



1
2
3
4

Document Number: DSP0811

Date: 2009-06-04

Version: 1.0.0

5 **Simple Identity Management Profile SM CLP**
6 **Command Mapping Specification**

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

10

11 Copyright notice

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

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

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

33

34

CONTENTS

35 Foreword 5

36 Introduction 6

37 1 Scope 7

38 2 Normative References..... 7

39 2.1 Approved References 7

40 2.2 Other References..... 7

41 3 Terms and Definitions..... 7

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

43 5 Recipes..... 9

44 5.1 IShowAccount 9

45 6 Mappings..... 10

46 6.1 CIM_Account 10

47 6.2 CIM_AccountManagementCapabilities..... 18

48 6.3 CIM_AccountManagementService 20

49 6.4 CIM_AccountOnSystem..... 22

50 6.5 CIM_AccountSettingData 24

51 6.6 CIM_AssignedIdentity 27

52 6.7 CIM_ConcreteDependency 30

53 6.8 CIM_ElementCapabilities 32

54 6.9 CIM_ElementSettingData 36

55 6.10 CIM_EnabledLogicalElementCapabilities..... 39

56 6.11 CIM_Group 41

57 6.12 CIM_HostedService 43

58 6.13 CIM_Identity 45

59 6.14 CIM_IdentityContext 47

60 6.15 CIM_MemberOfCollection 49

61 6.16 CIM_OwningCollectionElement 52

62 6.17 CIM_ServiceAffectsElement 55

63 6.18 CIM_SettingsDefineCapabilities 57

64 6.19 CIM_UserContact 60

65 ANNEX A (informative) Change Log 63

66

67 Tables

68 Table 1 – Local Recipes..... 9

69 Table 2 – Command Verb Requirements for CIM_Account 10

70 Table 3 – Command Verb Requirements for CIM_AccountManagementCapabilities..... 18

71 Table 4 – Command Verb Requirements for CIM_AccountManagementService 20

72 Table 5 – Command Verb Requirements for CIM_AccountOnSystem 22

73 Table 6 – Command Verb Requirements for CIM_AccountSettingData 24

74 Table 7 – Command Verb Requirements for CIM_AssignedIdentity 28

75 Table 8 – Command Verb Requirements for CIM_ConcreteDependency 30

76 Table 9 – Command Verb Requirements for CIM_ElementCapabilities 32

77 Table 10 – Command Verb Requirements for CIM_ElementSettingData 36

78 Table 11 – Command Verb Requirements for CIM_EnabledLogicalElementCapabilities..... 39

79 Table 12 – Command Verb Requirements for CIM_Group 41

80 Table 13 – Command Verb Requirements for CIM_HostedService 43

81	Table 14 – Command Verb Requirements for CIM_Identity.....	45
82	Table 15 – Command Verb Requirements for CIM_IdentityContext	47
83	Table 16 – Command Verb Requirements for CIM_MemberOfCollection	50
84	Table 17 – Command Verb Requirements for CIM_OwningCollectionElement	52
85	Table 18 – Command Verb Requirements for CIM_ServiceAffectsElement	55
86	Table 19 – Command Verb Requirements for CIM_SettingsDefineCapabilities	57
87	Table 20 – Command Verb Requirements for CIM_UserContact	61
88		

89

Foreword

90 The *Simple Identity Management Profile SM CLP Command Mapping Specification* (DSP0811) was
91 prepared by the Server Management Working Group.

92 **Conventions**

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

95 **Acknowledgements**

96 The authors wish to acknowledge the following participants from the DTMF Server Management Working
97 Group:

- 98 • Aaron Merkin – IBM
- 99 • Jon Hass – Dell
- 100 • Khachatur Papanyan – Dell
- 101 • Reddy Dasari – Dell
- 102 • Jeff Hilland – HP
- 103 • Christina Shaw – HP
- 104 • Vincent – Intel
- 105 • John Leung – Intel

106

Introduction

107 This document defines the SM CLP mapping for CIM elements described in the [Simple Identity](#)
108 [Management Profile](#). The information in this specification, combined with the [SM CLP-to-CIM Common](#)
109 [Mapping Specification 1.0](#), is intended to be sufficient to implement SM CLP commands relevant to the
110 classes, properties, and methods described in the [Simple Identity Management Profile](#) using CIM
111 operations.

112 The target audience for this specification is implementers of the SM CLP support for the [Simple Identity](#)
113 [Management Profile](#).

Simple Identity Management Profile SM CLP Command Mapping Specification

1 Scope

This specification contains the requirements for an implementation of the SM CLP to provide access to, and implement the behaviors of, the [Simple Identity Management Profile](#).

2 Normative References

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

2.1 Approved References

DMTF DSP1034, *Simple Identity Management Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1034_1.0.pdf

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

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

2.2 Other References

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

3 Terms and Definitions

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

3.1

can

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

3.2

cannot

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

3.3

conditional

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

- 145 **3.4**
146 **mandatory**
147 indicates requirements to be followed strictly in order to conform to the document and from which no
148 deviation is permitted
- 149 **3.5**
150 **may**
151 indicates a course of action permissible within the limits of the document
- 152 **3.6**
153 **need not**
154 indicates a course of action permissible within the limits of the document
- 155 **3.7**
156 **optional**
157 indicates a course of action permissible within the limits of the document
- 158 **3.8**
159 **shall**
160 indicates requirements to be followed strictly in order to conform to the document and from which no
161 deviation is permitted
- 162 **3.9**
163 **shall not**
164 indicates requirements to be followed strictly in order to conform to the document and from which no
165 deviation is permitted
- 166 **3.10**
167 **should**
168 indicates that among several possibilities, one is recommended as particularly suitable, without
169 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 170 **3.11**
171 **should not**
172 indicates that a certain possibility or course of action is deprecated but not prohibited

173 **4 Symbols and Abbreviated Terms**

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

- 175 **4.1**
176 **CIM**
177 Common Information Model
- 178 **4.2**
179 **CLP**
180 Command Line Protocol
- 181 **4.3**
182 **DMTF**
183 Distributed Management Task Force

- 184 **4.4**
 185 **IETF**
 186 Internet Engineering Task Force
- 187 **4.5**
 188 **SM**
 189 Server Management
- 190 **4.6**
 191 **SMI-S**
 192 Storage Management Initiative Specification
- 193 **4.7**
 194 **SNIA**
 195 Storage Networking Industry Association
- 196 **4.8**
 197 **UFsT**
 198 User Friendly selection Tag

199 **5 Recipes**

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

- 202 • smShowInstance()
- 203 • smShowInstances()
- 204 • smSetInstance()
- 205 • smShowAssociationInstances()
- 206 • smShowAssociationInstance()

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

209 **Table 1 – Local Recipes**

Recipe Name	Description	Definition
IShowAccount	Show an instance of CIM_Account	See 5.1.

210 The following subclauses detail Local Recipes defined for use in this mapping.

211 **5.1 IShowAccount**

212 **5.1.1 Description**

213 IShowAccount is a reusable recipe for displaying an instance of CIM_Account. A recipe is defined for re-
 214 use by the *show* and *create* verbs applied to CIM_Account.

215 5.1.2 Preconditions

216 \$account contains the instance of CIM_Account to display.

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

218 5.1.3 Pseudo Code

```

219 sub lShowAccount($account, #all)
220 {
221     #propertylist[] = NULL;
222     //if we're not displaying all of the properties, provide a list
223     if (false == #all)
224     {
225         #propertylist[] = { //all mandatory non-key properties };
226     }
227     &smShowInstance ( $account.GetObjectPath(), #propertyList[] );
228     &smEnd;
229 } //lShowAccount()

```

230 6 Mappings

231 The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in
 232 the [Simple Identity Management Profile](#). Requirements specified here related to support for a CLP verb
 233 for a particular class are solely within the context of this profile.

234 6.1 CIM_Account

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

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

241 **Table 2 – Command Verb Requirements for CIM_Account**

Command Verb	Requirement	Comments
create	May	See 6.1.2.
delete	May	See 6.1.3.
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.1.4.
show	Shall	See 6.1.5.
start	May	See 6.1.6.
stop	May	See 6.1.7.

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

243 6.1.1 Ordering of Results

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

- 246 • Results for CIM_Account are unordered; therefore, no algorithm is defined.

247 6.1.2 Create

248 This section describes how to implement the `create` verb when applied to an instance of CIM_Account.
249 Implementations may support the use of the `create` verb with CIM_Account.

250 The `create` verb is used to create an account on a system.

251 6.1.2.1 Create Specifying Properties

252 6.1.2.1.1 Command Form

```
253 create <CIM_Account> <propertyname1>=<propertyvalue1> <propertynamen>=<propertyvaluen>
```

254 6.1.2.1.2 CIM Requirements

255 See CIM_AccountManagementService.CreateAccount in the “CIM Elements” section of the [Simple](#)
256 [Identify Management Profile](#) for the list of modifiable properties.

257 6.1.2.1.3 Behavior Requirements

258 6.1.2.1.3.1 Preconditions

259 `$system` contains the instance of CIM_ComputerSystem that is the Resultant Target of the CLP
260 command. This is the container instance for the desired Account instance.

261 6.1.2.1.3.2 Pseudo Code

```
262 //1 Find the Service for the system to create the account
263 #Error = &smOpAssociators(
264     $system.getObjectPath(),
265     "CIM_HostedService",
266     "CIM_AccountManagementService",
267     NULL,
268     NULL,
269     NULL,
270     $SvcInstancePaths[])
271 if (0 != #Error.code)
272 {
273     &smProcessOpError (#Error);
274     //includes &smEnd;
275 }
276 $service = $SvcInstancePaths[0];
277 //2 populate the properties
278 $accountTemplate = smNewInstance ("CIM_Account");
279 for #i < n
280 {
281     $accountTemplate.<propertyname#i> = $accountTemplate.<propertyvalue#i>;
282 }
```

```
283 //3 invoke the method
284 $account = smNewInstance ("CIM_Account");
285 //build the parameter lists and invoke the method
286 %InArguments[] = {newArgument("AccountTemplate", $accountTemplate)}
287 %OutArguments[] = { newArgument("Account", $account.GetObjectPath()) };
288 //invoke method
289 #returnStatus = smOpInvokeMethod ($service.GetObjectPath(),
290     "CreateAccount",
291     %InArguments[],
292     %OutArguments[]);
293 //3 process return code to CLP Command Status
294 if (0 != #Error.code)
295     {
296     //method invocation failed
297     if ( (null != #Error.$error) && (null != #Error.$error[0]) )    {
298         // if the method invocation contains an embedded error
299         // use it for the Error for the overall job
300         &smAddError($job, #Error.$error[0]);
301         &smMakeCommandStatus($job);
302         &smEnd;
303     }
304     else if (#Error.code == 17)
305     {
306         //trap for CIM_METHOD_NOT_FOUND
307         //and make nice Unsupported msg.
308         //unsupported
309         $OperationError = smNewInstance("CIM_Error");
310         //CIM_ERR_NOT_SUPPORTED
311         $OperationError.CIMStatusCode = 7;
312         //Other
313         $OperationError.ErrorType = 1;
314         //Low
315         $OperationError.PerceivedSeverity = 2;
316         $OperationError.OwningEntity = DMTF:SMCLP;
317         $OperationError.MessageID = 0x00000001;
318         $OperationError.Message = "Operation is not supported.";
319         &smAddError($job, $OperationError);
320         &smMakeCommandStatus($job);
321         &smEnd;
322     }
323     else
324     {
325         //operation failed, but no detailed error instance, need to make one up
326         //make an Error instance and associate with job for Operation
327         $OperationError = smNewInstance("CIM_Error");
328         //CIM_ERR_FAILED
329         $OperationError.CIMStatusCode = 1;
330         //Software Error
331         $OperationError.ErrorType = 4;
```

```
332         //Unknown
333         $OperationError.PerceivedSeverity = 0;
334         $OperationError.OwningEntity = DMTF:SMCLP;
335         $OperationError.MessageID = 0x00000009;
336         $OperationError.Message = "An internal software error has occurred.";
337         &smAddError($job, $OperationError);
338         &smMakeCommandStatus($job);
339         &smEnd;
340     }
341 }//if CIM op failed
342 else if (0 == #returnStatus)
343     {
344         //completed successfully, show created instance
345         &lShowAccount($account, "false");
346         &smEnd;
347     }
348 else if (4 == #returnStatus)
349     {
350         //generic failure
351         $OperationError = smNewInstance("CIM_Error");
352         //CIM_ERR_FAILED
353         $OperationError.CIMStatusCode = 1;
354         //Other
355         $OperationError.ErrorType = 1;
356         //Low
357         $OperationError.PerceivedSeverity = 2;
358         $OperationError.OwningEntity = DMTF:SMCLP;
359         $OperationError.MessageID = 0x00000002;
360         $OperationError.Message = "Failed. No further information is available.";
361         &smAddError($job, $OperationError);
362         &smMakeCommandStatus($job);
363     }
364 else
365     {
366         //invalid parameter
367         $OperationError = smNewInstance("CIM_Error");
368         //CIM_ERR_FAILED
369         $OperationError.CIMStatusCode = 1;
370         //Other
371         $OperationError.ErrorType = 1;
372         //Low
373         $OperationError.PerceivedSeverity = 2;
374         $OperationError.OwningEntity = DMTF:SMCLP;
375         $OperationError.MessageID = 0x00000004;
376         $OperationError.Message = "One or more parameters specified are invalid.";
377         &smAddError($job, $OperationError);
378         &smMakeCommandStatus($job);
379         &smEnd;
380     }
```

381 6.1.3 Delete

382 This section describes how to implement the `delete` verb when applied to an instance of `CIM_Account`.
 383 Implementations may support the use of the `delete` verb with `CIM_Account`.

384 The `delete` command is used to remove an instance of `CIM_Account`.

385 6.1.3.1 Delete a Single Instance

386 Use the `delete` command to delete a single instance of `CIM_Account`.

387 6.1.3.1.1 Command Form

```
388 delete <CIM_Account single instance>
```

389 6.1.3.1.2 CIM Requirements

390 See `CIM_Account` in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of
 391 modifiable properties.

392 6.1.3.1.3 Behavior Requirements

```
393 $instance=<CIM_Account single instance>  
394 &smOpDeleteInstance ( $instance.GetObjectPath() );
```

395 6.1.4 Set

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

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

399 6.1.4.1 Set Account to Enabled but Offline

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

401 6.1.4.1.1 Command Form

```
402 set <CIM_Account single instance> RequestedState="Offline"
```

403 6.1.4.1.2 CIM Requirements

```
404 uint16 EnabledState;  
405 uint16 RequestedState;  
406 uint32 EnabledLogicalElement.RequestStateChange (  
407     [IN] uint16 RequestedState = "Offline",  
408     [OUT] REF CIM_ConcreteJob Job,  
409     [IN] datetime TimeoutPeriod );
```

410 6.1.4.1.3 Behavior Requirements

```
411 $instance=<CIM_Account single instance>  
412 //6 = Enabled but Offline  
413 smRequestStateChange ( $instance.GetObjectPath(), 6 );  
414 &smEnd;
```

415 6.1.4.2 General Usage of Set for a Single Property

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

418 The requirement for supporting modification of a property using this command form shall be equivalent to
419 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined
420 in the [Simple Identity Management Profile](#).

421 6.1.4.2.1 Command Form

```
422 set <CIM_Account single instance> <propertyname>=<propertyvalue>
```

423 6.1.4.2.2 CIM Requirements

424 See `CIM_Account` in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of
425 modifiable properties.

426 6.1.4.2.3 Behavior Requirements

```
427 $instance=<CIM_Account single instance>
428 #propertyNames[] = {<propertyname>};
429 #propertyValues[] = {<propertyvalue>};
430 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
431 &smEnd;
```

432 6.1.4.3 General Usage of Set for Multiple Properties

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

436 The requirement for supporting modification of a property using this command form shall be equivalent to
437 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined
438 in the [Simple Identity Management Profile](#).

439 6.1.4.3.1 Command Form

```
440 set <CIM_Account single instance> <propertyname1>=<propertyvalue1>
441 <propertynamen>=<propertyvaluen>
```

442 6.1.4.3.2 CIM Requirements

443 See `CIM_Account` in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of
444 modifiable properties.

445 6.1.4.3.3 Behavior Requirements

```
446 $instance=<CIM_Account single instance>
447 #propertyNames[] = {<propertyname>};
448 for #i < n
449 {
450     #propertyNames[#i] = <propertyname#i>
451     #propertyValues[#i] = <propertyvalue#i>
452 }
453 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
454 &smEnd;
```

455 6.1.5 Show

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

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

459 6.1.5.1 Show a Single Instance

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

461 6.1.5.1.1 Command Form

```
462 show <CIM_Account single instance>
```

463 6.1.5.1.2 CIM Requirements

464 See `CIM_Account` in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of
465 mandatory properties.

466 6.1.5.1.3 Behavior Requirements

467 6.1.5.1.3.1 Preconditions

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

469 6.1.5.1.3.2 Pseudo Code

```
470 $instance=<CIM_Account single instance>  
471 #propertylist[] = NULL;  
472 if (false == #all)  
473     {  
474         #propertylist[] = { //all mandatory non-key properties };  
475     }  
476 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );  
477 &smEnd;
```

478 6.1.5.2 Show Multiple Instances

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

481 6.1.5.2.1 Command Form

```
482 show <CIM_Account multiple instances>
```

483 6.1.5.2.2 CIM Requirements

484 See `CIM_Account` in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of
485 mandatory properties.

486 6.1.5.2.3 Behavior Requirements

487 6.1.5.2.3.1 Preconditions

488 `$containerInstance` contains the instance of `CIM_ComputerSystem` for which scoped accounts
489 (`CIM_Account` instances) are displayed. The [Simple Identity Management Profile](#) requires that the

490 CIM_Account instance be associated with its scoping system via an instance of the
491 CIM_AccountOnSystem association.

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

493 **6.1.5.2.3.2 Pseudo Code**

```
494 #propertylist[] = NULL;
495 if (false == #all)
496     {
497         #propertylist[] = { //all mandatory non-key properties };
498     }
499 &smShowInstances ( "CIM_Account", "CIM_AccountOnSystem",
500     $containerInstance.getObjectPath(), #propertylist[] );
501 &smEnd;
```

502 **6.1.6 Start**

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

505 The `start` verb is used to enable an account.

506 **6.1.6.1 Start a Single Instance**

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

508 **6.1.6.1.1 Command Form**

```
509 start <CIM_Account single instance>
```

510 **6.1.6.1.2 CIM Requirements**

```
511 uint16 EnabledState;
512 uint16 RequestedState;
513 //2 (Enabled)
514 uint32 EnabledLogicalElement.RequestStateChange (
515     [IN] uint16 RequestedState = 2,
516     [OUT] REF CIM_ConcreteJob Job,
517     [IN] datetime TimeoutPeriod );
```

518 **6.1.6.1.3 Behavior Requirements**

```
519 $instance=<CIM_Account single instance>
520 smStartRSC ( $instance.getObjectPath() );
521 &smEnd;
```

522 **6.1.7 Stop**

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

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

526 **6.1.7.1 Stop a Single Instance**

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

528 **6.1.7.1.1 Command Form**529

```
stop <CIM_Account single instance>
```

530 **6.1.7.1.2 CIM Requirements**

```
531 uint16 EnabledState;
532 uint16 RequestedState;
533 //3 (Disabled)
534 uint32 EnabledLogicalElement.RequestStateChange (
535     [IN] uint16 RequestedState = 3,
536     [OUT] REF CIM_ConcreteJob Job,
537     [IN] datetime TimeoutPeriod );
```

538 **6.1.7.1.3 Behavior Requirements**

```
539 $instance=<CIM_Account single instance>
540 smStopRSC ( $instance.getObjectPath() );
541 &smEnd;
```

542 **6.2 CIM_AccountManagementCapabilities**543 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

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

549 **Table 3 – Command Verb Requirements for CIM_AccountManagementCapabilities**

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	

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

552 **6.2.1 Ordering of Results**

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

- 555 • Results for `CIM_AccountManagementCapabilities` are unordered; therefore, no algorithm is
 556 defined.

557 **6.2.2 Show**

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

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

563 **6.2.2.1 Show a Single Instance**

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

566 **6.2.2.1.1 Command Form**

```
567 show <CIM_AccountManagementCapabilities single instance>
```

568 **6.2.2.1.2 CIM Requirements**

569 See `CIM_AccountManagementCapabilities` in the “CIM Elements” section of the [Simple Identify](#)
 570 [Management Profile](#) for the list of mandatory properties.

571 **6.2.2.1.3 Behavior Requirements**

572 **6.2.2.1.3.1 Preconditions**

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

574 **6.2.2.1.3.2 Pseudo Code**

```
575 $instance=<CIM_AccountManagementCapabilities 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 **6.2.2.2 Show Multiple Instances**

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

587 **6.2.2.2.1 Command Form**

```
588 show <CIM_AccountManagementCapabilities multiple instances>
```

589 **6.2.2.2.2 CIM Requirements**

590 See `CIM_AccountManagementCapabilities` in the “CIM Elements” section of the [Simple Identify](#)
 591 [Management Profile](#) for the list of mandatory properties.

592 **6.2.2.2.3 Behavior Requirements**593 **6.2.2.2.3.1 Preconditions**

594 \$containerInstance contains the instance of CIM_Group for which 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 **6.2.2.2.3.2 Pseudo Code**

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

607 **6.3 CIM_AccountManagementService**

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

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

614 **Table 4 – Command Verb Requirements for CIM_AccountManagementService**

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

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

616 **6.3.1 Ordering of Results**

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

- 619 • Results for CIM_AccountManagementService are unordered; therefore, no algorithm is defined.

620 **6.3.2 Show**

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

624 The `show` verb is used to display information about the `CIM_AccountManagementService`.

625 **6.3.2.1 Show a Single Instance**

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

628 **6.3.2.1.1 Command Form**

```
629 show <CIM_AccountManagementService single instance>
```

630 **6.3.2.1.2 CIM Requirements**

631 See `CIM_AccountManagementService` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
 632 for the list of mandatory properties.

633 **6.3.2.1.3 Behavior Requirements**

634 **6.3.2.1.3.1 Preconditions**

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

636 **6.3.2.1.3.2 Pseudo Code**

```
637 $instance=<CIM_AccountManagementService single instance>  

  638 #propertylist[] = NULL;  

  639 if (false == #all)  

  640 {  

  641     #propertylist[] = { "ElementName" };  

  642 }  

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

  644 &smEnd;
```

645 **6.3.2.2 Show Multiple Instances**

646 This command form is for the `show` verb applied to multiple instances of
 647 `CIM_AccountManagementService`. This command form corresponds to UFT-based selection within a
 648 scoping system.

649 **6.3.2.2.1 Command Form**

```
650 show <CIM_AccountManagementService multiple instances>
```

651 **6.3.2.2.2 CIM Requirements**

652 See `CIM_AccountManagementService` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
 653 for the list of mandatory properties.

654 **6.3.2.2.3 Behavior Requirements**655 **6.3.2.2.3.1 Preconditions**

656 \$containerInstance contains the instance of CIM_ComputerSystem for which scoped instances of
 657 the CIM_AccountManagementService are displayed. The [Simple Identity Management Profile](#) requires
 658 that the CIM_AccountManagementService instance be associated with its scoping system via an instance
 659 of the CIM_HostedService association.

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

661 **6.3.2.2.3.2 Pseudo Code**

```
662 #propertylist[] = NULL;
663 if (false == #all)
664     {
665         #propertylist[] = { "ElementName" };
666     }
667 &smShowInstances ( "CIM_AccountManagementService", "CIM_HostedService",
668     $containerInstance.getObjectPath(), #propertylist[] );
669 &smEnd;
```

670 **6.4 CIM_AccountOnSystem**

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

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

677 **Table 5 – Command Verb Requirements for CIM_AccountOnSystem**

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	

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

680 **6.4.1 Ordering of Results**

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

- 683 • Results for CIM_AccountOnSystem are unordered; therefore, no algorithm is defined.

684 **6.4.2 Show**

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

688 The `show` command is used to display information about the CIM_AccountOnSystem instance or
689 instances.

690 **6.4.2.1 Show Multiple Instances**

691 This command form is for the `show` verb applied to multiple instances. This command form corresponds
692 to a `show` command issued against CIM_AccountOnSystem where only one reference is specified and
693 the reference is to an instance of CIM_ComputerSystem.

694 **6.4.2.1.1 Command Form**

```
695 show <CIM_AccountOnSystem multiple instances>
```

696 **6.4.2.1.2 CIM Requirements**

697 See CIM_AccountOnSystem in the “CIM Elements” section of the [Simple Identify Management Profile](#) for
698 the list of mandatory properties.

699 **6.4.2.1.3 Behavior Requirements**

700 **6.4.2.1.3.1 Preconditions**

701 `$instance` contains the instance of CIM_ComputerSystem which is referenced by
702 CIM_AccountOnSystem.

703 **6.4.2.1.3.2 Pseudo Code**

```
704 &smShowAssociationInstances ( "CIM_AccountOnSystem", $instance.getObjectPath() );  
705 &smEnd;
```

706 **6.4.2.2 Show a Single Instance – CIM_Account Reference**

707 This command form is for the `show` verb applied to a single instance. This command form corresponds to
708 a `show` command issued against CIM_AccountOnSystem where the reference specified is to an instance
709 of CIM_Account. An instance is referenced by exactly one instance of CIM_AccountOnSystem.
710 Therefore, a single instance will be returned.

711 **6.4.2.2.1 Command Form**

```
712 show <CIM_AccountOnSystem single instance>
```

713 **6.4.2.2.2 CIM Requirements**

714 See CIM_AccountOnSystem in the “CIM Elements” section of the [Simple Identify Management Profile](#) for
715 the list of mandatory properties.

716 **6.4.2.2.3 Behavior Requirements**717 **6.4.2.2.3.1 Preconditions**

718 \$instance contains the instance of CIM_Account which is referenced by CIM_AccountOnSystem.

719 **6.4.2.2.3.2 Pseudo Code**

```
720 &smShowAssociationInstances ( "CIM_AccountOnSystem", $instance.GetObjectPath() );
721 &smEnd;
```

722 **6.4.2.3 Show a Single Instance – Both References**

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

726 **6.4.2.3.1 Command Form**

```
727 show <CIM_AccountOnSystem single instance>
```

728 **6.4.2.3.2 CIM Requirements**

729 See CIM_AccountOnSystem in the “CIM Elements” section of the [Simple Identify Management Profile](#) for
730 the list of mandatory properties.

731 **6.4.2.3.3 Behavior Requirements**732 **6.4.2.3.3.1 Preconditions**

733 \$instanceA contains the instance of CIM_ComputerSystem which is referenced by
734 CIM_AccountOnSystem.

735 \$instanceB contains the instance of CIM_Account which is referenced by CIM_AccountOnSystem.

736 **6.4.2.3.3.2 Pseudo Code**

```
737 &smShowAssociationInstance ( "CIM_AccountOnSystem", $instanceA.GetObjectPath(),
738     $instanceB.GetObjectPath() );
739 &smEnd;
```

740 **6.5 CIM_AccountSettingData**

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

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

747 **Table 6 – Command Verb Requirements for CIM_AccountSettingData**

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

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

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

750 6.5.1 Ordering of Results

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

- 753 • Results for *CIM_AccountSettingData* are unordered; therefore, no algorithm is defined.

754 6.5.2 Set

755 This section describes how to implement the *set* verb when it is applied to an instance of
756 *CIM_AccountSettingData*. Implementations may support the use of the *set* verb with
757 *CIM_AccountSettingData*.

758 The *set* verb is used to modify descriptive properties of the *CIM_AccountSettingData* instance.

759 6.5.2.1 General Usage of Set for a Single Property

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

762 The requirement for supporting modification of a property using this command form shall be equivalent to
763 the requirement for supporting modification of the property using the *ModifyInstance* operation as defined
764 in the [Simple Identity Management Profile](#).

765 6.5.2.1.1 Command Form

```
766 set <CIM_AccountSettingData single instance> <propertyname>=<propertyvalue>
```

767 6.5.2.1.2 CIM Requirements

768 See *CIM_AccountSettingData* in the “CIM Elements” section of the [Simple Identify Management Profile](#)
769 for the list of modifiable properties.

770 6.5.2.1.3 Behavior Requirements

```
771 $instance=<CIM_AccountSettingData single instance>
772 #propertyName[] = {<propertyname>};
773 #propertyValues[] = {<propertyvalue>};
774 &smSetInstance ( $instance, #propertyName[], #propertyValues[] );
775 &smEnd;
```

776 6.5.2.2 General Usage of Set for Multiple Properties

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

780 The requirement for supporting modification of a property using this command form shall be equivalent to
781 the requirement for supporting modification of the property using the `ModifyInstance` operation as defined
782 in the [Simple Identity Management Profile](#).

783 6.5.2.2.1 Command Form

```
784 set <CIM_AccountSettingData single instance> <propertyname1>=<propertyvalue1>  
785 <propertynamen>=<propertyvaluen>
```

786 6.5.2.2.2 CIM Requirements

787 See `CIM_AccountSettingData` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
788 for the list of mandatory properties.

789 6.5.2.2.3 Behavior Requirements

```
790 $instance=<CIM_AccountSettingData single instance>  
791 #propertyNames[] = {<propertyname>};  
792 for #i < n  
793 {  
794     #propertyNames[#i] = <propertname#i>  
795     #propertyValues[#i] = <propertyvalue#i>  
796 }  
797 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );  
798 &smEnd;
```

799 6.5.3 Show

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

803 The `show` verb is used to display information about `CIM_AccountSettingData`.

804 6.5.3.1 Show a Single Instance

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

806 6.5.3.1.1 Command Form

```
807 show <CIM_AccountSettingData single object>
```

808 6.5.3.1.2 CIM Requirements

809 See `CIM_AccountSettingData` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
810 for the list of mandatory properties.

811 6.5.3.1.3 Behavior Requirements

812 6.5.3.1.3.1 Preconditions

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

814 6.5.3.1.3.2 Pseudo Code

```

815 $instance=<CIM_AccountSettingData single object>
816 #propertylist[] = NULL;
817 if (false == #all)
818     {
819         #propertylist[] = { "ElementName", "AddressOrigin" }
820     }
821 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
822 &smEnd;

```

823 6.5.3.2 Show Multiple Instances

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

826 6.5.3.2.1 Command Form

```

827 show <CIM_AccountSettingData multiple objects>

```

828 6.5.3.2.2 CIM Requirements

829 See `CIM_AccountSettingData` in the “CIM Elements” section of the [Simple Identity Management Profile](#)
 830 for the list of mandatory properties.

831 6.5.3.2.3 Behavior Requirements

832 6.5.3.2.3.1 Preconditions

833 `$containerInstance` contains the instance of `CIM_ConcreteCollection` for which related
 834 `CIM_AccountSettingData` instances are displayed. SM ME Addressing requires that the
 835 `CIM_AccountSettingData` instance be associated with an instance of `CIM_ConcreteCollection` via an
 836 instance of the `CIM_MemberOfCollection` association.

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

838 6.5.3.2.3.2 Pseudo Code

```

839 #propertylist[] = NULL;
840 if (false == #all)
841     {
842         #propertylist[] = { "ElementName", "AddressOrigin" }
843     }
844 &smShowInstances ( "CIM_AccountSettingData", "CIM_OrderedComponent",
845     $containerInstance.getObjectPath(), #propertylist[] );
846 &smEnd;

```

847 6.6 CIM_AssignedIdentity

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

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

854

Table 7 – Command Verb Requirements for CIM_AssignedIdentity

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	

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

857 6.6.1 Ordering of Results

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

- 860 • Results for CIM_AssignedIdentity are unordered; therefore, no algorithm is defined.

861 6.6.2 Show

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

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

866 6.6.2.1 Show Multiple Instances – CIM_Identity Reference

867 This command form is for the `show` verb applied to multiple instances. This command form corresponds
868 to a `show` command issued against CIM_AssignedIdentity where only one reference is specified and the
869 reference is to an instance of CIM_Identity.

870 6.6.2.1.1 Command Form

```
871 show <CIM_AssignedIdentity multiple instances>
```

872 6.6.2.1.2 CIM Requirements

873 See CIM_AssignedIdentity in the “CIM Elements” section of the [Simple Identity Management Profile](#) for
874 the list of mandatory properties.

875 6.6.2.1.3 Behavior Requirements

876 6.6.2.1.3.1 Preconditions

877 \$instance contains the instance of CIM_Identity that is referenced by CIM_AssignedIdentity.

878 6.6.2.1.3.2 Pseudo Code

```
879 &smShowAssociationInstances ( "CIM_AssignedIdentity", $instance.getObjectPath() );  
880 &smEnd;
```

881 6.6.2.2 Show Multiple Instances – CIM_Account reference

882 This command form is for the `show` verb applied to multiple instances. This command form corresponds
883 to a `show` command issued against `CIM_AssignedIdentity` where only one reference is specified and the
884 reference is to an instance of `CIM_Account`.

885 6.6.2.2.1 Command Form

```
886 show <CIM_AssignedIdentity multiple instances>
```

887 6.6.2.2.2 CIM Requirements

888 See `CIM_AssignedIdentity` in the “CIM Elements” section of the [Simple Identify Management Profile](#) for
889 the list of mandatory properties.

890 6.6.2.2.3 Behavior Requirements**891 6.6.2.2.3.1 Preconditions**

892 `$instance` contains the instance of `CIM_Account` that is referenced by `CIM_AssignedIdentity`.

893 6.6.2.2.3.2 Pseudo Code

```
894 &smShowAssociationInstances ( "CIM_AssignedIdentity", $instance.getObjectPath() );  
895 &smEnd;
```

896 6.6.2.3 Show a Single Instance – CIM_UserContact or CIM_Group Reference

897 This command form is for the `show` verb applied to a single instance. This command form corresponds to
898 a `show` command issued against `CIM_AssignedIdentity` where the reference specified is to an instance of
899 `CIM_UserContact` or `CIM_Group`. A single instance of `CIM_UserContact` or `CIM_Group` can be
900 associated with each instance of a `CIM_Identity` sub-class. Therefore, a single instance will be returned.

901 6.6.2.3.1 Command Form

```
902 show <CIM_AssignedIdentity single instance>
```

903 6.6.2.3.2 CIM Requirements

904 See `CIM_AssignedIdentity` in the “CIM Elements” section of the [Simple Identify Management Profile](#) for
905 the list of mandatory properties.

906 6.6.2.3.3 Behavior Requirements**907 6.6.2.3.3.1 Preconditions**

908 `$instance` contains the instance of `CIM_UserContact` or `CIM_Group` which is referenced by
909 `CIM_AssignedIdentity`.

910 6.6.2.3.3.2 Pseudo Code

```
911 &smShowAssociationInstances ( "CIM_AssignedIdentity", $instance.getObjectPath() );  
912 &smEnd;
```

913 6.6.2.4 Show a Single Instance – Both References

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

917 6.6.2.4.1 Command Form

```
918 show <CIM_AssignedIdentity single instance>
```

919 6.6.2.4.2 CIM Requirements

920 See `CIM_AssignedIdentity` in the “CIM Elements” section of the [Simple Identity Management Profile](#) for
 921 the list of mandatory properties.

922 6.6.2.4.3 Behavior Requirements

923 6.6.2.4.3.1 Preconditions

924 `$instanceA` contains the instance of `CIM_Identity` which is referenced by `CIM_AssignedIdentity`.

925 `$instanceB` contains the instance of `CIM_UserContact`, `CIM_Group`, or `CIM_Account` which is
 926 referenced by `CIM_AssignedIdentity`.

927 6.6.2.4.3.2 Pseudo Code

```
928 &smShowAssociationInstance ( "CIM_AssignedIdentity", $instanceA.getObjectPath(),
929     $instanceB.getObjectPath() );
930 &smEnd;
```

931 6.7 CIM_ConcreteDependency

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

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

938 **Table 8 – Command Verb Requirements for CIM_ConcreteDependency**

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

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

941 6.7.1 Ordering of Results

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

- 944 • Results for CIM_ConcreteDependency are unordered; therefore, no algorithm is defined.

945 6.7.2 Show

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

949 The `show` command is used to display information about the CIM_ConcreteDependency instance or
950 instances.

951 6.7.2.1 Show Multiple Instances – CIM_ManagedElement reference

952 This command form is for the `show` verb applied to multiple instances. This command form corresponds
953 to a `show` command issued against CIM_ConcreteDependency where only one reference is specified and
954 the reference is to an instance of CIM_ManagedElement.

955 6.7.2.1.1 Command Form

```
956 show <CIM_ConcreteDependency multiple instances>
```

957 6.7.2.1.2 CIM Requirements

958 See CIM_ConcreteDependency in the “CIM Elements” section of the [Simple Identify Management Profile](#)
959 for the list of mandatory properties.

960 6.7.2.1.3 Behavior Requirements

961 6.7.2.1.3.1 Preconditions

962 `$instance` contains the instance of CIM_ManagedElement that is referenced by
963 CIM_ConcreteDependency.

964 6.7.2.1.3.2 Pseudo Code

```
965 &smShowAssociationInstances ( "CIM_ConcreteDependency", $instance.getObjectPath() );  
966 &smEnd;
```

967 6.7.2.2 Show a Single Instance – CIM_Identity Reference

968 This command form is for the `show` verb applied to a single instance. This command form corresponds to
969 a `show` command issued against CIM_ConcreteDependency where the reference specified is to an
970 instance of CIM_Identity.

971 6.7.2.2.1 Command Form

```
972 show <CIM_ConcreteDependency single instance>
```

973 6.7.2.2.2 CIM Requirements

974 See CIM_ConcreteDependency in the “CIM Elements” section of the [Simple Identify Management Profile](#)
975 for the list of mandatory properties.

976 **6.7.2.2.3 Behavior Requirements**977 **6.7.2.2.3.1 Preconditions**

978 `$instance` contains the instance of `CIM_Identity` which is referenced by `CIM_ConcreteDependency`.

979 **6.7.2.2.3.2 Pseudo Code**

```
980 &smShowAssociationInstances ( "CIM_ConcreteDependency", $instance.GetObjectPath() );
981 &smEnd;
```

982 **6.7.2.3 Show a Single Instance – Both References**

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

986 **6.7.2.3.1 Command Form**

```
987 show <CIM_ConcreteDependency single instance>
```

988 **6.7.2.3.2 CIM Requirements**

989 See `CIM_ConcreteDependency` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
 990 for the list of mandatory properties.

991 **6.7.2.3.3 Behavior Requirements**992 **6.7.2.3.3.1 Preconditions**

993 `$instanceA` contains the instance of `CIM_Identity` which is referenced by `CIM_ConcreteDependency`.

994 `$instanceB` contains the instance of `CIM_ManagedElement` which is referenced by
 995 `CIM_ConcreteDependency`.

996 **6.7.2.3.3.2 Pseudo Code**

```
997 &smShowAssociationInstance ( "CIM_ConcreteDependency", $instanceA.GetObjectPath(),
998     $instanceB.GetObjectPath() );
999 &smEnd;
```

1000 **6.8 CIM_ElementCapabilities**

1001 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1002 Table 9 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of
 1003 the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
 1004 verb and target. Table 9 is for informational purposes only; in case of a conflict between Table 9 and
 1005 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1006 information in Table 9.

1007 **Table 9 – Command Verb Requirements for CIM_ElementCapabilities**

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 6.8.2.
start	Not supported	
stop	Not supported	

1008 No mapping is defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,
1009 `reset`, `set`, `start`, and `stop`.

1010 6.8.1 Ordering of Results

1011 When results are returned for multiple instances of `CIM_ElementCapabilities`, implementations shall
1012 utilize the following algorithm to produce the natural (that is, default) ordering:

- 1013 • Results for `CIM_ElementCapabilities` are unordered; therefore, no algorithm is defined.

1014 6.8.2 Show

1015 This section describes how to implement the `show` verb when applied to an instance of
1016 `CIM_ElementCapabilities`. Implementations shall support the use of the `show` verb with
1017 `CIM_ElementCapabilities`.

1018 The `show` command is used to display information about the `CIM_ElementCapabilities` instance or
1019 instances.

1020 6.8.2.1 Show Multiple Instances – `CIM_EnabledLogicalElementCapabilities` Reference

1021 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1022 to a `show` command issued against `CIM_ElementCapabilities` where only one reference is specified and
1023 the reference is to an instance of `CIM_EnabledLogicalElementCapabilities`.

1024 6.8.2.1.1 Command Form

```
1025 show <CIM_ElementCapabilities multiple instances>
```

1026 6.8.2.1.2 CIM Requirements

1027 See `CIM_ElementCapabilities` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1028 for the list of mandatory properties.

1029 6.8.2.1.3 Behavior Requirements

1030 6.8.2.1.3.1 Preconditions

1031 `$instance` contains the instance of `CIM_EnabledLogicalElementCapabilities` which is referenced by
1032 `CIM_ElementCapabilities`.

1033 6.8.2.1.3.2 Pseudo Code

```
1034 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );  
1035 &smEnd;
```

1036 6.8.2.2 Show a Single Instance – CIM_Account Reference

1037 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1038 a `show` command issued against `CIM_ElementCapabilities` where the reference specified is to an
1039 instance of `CIM_Account`.

1040 6.8.2.2.1 Command Form

```
1041 show <CIM_ElementCapabilities single instance>
```

1042 6.8.2.2.2 CIM Requirements

1043 See `CIM_ElementCapabilities` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1044 for the list of mandatory properties.

1045 6.8.2.2.3 Behavior Requirements

1046 6.8.2.2.3.1 Preconditions

1047 `$instance` contains the instance of `CIM_Account`, which is referenced by `CIM_ElementCapabilities`.

1048 6.8.2.2.3.2 Pseudo Code

```
1049 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );  
1050 &smEnd;
```

1051 6.8.2.3 Show Multiple Instances – CIM_AccountManagementCapabilities Reference

1052 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1053 to a `show` command issued against `CIM_ElementCapabilities` where only one reference is specified and
1054 the reference is to an instance of `CIM_EnabledLogicalElementCapabilities`.

1055 6.8.2.3.1 Command Form

```
1056 show <CIM_ElementCapabilities multiple instances>
```

1057 6.8.2.3.2 CIM Requirements

1058 See `CIM_ElementCapabilities` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1059 for the list of mandatory properties.

1060 6.8.2.3.3 Behavior Requirements

1061 6.8.2.3.3.1 Preconditions

1062 `$instance` contains the instance of `CIM_AccountManagementCapabilities` which is referenced by
1063 `CIM_ElementCapabilities`.

1064 6.8.2.3.3.2 Pseudo Code

```
1065 &smShowAssociationInstances ( "CIM_ElementCapabilities", $instance.getObjectPath() );  
1066 &smEnd;
```

1067 6.8.2.4 Show a Single Instance – CIM_AccountManagementService Reference

1068 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1069 a `show` command issued against `CIM_ElementCapabilities` where the reference specified is to an
1070 instance of `CIM_Account`.

1071 **6.8.2.4.1 Command Form**1072 `show <CIM_ElementCapabilities single instance>`1073 **6.8.2.4.2 CIM Requirements**1074 See CIM_ElementCapabilities in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1075 for the list of mandatory properties.1076 **6.8.2.4.3 Behavior Requirements**1077 **6.8.2.4.3.1 Preconditions**1078 \$instance contains the instance of CIM_AccountManagementService which is referenced by
1079 CIM_ElementCapabilities.1080 **6.8.2.4.3.2 Pseudo Code**1081 `&smShowAssociationInstances ("CIM_ElementCapabilities", $instance.getObjectPath());`
1082 `&smEnd;`1083 **6.8.2.5 Show a Single Instance – CIM_EnabledLogicalElementCapabilities and CIM_Account**
1084 **References**1085 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1086 a `show` command issued against CIM_ElementCapabilities where both references are specified and
1087 therefore the desired instance is unambiguously identified.1088 **6.8.2.5.1 Command Form**1089 `show <CIM_ElementCapabilities single instance>`1090 **6.8.2.5.2 CIM Requirements**1091 See CIM_ElementCapabilities in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1092 for the list of mandatory properties.1093 **6.8.2.5.3 Behavior Requirements**1094 **6.8.2.5.3.1 Preconditions**1095 \$instanceA contains the instance of CIM_EnabledLogicalElementCapabilities which is referenced by
1096 CIM_ElementCapabilities.

1097 \$instanceB contains the instance of CIM_Account which is referenced by CIM_ElementCapabilities.

1098 **6.8.2.5.3.2 Pseudo Code**1099 `&smShowAssociationInstance ("CIM_ElementCapabilities", $instanceA.getObjectPath(),`
1100 `$instanceB.getObjectPath());`
1101 `&smEnd;`1102 **6.8.2.6 Show a Single Instance – CIM_AccountManagementCapabilities and**
1103 **CIM_AccountManagementService References**1104 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1105 a `show` command issued against CIM_ElementCapabilities where both references are specified and
1106 therefore the desired instance is unambiguously identified.

1107 **6.8.2.6.1 Command Form**1108 `show <CIM_ElementCapabilities single instance>`1109 **6.8.2.6.2 CIM Requirements**1110 See CIM_ElementCapabilities in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1111 for the list of mandatory properties.1112 **6.8.2.6.3 Behavior Requirements**1113 **6.8.2.6.3.1 Preconditions**1114 \$instanceA contains the instance of CIM_AccountManagementCapabilities which is referenced by
1115 CIM_ElementCapabilities.1116 \$instanceB contains the instance of CIM_AccountManagementService which is referenced by
1117 CIM_ElementCapabilities.1118 **6.8.2.6.3.2 Pseudo Code**1119 `&smShowAssociationInstance ("CIM_ElementCapabilities", $instanceA.getObjectPath(),`
1120 `$instanceB.getObjectPath());`
1121 `&smEnd;`1122 **6.9 CIM_ElementSettingData**1123 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).1124 Table 10 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
1125 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
1126 verb and target. Table 10 is for informational purposes only; in case of a conflict between Table 10 and
1127 requirements detailed in the following sections, the text detailed in the following sections supersedes the
1128 information in Table 10.1129 **Table 10 – Command Verb Requirements for CIM_ElementSettingData**

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

1130 No mappings are defined for the following verbs for the specified target: create, delete, dump, load,
1131 reset, set, start, and stop.

1132 6.9.1 Ordering of Results

1133 When results are returned for multiple instances of CIM_ElementSettingData, implementations shall
1134 utilize the following algorithm to produce the natural (that is, default) ordering:

- 1135 • Results for CIM_ElementSettingData are unordered; therefore, no algorithm is defined.

1136 6.9.2 Set

1137 This section describes how to implement the set verb when applied to an instance of
1138 CIM_ElementSettingData. Implementations shall support the use of the set verb for an instance of
1139 CIM_ElementSettingData which references an instance of CIM_AccountSettingData.

1140 6.9.2.1 Set IsNext Property

1141 This command form is for when the set verb is used to change the value of the IsNext property of an
1142 instance of CIM_ElementSettingData that associates an instance of CIM_AccountSettingData with an
1143 instance of CIM_AccountManagementService. The only valid input value is 1 (Is Next).

1144 6.9.2.1.1 Command Form

```
1145 set <CIM_ElementSettingData single instance> IsNextt=<propvalue>
```

1146 6.9.2.1.2 CIM Requirements

1147 See CIM_ElementSettingData, CIM_AccountSettingData, and CIM_AccountManagementService in the
1148 “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of mandatory properties.

1149 6.9.2.1.3 Behavior Requirements

```
1150 $instance=<CIM_ElementSettingData single instance>
1151 $instance=<CIM_AccountSettingData single instance>
1152 #propertyName[] = {<propertyname>};
1153 #propertyValues[] = {<propertyvalue>};
1154 &smSetInstance ( $instance, #propertyName[], #propertyValues[] );
1155 &smEnd;
```

1156 6.9.3 Show

1157 This section describes how to implement the show verb when applied to an instance of
1158 CIM_ElementSettingData. Implementations shall support the use of the show verb with
1159 CIM_ElementSettingData.

1160 The show command is used to display information about the CIM_ElementSettingData instance or
1161 instances.

1162 6.9.3.1 Show Multiple Instances – CIM_AccountManagementService

1163 This command form is for the show verb applied to multiple instances. This command form corresponds
1164 to a show command issued against CIM_ElementSettingData where only one reference is specified and
1165 the reference is to an instance of CIM_AccountManagementService.

1166 6.9.3.1.1 Command Form

```
1167 show <CIM_ElementSettingData multiple objects>
```

1168 **6.9.3.1.2 CIM Requirements**

1169 See CIM_ElementSettingData in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1170 for the list of mandatory properties.

1171 **6.9.3.1.3 Behavior Requirements**1172 **6.9.3.1.3.1 Preconditions**

1173 \$instance contains the instance of CIM_IPProtocolEndpoint which is referenced by
1174 CIM_ElementSettingData.

1175 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1176 **6.9.3.1.3.2 Pseudo Code**

```
1177 #propertylist[] = NULL;
1178 if (#all == false)
1179     {
1180         #propertylist[] = { "IsNext" };
1181     }
1182 &smShowAssociationInstances ( "CIM_ElementSettingData", $instance.getObjectPath(),
1183     #propertylist[] );
1184 &smEnd;
```

1185 **6.9.3.2 Show Multiple Instances – CIM_AccountSettingData Reference**

1186 This command form is for the show verb applied to multiple instances. This command form corresponds
1187 to a show command issued against CIM_ElementSettingData where only one reference is specified and
1188 the reference is to an instance of CIM_AccountSettingData.

1189 **6.9.3.2.1 Command Form**

```
1190 show <CIM_ElementSettingData multiple objects>
```

1191 **6.9.3.2.2 CIM Requirements**

1192 See CIM_ElementSettingData in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1193 for the list of mandatory properties.

1194 **6.9.3.2.3 Behavior Requirements**1195 **6.9.3.2.3.1 Preconditions**

1196 \$instance contains the instance of CIM_AccountSettingData which is referenced by
1197 CIM_ElementSettingData.

1198 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1199 **6.9.3.2.3.2 Pseudo Code**

```
1200 #propertylist[] = NULL;
1201 if (#all == false)
1202     {
1203         #propertylist[] = { "IsNext" };
1204     }
1205 &smShowAssociationInstances ( "CIM_ElementSettingData", $instance.getObjectPath(),
1206     #propertylist[] );
1207 &smEnd;
```

1208 **6.9.3.3 Show a Single Instance – Both References**

1209 This command form is for the `show` verb applied to a single instance. This command form corresponds to
 1210 a `show` command issued against `CIM_ElementSettingData` where both references are specified and
 1211 therefore the desired instance is unambiguously identified.

1212 **6.9.3.3.1 Command Form**

1213 `show <CIM_ElementSettingData single object>`

1214 **CIM Requirements**

1215 See `CIM_ElementSettingData` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
 1216 for the list of mandatory properties.

1217 **6.9.3.3.2 Behavior Requirements**

1218 **6.9.3.3.2.1 Preconditions**

1219 `$instanceA` contains the instance of `CIM_AccountManagementService` that is referenced by
 1220 `CIM_ElementSettingData`.

1221 `$instanceB` contains the instance of `CIM_AccountSettingData` referenced by `CIM_ElementSettingData`.

1222 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

1223 **6.9.3.3.2.2 Pseudo Code**

```
1224 #propertylist[] = NULL;
1225 if (#all == false)
1226     {
1227         #propertylist[] = { "IsNext" };
1228     }
1229 &smShowAssociationInstance ( "CIM_ElementSettingData", $instanceA.getObjectPath(),
1230     $instanceB.getObjectPath(), #propertylist[] );
1231 &smEnd;
```

1232 **6.10 CIM_EnabledLogicalElementCapabilities**

1233 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1234 Table 11 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
 1235 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
 1236 verb and target. Table 11 is for informational purposes only; in case of a conflict between Table 11 and
 1237 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1238 information in Table 11.

1239 **Table 11 – Command Verb Requirements for CIM_EnabledLogicalElementCapabilities**

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

Command Verb	Requirement	Comments
show	Shall	See 6.10.2.
start	Not supported	
stop	Not supported	

1240 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
1241 reset, set, start, and stop.

1242 6.10.1 Ordering of Results

1243 When results are returned for multiple instances of CIM_EnabledLogicalElementCapabilities,
1244 implementations shall utilize the following algorithm to produce the natural (that is, default) ordering:

- 1245 • Results for CIM_EnabledLogicalElementCapabilities are unordered; therefore, no algorithm is
1246 defined.

1247 6.10.2 Show

1248 This section describes how to implement the `show` verb when applied to an instance of
1249 CIM_EnabledLogicalElementCapabilities. Implementations shall support the use of the `show` verb with
1250 CIM_EnabledLogicalElementCapabilities.

1251 The `show` verb is used to display information about an instance or instances of the
1252 CIM_EnabledLogicalElementCapabilities class.

1253 6.10.2.1 Show a Single Instance

1254 This command form is for the `show` verb applied to a single instance of
1255 CIM_EnabledLogicalElementCapabilities.

1256 6.10.2.1.1 Command Form

```
1257 show <CIM_EnabledLogicalElementCapabilities single instance>
```

1258 6.10.2.1.2 CIM Requirements

1259 See CIM_EnabledLogicalElementCapabilities in the “CIM Elements” section of the [Simple Identify](#)
1260 [Management Profile](#) for the list of mandatory properties.

1261 6.10.2.1.3 Behavior Requirements

1262 6.10.2.1.3.1 Preconditions

1263 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1264 6.10.2.1.3.2 Pseudo Code

```
1265 $instance=<CIM_EnabledLogicalElementCapabilities single instance>
1266 #propertylist[] = NULL;
1267 if ( false == #all)
1268     {
1269         #propertylist[] = { //all mandatory non-key properties}
1270     }
1271 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
1272 &smEnd;
```


1273 **6.10.2.2 Show Multiple Instances**

1274 This command form is for the `show` verb applied to multiple instances of
 1275 `CIM_EnabledLogicalElementCapabilities`. This command form corresponds to UFS-T-based selection
 1276 within a capabilities collection.

1277 **6.10.2.2.1 Command Form**

```
1278 show <CIM_EnabledLogicalElementCapabilities multiple instances>
```

1279 **6.10.2.2.2 CIM Requirements**

1280 See `CIM_EnabledLogicalElementCapabilities` in the “CIM Elements” section of the [Simple Identify](#)
 1281 [Management Profile](#) for the list of mandatory properties.

1282 **6.10.2.2.3 Behavior Requirements**

1283 **6.10.2.2.3.1 Preconditions**

1284 `$containerInstance` contains the instance of `CIM_Group` for which contained `CIM_Capabilities`
 1285 instances are displayed. `CIM_Capabilities` instances are addressed via an aggregating instance of
 1286 `CIM_Group`.

1287 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

1288 **6.10.2.2.3.2 Pseudo Code**

```
1289 #propertylist[] = NULL;
1290 if ( false == #all)
1291 {
1292     #propertylist[] = { //all mandatory non-key properties }
1293 }
1294 &smShowInstances ( "CIM_EnabledLogicalElementCapabilities", "CIM_MemberOfCollection",
1295     $containerInstance.GetObjectPath(), #propertylist[] );
1296 &smEnd;
```

1297 **6.11 CIM_Group**

1298 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1299 Table 12 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
 1300 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and
 1301 target. Table 12 is for informational purposes only; in case of a conflict between Table 12 and
 1302 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1303 information in Table 12.

1304 **Table 12 – Command Verb Requirements for CIM_Group**

Command Verb	Requirement	Comments
create	Not supported	None
delete	Not supported	None
dump	Not supported	None
load	Not supported	None
reset	Not supported	None

Command Verb	Requirement	Comments
set	Not supported	None
show	Shall	See 6.11.2.
start	Not supported	None
stop	Not supported	None

1305 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
1306 reset, set, start, and stop.

1307 6.11.1 Ordering of Results

1308 When results are returned for multiple instances of CIM_Group, implementations shall utilize the following
1309 algorithm to produce the natural (that is, default) ordering:

- 1310 • Results for CIM_Group are unordered; therefore, no algorithm is defined.

1311 6.11.2 Show

1312 This section describes how to implement the `show` verb when applied to an instance of CIM_Group.
1313 Implementations shall support the use of the `show` verb with CIM_Group.

1314 6.11.2.1 Show a Single Instance

1315 Use this command to show a single instance of CIM_Group.

1316 6.11.2.1.1 Command Form

```
1317 show <CIM_Group single instance>
```

1318 6.11.2.1.2 CIM Requirements

1319 See CIM_Group in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of
1320 mandatory properties.

1321 6.11.2.1.3 Behavior Requirements

1322 6.11.2.1.3.1 Preconditions

1323 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1324 6.11.2.1.3.2 Pseudo Code

```
1325 $instance=<CIM_Group single instance>
1326 #propertylist[] = NULL;
1327 if (false == #all)
1328     {
1329         #propertylist[] = { //all mandatory non-key properties }
1330     }
1331 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
1332 &smEnd;
```

1333 6.11.2.2 Show Multiple Instances

1334 Use this command to show multiple instances of CIM_Group.

1335 **6.11.2.2.1 Command Form**

1336 `show <CIM_Group multiple instances>`

1337 **6.11.2.2.2 CIM Requirements**

1338 See CIM_Group in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of
 1339 mandatory properties.

1340 **6.11.2.2.3 Behavior Requirements**

1341 **6.11.2.2.3.1 Preconditions**

1342 \$containerInstance contains the instance of CIM_ComputerSystem for which related CIM_Group
 1343 instances are displayed.

1344 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1345 **6.11.2.2.3.2 Pseudo Code**

```

1346 #propertylist[] = NULL;
1347 if (false == #all)
1348     {
1349         #propertylist[] = { //all mandatory non-key properties };
1350     }
1351 &smShowInstances ( "CIM_Group", "CIM_OwningCollectionElement",
1352     $containerInstance.getObjectPath(), #propertylist[] );
1353 &smEnd;
```

1354 **6.12 CIM_HostedService**

1355 The cd and help verbs shall be supported as described in [DSP0216](#).

1356 Table 13 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
 1357 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
 1358 verb and target. Table 13 is for informational purposes only; in case of a conflict between Table 13 and
 1359 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1360 information in Table 13.

1361 **Table 13 – Command Verb Requirements for CIM_HostedService**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.12.2.
start	Not supported	
stop	Not supported	

1362 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
 1363 reset, set, start, and stop.

1364 6.12.1 Ordering of Results

1365 When results are returned for multiple instances of CIM_HostedService, implementations shall utilize the
1366 following algorithm to produce the natural (that is, default) ordering:

- 1367 • Results for CIM_HostedService are unordered; therefore, no algorithm is defined.

1368 6.12.2 Show

1369 This section describes how to implement the `show` verb when applied to an instance of
1370 CIM_HostedService. Implementations shall support the use of the `show` verb with CIM_HostedService.

1371 The `show` command is used to display information about the CIM_HostedService instance or instances.

1372 6.12.2.1 Show Multiple Instances – CIM_ComputerSystem Reference

1373 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1374 to a `show` command issued against CIM_HostedService where only one reference is specified and the
1375 reference is to an instance of CIM_ComputerSystem.

1376 6.12.2.1.1 Command Form

```
1377 show <CIM_HostedService multiple instances>
```

1378 6.12.2.1.2 CIM Requirements

1379 See CIM_HostedService in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the
1380 list of mandatory properties.

1381 6.12.2.1.3 Behavior Requirements

1382 6.12.2.1.3.1 Preconditions

1383 `$instance` contains the instance of CIM_ComputerSystem which is referenced by CIM_HostedService.

1384 6.12.2.1.3.2 Pseudo Code

```
1385 &smShowAssociationInstances ( "CIM_HostedService", $instance.GetObjectPath() );  
1386 &smEnd;
```

1387 6.12.2.2 Show a Single Instance – CIM_AccountManagementService Reference

1388 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1389 a `show` command issued against CIM_HostedService where the reference specified is to an instance of
1390 CIM_AccountManagementService. An instance of CIM_AccountManagementService is referenced by
1391 exactly one instance of CIM_HostedService. Therefore, a single instance will be returned.

1392 6.12.2.2.1 Command Form

```
1393 show <CIM_HostedService single instance>
```

1394 6.12.2.2.2 CIM Requirements

1395 See CIM_HostedService in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the
1396 list of mandatory properties.

1397 6.12.2.2.3 Behavior Requirements

1398 6.12.2.2.3.1 Preconditions

1399 \$instance contains the instance of CIM_AccountManagementService which is referenced by
1400 CIM_HostedService.

1401 6.12.2.2.3.2 Pseudo Code

```
1402 &smShowAssociationInstances ( "CIM_HostedService", $instance.getObjectPath() );
1403 &smEnd;
```

1404 6.12.2.3 Show a Single Instance – Both References

1405 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1406 a `show` command issued against CIM_HostedService where both references are specified and therefore
1407 the desired instance is unambiguously identified.

1408 6.12.2.3.1 Command Form

```
1409 show <CIM_HostedService single instance>
```

1410 6.12.2.3.2 CIM Requirements

1411 See CIM_HostedService in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the
1412 list of mandatory properties.

1413 6.12.2.3.3 Behavior Requirements

1414 6.12.2.3.3.1 Preconditions

1415 \$instanceA contains the instance of CIM_ComputerSystem which is referenced by
1416 CIM_HostedService.

1417 \$instanceB contains the instance of CIM_AccountManagementService which is referenced by
1418 CIM_HostedService.

1419 6.12.2.3.3.2 Pseudo Code

```
1420 &smShowAssociationInstance ( "CIM_HostedService", $instanceA.getObjectPath(),
1421     $instanceB.getObjectPath() );
1422 &smEnd;
```

1423 6.13 CIM_Identity

1424 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1425 Table 14 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
1426 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
1427 verb and target. Table 14 is for informational purposes only; in case of a conflict between Table 14 and
1428 requirements detailed in the following sections, the text detailed in the following sections supersedes the
1429 information in Table 14.

1430 **Table 14 – Command Verb Requirements for CIM_Identity**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	

Command Verb	Requirement	Comments
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.13.2.
start	Not supported	
stop	Not supported	

1431 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
1432 reset, set, show, start, and stop.

1433 6.13.1 Ordering of Results

1434 When results are returned for multiple instances of CIM_Identity, implementations shall utilize the
1435 following algorithm to produce the natural (that is, default) ordering:

- 1436 • Results for CIM_Identity are unordered; therefore, no algorithm is defined.

1437 6.13.2 Show

1438 This section describes how to implement the `show` verb when applied to an instance of CIM_Identity.
1439 Implementations shall support the use of the `show` verb with CIM_Identity.

1440 6.13.2.1 Show a Single Instance

1441 Use this verb to show a single instance of CIM_Identity.

1442 6.13.2.1.1 Command Form

```
1443 show <CIM_Identity single instance>
```

1444 6.13.2.1.2 CIM Requirements

1445 See CIM_Identity in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of
1446 mandatory properties.

1447 6.13.2.1.3 Behavior Requirements

1448 6.13.2.1.3.1 Preconditions

1449 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1450 6.13.2.1.3.2 Pseudo Code

```
1451 $instance=<CIM_Identity single instance>
1452 #propertylist[] = NULL;
1453 if (false == #all)
1454 {
1455     #propertylist[] = { //all mandatory non-key properties }
1456 }
1457 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
1458 &smEnd;
```

1459 **6.13.2.2 Show Multiple Instances**

1460 Use this verb to show multiple instances of CIM_Identity.

1461 **6.13.2.2.1 Command Form**

1462 `show <CIM_Identity multiple instances>`

1463 **6.13.2.2.2 CIM Requirements**

1464 See CIM_Identity in the “CIM Elements” section of the [Simple Identity Management Profile](#) for the list of
1465 mandatory properties.

1466 **6.13.2.2.3 Behavior Requirements**

1467 **6.13.2.2.3.1 Preconditions**

1468 \$containerInstance contains the instance of CIM_Account, CIM_Group, or CIM_UserContact for
1469 which related CIM_Identity instances are displayed.

1470 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1471 **6.13.2.2.3.2 Pseudo Code**

```
1472 #propertylist[] = NULL;
1473 if (false == #all)
1474     {
1475     //no properties to show
1476     #propertylist[] = { };
1477     }
1478 &smShowInstances ( "CIM_Identity", "CIM_AssignedIdentity",
1479     $containerInstance.getObjectPath(), #propertylist[] );
1480 &smEnd;
```

1481 **6.14 CIM_IdentityContext**

1482 The cd and help verbs shall be supported as described in [DSP0216](#).

1483 Table 15 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
1484 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
1485 verb and target. Table 15 is for informational purposes only; in case of a conflict between Table 15 and
1486 requirements detailed in the following sections, the text detailed in the following sections supersedes the
1487 information in Table 15.

1488 **Table 15 – Command Verb Requirements for CIM_IdentityContext**

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

Command Verb	Requirement	Comments
show	Shall	See 6.14.2.
start	Not supported	
stop	Not supported	

1489 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
1490 reset, set, start, and stop.

1491 6.14.1 Ordering of Results

1492 When results are returned for multiple instances of CIM_IdentityContext, implementations shall utilize the
1493 following algorithm to produce the natural (that is, default) ordering:

- 1494 • Results for CIM_IdentityContext are unordered; therefore, no algorithm is defined.

1495 6.14.2 Show

1496 This section describes how to implement the `show` verb when applied to an instance of
1497 CIM_IdentityContext. Implementations shall support the use of the `show` verb with CIM_IdentityContext.

1498 The `show` command is used to display information about the CIM_IdentityContext instance or instances.

1499 6.14.2.1 Show Multiple Instances – CIM_ManagedElement Reference

1500 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1501 to a `show` command issued against CIM_IdentityContext where only one reference is specified and the
1502 reference is to an instance of CIM_ManagedElement.

1503 6.14.2.1.1 Command Form

```
1504 show <CIM_IdentityContext multiple instances>
```

1505 6.14.2.1.2 CIM Requirements

1506 See CIM_IdentityContext in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the
1507 list of mandatory properties.

1508 6.14.2.1.3 Behavior Requirements

1509 6.14.2.1.3.1 Preconditions

1510 `$instance` contains the instance of CIM_Role that is referenced by CIM_IdentityContext.

1511 6.14.2.1.3.2 Pseudo Code

```
1512 &smShowAssociationInstances ( "CIM_IdentityContext", $instance.getObjectPath() );  
1513 &smEnd;
```

1514 6.14.2.2 Show Multiple Instances – CIM_Identity Reference

1515 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1516 a `show` command issued against CIM_IdentityContext where the reference specified is to an instance of
1517 CIM_Identity.

1518 6.14.2.2.1 Command Form

```
1519 show <CIM_IdentityContext multiple instances>
```

1520 6.14.2.2.2 CIM Requirements

1521 See CIM_IdentityContext in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the
1522 list of mandatory properties.

1523 6.14.2.2.3 Behavior Requirements

1524 6.14.2.2.3.1 Preconditions

1525 \$instance contains the instance of CIM_Identity which is referenced by CIM_IdentityContext.

1526 6.14.2.2.3.2 Pseudo Code

```
1527 &smShowAssociationInstances ( "CIM_IdentityContext", $instance.getObjectPath() );  
1528 &smEnd;
```

1529 6.14.2.3 Show a Single Instance – Both References

1530 This command form is for the show verb applied to a single instance. This command form corresponds to
1531 a show command issued against CIM_IdentityContext where both references are specified and therefore
1532 the desired instance is unambiguously identified.

1533 6.14.2.3.1 Command Form

```
1534 show <CIM_IdentityContext single instance>
```

1535 6.14.2.3.2 CIM Requirements

1536 See CIM_IdentityContext in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the
1537 list of mandatory properties.

1538 6.14.2.3.3 Behavior Requirements

1539 6.14.2.3.3.1 Preconditions

1540 \$instanceA contains the instance of CIM_Identity which is referenced by CIM_IdentityContext.

1541 \$instanceB contains the instance of CIM_ManagedElement which is referenced by
1542 CIM_IdentityContext.

1543 6.14.2.3.3.2 Pseudo Code

```
1544 &smShowAssociationInstance ( "CIM_IdentityContext", $instanceA.getObjectPath(),  
1545     $instanceB.getObjectPath() );  
1546 &smEnd;
```

1547 6.15 CIM_MemberOfCollection

1548 The cd and help verbs shall be supported as described in [DSP0216](#).

1549 Table 16 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
1550 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and
1551 target. Table 16 is for informational purposes only; in case of a conflict between Table 16 and
1552 requirements detailed in the following sections, the text detailed in the following sections supersedes the
1553 information in Table 16.

1554

Table 16 – Command Verb Requirements for CIM_MemberOfCollection

Command Verb	Requirement	Comments
create	Not supported	None
delete	Not supported	None
dump	Not supported	None
load	Not supported	None
reset	Not supported	None
set	Not supported	None
show	Shall	See 6.15.2.
start	Not supported	None
stop	Not supported	None

1555 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
1556 reset, set, start, and stop.

1557 6.15.1 Ordering of Results

1558 When results are returned for multiple instances of CIM_MemberOfCollection, implementations shall
1559 utilize the following algorithm to produce the natural (that is, default) ordering:

- 1560 • Results for CIM_MemberOfCollection are unordered; therefore, no algorithm is defined.

1561 6.15.2 Show

1562 This section describes how to implement the `show` verb when applied to an instance of
1563 CIM_MemberOfCollection. Implementations shall support the use of the `show` verb with
1564 CIM_MemberOfCollection.

1565 6.15.2.1 Show a Single Instance – Both References

1566 This command form is for the `show` command applied to CIM_MemberOfCollection where both
1567 references are specified. Therefore, exactly one instance is shown.

1568 6.15.2.1.1 Command Form

```
1569 show <CIM_MemberOfCollection single instance>
```

1570 6.15.2.1.2 CIM Requirements

1571 See CIM_MemberOfCollection in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1572 for the list of mandatory properties.

1573 6.15.2.1.3 Behavior Requirements

1574 6.15.2.1.3.1 Preconditions

1575 `$instanceA` contains one of the instances of CIM_Group referenced by CIM_MemberOfCollection.

1576 `$instanceB` contains one of the instances of CIM_Identity referenced by CIM_MemberOfCollection.

1577 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

1578 6.15.2.1.3.2 Pseudo Code

```

1579 #propertylist[] = NULL;
1580 if ( false == #all)
1581     {
1582         #propertylist[] = { //all mandatory non-key properties};
1583     }
1584 &smShowAssociationInstance ( "CIM_MemberOfCollection", $instanceA.getObjectPath(),
1585     $instanceB.getObjectPath(), #propertylist[] );
1586 &smEnd;

```

1587 6.15.2.2 Show Multiple Instances – CIM_Group Reference

1588 This command form is for the `show` command applied to an instance of `CIM_MemberOfCollection` where
 1589 only the reference to an instance of `CIM_Group` is specified. Zero or more instances of
 1590 `CIM_MemberOfCollection` can reference a single instance of `CIM_Group`.

1591 6.15.2.2.1 Command Form

```

1592 show <CIM_MemberOfCollection multiple instances>

```

1593 6.15.2.2.2 CIM Requirements

1594 See `CIM_MemberOfCollection` in the “CIM Elements” section of the [Simple Identity Management Profile](#)
 1595 for the list of mandatory properties.

1596 6.15.2.2.3 Behavior Requirements

1597 6.15.2.2.3.1 Preconditions

1598 `$instance` contains the instance of `CIM_Group` that is referenced by `CIM_MemberOfCollection`.

1599 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

1600 6.15.2.2.3.2 Pseudo Code

```

1601 #propertylist[] = NULL;
1602 if ( false == #all)
1603     {
1604         #propertylist[] = { //all mandatory non-key properties};
1605     }
1606 &smShowAssociationInstances ( "CIM_MemberOfCollection", $instance.getObjectPath(),
1607     #propertylist[] );
1608 &smEnd;

```

1609 6.15.2.3 Show Multiple Instances – CIM_Identity Reference

1610 This command form is for the `show` command applied to `CIM_MemberOfCollection` where only the
 1611 reference to an instance of `CIM_Identity` is specified. An instance of `CIM_Identity` can be referenced by
 1612 zero or more instances of `CIM_MemberOfCollection`.

1613 6.15.2.3.1 Command Form

```

1614 show <CIM_MemberOfCollection single instances>

```

1615 **6.15.2.3.2 CIM Requirements**

1616 See CIM_MemberOfCollection in the “CIM Elements” section of the [Simple Identify Management Profile](#)
 1617 for the list of mandatory properties.

1618 **6.15.2.3.3 Behavior Requirements**1619 **6.15.2.3.3.1 Preconditions**

1620 \$instance contains the instance of CIM_Identity that is referenced by CIM_MemberOfCollection.

1621 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1622 **6.15.2.3.3.2 Pseudo Code**

```
1623 #propertylist[] = NULL;
1624 if ( false == #all)
1625 {
1626     #propertylist[] = { //all mandatory non-key properties};
1627 }
1628 &smShowAssociationInstances ( "CIM_MemberOfCollection", $instance.getObjectPath(),
1629     #propertylist[] );
1630 &smEnd;
```

1631 **6.16 CIM_OwningCollectionElement**

1632 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1633 Table 17 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
 1634 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and
 1635 target. Table 17 is for informational purposes only; in case of a conflict between Table 17 and
 1636 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1637 information in Table 17.

1638 **Table 17 – Command Verb Requirements for CIM_OwningCollectionElement**

Command Verb	Requirement	Comments
create	Not supported	None
delete	Not supported	None
dump	Not supported	None
load	Not supported	None
reset	Not supported	None
set	Not supported	None
show	Shall	See 6.16.2.
start	Not supported	None
stop	Not supported	None

1639 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
 1640 reset, set, start, and stop.

1641 **6.16.1 Ordering of Results**

1642 When results are returned for multiple instances of CIM_OwningCollectionElement, implementations shall
1643 utilize the following algorithm to produce the natural (that is, default) ordering:

- 1644 • Results for CIM_OwningCollectionElement are unordered; therefore, no algorithm is defined.

1645 **6.16.2 Show**

1646 This section describes how to implement the `show` verb when applied to an instance of
1647 CIM_OwningCollectionElement. Implementations shall support the use of the `show` verb with
1648 CIM_OwningCollectionElement.

1649 **6.16.2.1 Show a Single Instance – Both References**

1650 This command form is for the `show` command applied to CIM_OwningCollectionElement where both
1651 references are specified. Therefore, a single instance will be returned.

1652 **6.16.2.1.1 Command Form**

```
1653 show <CIM_OwningCollectionElement single instance>
```

1654 **6.16.2.1.2 CIM Requirements**

1655 See CIM_OwningCollectionElement in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1656 for the list of mandatory properties.

1657 **6.16.2.1.3 Behavior Requirements**

1658 **6.16.2.1.3.1 Preconditions**

1659 `$instanceA` contains one of the instances of <CIM_Group> that is referenced by
1660 CIM_OwningCollectionElement.

1661 `$instanceB` contains one of the instances of <CIM_ComputerSystem> that referenced by
1662 CIM_OwningCollectionElement.

1663 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

1664 **6.16.2.1.3.2 Pseudo Code**

```
1665 #propertylist[] = NULL;
1666 if ( false == #all)
1667     {
1668         #propertylist[] = { //all mandatory non-key properties};
1669     }
1670 &smShowAssociationInstance ( "CIM_OwningCollectionElement",
1671     $instanceA.getObjectPath(), $instanceB.getObjectPath(), #propertylist[] );
1672 &smEnd;
```

1673 **6.16.2.2 Show Multiple Instances – CIM_ComputerSystem reference**

1674 This command form is for the `show` command applied to CIM_OwningCollectionElement where only the
1675 reference to an instance of CIM_ComputerSystem is specified. An instance of CIM_ComputerSystem can
1676 be referenced by multiple instances of CIM_OwningCollectionElement.

1677 6.16.2.2.1 Command Form

```
1678 show <CIM_OwningCollectionElement multiple instances>
```

1679 6.16.2.2.2 CIM Requirements

1680 See CIM_OwningCollectionElement in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of mandatory properties.
1681

1682 6.16.2.2.3 Behavior Requirements**1683 6.16.2.2.3.1 Preconditions**

1684 \$instance contains the instance of <CIM_ComputerSystem> that is referenced by
1685 CIM_OwningCollectionElement.

1686 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1687 6.16.2.2.3.2 Pseudo Code

```
1688 #propertylist[] = NULL;  
1689 if ( false == #all)  
1690 {  
1691     #propertylist[] = {/all mandatory non-key properties};  
1692 }  
1693 &smShowAssociationInstances ( "CIM_OwningCollectionElement",  
1694     $instance.getObjectPath(), #propertylist[] );  
1695 &smEnd;
```

1696 6.16.2.3 Show a Single Instance – CIM_Group Reference

1697 This command form is for the show command applied to CIM_OwningCollectionElement where only the
1698 reference to an instance of CIM_Group is specified. An instance of CIM_Group is referenced by exactly
1699 one instance of CIM_OwningCollectionElement. Therefore, a single instance is returned.

1700 6.16.2.3.1 Command Form

```
1701 show <CIM_OwningCollectionElement single instances>
```

1702 6.16.2.3.2 CIM Requirements

1703 See CIM_OwningCollectionElement in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of mandatory properties.
1704

1705 6.16.2.3.3 Behavior Requirements**1706 6.16.2.3.3.1 Preconditions**

1707 \$instance contains the instance of <CIM_Group> that is referenced by CIM_OwningCollectionElement.

1708 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1709 **6.16.2.3.3.2 Pseudo Code**

```

1710 #propertylist[] = NULL;
1711 if ( false == #all)
1712     {
1713         #propertylist[] = {/all mandatory non-key properties};
1714     }
1715 &smShowAssociationInstances ( "CIM_OwningCollectionElement",
1716     $instance.getObjectPath(), #propertylist[] );
1717 &smEnd;
    
```

1718 **6.17 CIM_ServiceAffectsElement**

1719 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1720 Table 18 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
 1721 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
 1722 verb and target. Table 18 is for informational purposes only; in case of a conflict between Table 18 and
 1723 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1724 information in Table 18.

1725 **Table 18 – Command Verb Requirements for CIM_ServiceAffectsElement**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.17.2.
start	Not supported	
stop	Not supported	

1726 The following verbs shall not be supported for the specified target: `create`, `delete`, `dump`, `load`,
 1727 `reset`, `set`, `start`, and `stop`.

1728 **6.17.1 Ordering of Results**

1729 When results are returned for multiple instances of `CIM_ServiceAffectsElement`, implementations shall
 1730 utilize the following algorithm to produce the natural (that is, default) ordering:

- 1731 • Results for `CIM_ServiceAffectsElement` are unordered; therefore, no algorithm is defined.

1732 **6.17.2 Show**

1733 This section describes how to implement the `show` verb when applied to an instance of
 1734 `CIM_ServiceAffectsElement`. Implementations shall support the use of the `show` verb with
 1735 `CIM_ServiceAffectsElement`.

1736 The `show` command is used to display information about the `CIM_ServiceAffectsElement` instance or
1737 instances.

1738 **6.17.2.1 Show a Single Instance – CIM_Identity**

1739 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1740 a `show` command issued against `CIM_ServiceAffectsElement` where the reference specified is to an
1741 instance of `CIM_Identity`. An instance of `CIM_Identity` is referenced by exactly one instance of
1742 `CIM_ServiceAffectsElement`. Therefore, a single instance will be returned.

1743 **6.17.2.1.1 Command Form**

```
1744 show <CIM_ServiceAffectsElement single instance>
```

1745 **6.17.2.1.2 CIM Requirements**

1746 See `CIM_ServiceAffectsElement` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1747 for the list of mandatory properties.

1748 **6.17.2.1.3 Behavior Requirements**

1749 **6.17.2.1.3.1 Preconditions**

1750 `$instance` contains the instance of `CIM_Identity` which is referenced by `CIM_ServiceAffectsElement`.

1751 **6.17.2.1.3.2 Pseudo Code**

```
1752 &smShowAssociationInstances ( "CIM_ServiceAffectsElement",  
1753     $instance.getObjectPath() );  
1754 &smEnd;
```

1755 **6.17.2.2 Show a Single Instance – Both References**

1756 This command form is for the `show` verb applied to a single instance. This command form corresponds to
1757 a `show` command issued against `CIM_ServiceAffectsElement` where both references are specified and
1758 therefore the desired instance is unambiguously identified.

1759 **6.17.2.2.1 Command Form**

```
1760 show <CIM_ServiceAffectsElement single instance>
```

1761 **6.17.2.2.2 CIM Requirements**

1762 See `CIM_ServiceAffectsElement` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
1763 for the list of mandatory properties.

1764 **6.17.2.2.3 Behavior Requirements**

1765 **6.17.2.2.3.1 Preconditions**

1766 `$instanceA` contains the instance of `CIM_AccountManagementService` that is referenced by
1767 `CIM_ServiceAffectsElement`.

1768 `$instanceB` contains the instance of `CIM_Identity` that is referenced by `CIM_ServiceAffectsElement`.

1769 **6.17.2.2.3.2 Pseudo Code**

```
1770 &smShowAssociationInstance ( "CIM_ServiceAffectsElement", $instanceA.getObjectPath(),
1771     $instanceB.getObjectPath() );
1772 &smEnd;
```

1773 **6.18 CIM_SettingsDefineCapabilities**

1774 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1775 Table 19 lists each SM CLP verb, the required level of support for the verb in conjunction with instances
 1776 of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the
 1777 verb and target. Table 19 is for informational purposes only; in case of a conflict between Table 19 and
 1778 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1779 information in Table 19.

1780 **Table 19 – Command Verb Requirements for CIM_SettingsDefineCapabilities**

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 6.18.2.
start	Not supported	
stop	Not supported	

1781 No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,
 1782 `reset`, `set`, `start`, and `stop`.

1783 **6.18.1 Ordering of Results**

1784 When results are returned for multiple instances of `CIM_SettingsDefineCapabilities`, implementations
 1785 shall utilize the following algorithm to produce the natural (that is, default) ordering:

- 1786 • Results for `CIM_SettingsDefineCapabilities` are unordered; therefore, no algorithm is defined.

1787 **6.18.2 Show**

1788 This section describes how to implement the `show` verb when applied to an instance of
 1789 `CIM_SettingsDefineCapabilities`. Implementations shall support the use of the `show` verb with
 1790 `CIM_SettingsDefineCapabilities`.

1791 The `show` command is used to display information about the `CIM_SettingsDefineCapabilities` instance or
 1792 instances.

1793 6.18.2.1 Show Multiple Instances – CIM_AccountManagementCapabilities

1794 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1795 to a `show` command issued against `CIM_SettingsDefineCapabilities` where only one reference is
1796 specified and the reference is to an instance of `CIM_AccountManagementCapabilities`.

1797 6.18.2.1.1 Command Form

```
1798 show <CIM_SettingsDefineCapabilities multiple objects>
```

1799 6.18.2.1.2 CIM Requirements

1800 See `CIM_SettingsDefineCapabilities` in the “CIM Elements” section of the [Simple Identify Management](#)
1801 [Profile](#) for the list of mandatory properties.

1802 6.18.2.1.3 Behavior Requirements

1803 6.18.2.1.3.1 Preconditions

1804 `$instance` contains the instance of `CIM_AccountManagementCapabilities`.

1805 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

1806 6.18.2.1.3.2 Pseudo Code

```
1807 #propertylist[] = NULL;  
1808 if (#all == false)  
1809 {  
1810     #propertylist[] = { //all mandatory non-key properties };  
1811 }  
1812 &smShowAssociationInstances ( "CIM_SettingsDefineCapabilities",  
1813     $instance.getObjectPath(), #propertylist[] );  
1814 &smEnd;
```

1815 6.18.2.2 Show Multiple Instances – CIM_EnabledLogicalElementCapabilities

1816 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1817 to a `show` command issued against `CIM_SettingsDefineCapabilities` where only one reference is
1818 specified and the reference is to an instance of `CIM_EnabledLogicalElementCapabilities`.

1819 6.18.2.2.1 Command Form

```
1820 show <CIM_SettingsDefineCapabilities multiple objects>
```

1821 6.18.2.2.2 CIM Requirements

1822 See `CIM_SettingsDefineCapabilities` in the “CIM Elements” section of the [Simple Identify Management](#)
1823 [Profile](#) for the list of mandatory properties.

1824 6.18.2.2.3 Behavior Requirements

1825 6.18.2.2.3.1 Preconditions

1826 `$instance` contains the instance of `IM_EnabledLogicalElementCapabilities`.

1827 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

1828 6.18.2.2.3.2 Pseudo Code

```

1829 #propertylist[] = NULL;
1830 if (#all == false)
1831     {
1832         #propertylist[] = { //all mandatory non-key properties };
1833     }
1834 &smShowAssociationInstances ( "CIM_SettingsDefineCapabilities",
1835     $instance.getObjectPath(), #propertylist[] );
1836 &smEnd;

```

1837 6.18.2.3 Show Multiple Instances – CIM_AccountSettingData Reference

1838 This command form is for the `show` verb applied to multiple instances. This command form corresponds
 1839 to a `show` command issued against `CIM_SettingsDefineCapabilities` where only one reference is
 1840 specified and the reference is to an instance of `CIM_AccountSettingData`.

1841 6.18.2.3.1 Command Form

```
1842 show <CIM_SettingsDefineCapabilities multiple objects>
```

1843 6.18.2.3.2 CIM Requirements

1844 See `CIM_SettingsDefineCapabilities` in the “CIM Elements” section of the [Simple Identify Management Profile](#)
 1845 for the list of mandatory properties.

1846 6.18.2.3.3 Behavior Requirements

1847 6.18.2.3.3.1 Preconditions

1848 `$instance` contains the instance of `CIM_AccountSettingData` which is referenced by
 1849 `CIM_SettingsDefineCapabilities`.

1850 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

1851 6.18.2.3.3.2 Pseudo Code

```

1852 #propertylist[] = NULL;
1853 if (#all == false)
1854     {
1855         #propertylist[] = { //all mandatory non-key properties };
1856     }
1857 &smShowAssociationInstances ( "CIM_SettingsDefineCapabilities",
1858     $instance.getObjectPath(), #propertylist[] );
1859 &smEnd;

```

1860 6.18.2.4 Show Single Instances – CIM_AccountManagementCapabilities Reference with Other 1861 Reference to CIM_AccountSettingData

1862 This command form is for the `show` verb applied to multiple instances. This command form corresponds
 1863 to a `show` command issued against `CIM_SettingsDefineCapabilities` where a reference is specified to an
 1864 instance of `CIM_AccountManagementCapabilities` and the other reference is to an instance of
 1865 `CIM_AccountSettingData`.

1866 6.18.2.4.1 Command Form

```
1867 show <CIM_SettingsDefineCapabilities multiple objects>
```

1868 6.18.2.4.2 CIM Requirements

1869 See CIM_SettingsDefineCapabilities in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of mandatory properties.

1871 6.18.2.4.3 Behavior Requirements

1872 6.18.2.4.3.1 Preconditions

1873 \$instanceA contains the instance of CIM_AccountManagementCapabilities which is referenced by
1874 CIM_SettingsDefineCapabilities.

1875 \$instanceB contains the instance of CIM_AccountSettingData that is referenced by
1876 CIM_SettingsDefineCapabilities.

1877 6.18.2.4.3.2 Pseudo Code

```
1878 &smShowAssociationInstance ( "CIM_SettingsDefineCapabilities",  
1879     $instanceA.getObjectPath(), $instanceB.getObjectPath() );  
1880 &smEnd;
```

1881 6.18.2.5 Show Single Instances – CIM_EnabledLogicalElementCapabilities Reference with Other 1882 Reference to CIM_AccountSettingData

1883 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1884 to a `show` command issued against CIM_SettingsDefineCapabilities where a reference is specified to an
1885 instance of CIM_EnabledLogicalElementCapabilities and the other reference is to an instance of
1886 CIM_AccountSettingData.

1887 6.18.2.5.1 Command Form

```
1888 show <CIM_SettingsDefineCapabilities multiple objects>
```

1889 6.18.2.5.2 CIM Requirements

1890 See CIM_SettingsDefineCapabilities in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the list of mandatory properties.

1892 6.18.2.5.3 Behavior Requirements

1893 6.18.2.5.3.1 Preconditions

1894 \$instanceA contains the instance of CIM_EnabledLogicalElementCapabilities which is referenced by
1895 CIM_SettingsDefineCapabilities.

1896 \$instanceB contains the instance of CIM_AccountSettingData that is referenced by
1897 CIM_SettingsDefineCapabilities.

1898 6.18.2.5.3.2 Pseudo Code

```
1899 &smShowAssociationInstance ( "CIM_SettingsDefineCapabilities",  
1900     $instanceA.getObjectPath(), $instanceB.getObjectPath() );  
1901 &smEnd;
```

1902 6.19 CIM_UserContact

1903 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

1904 Table 20 lists each SM CLP verb, the required level of support for the verb in conjunction with the target
 1905 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and
 1906 target. Table 20 is for informational purposes only; in case of a conflict between Table 20 and
 1907 requirements detailed in the following sections, the text detailed in the following sections supersedes the
 1908 information in Table 20.

1909 **Table 20 – Command Verb Requirements for CIM_UserContact**

Command Verb	Requirement	Comments
create	Not supported	None
delete	Not supported	None
dump	Not supported	None
load	Not supported	None
reset	Not supported	None
set	Not supported	None
show	Shall	See 6.19.2.
start	Not supported	None
stop	Not supported	None

1910 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,
 1911 reset, set, start, and stop.

1912 **6.19.1 Ordering of Results**

1913 When results are returned for multiple instances of CIM_UserContact, implementations shall utilize the
 1914 following algorithm to produce the natural (that is, default) ordering:

- 1915 • Results for CIM_UserContact are unordered; therefore, no algorithm is defined.

1916 **6.19.2 Show**

1917 This section describes how to implement the show verb when applied to an instance of
 1918 CIM_UserContact. Implementations shall support the use of the show verb with CIM_UserContact.

1919 **6.19.2.1 Show a Single Instance**

1920 Show a single instance of CIM_UserContact.

1921 **6.19.2.1.1 Command Form**

1922 `show <CIM_UserContact single instance>`

1923 **6.19.2.1.2 CIM Requirements**

1924 See CIM_UserContact in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the
 1925 list of mandatory properties.

1926 **6.19.2.1.3 Behavior Requirements**

1927 **6.19.2.1.3.1 Preconditions**

1928 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1929 **6.19.2.1.3.2 Pseudo Code**

```
1930 $instance=<CIM_UserContact single instance>
1931 #propertylist[] = NULL;
1932 if (false == #all)
1933     {
1934         #propertylist[] = { //all mandatory non-key properties }
1935     }
1936 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
1937 &smEnd;
```

1938 **6.19.2.2 Show Multiple Instances**

1939 Show multiple instances of CIM_UserContact.

1940 **6.19.2.2.1 Command Form**

```
1941 show <CIM_UserContact multiple instances>
```

1942 **6.19.2.2.2 CIM Requirements**

1943 See CIM_UserContact in the “CIM Elements” section of the [Simple Identify Management Profile](#) for the
1944 list of mandatory properties.

1945 **6.19.2.2.3 Behavior Requirements**1946 **6.19.2.2.3.1 Preconditions**

1947 \$containerInstance contains the instance of CIM_Identity for which related CIM_UserContact
1948 instances are displayed.

1949 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

1950 **6.19.2.2.3.2 Pseudo Code**

```
1951 #propertylist[] = NULL;
1952 if (false == #all)
1953     {
1954         #propertylist[] = { //all mandatory non-key properties };
1955     }
1956 &smShowInstances ( "CIM_UserContact", "CIM_AssignedIdentity",
1957     $containerInstance.getObjectPath(), #propertylist[] );
1958 &smEnd;
```

1959

1960
 1961
 1962
 1963
 1964

ANNEX A (informative)

Change Log

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

1965