an instance of
580 `CIM_PlatformWatchdogService`. The `set` verb is used to set properties on an instance of
581 `CIM_PlatformWatchdogService`.

582 Implementations may support the use of the `set` verb with `CIM_PlatformWatchdogService`.

583 6.6.2.1 General Usage of Set for a Single Property

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

586 The requirements for supporting modification of a property using this command form shall be equivalent
587 to the requirement for supporting modification of the property using the `ModifyInstance` operation as
588 defined in the [Platform Watchdog Profile](#).

589 6.6.2.1.1 Command Form

```
590 set <CIM_PlatformWatchdogService single object> <propertyname>=<propertyvalue>
```

591 6.6.2.1.2 CIM Requirements

592 See `CIM_PlatformWatchdogService` in the “CIM Elements” section of the [Platform Watchdog Profile](#) for
593 the list of mandatory properties.

594 6.6.2.1.3 Behavior Requirements

```
595 $instance=<CIM_PlatformWatchdogService single object>
```

```
596 #propertyName[] = <propertyname>
```

```

597 #propertyValues[] = <propertyvalue>
598 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
599 &smEnd;

```

600 6.6.2.2 General Usage of Set for Multiple Properties

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

603 The requirements for supporting modification of a property using this command form shall be equivalent
604 to the requirement for supporting modification of the property using the `ModifyInstance` operation as
605 defined in the [Platform Watchdog Profile](#).

606 6.6.2.2.1 Command Form

```

607 set <CIM_PlatformWatchdogService single object> <propertyname1>=<propertyvalue1>
608 <propertynameN>=<propertyvalueN>

```

609 6.6.2.2.2 CIM Requirements

610 See `CIM_PlatformWatchdogService` in the “CIM Elements” section of the [Platform Watchdog Profile](#) for
611 the list of mandatory properties.

612 6.6.2.2.3 Behavior Requirements

613 6.6.2.2.3.1 Preconditions

614 `$instance` represents the instance of `CIM_PlatformWatchdogService`.

615 6.6.2.2.3.2 Pseudo Code

```

616 for #i < n
617 {
618     #propertyNames[#i] = <propertyname#i>
619     #propertyValues[#i] = <propertyvalue#i>
620 }
621 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
622 &smEnd;

```

623 6.6.3 Show

624 The `show` verb is used to display information about instances of `CIM_PlatformWatchdogService`.
625 Implementations shall support the use of the `show` verb with `CIM_PlatformWatchdogService`.

626 6.6.3.1 Show Command Form for a Single Instance

627 This command form is to show a single instance of `CIM_PlatformWatchdogService`.

628 6.6.3.1.1 Command Form

```

629 show <CIM_PlatformWatchdogService single object>

```

630 6.6.3.1.2 CIM Requirements

631 See `CIM_PlatformWatchdogService` in the “CIM Elements” section of the [Platform Watchdog Profile](#) for
632 the list of mandatory properties.

633 6.6.3.1.3 Behavior Requirements

634 6.6.3.1.3.1 Preconditions

635 `$instance` represents the instance of `CIM_PlatformWatchdogService`.

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

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

638 6.6.3.1.3.2 Pseudo Code

```
639 if (false != #all) { #propertylist[] = NULL; }
640 &smShowInstance ( $instance.GetObjectPath(), #propertylist[] );
641 &smEnd;
```

642 6.6.3.2 Show Command Form for Multiple Instances

643 This command form is used to show multiple instances of `CIM_PlatformWatchdogService`.

644 6.6.3.2.1 Command Form

```
645 show <CIM_PlatformWatchdogService multiple objects>
```

646 6.6.3.2.2 CIM Requirements

647 See `CIM_PlatformWatchdogService` in the “CIM Elements” section of the [Platform Watchdog Profile](#) for
648 the list of mandatory properties.

649 6.6.3.2.3 Behavior Requirements

650 6.6.3.2.3.1 Preconditions

651 `$containerInstance` represents the instance of `CIM_ComputerSystem` which represents the
652 container system and is associated with the target instance of `CIM_PlatformWatchdogService` through
653 the `CIM_HostedService` association.

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

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

656 6.6.3.2.3.2 Pseudo Code

```
657 if (false != #all) { #propertylist[] = NULL; }
658 &smShowInstances ( "CIM_PlatformWatchdogService", "CIM_HostedService",
659     $containerInstance.GetObjectPath(), #propertylist[] );
660 &smEnd;
```

661 6.6.4 Start

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

665 6.6.4.1 Start Command Form for a Single Instance

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

667 **6.6.4.1.1 Command Form**

668 `start <CIM_PlatformWatchdogService single object>`

669 **6.6.4.1.2 CIM Requirements**

670 See CIM_PlatformWatchdogService in the “CIM Elements” section of the [Platform Watchdog Profile](#) for
671 the list of mandatory properties.

672 **6.6.4.1.3 Behavior Requirements**

673 `$instance=<CIM_PlatformWatchdogService single object>`
674 `smStartRSC ($instance.GetObjectPath());`
675 `&smEnd;`

676 **6.6.5 Stop**

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

680 **6.6.5.1 Stop Command Form for a Single Instance**

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

682 **6.6.5.1.1 Command Form**

683 `stop <CIM_PlatformWatchdogService single object>`

684 **6.6.5.1.2 CIM Requirements**

685 See CIM_PlatformWatchdogService in the “CIM Elements” section of the [Platform Watchdog Profile](#) for
686 the list of mandatory properties.

687 **6.6.5.1.3 Behavior Requirements**

688 `$instance=<CIM_PlatformWatchdogService single object>`
689 `smRequestStateChange ($instance.GetObjectPath(), "Disabled");`
690 `&smEnd;`

691 **6.7 CIM_PlatformWatchdogServiceCapabilities**

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

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

698 **Table 7 – Command Verb Requirements for CIM_PlatformWatchdogServiceCapabilities**

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

Command Verb	Requirement	Comments
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.7.2.
start	Not supported	
stop	Not supported	

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

701 6.7.1 Ordering of Results

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

- 704 • Results for CIM_PlatformWatchdogServiceCapabilities are unordered; therefore, no algorithm is
705 defined.

706 6.7.2 Show

707 This section describes how to implement the `show` verb when applied to an instance of
708 CIM_PlatformWatchdogServiceCapabilities. The `show` verb is used to display information about instances
709 of CIM_PlatformWatchdogServiceCapabilities.

710 Implementations shall support the use of the `show` verb with CIM_PlatformWatchdogServiceCapabilities.

711 6.7.2.1 Show Command Form for a Single Instance

712 This command form is used to show a single instance of CIM_PlatformWatchdogServiceCapabilities.

713 6.7.2.1.1 Command Form

714 `show <CIM_PlatformWatchdogServiceCapabilities single object>`

715 6.7.2.1.2 CIM Requirements

716 See CIM_PlatformWatchdogServiceCapabilities in the “CIM Elements” section of the [Platform Watchdog](#)
717 [Profile](#) for the list of mandatory properties.

718 6.7.2.1.3 Behavior Requirements

719 6.7.2.1.3.1 Preconditions

720 `$instance` represents the instance of CIM_PlatformWatchdogServiceCapabilities.

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

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

723 **6.7.2.1.3.2 Pseudo Code**

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

727 **6.7.2.2 Show Multiple Instances**

728 This command form is used to show multiple instances of CIM_PlatformWatchdogServiceCapabilities.

729 **6.7.2.2.1 Command Form**

```
730 show <CIM_PlatformWatchdogServiceCapabilities multiple objects>
```

731 **6.7.2.2.2 CIM Requirements**

732 See CIM_PlatformWatchdogServiceCapabilities in the “CIM Elements” section of the [Platform Watchdog Profile](#) for the list of mandatory properties.

734 **6.7.2.2.3 Behavior Requirements**735 **6.7.2.2.3.1 Preconditions**

736 \$containerInstance represents the instance of CIM_ConcreteCollection with ElementName property
737 that contains “Capabilities” and is associated with the targeted instances of
738 CIM_PlatformWatchdogServiceCapabilities through the CIM_MemberOfCollection association.

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

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

741 **6.7.2.2.3.2 Pseudo Code**

```
742 if (false != #all) { #propertylist[] = NULL; }
743 &smShowInstances ( "CIM_PlatformWatchdogServiceCapabilities",
744   "CIM_MemberOfCollection", $containerInstance.getObjectPath(), #propertylist[] );
745 &smEnd;
```

746
747
748
749
750

ANNEX A
(informative)

Change Log

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

751