



1  
2  
3  
4

**Document Number: DSP0842**

**Date: 2009-06-04**

**Version: 1.0.0**

5  
6

# **OS Status Profile SM CLP Command Mapping Specification**

7  
8  
9

**Document Type: Specification**

**Document Status: DMTF Standard**

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

# CONTENTS

34 Foreword ..... 5

35 Introduction ..... 6

36 1 Scope ..... 7

37 2 Normative References..... 7

38 2.1 Approved References ..... 7

39 2.2 Other References..... 7

40 3 Terms and Definitions..... 7

41 4 Symbols and Abbreviated Terms..... 8

42 5 Recipes..... 9

43 6 Mappings..... 9

44 6.1 CIM\_OperatingSystem..... 9

45 6.2 CIM\_RunningOS..... 14

46 6.3 CIM\_InstalledOS..... 16

47 6.4 CIM\_ElementCapabilities ..... 19

48 6.5 CIM\_OperatingSystemCapabilities..... 21

49 ANNEX A (informative) Change Log ..... 24

50

## 51 Tables

52 Table 1 – Command Verb Requirements for CIM\_OperatingSystem..... 9

53 Table 2 – Command Verb Requirements for CIM\_RunningOS..... 14

54 Table 3 – Command Verb Requirements for CIM\_InstalledOS..... 17

55 Table 4 – Command Verb Requirements for CIM\_ElementCapabilities ..... 19

56 Table 5 – Command Verb Requirements for CIM\_OperatingSystemCapabilities..... 21

57



59

## Foreword

60 The *OS Status Profile SM CLP Command Mapping Specification* (DSP0842) was prepared by the Server  
61 Management Working Group.

### 62 **Conventions**

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

### 65 **Acknowledgements**

- 66 • Aaron Merkin – IBM
- 67 • Jon Hass – Dell
- 68 • Khachatur Papanyan – Dell
- 69 • Jeff Hilland – HP
- 70 • Joel Clark – Intel

71

72

## Introduction

73 This document defines the SM CLP mapping for CIM elements described in the [OS Status Profile](#). The  
74 information in this specification, combined with the *SM CLP-to-CIM Common Mapping Specification 1.0*  
75 ([DSP0216](#)), is intended to be sufficient to implement SM CLP commands relevant to the classes,  
76 properties, and methods described in the [OS Status Profile](#) using CIM operations.

77 The target audience for this specification is implementers of the SM CLP support for the [OS Status](#)  
78 [Profile](#).

# 79 OS Status Profile SM CLP Command Mapping Specification

## 80 1 Scope

81 This specification contains the requirements for an implementation of the SM CLP to provide access to,  
82 and implement the behaviors of, the [OS Status Profile](#).

## 83 2 Normative References

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

### 87 2.1 Approved References

88 DMTF DSP1029, *OS Status Profile 1.0*,  
89 [http://www.dmtf.org/standards/published\\_documents/DSP1029\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1029_1.0.pdf)

90 DMTF DSP0216, *SM CLP-to-CIM Common Mapping Specification 1.0*,  
91 [http://www.dmtf.org/standards/published\\_documents/DSP0216\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0216_1.0.pdf)

92 SNIA, *Storage Management Initiative Specification (SMI-S) 1.1.0*,  
93 [http://www.snia.org/tech\\_activities/standards/curr\\_standards/smi](http://www.snia.org/tech_activities/standards/curr_standards/smi)

### 94 2.2 Other References

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

## 97 3 Terms and Definitions

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

### 99 3.1

#### 100 **can**

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

### 102 3.2

#### 103 **cannot**

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

### 105 3.3

#### 106 **conditional**

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

- 109 **3.4**  
110 **mandatory**  
111 indicates requirements to be followed strictly in order to conform to the document and from which no  
112 deviation is permitted
- 113 **3.5**  
114 **may**  
115 indicates a course of action permissible within the limits of the document
- 116 **3.6**  
117 **need not**  
118 indicates a course of action permissible within the limits of the document
- 119 **3.7**  
120 **optional**  
121 indicates a course of action permissible within the limits of the document
- 122 **3.8**  
123 **shall**  
124 indicates requirements to be followed strictly in order to conform to the document and from which no  
125 deviation is permitted
- 126 **3.9**  
127 **shall not**  
128 indicates requirements to be followed strictly in order to conform to the document and from which no  
129 deviation is permitted
- 130 **3.10**  
131 **should**  
132 indicates that among several possibilities, one is recommended as particularly suitable, without  
133 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 134 **3.11**  
135 **should not**  
136 indicates that a certain possibility or course of action is deprecated but not prohibited

## 137 **4 Symbols and Abbreviated Terms**

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

- 139 **4.1**  
140 **CIM**  
141 Common Information Model
- 142 **4.2**  
143 **CLP**  
144 Command Line Protocol
- 145 **4.3**  
146 **DMTF**  
147 Distributed Management Task Force



148 **4.4**  
 149 **SM**  
 150 Server Management

151 **4.5**  
 152 **SMI-S**  
 153 Storage Management Initiative Specification

154 **4.6**  
 155 **SNIA**  
 156 Storage Networking Industry Association

## 157 **5 Recipes**

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

- 160 • smShowInstance()
- 161 • smShowInstances()
- 162 • smSetInstance()
- 163 • smShowAssociationInstances()
- 164 • smShowAssociationInstance()

## 165 **6 Mappings**

166 The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in  
 167 the [OS Status Profile](#). Requirements specified here related to support for a CLP verb for a particular class  
 168 are solely within the context of this profile.

### 169 **6.1 CIM\_OperatingSystem**

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

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

176 **Table 1 – Command Verb Requirements for CIM\_OperatingSystem**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	May	See 6.1.2.
set	May	See 6.1.3.

Command Verb	Requirement	Comments
show	Shall	See 6.1.4.
start	Not supported	
stop	May	See 6.1.5.

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

### 178 6.1.1 Ordering of Results

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

- 181 • Results for CIM\_OperatingSystem are unordered; therefore, no algorithm is defined.

### 182 6.1.2 Reset

183 This section describes how to implement the `reset` verb when applied to an instance of  
184 CIM\_OperatingSystem. Implementations may support the use of the `reset` verb with  
185 CIM\_OperatingSystem.

#### 186 6.1.2.1 General Usage of Reset for a Single Property

##### 187 6.1.2.1.1 Command Form

```
188 reset <CIM_OperatingSystem single instance>
```

##### 189 6.1.2.1.2 CIM Requirements

```
190 uint16 EnabledState;  
191 uint16 RequestedState;  
192 uint32 CIM_OperatingSystem.RequestStateChange (  
193     [IN] uint16 RequestedState,  
194     [OUT] REF CIM_ConcreteJob Job,  
195     [IN] datetime TimeoutPeriod );
```

##### 196 6.1.2.1.3 Behavior Requirements

###### 197 6.1.2.1.3.1 Preconditions

198 `$instance` represents the targeted instance of CIM\_OperatingSystem.

```
199 $instance=<CIM_OperatingSystem single instance>
```

###### 200 6.1.2.1.3.2 Pseudo Code

```
201 &smResetRSC ( $instance.getObjectPath() );  
202 &smEnd;
```

### 203 6.1.3 Set

204 This section describes how to implement the `set` verb when it is applied to an instance of  
205 CIM\_OperatingSystem. Implementations may support the use of the `set` verb with  
206 CIM\_OperatingSystem.

### 207 6.1.3.1 Set requestedstate

208 This command form is for the `set` verb applied to a single instance of `CIM_OperatingSystem`.

#### 209 6.1.3.1.1 Command Form

```
210 set <CIM_OperatingSystem single instance> RequestedState=<requestedstate>
```

#### 211 6.1.3.1.2 CIM Requirements

```
212 uint16 EnabledState;
213 uint16 RequestedState;
214 uint32 EnabledLogicalElement.RequestStateChange (
215     [IN] uint16 RequestedState,
216     [OUT] REF CIM_ConcreteJob Job,
217     [IN] datetime TimeoutPeriod );
```

#### 218 6.1.3.1.3 Behavior Requirements

##### 219 6.1.3.1.3.1 Preconditions

```
220 $instance=<CIM_OperatingSystem single instance>
```

##### 221 6.1.3.1.3.2 Pseudo Code

```
222 &smRequestStateChange ( $instance.getObjectPath(), <requestedstate> );
223 &smEnd;
```

### 224 6.1.3.2 General Usage of Set for a Single Property

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

227 The requirement for supporting modification of a property using this command form shall be equivalent to  
228 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined  
229 in the [OS Status Profile](#).

#### 230 6.1.3.2.1 Command Form

```
231 set <CIM_OperatingSystem single instance> <propertyname>=<propertyvalue>
```

#### 232 6.1.3.2.2 CIM Requirements

233 See `CIM_OperatingSystem` in the “CIM Elements” section of the [OS Status Profile](#) for the list of  
234 modifiable properties.

#### 235 6.1.3.2.3 Behavior Requirements

##### 236 6.1.3.2.3.1 Preconditions

```
237 $instance=<CIM_OperatingSystem single instance>
```

##### 238 6.1.3.2.3.2 Pseudo Code

```
239 #propertyName[] = {<propertyname>};
240 #propertyValues[] = {<propertyvalue>};
241 &smSetInstance ( $instance, #propertyName[], #propertyValues[] );
242 &smEnd;
```

### 243 6.1.3.3 General Usage of Set for Multiple Properties

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

247 The requirement for supporting modification of a property using this command form shall be equivalent to  
248 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined  
249 in the [OS Status Profile](#).

#### 250 6.1.3.3.1 Command Form

```
251 set <CIM_OperatingSystem single instance> <propertyname1>=<propertyvalue1>  
252 <propertynamen>=<propertyvaluen>
```

#### 253 6.1.3.3.2 CIM Requirements

254 See `CIM_OperatingSystem` in the “CIM Elements” section of the [OS Status Profile](#) for the list of supported  
255 properties.

#### 256 6.1.3.3.3 Behavior Requirements

##### 257 6.1.3.3.3.1 Preconditions

```
258 $instance=<CIM_OperatingSystem single instance>
```

##### 259 6.1.3.3.3.2 Pseudo Code

```
260 #propertyName[] = {<propertyname>};  
261 for #i < n  
262 {  
263     #propertyName[#i] = <propertyname#i>  
264     #propertyValues[#i] = <propertyvalue#i>  
265 }  
266 &smSetInstance ( $instance, #propertyName[], #propertyValues[] );  
267 &smEnd;
```

### 268 6.1.4 Show

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

272 The `show` verb is used to display information about the account.

#### 273 6.1.4.1 Show a Single Instance

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

##### 275 6.1.4.1.1 Command Form

```
276 show <CIM_OperatingSystem single instance>
```

##### 277 6.1.4.1.2 CIM Requirements

278 See `CIM_OperatingSystem` in the “CIM Elements” section of the [OS Status Profile](#) for the list of  
279 mandatory properties.

### 280 6.1.4.1.3 Behavior Requirements

#### 281 6.1.4.1.3.1 Preconditions

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

#### 283 6.1.4.1.3.2 Pseudo Code

```
284 $instance=<CIM_OperatingSystem single instance>
285 #propertylist[] = NULL;
286 if (false == #all)
287     {
288         #propertylist[] = { //all mandatory non-key properties };
289     }
290 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
291 &smEnd;
```

### 292 6.1.4.2 Show Multiple Instances

293 This command form is for the show verb applied to multiple instances of CIM\_OperatingSystem. This  
294 command form corresponds to UfsT-based selection within a scoping system.

#### 295 6.1.4.2.1 Command Form

```
296 show <CIM_OperatingSystem multiple instances>
```

#### 297 6.1.4.2.2 CIM Requirements

298 See CIM\_OperatingSystem in the “CIM Elements” section of the [OS Status Profile](#) for the list of  
299 mandatory properties.

### 300 6.1.4.2.3 Behavior Requirements

#### 301 6.1.4.2.3.1 Preconditions

302 \$containerInstance contains the instance of CIM\_ComputerSystem for which we are displaying scoped  
303 accounts (CIM\_OperatingSystem instances). The [OS Status Profile](#) requires that the  
304 CIM\_OperatingSystem instance be associated with its scoping system via an instance of the  
305 CIM\_OperatingSystemOnSystem association.

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

#### 307 6.1.4.2.3.2 Pseudo Code

```
308 #propertylist[] = NULL;
309 if (false == #all)
310     {
311         #propertylist[] = { //all mandatory non-key properties };
312     }
313 &smShowInstances ( "CIM_OperatingSystem", "CIM_OperatingSystemOnSystem",
314     $containerInstance.getObjectPath(), #propertylist[] );
315 &smEnd;
```

316 **6.1.5 Stop**

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

320 The `stop` verb is used to disable an account.

321 **6.1.5.1 Stop a Single Instance**

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

323 **6.1.5.1.1 Command Form**

```
324 stop <CIM_OperatingSystem single instance>
```

325 **6.1.5.1.2 CIM Requirements**

```
326 uint16 EnabledState;
327 uint16 RequestedState;
328 //3 (Disabled)
329 uint32 EnabledLogicalElement.RequestStateChange (
330     [IN] uint16 RequestedState = 3,
331     [OUT] REF CIM_ConcreteJob Job,
332     [IN] datetime TimeoutPeriod );
```

333 **6.1.5.1.3 Behavior Requirements**334 **6.1.5.1.3.1 Preconditions**

```
335 $instance=<CIM_OperatingSystem single instance>
```

336 **6.1.5.1.3.2 Pseudo Code**

```
337 smStopRSC ( $instance.getObjectPath() );
338 &smEnd;
```

339 **6.2 CIM\_RunningOS**

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

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

346 **Table 2 – Command Verb Requirements for CIM\_RunningOS**

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 6.2.2.
start	Not supported	
stop	Not supported	

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

### 349 6.2.1 Ordering of Results

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

- 352 • Results for `CIM_RunningOS` are unordered; therefore, no algorithm is defined.

### 353 6.2.2 Show

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

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

#### 357 6.2.2.1 Show Multiple Instances

358 This command form is for the `show` verb applied to multiple instances. This command form corresponds  
 359 to a `show` command issued against `CIM_RunningOS` where only one reference is specified and the  
 360 reference is to an instance of `CIM_ComputerSystem`.

##### 361 6.2.2.1.1 Command Form

```
362 show <CIM_RunningOS multiple instances>
```

##### 363 6.2.2.1.2 CIM Requirements

364 See `CIM_RunningOS` in the “CIM Elements” section of the [OS Status Profile](#) for the list of mandatory  
 365 properties.

##### 366 6.2.2.1.3 Behavior Requirements

###### 367 6.2.2.1.3.1 Preconditions

368 `$instance` contains the instance of `CIM_ComputerSystem` which is referenced by `CIM_RunningOS`.

###### 369 6.2.2.1.3.2 Pseudo Code

```
370 &smShowAssociationInstances ( "CIM_RunningOS", $instance.getObjectPath() );  

  371 &smEnd;
```

#### 372 6.2.2.2 Show a Single Instance – `CIM_OperatingSystem` Reference

373 This command form is for the `show` verb applied to a single instance. This command form corresponds to  
 374 a `show` command issued against `CIM_RunningOS` where the reference specified is to an instance of  
 375 `CIM_OperatingSystem`. An instance is referenced by exactly one instance of `CIM_RunningOS`. Therefore,  
 376 a single instance will be returned.

### 377 6.2.2.2.1 Command Form

```
378 show <CIM_RunningOS single instance>
```

### 379 6.2.2.2.2 CIM Requirements

380 See CIM\_RunningOS in the “CIM Elements” section of the [OS Status Profile](#) for the list of mandatory  
381 properties.

### 382 6.2.2.2.3 Behavior Requirements

#### 383 6.2.2.2.3.1 Preconditions

384 \$instance contains the instance of CIM\_OperatingSystem which is referenced by CIM\_RunningOS.

#### 385 6.2.2.2.3.2 Pseudo Code

```
386 &smShowAssociationInstances ( "CIM_RunningOS", $instance.getObjectPath() );  
387 &smEnd;
```

### 388 6.2.2.3 Show a Single Instance – Both References

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

#### 392 6.2.2.3.1 Command Form

```
393 show <CIM_RunningOS single instance>
```

#### 394 6.2.2.3.2 CIM Requirements

395 See CIM\_RunningOS in the “CIM Elements” section of the [OS Status Profile](#) for the list of mandatory  
396 properties.

#### 397 6.2.2.3.3 Behavior Requirements

##### 398 6.2.2.3.3.1 Preconditions

399 \$instanceA contains the instance of CIM\_ComputerSystem which is referenced by CIM\_RunningOS.

400 \$instanceB contains the instance of CIM\_OperatingSystem which is referenced by CIM\_RunningOS.

##### 401 6.2.2.3.3.2 Pseudo Code

```
402 &smShowAssociationInstance ( "CIM_RunningOS", $instanceA.getObjectPath(),  
403     $instanceB.getObjectPath() );  
404 &smEnd;
```

## 405 6.3 CIM\_InstalledOS

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

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



412

**Table 3 – Command Verb Requirements for CIM\_InstalledOS**

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	

413 No mappings are defined for the following verbs for the specified target: *create*, *delete*, *dump*, *load*,  
 414 *reset*, *set*, *start*, and *stop*.

### 415 6.3.1 Ordering of Results

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

- 418 • Results for CIM\_InstalledOS are unordered; therefore, no algorithm is defined.

### 419 6.3.2 Show

420 This section describes how to implement the *show* verb when applied to an instance of CIM\_InstalledOS.  
 421 Implementations shall support the use of the *show* verb with CIM\_InstalledOS.

422 The *show* command is used to display information about the CIM\_InstalledOS instance or instances.

#### 423 6.3.2.1 Show Multiple Instances

424 This command form is for the *show* verb applied to multiple instances. This command form corresponds  
 425 to a *show* command issued against CIM\_InstalledOS where only one reference is specified and the  
 426 reference is to an instance of CIM\_ComputerSystem.

##### 427 6.3.2.1.1 Command Form

```
428 show <CIM_InstalledOS multiple instances>
```

##### 429 6.3.2.1.2 CIM Requirements

430 See CIM\_InstalledOS in the “CIM Elements” section of the [OS Status Profile](#) for the list of mandatory  
 431 properties.

##### 432 6.3.2.1.3 Behavior Requirements

###### 433 6.3.2.1.3.1 Preconditions

434 \$instance contains the instance of CIM\_ComputerSystem which is referenced by CIM\_InstalledOS.

**435 6.3.2.1.3.2 Pseudo Code**

```
436 &smShowAssociationInstances ( "CIM_InstalledOS", $instance.GetObjectPath() );  
437 &smEnd;
```

**438 6.3.2.2 Show a Single Instance – CIM\_OperatingSystem Reference**

439 This command form is for the `show` verb applied to a single instance. This command form corresponds to  
440 a `show` command issued against `CIM_InstalledOS` where the reference specified is to an instance of  
441 `CIM_OperatingSystem`. An instance is referenced by exactly one instance of `CIM_InstalledOS`. Therefore,  
442 a single instance will be returned.

**443 6.3.2.2.1 Command Form**

```
444 show <CIM_InstalledOS single instance>
```

**445 6.3.2.2.2 CIM Requirements**

446 See `CIM_InstalledOS` in the “CIM Elements” section of the [OS Status Profile](#) for the list of mandatory  
447 properties.

**448 6.3.2.2.3 Behavior Requirements****449 6.3.2.2.3.1 Preconditions**

450 `$instance` contains the instance of `CIM_OperatingSystem` which is referenced by `CIM_InstalledOS`.

**451 6.3.2.2.3.2 Pseudo Code**

```
452 &smShowAssociationInstances ( "CIM_InstalledOS", $instance.GetObjectPath() );  
453 &smEnd;
```

**454 6.3.2.3 Show a Single Instance – Both References**

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

**458 6.3.2.3.1 Command Form**

```
459 show <CIM_InstalledOS single instance>
```

**460 6.3.2.3.2 CIM Requirements**

461 See `CIM_InstalledOS` in the “CIM Elements” section of the [OS Status Profile](#) for the list of mandatory  
462 properties.

**463 6.3.2.3.3 Behavior Requirements****464 6.3.2.3.3.1 Preconditions**

465 `$instanceA` contains the instance of `CIM_ComputerSystem` which is referenced by `CIM_InstalledOS`.

466 `$instanceB` contains the instance of `CIM_OperatingSystem` which is referenced by `CIM_InstalledOS`.

**467 6.3.2.3.3.2 Pseudo Code**

```
468 &smShowAssociationInstance ( "CIM_InstalledOS", $instanceA.GetObjectPath(),  
469     $instanceB.GetObjectPath() );  
470 &smEnd;
```

## 471 6.4 CIM\_ElementCapabilities

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

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

478 **Table 4 – 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 6.4.2.
start	Not supported	
stop	Not supported	

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

### 481 6.4.1 Ordering of Results

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

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

### 485 6.4.2 Show

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

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

#### 491 6.4.2.1 Show Multiple Instances – CIM\_OperatingSystemCapabilities Reference

492 This command form is for the `show` verb applied to multiple instances. This command form corresponds  
 493 to a `show` command issued against `CIM_ElementCapabilities` where only one reference is specified and  
 494 the reference is to an instance of `CIM_OperatingSystemCapabilities`.

##### 495 6.4.2.1.1 Command Form

496 `show <CIM_ElementCapabilities multiple instances>`

#### 497 6.4.2.1.2 CIM Requirements

498 See CIM\_ElementCapabilities in the “CIM Elements” section of the [OS Status Profile](#) for the list of  
499 mandatory properties.

#### 500 6.4.2.1.3 Behavior Requirements

##### 501 6.4.2.1.3.1 Preconditions

502 \$instance contains the instance of CIM\_OperatingSystemCapabilities which is referenced by  
503 CIM\_ElementCapabilities.

##### 504 6.4.2.1.3.2 Pseudo Code

```
505 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );  
506 &smEnd;
```

#### 507 6.4.2.2 Show a Single Instance – CIM\_OperatingSystem Reference

508 This command form is for the `show` verb applied to a single instance. This command form corresponds to  
509 a `show` command issued against CIM\_ElementCapabilities where the reference specified is to an  
510 instance of CIM\_OperatingSystem.

##### 511 6.4.2.2.1 Command Form

```
512 show <CIM_ElementCapabilities single instance>
```

##### 513 6.4.2.2.2 CIM Requirements

514 See CIM\_ElementCapabilities in the “CIM Elements” section of the [OS Status Profile](#) for the list of  
515 mandatory properties.

##### 516 6.4.2.2.3 Behavior Requirements

###### 517 6.4.2.2.3.1 Preconditions

518 \$instance contains the instance of CIM\_OperatingSystem, which is referenced by  
519 CIM\_ElementCapabilities.

###### 520 6.4.2.2.3.2 Pseudo Code

```
521 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );  
522 &smEnd;
```

#### 523 6.4.2.3 Show a Single Instance – CIM\_OperatingSystemCapabilities and CIM\_OperatingSystem 524 References

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

##### 528 6.4.2.3.1 Command Form

```
529 show <CIM_ElementCapabilities single instance>
```

##### 530 6.4.2.3.2 CIM Requirements

531 See CIM\_ElementCapabilities in the “CIM Elements” section of the [OS Status Profile](#) for the list of  
532 mandatory properties.

### 533 6.4.2.3.3 Behavior Requirements

#### 534 6.4.2.3.3.1 Preconditions

535 \$instanceA contains the instance of CIM\_OperatingSystemCapabilities which is referenced by  
536 CIM\_ElementCapabilities.

537 \$instanceB contains the instance of CIM\_OperatingSystem which is referenced by  
538 CIM\_ElementCapabilities.

#### 539 6.4.2.3.3.2 Pseudo Code

```
540 &smShowAssociationInstance ( "CIM_ElementCapabilities", $instanceA.getObjectPath(),
541     $instanceB.getObjectPath() );
542 &smEnd;
```

## 543 6.5 CIM\_OperatingSystemCapabilities

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

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

550 **Table 5 – Command Verb Requirements for CIM\_OperatingSystemCapabilities**

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	

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

### 553 6.5.1 Ordering of Results

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

- 556 • Results for CIM\_OperatingSystemCapabilities are unordered; therefore, no algorithm is defined.

557 **6.5.2 Show**

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

561 The `show` verb is used to display information about an instance or instances of the  
 562 `CIM_OperatingSystemCapabilities` class.

563 **6.5.2.1 Show a Single Instance**

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

566 **6.5.2.1.1 Command Form**

```
567 show <CIM_OperatingSystemCapabilities single instance>
```

568 **6.5.2.1.2 CIM Requirements**

569 See `CIM_OperatingSystemCapabilities` in the “CIM Elements” section of the [OS Status Profile](#) for the list  
 570 of mandatory properties.

571 **6.5.2.1.3 Behavior Requirements**572 **6.5.2.1.3.1 Preconditions**

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

574 **6.5.2.1.3.2 Pseudo Code**

```
575 $instance=<CIM_OperatingSystemCapabilities single instance>  

  576 #propertylist[] = NULL;  

  577 if (false == #all)  

  578 {  

  579     #propertylist[] = {/all mandatory non-key properties}  

  580 }  

  581 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );  

  582 &smEnd;
```

583 **6.5.2.2 Show Multiple Instances**

584 This command form is for the `show` verb applied to multiple instances of  
 585 `CIM_OperatingSystemCapabilities`. This command form corresponds to UfsT-based selection within a  
 586 `capabilities` collection.

587 **6.5.2.2.1 Command Form**

```
588 show <CIM_OperatingSystemCapabilities multiple instances>
```

589 **6.5.2.2.2 CIM Requirements**

590 See `CIM_OperatingSystemCapabilities` in the “CIM Elements” section of the [OS Status Profile](#) for the list  
 591 of mandatory properties.

**592 6.5.2.2.3 Behavior Requirements****593 6.5.2.2.3.1 Preconditions**

594 \$containerInstance contains the instance of CIM\_Group for which the contained CIM\_Capabilities  
595 instances are displayed. CIM\_Capabilities instances are addressed via an aggregating instance of  
596 CIM\_Group.

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

**598 6.5.2.2.3.2 Pseudo Code**

```
599 #propertylist[] = NULL;  
600 if (false == #all)  
601     {  
602         #propertylist[] = { //all mandatory non-key properties }  
603     }  
604 &smShowInstances ( "CIM_OperatingSystemCapabilities", "CIM_MemberOfCollection",  
605     $containerInstance.getObjectPath(), #propertylist[] );  
606 &smEnd;
```

607

608  
609  
610  
611  
612

**ANNEX A**  
(informative)

**Change Log**

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

613