



1
2 **Document Number: DSP1005**
3 **Date: 2009-06-17**
4 **Version: 1.0.0**

5 **Command Line Protocol Service Profile**

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

9 Copyright Notice

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

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

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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

CONTENTS

32	Foreword	7
33	Introduction	8
34	1 Scope	9
35	2 Normative References.....	9
36	2.1 Approved References	9
37	2.2 Other References.....	9
38	3 Terms and Definitions	10
39	4 Symbols and Abbreviated Terms	11
40	5 Synopsis.....	11
41	6 Description	12
42	6.1 CLP Session Lifecycle	14
43	7 Implementation Requirements	14
44	7.1 Representing a CLP Service	14
45	7.2 Representing a CLP Session.....	16
46	7.3 Relationship with Transport Services (Optional)	17
47	7.4 SM CLP Admin Domain.....	17
48	7.5 CLP User Account Management (Mandatory).....	17
49	7.6 CLP Operations and Queue	20
50	8 Methods.....	21
51	8.1 CIM_ProtocolService.RequestStateChange()	21
52	8.2 CIM_ConcreteJob.RequestStateChange()	22
53	8.3 CIM_ConcreteJob.GetError().....	23
54	8.4 Profile Conventions for Operations.....	23
55	8.5 CIM_BindsTo	23
56	8.6 CIM_CLPCapabilities.....	24
57	8.7 CIM_CLPSettingData.....	24
58	8.8 CIM_CLPProtocolEndpoint.....	24
59	8.9 CIM_ConcreteJob	25
60	8.10 CIM_ElementCapabilities	25
61	8.11 CIM_ElementSettingData	26
62	8.12 CIM_Error	26
63	8.13 CIM_HostedAccessPoint	27
64	8.14 CIM_HostedJobDestination	27
65	8.15 CIM_HostedService	27
66	8.16 CIM_JobDestinationJobs	28
67	8.17 CIM_JobQueue.....	28
68	8.18 CIM_OwningJobElement	28
69	8.19 CIM_ProtocolService	28
70	8.20 CIM_ProvidesEndpoint	29
71	8.21 CIM_ServiceAccessBySAP	29
72	8.22 ServiceAffectsElement.....	29
73	9 Use Cases.....	30
74	9.1 Object Diagrams	30
75	9.2 Modifying Active Session Settings.....	37
76	9.3 Disabling the CLP Service	38
77	9.4 Determining the CLP Service Capabilities.....	38
78	9.5 Determining If ElementName Can Be Modified.....	38
79	9.6 Determining If State Management Is Supported	38
80	10 CIM Elements.....	38
81	10.1 CIM_BindsTo	39
82	10.2 CIM_CLPCapabilities.....	40
83	10.3 CIM_CLPProtocolEndpoint.....	40

84	10.4 CIM_CLPSettingData – Specification Default Configuration	41
85	10.5 CIM_CLPSettingData.....	41
86	10.6 CIM_ConcreteDependency – (Access Ingress)	42
87	10.7 CIM_ConcreteJob	42
88	10.8 CIM_ElementCapabilities	42
89	10.9 CIM_ElementSettingData – CLP Service	43
90	10.10 CIM_ElementSettingData – CLP Session	43
91	10.11 CIM_Error	43
92	10.12 CIM_HostedAccessPoint	44
93	10.13 CIM_HostedJobDestination	44
94	10.14 CIM_HostedService	44
95	10.15 CIM_IdentityContext	45
96	10.16 CIM_JobDestinationJobs	45
97	10.17 CIM_JobQueue	45
98	10.18 CIM_OwningCollectionElement	45
99	10.19 CIM_OwningJobElement – CLP Service	46
100	10.20 CIM_OwningJobElement – CLP Session	46
101	10.21 CIM_Privilege (Administrator)	46
102	10.22 CIM_Privilege (Operator)	47
103	10.23 CIM_Privilege (Read Only)	47
104	10.24 CIM_ProtocolService	47
105	10.25 CIM_ProvidesEndpoint	48
106	10.26 CIM_RegisteredProfile	48
107	10.27 CIM_Role (Administrator)	48
108	10.28 CIM_Role (Operator)	48
109	10.29 CIM_Role (Read Only)	49
110	10.30 CIM_RoleLimitedToTarget	49
111	10.31 CIM_ServiceAccessBySAP	49
112	10.32 CIM_ServiceAffectsElement – AdminDomain	50
113	10.33 CIM_ServiceAffectsElement – Job Queue	50
114	ANNEX A (informative) Change Log	51
115		

Figures

117	Figure 1 – Command Line Protocol Service Profile: Class Diagram	13
118	Figure 2 – Registered Profile	31
119	Figure 3 – CLP Service Accessible via SSH.....	31
120	Figure 4 – One Active Session	32
121	Figure 5 – Session with Admin Domain and Job Queue	33
122	Figure 6 – Single Operation Executing	34
123	Figure 7 – CLP Command Used to Terminate Running CLP Operation	35
124	Figure 8 – CLP Roles	36
125	Figure 9 – Active Session with Security Principal	37
126		

Tables

128	Table 1 – Referenced Profiles	12
129	Table 2 – CIM_ProtocolService.RequestStateChange() Method: Return Code Values	22
130	Table 3 – CIM_ProtocolService.RequestStateChange() Method: Parameters	22
131	Table 4 – CIM_ConcreteJob.RequestStateChange() Method: Return Code Values	22
132	Table 5 – CIM_ConcreteJob.RequestStateChange() Method: Parameters	23
133	Table 6 – CIM_ConcreteJob.GetError() Method: Return Code Values	23
134	Table 7 – CIM_ConcreteJob.GetError() Method: Parameters	23

135	Table 8 – Operations: CIM_BindsTo.....	24
136	Table 9 – Operations: CIM_CLPSettingData	24
137	Table 10 – Operations: CIM_CLPProtocolEndpoint	24
138	Table 11 – Operations: CIM_ConcreteJob	25
139	Table 12 – Operations: CIM_ElementCapabilities	26
140	Table 13 – Operations: CIM_ElementSettingData.....	26
141	Table 14 – Operations: CIM_HostedAccessPoint	27
142	Table 15 – Operations: CIM_HostedJobDestination	27
143	Table 16 – Operations: CIM_HostedService	27
144	Table 17 – Operations: CIM_JobDestinationJobs	28
145	Table 18 – Operations: CIM_OwningJobElement	28
146	Table 19 – Operations: CIM_ProtocolService.....	28
147	Table 20 – Operations: CIM_ProvidesEndpoint	29
148	Table 21 – Operations: CIM_ServiceAccessBySAP	29
149	Table 22 – Operations: CIM_ServiceAffectsElement	30
150	Table 23 – CIM Elements: Command Line Protocol Service Profile	38
151	Table 24 – Class: CIM_BindsTo	39
152	Table 25 – Class: CIM_CLPCapabilities.....	40
153	Table 26 – Class: CIM_CLPProtocolEndpoint.....	40
154	Table 27 – Class: CIM_CLPSettingData.....	41
155	Table 28 – Class: CIM_CLPSettingData.....	41
156	Table 29 – Class: CIM_ConcreteDependency – (Access Ingress).....	42
157	Table 30 – Class: CIM_ConcreteJob	42
158	Table 31 – Class: CIM_ElementCapabilities.....	42
159	Table 32 – Class: CIM_ElementSettingData (CLP Service).....	43
160	Table 33 – Class: CIM_ElementSettingData (CLP Session).....	43
161	Table 34 – Class: CIM_Error.....	43
162	Table 35 – Class: CIM_HostedAccessPoint	44
163	Table 36 – Class: CIM_HostedJobDestination	44
164	Table 37 – Class: CIM_HostedService	44
165	Table 38 – Class: CIM_IdentityContext	45
166	Table 39 – Class: CIM_JobDestinationJobs	45
167	Table 40 – Class: CIM_JobQueue	45
168	Table 41 – Class: CIM_OwningCollectionElement	46
169	Table 42 – Class: CIM_OwningJobElement	46
170	Table 43 – Class: CIM_OwningJobElement	46
171	Table 44 – Class: CIM_Privilege (Administrator)	46
172	Table 45 – Class: CIM_Privilege (Operator)	47
173	Table 46 – Class: CIM_Privilege (Read Only)	47
174	Table 47 – Class: CIM_ProtocolService	47
175	Table 48 – Class: CIM_ProvidesEndpoint	48
176	Table 49 – Class: CIM_RegisteredProfile.....	48
177	Table 50 – Class: CIM_Role (Administrator)	48
178	Table 51 – Class: CIM_Role (Operator)	49
179	Table 52 – Class: CIM_Role (Read Only)	49
180	Table 53 – Class: CIM_RoleLimitedToTarget.....	49
181	Table 54 – Class: CIM_ServiceAccessBySAP	49
182	Table 55 – Class: CIM_ServiceAffectsElement	50
183	Table 56 – Class: CIM_ServiceAffectsElement	50

184

185

186

Foreword

187 The *Command Line Protocol Service Profile* (DSP1005) was prepared by the Physical Platform Profiles
188 Working Group and the Server Management Working Group of the DMTF.

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

191 Acknowledgments

192 The authors wish to acknowledge the following people.

193 Contributors:

194 • Aaron Merkin

195 Participants from the DMTF Server Management Working Group:

196 • Jon Hass – Dell

197 • Khachatur Papanyan – Dell

198 • Enoch Suen – Dell

199 • Jeff Hilland – HP

200 • Christina Shaw – HP

201 • Perry Vincent – Intel

202 • John Leung – Intel

203

204

Introduction

205 The information in this specification should be sufficient for a provider or consumer of this data to identify
206 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to
207 represent and manage a CLP service, its associated configuration information, and any active
208 connections.

209 The target audience for this specification is implementers who are writing CIM-based providers or
210 consumers of management interfaces that represent the component described in this document.

211 Command Line Protocol Service Profile

212 1 Scope

213 The *Command Line Protocol Service Profile* extends the management capability of referencing profiles by
214 adding the capability to represent a CLP service and its associated sessions. A CLP service is logical
215 entity that provides management access through the *Server Management Command Line Profile* defined
216 in the [*Server Management Command Line Protocol Specification*](#).

217 2 Normative References

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

221 2.1 Approved References

222 DMTF DSP0004, *CIM Infrastructure Specification 2.3*,
http://www.dmtf.org/standards/published_documents/DSP0004_2.3.pdf

224 DMTF DSP0200, *CIM Operations over HTTP 1.2*,
http://www.dmtf.org/standards/published_documents/DSP0200_1.2.pdf

226 DMTF DSP0214, *Server Management Command Line Protocol Specification 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0214_1.0.pdf

228 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf

230 DMTF DSP1007, *SM CLP Admin Domain Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1007_1.0.pdf

232 DMTF DSP1016, *Telnet Service Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1016_1.0.pdf

234 DMTF DSP1017, *SSH Service Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1017_1.0.pdf

236 DMTF DSP1033, *Profile Registration Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

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

240 DMTF DSP1039, *Role Based Authorization Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1039_1.0.pdf

242 2.2 Other References

243 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>

245 3 Terms and Definitions

246 For the purposes of this document, the following terms and definitions apply. For the purposes of this
247 document, the terms and definitions given in [Profile Registration Profile](#) and [Management Profile](#)
248 [Specification Usage Guide](#) also apply.

249 **3.1**

250 **can**

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

252 **3.2**

253 **cannot**

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

255 **3.3**

256 **conditional**

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

259 **3.4**

260 **mandatory**

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

263 **3.5**

264 **may**

265 indicates a course of action permissible within the limits of the document

266 **3.6**

267 **need not**

268 indicates a course of action permissible within the limits of the document

269 **3.7**

270 **optional**

271 indicates a course of action permissible within the limits of the document

272 **3.8**

273 **referencing profile**

274 indicates a profile that owns the definition of this class and can include a reference to this profile in its
275 "Referenced Profiles" table

276 **3.9**

277 **shall**

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

280 **3.10**

281 **shall not**

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

284 **3.11**

285 **should**

286 indicates that among several possibilities, one is recommended as particularly suitable, without
287 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

288 **3.12**
289 **should not**
290 indicates that a certain possibility or course of action is deprecated but not prohibited
291 **3.13**
292 **unspecified**
293 indicates that this profile does not define any constraints for the referenced CIM element or operation

294 **4 Symbols and Abbreviated Terms**

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

296 **4.1**
297 **CIM**
298 Common Information Model

299 **4.2**
300 **CLP**
301 Command Line Protocol

302 **4.3**
303 **IP**
304 Internet Protocol

305 **4.4**
306 **TCP**
307 Transmission Control Protocol

308 **5 Synopsis**

309 **Profile Name:** Command Line Protocol Service

310 **Version:** 1.0.0

311 **Organization:** DMTF

312 **CIM Schema Version:** 2.22

313 **Central Class:** CIM_ProtocolService

314 **Scoping Class:** CIM_ComputerSystem

315 The *Command Line Protocol Service Profile* extends the management capability of referencing profiles by
316 adding the capability to represent a CLP service in a managed system. This profile includes a
317 specification of the CLP service, its associated configuration, and any active sessions.

318 Table 1 identifies profiles on which this profile has a dependency.

319

Table 1 – Referenced Profiles

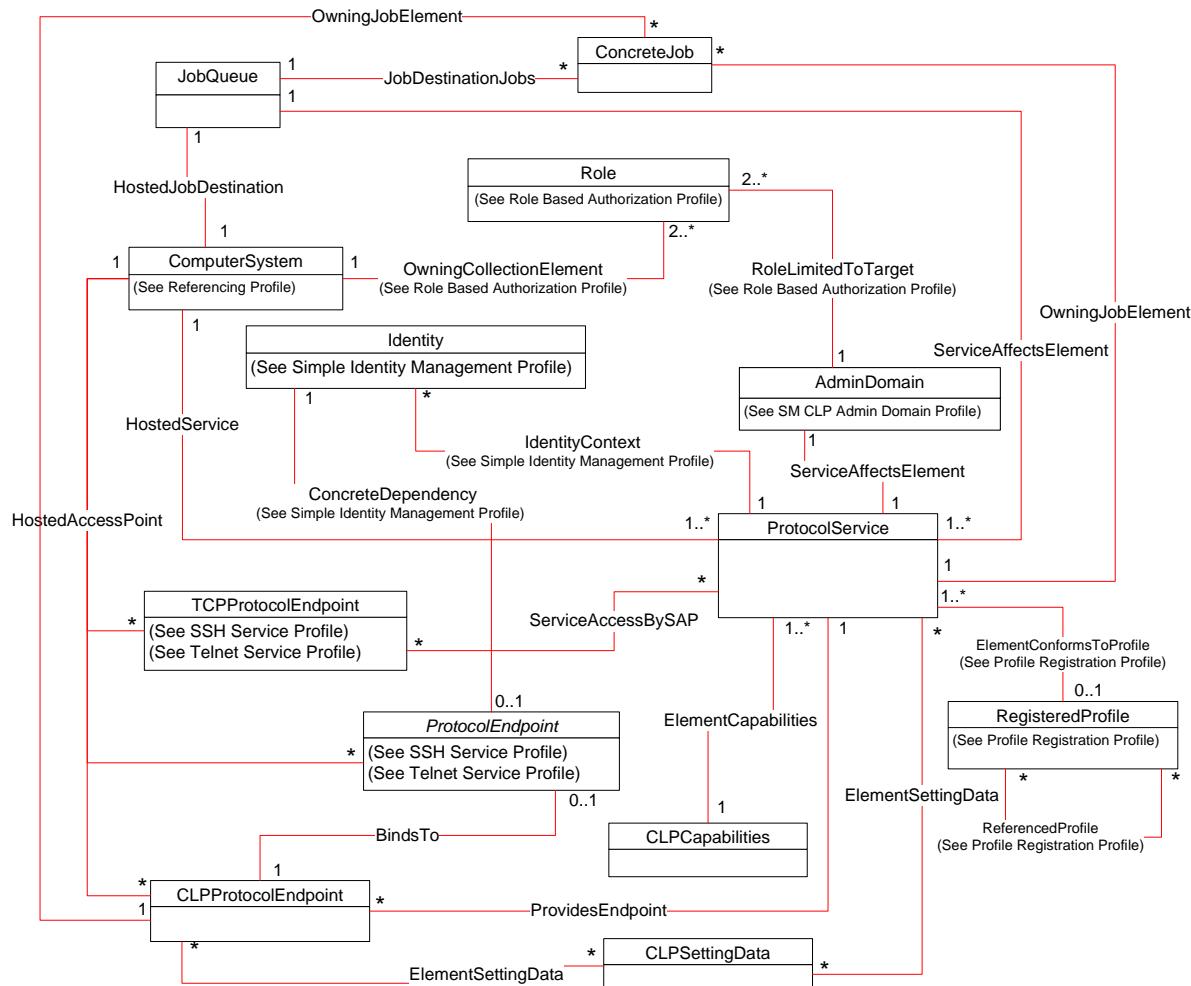
Profile Name	Organization	Version	Relationship	Behavior
Simple Identity Management	DMTF	1.0	Mandatory	See section 7.5.
Profile Registration	DMTF	1.0	Mandatory	None
Role Based Authorization	DMTF	1.0	Mandatory	See section 7.5.
SM CLP Admin Domain	DMTF	1.0	Mandatory	See section 7.4.
SSH Service	DMTF	1.0	Optional	See section 7.3.
Telnet Service	DMTF	1.0	Optional	See section 7.3.

320 The Central Class for the *Command Line Protocol Service Profile* shall be the CIM_ProtocolService class.
 321 The Central Instance of the *Command Line Protocol Service Profile* shall be an instance of
 322 CIM_ProtocolService. The Scoping Class for the *Command Line Protocol Service Profile* shall be
 323 CIM_ComputerSystem. The Scoping Instance of the *Command Line Protocol Service Profile* shall be the
 324 instance of CIM_ComputerSystem to which the Central Instance is associated through an instance of the
 325 CIM_HostedService association.

326 **6 Description**

327 The *Command Line Protocol Service Profile* describes a CLP service, its associated configuration, and
 328 active sessions. A CLP Service is a conceptual entity that supports the protocol specified in *Server*
 329 *Management Command Line Protocol Specification* ([DSP0214](#)). [DSP0214](#) identifies numerous
 330 requirements for configuration and management functionality of the CLP service itself. The *Command*
 331 *Line Protocol Service Profile* specifies how to satisfy these requirements through representation and
 332 manipulation of CIM elements.

333 Figure 1 represents the class schema for the *Command Line Protocol Service Profile*.



334

335

Figure 1 – Command Line Protocol Service Profile: Class Diagram

336 The *Command Line Protocol Service Profile* extends the management capability of referencing profiles by
 337 adding the capability to represent a CLP service hosted on a managed system. Functionality within the
 338 scope of this profile includes:

- 339
 - representation of the CLP service
 - 340 • representation of active CLP sessions at the CLP server
 - 341 • configuration of the CLP service
 - 342 • configuration of the CLP sessions from the CLP server

343 Functionality explicitly excluded from the scope of this profile includes modeling of the CLP session at the
 344 CLP client.

345 This profile represents the capabilities of the CLP service, the current configuration of the CLP service,
 346 active sessions, and the default settings when new sessions are activated. The CLP service is
 347 represented by an instance of CIM_ProtocolService. The capabilities of the CLP service are represented
 348 by an instance of CIM_CLPCapabilities. The current configuration of the CLP service is modeled with the
 349 properties from the instance of CIM_ProtocolService. Each active session with the CLP service is
 350 represented by an instance of CIM_CLPProtocolEndpoint. The current configuration of an active session
 351 is reflected in the values of the properties from the CIM_CLPProtocolEndpoint. CIM_CLPSettingData

352 represents a complete configuration that a CLP session could have. When associated with
353 CIM_ProtocolService, an instance of CIM_CLPSettingData contains a configuration that could be in effect
354 for a CLP session when it is first established. When associated with a CIM_CLPProtocolEndpoint, an
355 instance of CIM_CLPSettingData contains the configuration that was in effect when the CLP session was
356 established.

357 [DSP0214](#) specifies requirements for the management of CLP operations. CLP operations and the job
358 queue are modeled using CIM_ConcreteJob and CIM_JobQueue respectively. See section 7.6.

359 **6.1 CLP Session Lifecycle**

360 When a CLP session is established with the CLP service, an instance of CIM_CLPProtocolEndpoint is
361 created. The CIM_CLPProtocolEndpoint instance exists for the duration of the CLP session that it
362 represents. When the CLP session is ended, the CIM_CLPProtocolEndpoint will be removed. When the
363 CIM_CLPProtocolEndpoint is explicitly deleted through an intrinsic DeleteInstance operation, the CLP
364 session is ended.

365 **7 Implementation Requirements**

366 This section details the requirements related to the arrangement of instances and properties of instances
367 for implementations of this profile.

368 **7.1 Representing a CLP Service**

369 An instance of CIM_ProtocolService shall represent the CLP service being modeled.

370 **7.1.1 CIM_ProtocolService.Protocol**

371 The Protocol property of the CIM_ProtocolService instance shall have a value of 4 (CLP).

372 **7.1.2 CLP Service Capabilities**

373 An instance of CIM_CLPCapabilities shall be associated with the CIM_ProtocolService instance through
374 an instance of CIM_ElementCapabilities. This instance of CIM_CLPCapabilities shall represent the
375 capabilities of the CLP service.

376 **7.1.2.1 Maximum Number of Sessions Supported**

377 A value of zero for the CIM_CLPCapabilities.MaxConnections property shall indicate that the CLP service
378 does not enforce a limit on the number of concurrent sessions supported. The underlying transport
379 protocol may enforce a limit on the number of concurrent session which would in turn result in a restriction
380 in the number of concurrent CLP sessions. When the optional behavior of modeling the underlying
381 transport is implemented (see section 7.3), the CIM_ProtocolServiceCapabilities.MaxConnections
382 property for the CIM_ProtocolService instance representing the transport will indicate the maximum
383 number of concurrent sessions for the transport.

384 **7.1.3 Managing the CLP Service's State**

385 This section describes the usage of the RequestedState and EnabledState properties to represent the
386 state of an instance of CIM_ProtocolService.

387 **7.1.3.1 State Management Supported**

388 Exactly one instance of CIM_CLPCapabilities shall be associated with an instance of
389 CIM_ProtocolService, which indicates support for managing the state of the CLP service.

390 Support for managing the state of the CLP service is conditional behavior. This section describes the CIM
391 elements and behaviors that shall be implemented when this behavior is supported.

392 **7.1.3.2 CIM_ProtocolService.RequestStateChange() Supported**

393 When the CIM_CLPCapabilities.RequestedStatesSupported property contains at least one value, the
394 CIM_ProtocolService.RequestStateChange() method shall be implemented and supported. The
395 CIM_ProtocolService.RequestStateChange() method shall not return a value of 1 (Unspecified).

396 **7.1.3.3 CIM_ProtocolService.RequestedState**

397 When state management is supported, the RequestedState property shall be supported. When state
398 management is Unspecified, the RequestedState property may be supported.

399 Upon successful invocation of the CIM_ProtocolService.RequestStateChange() method, the value of the
400 RequestedState property shall be the value of the RequestedState parameter. If the method is not
401 successfully invoked, the value of the RequestedState property is indeterminate. When the
402 RequestedStatesSupported property of the associated instance of CIM_CLPCapabilities contains one or
403 more values, the RequestedState property shall have one of the values specified or 5 (No Change).
404 When the RequestedStatesProperty of the associated instance of
405 CIM_EnabledLogicalElementCapabilities does not contain any values, the RequestedState property shall
406 have the value of 12 (Not Applicable).

407 **7.1.3.4 EnabledState**

408 When the RequestedState parameter has a value of 2 (Enabled) or 3 (Disabled), upon successful
409 completion of the CIM_ProtocolService.RequestStateChange() method, the value of the EnabledState
410 property shall equal the value of the RequestedState property. If the method does not complete
411 successfully, the value of the EnabledState property is indeterminate. The EnabledState property shall
412 have the value 2 (Enabled), 3 (Disabled), or 5 (Not Applicable).

413 **7.1.3.5 Indicating State Management Support with CIM_CLPCapabilities**

414 When state management is supported, the RequestedStatesSupported property of the
415 CIM_CLPCapabilities instance associated with the CIM_ProtocolService instance via an instance of
416 CIM_ElementCapabilities shall contain at least one value. The RequestedStatesSupported property may
417 have zero or more of the following values: 2 (Enabled), 3 (Disabled), or 11 (Reset).

418 **7.1.4 CIM_ProtocolService ElementName Constraints**

419 The ElementName property of CIM_ProtocolService may be modifiable by a client or it may have a fixed
420 value.

421 **7.1.4.1 ElementName Is Not Modifiable**

422 When an implementation does not support modification of the ElementName property by a client, the
423 ElementName property shall be formatted as a free-form string of variable length (pattern ".*").

424 **7.1.4.2 ElementName Is Modifiable**

425 The CIM_ProtocolService.ElementName property may be modified by a client. This is conditional
426 behavior. This section describes the CIM elements and behavioral requirements when an implementation
427 supports client modification of the CIM_ProtocolService.ElementName property.

428 **7.1.4.2.1 CIM_CLPCapabilities.ElementNameEditSupported**

429 This property shall have a value of TRUE when the implementation supports client modification of the
430 CIM_ProtocolService.ElementName property.

431 **7.1.4.2.2 CIM_CLPCapabilities.MaxElementNameLen**

432 The MaxElementNameLen property shall be implemented when the ElementNameEditSupported
433 property has a value of TRUE. The MaxElementNameLen property shall indicate the maximum length of
434 a string that the implementation will accept as a value for the ElementName property of the associated
435 CIM_ProtocolService instance.

436 **7.2 Representing a CLP Session**

437 Each active session with the CLP service shall be represented with an instance of
438 CIM_CLPProtocolEndpoint.

439 **7.2.1 Relationship with Service**

440 An instance of CIM_ProvidesEndpoint shall associate the CIM_ProtocolService with the
441 CIM_CLPProtocolEndpoint.

442 **7.2.2 Specification Default Configuration**

443 [DSP0214](#) defines default values for each session attribute that is required to be maintained. This is the
444 specification default configuration and shall be represented by an instance of CIM_CLPSettingData
445 implemented as specified in section 10.4. This instance of CIM_CLPSettingData shall be associated with
446 the Central Instance through the CIM_ElementSettingData association where the IsDefault property of the
447 CIM_ElementSettingData instance has the value 1 (Is Default).

448 **7.2.3 Session Default Configuration**

449 When a CLP session is created, it will have an initial configuration. Implementations can indicate to
450 clients the configuration that will be assigned to a session. An implementation can also indicate to clients
451 the configuration that an active session had when the session was first established. This specification
452 does not identify requirements for representing the configuration that will be utilized when a specific user
453 establishes a session.

454 **7.2.3.1 Configuration that Will Be Assigned (Optional)**

455 The same initial configuration may be assigned for all CLP sessions spawned. This is optional behavior.
456 When the implementation assigns the same initial configuration for all CLP sessions, the configuration
457 that a session will have when it is established shall be represented by an instance of
458 CIM_CLPSettingData associated with the CIM_ProtocolService through an instance of
459 CIM_ElementSettingData where the IsNext property of the CIM_ElementSettingData instance has a value
460 of 1 (Is Next).

461 **7.2.3.2 Initial Configuration of a Session (Optional)**

462 The initial configuration of a session may be modeled. This is optional behavior. When the configuration
463 that a session had when it was established is modeled, it shall be represented by an instance of
464 CIM_CLPSettingData associated with the CIM_CLPProtocolEndpoint through an instance of
465 CIM_ElementSettingData where the IsCurrent property of the CIM_ElementSettingData instance has a
466 value of 1 (Is Current).

467 It is not necessary that there be a discrete copy of CIM_CLPSettingData for each active session. It is only
468 necessary that the CIM_CLPSettingData associated with the CIM_CLPProtocolEndpoint accurately
469 reflect the initial configuration of the session.

470 **7.3 Relationship with Transport Services (Optional)**

471 [DSP0214](#) indicates support for accessing the CLP using either SSH or Telnet as the transport protocol.
472 The ability to access the CLP through SSH or Telnet may be modeled. When the ability to access the
473 CLP over SSH is modeled, the [SSH Service Profile](#) shall be implemented. When the ability to access the
474 CLP over Telnet is modeled, the [Telnet Service Profile](#) shall be implemented. When the ability to access
475 the CLP through a transport protocol is modeled, the behavior in the following sections shall be
476 implemented.

477 **7.3.1 Access via SSH**

478 A CLP implementation may be accessible via SSH. When the SSH service underlying the CLP service is
479 modeled the requirements in this section shall be met.

480 The [SSH Service Profile](#) shall be implemented. The optional behavior specified in section 7.1.5.1.1 of the
481 [SSH Service Profile](#) should be implemented. After a CLP session has been established through an SSH
482 session, there shall be an instance of CIM_BindsTo that associates the CIM_CLPProtocolEndpoint
483 representing the CLP session with the CIM_SSHProtocolEndpoint representing the SSH session.

484 **7.3.2 Access via Telnet**

485 A CLP implementation may be accessible via Telnet. When the Telnet service underlying the CLP service
486 is modeled the requirements in this section shall be met.

487 The [Telnet Service Profile](#) shall be implemented. The optional behavior specified in section 7.2.2.1 of
488 [Telnet Service Profile](#) should be implemented. After a CLP session has been established through a
489 Telnet session, there shall be an instance of CIM_BindsTo that associates the CIM_CLPProtocolEndpoint
490 representing the CLP session with the CIM_TelnetProtocolEndpoint representing the Telnet session.

491 **7.3.3 Port(s) Through Which the CLP Can Be Accessed**

492 When the optional behavior specified in section 7.1.5.1.1 of [SSH Service Profile](#) or the optional behavior
493 specified in section 7.2.2.1 of [Telnet Service Profile](#) is implemented, there will be at least one instance of
494 CIM_TCPIPProtocolEndpoint.

495 For each instance of CIM_TCPIPProtocolEndpoint, if establishing a connection to the transport protocol
496 through the port represented by the instance of CIM_TCPIPProtocolEndpoint will establish, or enable the
497 establishment of, a CLP session, the Central Instance shall be associated to the instance of
498 CIM_TCPIPProtocolEndpoint through an instance of CIM_ServiceAccessBySAP.

499 **7.4 SM CLP Admin Domain**

500 The *SM CLP Admin Domain Profile* shall be implemented. There shall be an instance of
501 CIM_ServiceAffectsElement that associates the Central Instance of this profile with the Central Instance
502 of the *SM CLP Admin Domain Profile*.

503 **7.5 CLP User Account Management (Mandatory)**

504 This section describes the requirements for representing CLP groups and authorization. The [Simple](#)
505 [Identify Management Profile](#) and the [Role Based Authorization Profile](#) shall be implemented.

506 **7.5.1 Constraining Role Usage**

507 This section details constraints on associations defined in the [Role Based Authorization Profile](#) that are
508 used to indicate the scope of an instance of CIM_Role.

509 **7.5.1.1 Role Scope**

510 Each instance of CIM_Role implemented as defined in sections 7.5.2.1, 7.5.3.1, and 7.5.4.1 shall be
511 associated with the CIM_AdminDomain instance specified in section 7.4 through the
512 CIM_RoleLimitedToTarget association.

513 **7.5.1.2 Role Ownership**

514 Each instance of CIM_Role implemented as defined in sections 7.5.2.1, 7.5.3.1, and 7.5.4.1 shall be
515 associated with the Scoping Instance through the CIM_OwningCollectionElement association.

516 **7.5.2 Representing the CLP Administrator Role and Privileges**

517 This section details the requirements for representing an SM CLP Administrator Role.

518 **7.5.2.1 Administrator Role**

519 There shall be an instance of CIM_Role implemented as specified in section 0.

520 **7.5.2.2 Administrator Privileges**

521 There shall be an instance of CIM_Privilege associated with the instance of CIM_Role defined in
522 section 7.5.2.1 through the CIM_MemberOfCollection association. There shall not be any other instances
523 of CIM_Privilege associated with instance of CIM_Role defined in section 7.5.2.1. The instance of
524 CIM_Privilege shall be implemented as specified in section 10.21.

525 **7.5.2.2.1 CIM_Privilege.Activities**

526 For each array index of the CIM_Privilege.ActivityQualifiers property that contains a value, the
527 corresponding index of the CIM_Privilege.Activities property shall contain the value 7 (Execute).

528 **7.5.2.2.2 CIM_Privilege.ActivityQualifiers**

529 The CIM_Privilege.ActivityQualifiers property shall contain the values {"cd", "exit", "help", "show", "version",
530 "reset", "start", "stop", "set", "load", "dump", "create", "delete"} and may contain additional values.

531 **7.5.3 Representing the CLP Operator Role and Privileges (Optional)**

532 This section details the requirements for representing an SM CLP User Role.

533 **7.5.3.1 Operator Role**

534 There shall be an instance of CIM_Role implemented as specified in section 10.28.

535 **7.5.3.2 Operator Privileges**

536 There shall be an instance of CIM_Privilege associated with the instance of CIM_Role defined in
537 section 7.5.3.1 through the CIM_MemberOfCollection association. There shall not be any other instances
538 of CIM_Privilege associated with instance of CIM_Role defined in section 7.5.3.1. The instance of
539 CIM_Privilege shall be implemented as specified in section 10.22.

540 **7.5.3.2.1 CIM_Privilege.Activities**

541 For each array index of the CIM_Privilege.ActivityQualifiers property that contains a value, the
542 corresponding index of the CIM_Privilege.Activities property shall contain the value 7 (Execute).

543 **7.5.3.2.2 CIM_Privilege.ActivityQualifiers**

544 The CIM_Privilege.ActivityQualifiers property shall contain the values {"cd", "exit", "help", "show", "version",
545 "reset", "start", "stop", "set", "load", "dump"}. The CIM_Privilege.ActivityQualifiers property shall not contain
546 the values {"create", "delete"}. The CIM_Privilege.ActivityQualifiers property may contain additional
547 values.

548 **7.5.4 Representing the CLP Read Only Role and Privileges**

549 This section details the requirements for representing an SM CLP Read Only Role.

550 **7.5.4.1 Read Only Role**

551 There shall be an instance of CIM_Role implemented as specified in section 10.29.

552 **7.5.4.2 Read Only Privileges**

553 There shall be an instance of CIM_Privilege associated with the instance of CIM_Role defined in
554 section 7.5.4.1 through the CIM_MemberOfCollection association. There shall not be any other instances
555 of CIM_Privilege associated with instance of CIM_Role defined in section 7.5.4.1. The instance of
556 CIM_Privilege shall be implemented as specified in section 10.23.

557 **7.5.4.2.1 CIM_Privilege.Activities**

558 For each array index of the CIM_Privilege.ActivityQualifiers property that contains a value, the
559 corresponding index of the CIM_Privilege.Activities property shall contain the value 7 (Execute).

560 **7.5.4.2.2 CIM_Privilege.ActivityQualifiers**

561 The CIM_Privilege.ActivityQualifiers property shall contain the values {"cd", "exit", "help", "show",
562 "version"}. The CIM_Privilege.ActivityQualifiers property shall not contain the values {"reset", "start",
563 "stop", "set", "load", "dump", "create", "delete"}. The CIM_Privilege.ActivityQualifiers property may contain
564 additional values.

565 **7.5.5 CLP Security Principals**

566 An instance of CIM_Identity representing the security principal of the CLP user shall exist or be
567 instantiated when a CLP session is established and a CLP user has been authorized.

568 **7.5.5.1 Identity Context**

569 An instance of CIM_Identity that represents the security principal of a CLP User shall be associated with
570 the Central Instance of this profile through the CIM_IdentityContext association.

571 **7.5.5.2 Security Principal for Session — Optional**

572 When the optional behavior specified in section 7.3 is implemented, the instance of CIM_Identity that
573 corresponds to the security principal that was authenticated when the transport session underlying the
574 CLP session was established shall be associated with the instance of CIM_TelnetProtocolEndpoint or
575 CIM_SSHProtocolEndpoint that represents the transport session through the CIM_ConcreteDependency
576 association. The value of the CIM_ConcreteDependency.Antecedent property shall be a reference to the
577 instance of CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint.

578 **7.5.6 CLP Authorized Role Management**

579 There shall be an instance of CIM_RoleBasedAuthorizationService associated to each instance of
580 CIM_Role implemented as defined in sections 7.5.2, 7.5.3, and 7.5.4 through the
581 CIM_ServiceAffectsElement association.

- 582 The instance of CIM_RoleBasedManagementCapabilities associated with the instance of
583 CIM_RoleBasedAuthorizationService shall follow these requirements:
- 584 • The CIM_RoleBasedManagementCapabilities.SharedPrivilegeSupported property shall be set
585 FALSE.
- 586 • The CIM_RoleBasedManagementCapabilities.ActivitiesSupported property shall have value of
587 {7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute),
588 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute)}.
- 589 • The CIM_RoleBasedManagementCapabilities.ActivityQualifiersSupported property shall have
590 value {"cd", "exit", "help", "show", "version", "reset", "start", "stop", "set", "load", "dump", "create",
591 "delete"}.
- 592 • The CIM_RoleBasedManagementCapabilities.QualifierFormatsSupported property shall have
593 value {9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line
594 Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line
595 Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line
596 Instruction), 9 (Command Line Instruction)}.
- 597 • The SupportedMethods property shall be a zero length array.

599 **7.5.6.1 OEM Extensions**

600 The CIM_RoleBasedManagementCapabilities.ActivityQualifiersSupported property shall contain a value
601 that corresponds to each OEM SM CLP verb supported. The corresponding array index of the
602 CIM_RoleBasedManagementCapabilities.ActivitiesSupported property shall have value of {7(Execute)}
603 and the corresponding array index of the
604 CIM_RoleBasedManagementCapabilities.QualifierFormatsSupported property shall have value {9
605 (Command Line Instruction)}.

606 **7.6 CLP Operations and Queue**

607 This section describes the requirements for modeling the CLP job queue and operations.

608 **7.6.1 Job Queue**

609 There shall be exactly one instance of CIM_JobQueue associated with the Scoping Instance through the
610 CIM_HostedJobDestination. The Central Instance shall be associated with the CIM_JobQueue instance
611 through the CIM_ServiceAffectsElement association.

612 **7.6.2 Representing a CLP Operation**

613 Each CLP command job shall be modeled with an instance of CIM_ConcreteJob. The instance of
614 CIM_ConcreteJob shall be associated with the instance of CIM_JobQueue through the
615 CIM_JobDestinationJobs association.

616 When an instance of CIM_ConcreteJob is created to represent a CLP operation, the initial value of the
617 CIM_ConcreteJob.TimeBeforeRemoval property shall be the datetime representation of the value of the
618 CIM_CLPProtocolEndpoint.KeepTime property of the instance of CIM_CLPProtocolEndpoint that
619 represents the session through which the CLP command that resulted in the CLP operation was received.

620 The killing of a job may be supported; this is job and implementation specific. See section 8.2.

621 7.6.2.1 CIM_ConcreteJob.JobState

622 When a CIM_ConcreteJob is created the JobState property shall have the value 4 (Running). When a
623 CLP operation completes successfully, the JobState property shall have the value 7 (Completed). When a
624 CLP operation fails, the JobState property shall have the value 10 (Exception). When a CLP Operation is
625 killed using the CIM_ConcreteJob.RequestStateChange() method where the RequestedState parameter
626 has the value 5 (Kill), the JobState property shall have the value 9 (Killed). When a CLP Operation is
627 terminated using the CIM_ConcreteJob.RequestStateChange() method where the RequestedState
628 parameter has the value 5 (Terminate), the JobState property shall have the value 8 (Terminated).

629 7.6.2.2 CIM_ConcreteJob.ElementName

630 The ElementName property of an instance of CIM_ConcreteJob shall contain the Job Id for the modeled
631 CLP operation. The property shall contain one or more digits matching the pattern "(1234567890)+".

632 7.6.3 Representing a CLP Operation Error

633 When a CLP operation fails, the CIM_ConcreteJob.JobState property of the CIM_ConcreteJob that
634 represents the operation shall have the value 10 (Exception). The CIM_ConcreteJob.OperationalStatus
635 property shall have a value other than 2 (Ok). The CIM_ConcreteJob.GetError() method can be used to
636 retrieve an instance of CIM_Error detailing the cause of the operation failure. See section 8.3 for
637 information on CIM_ConcreteJob.GetError().

638 7.6.3.1 Representing a Message (Optional)

639 An instance of CIM_Error may convey a Standard Message or vendor defined message. When an
640 instance of CIM_Error is used to convey a message, the OwningEntity, MessageID, and Message
641 properties shall be implemented and the MessageArguments property may be implemented. When an
642 instance of CIM_Error is not used to convey a message, the OwningEntity, MessageID, Message, and
643 MessageArguments properties shall not be implemented.

644 8 Methods

645 This section details the requirements for supporting intrinsic operations and extrinsic methods for the CIM
646 elements defined by this profile.

647 8.1 CIM_ProtocolService.RequestStateChange()

648 CIM_ProtocolService.RequestStateChange() method invocation will change the element's state to the
649 value specified in the RequestedState parameter. The Enabled and Disabled values of the
650 RequestedState parameter correspond to enabling or disabling the functionality represented by the
651 instance of CIM_ProtocolService. A value of 2 (Enabled) shall correspond to a request to enable the
652 functionality. A value of 3 (Disabled) shall correspond to a request to disable the functionality. A value of
653 11 (Reset) shall initiate a reset of the CLP service.

654 See section 7.1.3 for information about the effect of this method on the RequestedState property.

655 The method shall be considered successful if the availability of the functionality upon completion of the
656 method corresponds to the desired availability indicated by the RequestedState parameter. It is not
657 necessary that an actual change in state occur for the method to be considered successful. It is sufficient
658 that the resultant state be equal to the requested state. Upon successful completion of the method, the
659 Return Value shall be zero.

660 See section 7.1.3.4 for information about the effect of this method on the EnabledState property.

661 Detailed requirements of the RequestStateChange() method are specified in Table 2 and Table 3.

662 No standard messages are defined.

663 Invoking the CIM_ProtocolService.RequestStateChange() method multiple times could result in earlier
 664 requests being overwritten or lost.

665 **Table 2 – CIM_ProtocolService.RequestStateChange() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is unsupported in the implementation.
2	Error occurred.
0x1000	Job started: REF returned to started CIM_ConcreteJob.

666 **Table 3 – CIM_ProtocolService.RequestStateChange() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN,	RequestedState	uint16	Valid state values : 2 (Enabled) 3 (Disabled) 11 (Reset)
OUT	Job	CIM_ConcreteJob REF	Returned if job started
IN	TimeoutPeriod	datetime	Client specified maximum amount of time the transition to a new state is supposed to take: 0 or NULL – No time requirements <interval> – Maximum time allowed

667 **8.1.1 CIM_ProtocolService.RequestStateChange() ConditionalSupport**

668 When the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least
 669 one value, the CIM_ProtocolService.RequestStateChange() method shall be implemented and
 670 supported. The CIM_ProtocolService.RequestStateChange() method shall not return a value of 1
 671 (Unsupported).

672 **8.2 CIM_ConcreteJob.RequestStateChange()**

673 The CIM_ConcreteJob.RequestStateChange() may be used to request that the CLP operation modeled
 674 by the CIM_ConcreteJob instance is terminated. See section 7.6.2.1 for the effect of this method on the
 675 JobState property.

676 No standard messages are defined.

677 **Table 4 – CIM_ConcreteJob.RequestStateChange() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is unsupported in the implementation.
2	Error occurred.

678 **Table 5 – CIM_ConcreteJob.RequestStateChange() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	RequestedState	uint16	Valid state values : 5 (Kill) 4 (Terminate)

679 Note that the use of the TimeoutPeriod parameter is not supported.

680 **8.3 CIM_ConcreteJob.GetError()**681 The GetError() method is used to retrieve the instance of CIM_Error that contains details of why the
682 operation failed if it did. This method shall be supported when the CIM_ConcreteJob.JobStatus has the
683 value 10 (Exception) and shall not return a value of 1.

684 No standard messages are defined.

685 **Table 6 – CIM_ConcreteJob.GetError() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is unsupported in the implementation.
2	Error occurred.

686 **Table 7 – CIM_ConcreteJob.GetError() Method: Parameters**

Qualifiers	Name	Type	Description/Values
OUT	Error	String	Embedded instance of CIM_Error

687 **8.4 Profile Conventions for Operations**688 For each profile class (including associations), the implementation requirements for operations, including
689 those in the following default list, are specified in class-specific subclauses of this clause.

690 The default list of operations is as follows:

- 691 • GetInstance
- 692 • Associators
- 693 • AssociatorNames
- 694 • References
- 695 • ReferenceNames
- 696 • EnumerateInstances
- 697 • EnumerateInstanceNames

698 **8.5 CIM_BindsTo**699 Table 8 lists implementation requirements for operations. If implemented, these operations shall be
700 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 8, all operations in
701 the default list in 8.4 shall be implemented as defined in [DSP0200](#).

702 NOTE: Related profiles may define additional requirements on operations for the profile class.

703

Table 8 – Operations: CIM_BindsTo

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

704 **8.6 CIM_CLPCapabilities**

705 All operations in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

706 NOTE: Related profiles may define additional requirements on operations for the profile class.

707 **8.7 CIM_CLPSettingData**

708 Table 9 lists implementation requirements for operations. If implemented, these operations shall be
 709 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 9, all operations in
 710 the default list in 8.4 shall be implemented as defined in [DSP0200](#).

711 NOTE: Related profiles may define additional requirements on operations for the profile class.

712 Table 9 – Operations: CIM_CLPSettingData

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.7.1.	None

713 **8.7.1 CIM_CLPSettingData – ModifyInstance**

714 When the CIM_CLPSettingData instance is associated with the CIM_ProtocolService instance through an
 715 instance of CIM_ElementSettingData and the value of the IsDefault property of the
 716 CIM_ElementSettingData instance that associates the CIM_CLPSettingData with the
 717 CIM_ProtocolService has a value of 1 (Is Default), the ModifyInstance operation shall not be supported.

718 When the CIM_CLPSettingData instance is not associated with an instance of CIM_ProtocolService
 719 through an instance of CIM_ElementSettingData where the IsDefault property has a value of 1 (Is
 720 Default), the ModifyInstance operation may be supported for the CIM_CLPSettingData instance.

721 **8.8 CIM_CLPProtocolEndpoint**

722 Table 10 lists implementation requirements for operations. If implemented, these operations shall be
 723 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 10, all operations
 724 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

725 NOTE: Related profiles may define additional requirements on operations for the profile class.

726 Table 10 – Operations: CIM_CLPProtocolEndpoint

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.8.1.	None
DeleteInstance	Optional. See section 8.8.2.	None

727 **8.8.1 ModifyInstance**

728 The ModifyInstance operation may be supported for an instance of CIM_CLPProtocolEndpoint. When the
 729 ModifyInstance operation is supported for an CIM_CLPProtocolEndpoint instance, the ModifyInstance
 730 operation shall not modify the following properties:

- 731 • NameFormat
 732 • ProtocolIFTType
 733 • OtherTypeDescription

734 **8.8.2 DeleteInstance**

735 The DeleteInstance operation may be supported for instances of CIM_CLPProtocolEndpoint. When the
 736 DeleteInstance operation is invoked against an instance, the corresponding CLP session shall be
 737 terminated prior to deleting the CIM_CLPProtocolEndpoint instance. The implementation shall also
 738 remove any association instances that reference the CIM_CLPProtocolEndpoint.

739 **8.9 CIM_ConcreteJob**

740 Table 11 lists implementation requirements for operations. If implemented, these operations shall be
 741 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 11, all operations
 742 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

743 NOTE: Related profiles may define additional requirements on operations for the profile class.

744 **Table 11 – Operations: CIM_ConcreteJob**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.9.1.	None
DeleteInstance	Optional. See section 8.9.2.	None

745 **8.9.1 ModifyInstance**

746 The ModifyInstance operation may be supported for an instance of CIM_ConcreteJob. When the
 747 ModifyInstance operation is supported, the TimeBeforeRemoval and DeleteOnCompletion properties
 748 shall be writable.

749 **8.9.2 DeleteInstance**

750 The DeleteInstance operation may be supported for an instance of CIM_ConcreteJob. When the
 751 DeleteInstance operation is supported, the DeleteInstance operation shall fail when the
 752 CIM_ConcreteJob.JobStatus property has the value 4 (Running).

753 **8.10 CIM_ElementCapabilities**

754 Table 12 lists implementation requirements for operations. If implemented, these operations shall be
 755 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 12, all operations
 756 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

757 NOTE: Related profiles may define additional requirements on operations for the profile class.

758

Table 12 – Operations: CIM_ElementCapabilities

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

759 **8.11 CIM_ElementSettingData**

760 Table 13 lists implementation requirements for operations. If implemented, these operations shall be
 761 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 13, all operations
 762 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

763 NOTE: Related profiles may define additional requirements on operations for the profile class.

764 **Table 13 – Operations: CIM_ElementSettingData**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.11.1.	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

765 **8.11.1 CIM_ElementSettingData – ModifyInstance**

766 When an instance of CIM_ElementSettingData associates an instance of CIM_CLPSettingData with an
 767 instance of CIM_CLPProtocolEndpoint, the following rules shall govern the behavior of the
 768 ModifyInstance operation:

- 769 • The ModifyInstance operation shall not allow the IsDefault property to be modified.
- 770 • The ModifyInstance operation shall not allow the IsCurrent property to be modified.
- 771 • When the ModifyInstance operation is used to modify the IsNext property to have a value of 1
 772 (Is Next), the ModifyInstance operation shall implement the following behavior:
 - 773 – The ModifyInstance operation shall find all other instances of CIM_ElementSettingData
 774 that associate an CIM_CLPSettingData instance with the CIM_CLPProtocolEndpoint
 775 instance referenced by the target instance of CIM_ElementSettingData.
 - 776 – For each instance of CIM_ElementSettingData found, the ModifyInstance operation shall
 777 modify the value of its IsNext property to have a value of 2 (Is Not Next).

778 **8.12 CIM_Error**

779 The sole usage of CIM_Error in this profile is as a template for an embedded instance. Therefore, none of
 780 the operations in the default list in 8.4 shall be supported as described by [DSP0200](#).

781 **8.13 CIM_HostedAccessPoint**

782 Table 14 lists implementation requirements for operations. If implemented, these operations shall be
 783 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 14, all operations
 784 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

785 NOTE: Related profiles may define additional requirements on operations for the profile class.

786 **Table 14 – Operations: CIM_HostedAccessPoint**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

787 **8.14 CIM_HostedJobDestination**

788 Table 15 lists implementation requirements for operations. If implemented, these operations shall be
 789 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 15, all operations
 790 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

791 NOTE: Related profiles may define additional requirements on operations for the profile class.

792 **Table 15 – Operations: CIM_HostedJobDestination**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

793 **8.15 CIM_HostedService**

794 Table 16 lists implementation requirements for operations. If implemented, these operations shall be
 795 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 16, all operations
 796 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

797 NOTE: Related profiles may define additional requirements on operations for the profile class.

798 **Table 16 – Operations: CIM_HostedService**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

799 **8.16 CIM_JobDestinationJobs**

800 Table 17 lists implementation requirements for operations. If implemented, these operations shall be
 801 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 17, all operations
 802 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

803 NOTE: Related profiles may define additional requirements on operations for the profile class.

804 **Table 17 – Operations: CIM_JobDestinationJobs**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

805 **8.17 CIM_JobQueue**

806 All operations in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

807 NOTE: Related profiles may define additional requirements on operations for the profile class.

808 **8.18 CIM_OwningJobElement**

809 Table 18 lists implementation requirements for operations. If implemented, these operations shall be
 810 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 18, all operations
 811 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

812 NOTE: Related profiles may define additional requirements on operations for the profile class.

813 **Table 18 – Operations: CIM_OwningJobElement**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

814 **8.19 CIM_ProtocolService**

815 Table 19 lists implementation requirements for operations. If implemented, these operations shall be
 816 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 19, all operations
 817 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

818 NOTE: Related profiles may define additional requirements on operations for the profile class.

819 **Table 19 – Operations: CIM_ProtocolService**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.19.1.	None

820 **8.19.1 CIM_ProtocolService — ModifyInstance**

821 When the ElementNameEditSupported property of the CIM_CLPCapabilities has a value of TRUE, the
 822 ModifyInstance operation shall allow the value of the ElementName property of the CIM_ProtocolService
 823 instance to be modified. The ModifyInstance operation shall enforce the length restriction specified in the
 824 MaxElementNameLen property of the CIM_CLPCapabilities.

825 When the ElementNameEditSupported property of the CIM_CLPCapabilities has a value of FALSE, the
 826 ModifyInstance operation shall not change the value of the ElementName property of the
 827 CIM_ProtocolService instance.

828 **8.20 CIM_ProvidesEndpoint**

829 Table 20 lists implementation requirements for operations. If implemented, these operations shall be
 830 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 20, all operations
 831 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

832 NOTE: Related profiles may define additional requirements on operations for the profile class.

833 **Table 20 – Operations: CIM_ProvidesEndpoint**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

834 **8.21 CIM_ServiceAccessBySAP**

835 Table 21 lists implementation requirements for operations. If implemented, these operations shall be
 836 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 21, all operations
 837 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

838 NOTE: Related profiles may define additional requirements on operations for the profile class.

839 **Table 21 – Operations: CIM_ServiceAccessBySAP**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

840 **8.22 ServiceAffectsElement**

841 Table 22 lists implementation requirements for operations. If implemented, these operations shall be
 842 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 22, all operations
 843 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

844 NOTE: Related profiles may define additional requirements on operations for the profile class.

845

Table 22 – Operations: CIM_ServiceAffectsElement

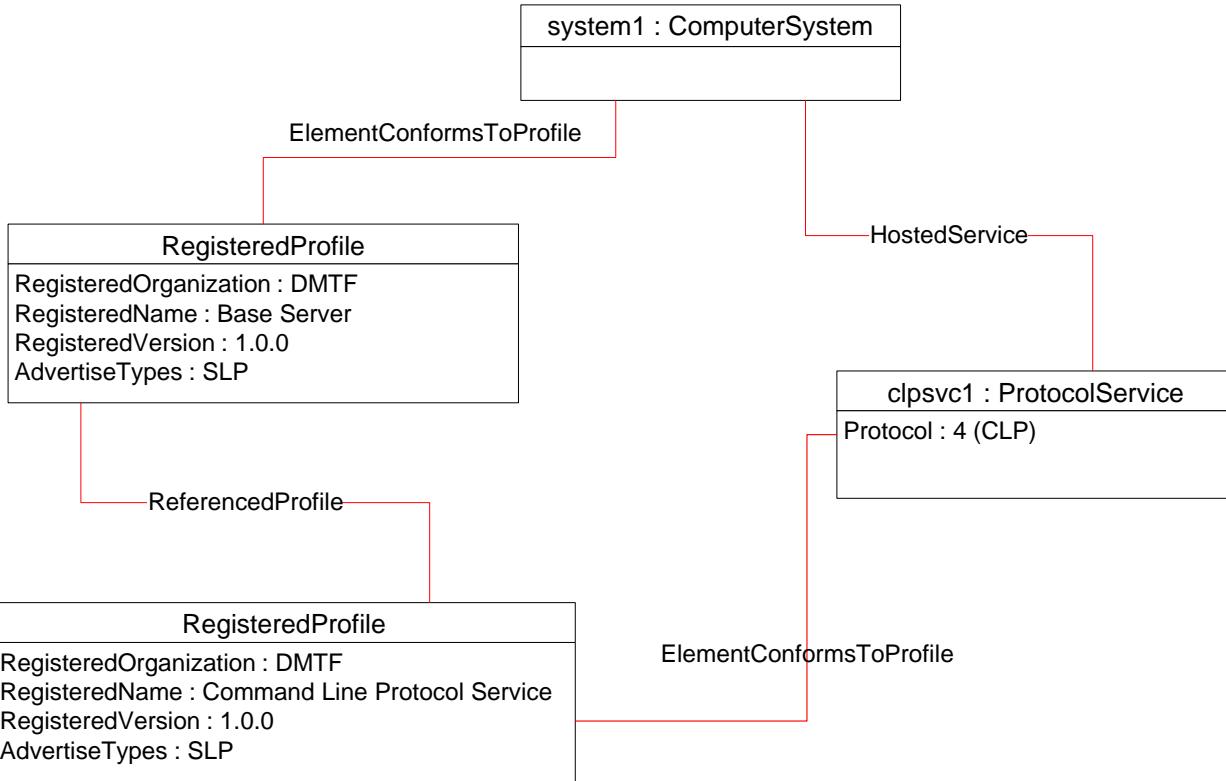
Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

846 **9 Use Cases**

847 The following sections outline common use cases for client interaction with the *Command Line Protocol*
 848 *Service Profile*.

849 **9.1 Object Diagrams**

850 The object diagram in Figure 2 shows how instances of CIM_RegisteredProfile are used to identify the
 851 version of the *Command Line Protocol Service Profile* with which an instance of CIM_ProtocolService and
 852 its associated instances are conformant. An instance of CIM_RegisteredProfile exists for each profile that
 853 is instrumented in the system. One instance of CIM_RegisteredProfile identifies the “DMTF Base Server
 854 *Profile* version 1.0”. The other instance identifies the “DMTF *Command Line Protocol Service Profile*
 855 version 1.0”. The CIM_ProtocolService instance is scoped to an instance of CIM_ComputerSystem. This
 856 instance of CIM_ComputerSystem is conformant with the DMTF *Base Server Profile* version 1.0 as
 857 indicated by the CIM_ElementConformsToProfile association to the CIM_RegisteredProfile instance. This
 858 implementation is using the Central Class Methodology. The CIM_ProtocolService instance is conformant
 859 with this profile as indicated by the CIM_ElementConformsToProfile association between the instance
 860 and the instance of CIM_RegisteredProfile that identifies this profile.

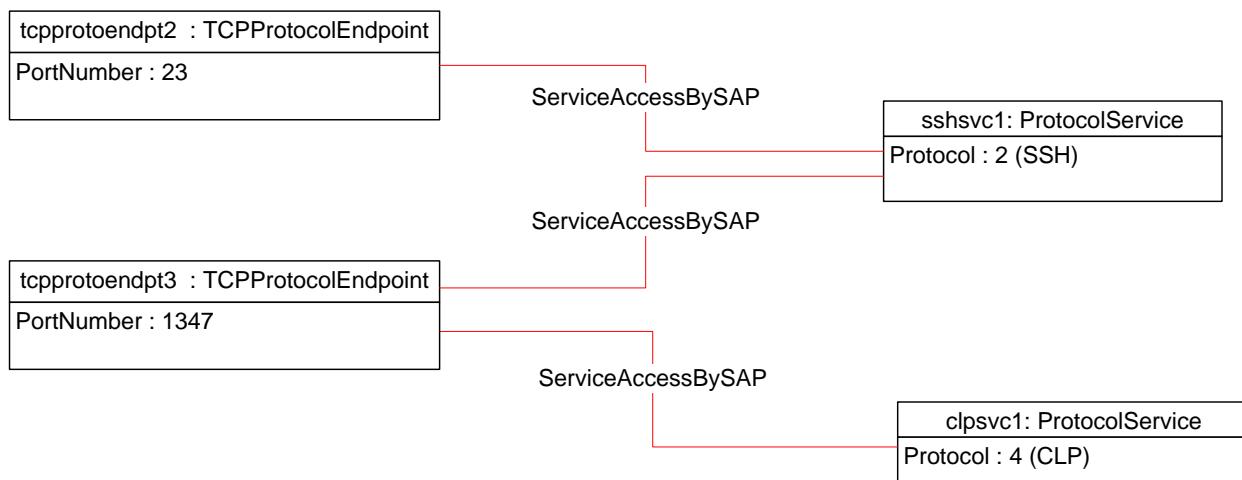


861

862

Figure 2 – Registered Profile

863 Figure 3 is an object diagram illustrating a CLP service accessible through SSH. The SSH service is
 864 listening on ports 23 and 1347. The CLP service is accessible through SSH sessions established by
 865 connecting to port 1347. This is indicated by the CIM_ServiceAccessBySAP between the clpsvc1 and
 866 tcpprotoendpt3. The object diagram does not indicate whether the CLP session is automatically
 867 initiated upon establishment of the SSH session or requires a manual step on the part of the user.

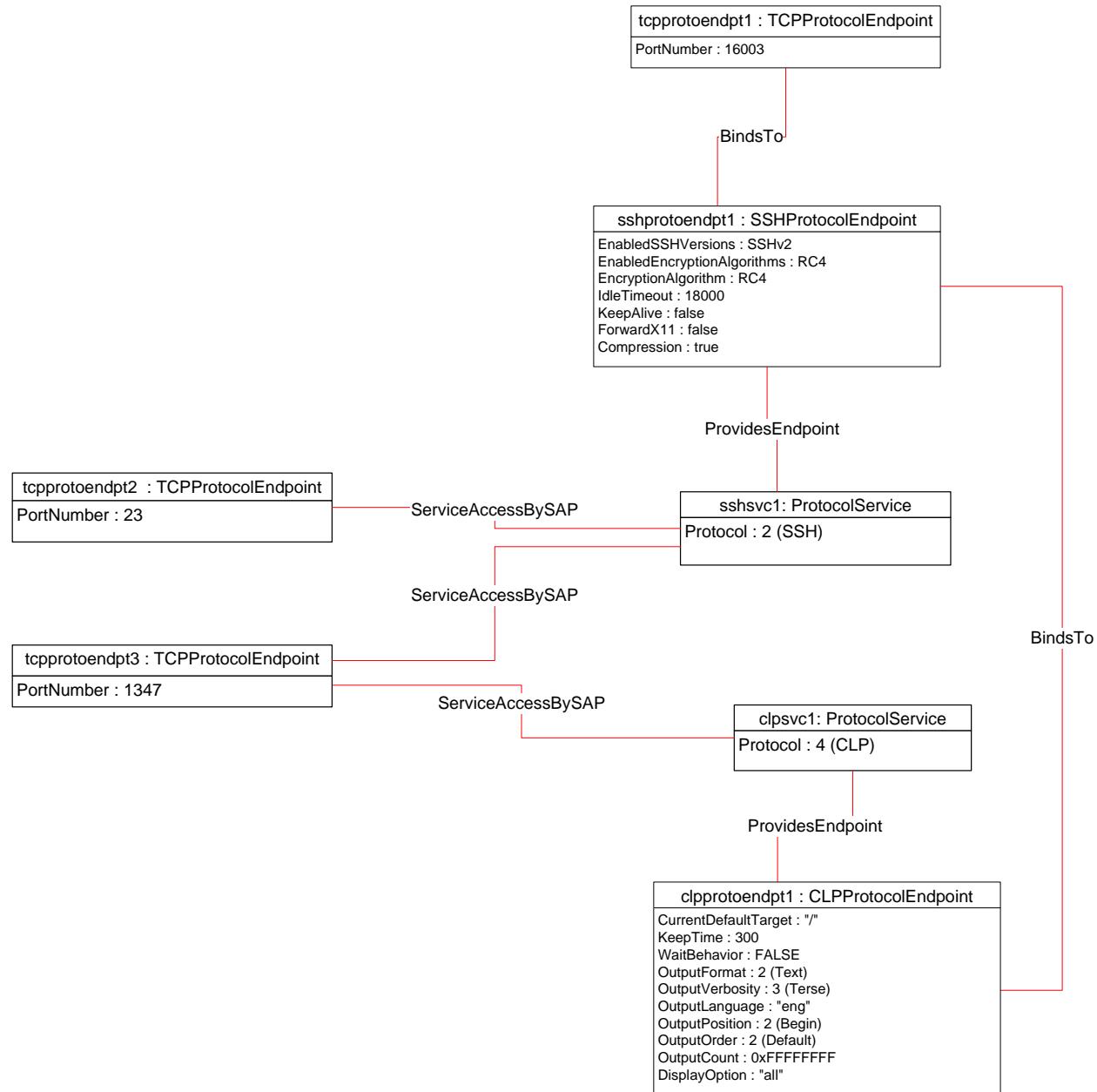


868

869

Figure 3 – CLP Service Accessible via SSH

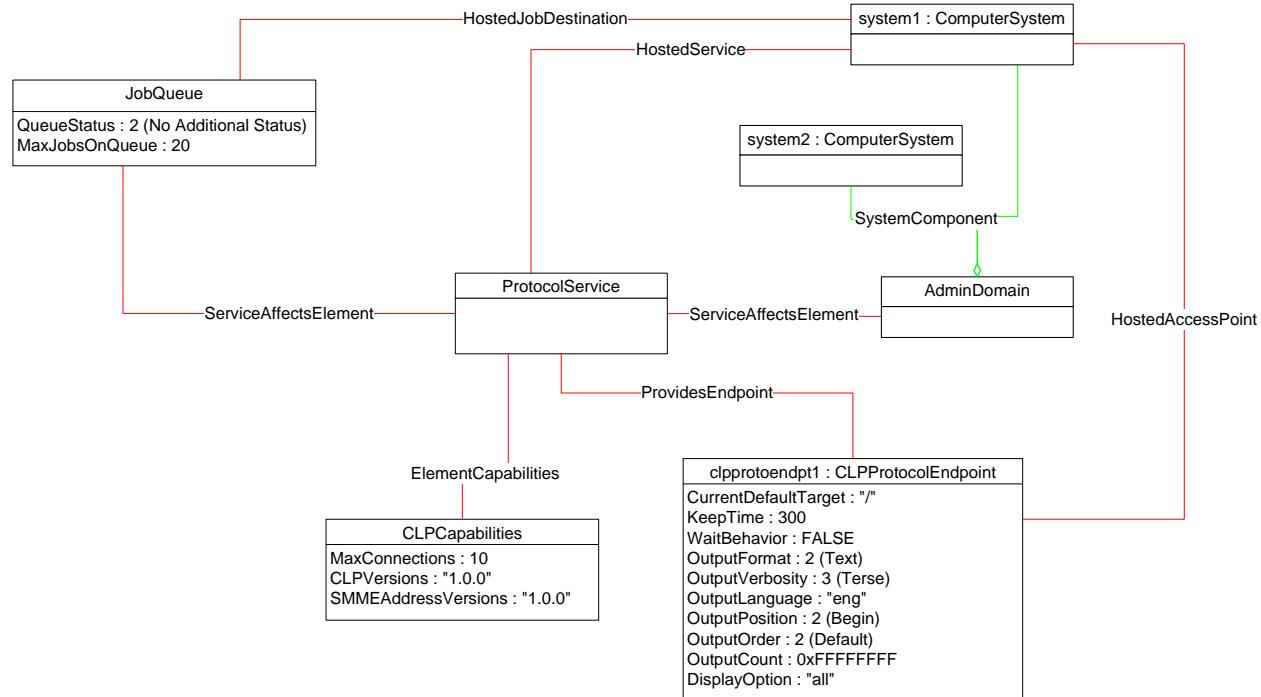
870 Figure 4 illustrates a single CLP session (`clppprotoendpt1`) established through an SSH session
 871 (`sshprotoendpt1`). This is indicated by the CIM_BindsTo association that references the two instances.



872

Figure 4 – One Active Session

874 The object diagram in Figure 5 provides a more complete view of a CLP implementation that is capable of
 875 managing two systems. The manageable systems are each represented by an instance of
 876 CIM_ComputerSystem aggregated into the CIM_AdminDomain instance. The
 877 CIM_ServiceAffectsElement association identifies the CIM_AdminDomain as the management domain of
 878 this CLP service. A single CLP session is active and there are no CLP operations currently executing.

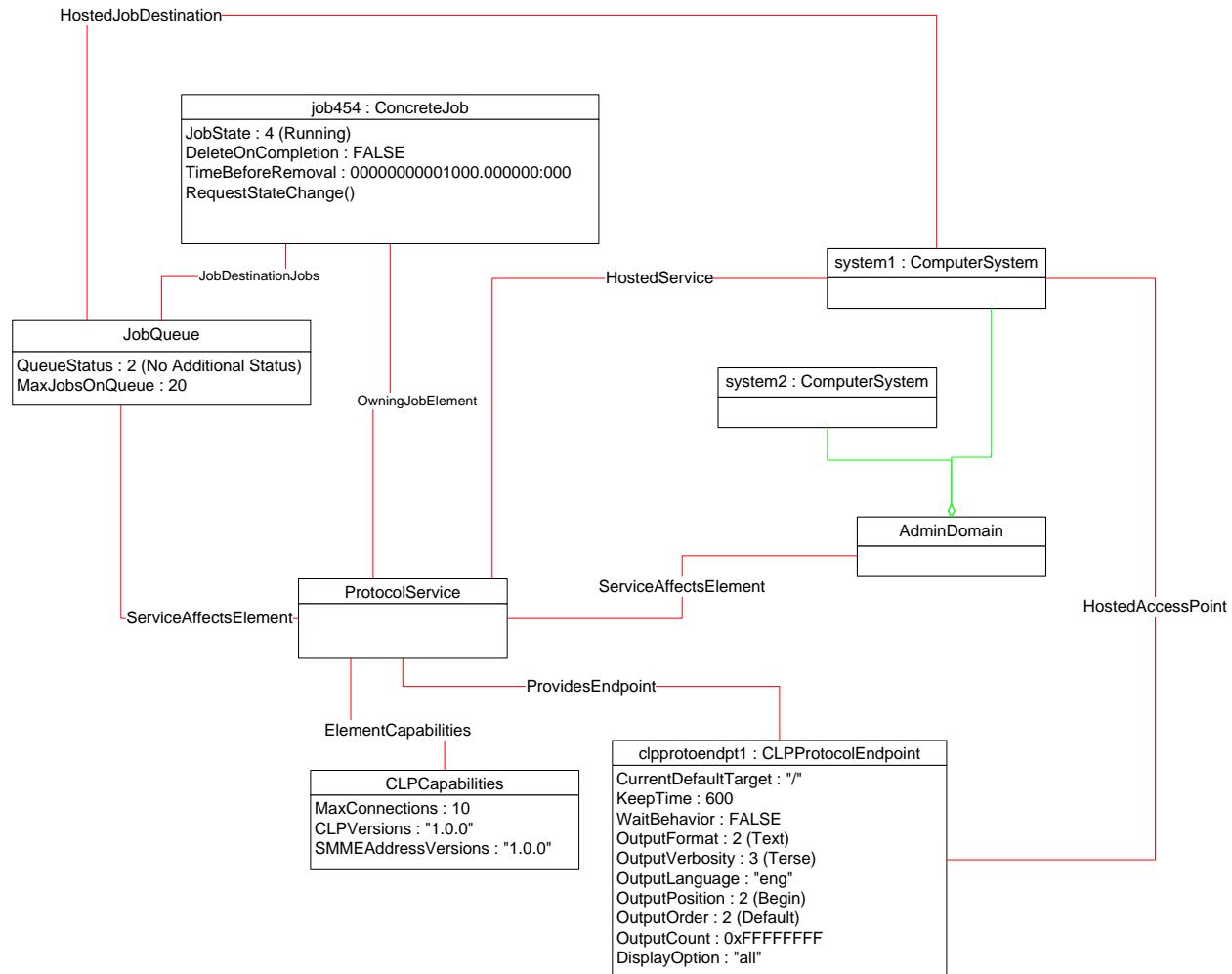


879

880

Figure 5 – Session with Admin Domain and Job Queue

881 Figure 6 is an object diagram for the same CLP service as in Figure 5. Note that the KeepTime property
 882 on clpprotoendpt1 has been changed from the specification default value of 300 seconds to a new
 883 value of 600 seconds prior to the initiation of the CLP operation which is modeled by job454. The CLP
 884 operation represented by job454 is currently executing as indicated by the value of the JobState
 885 property.

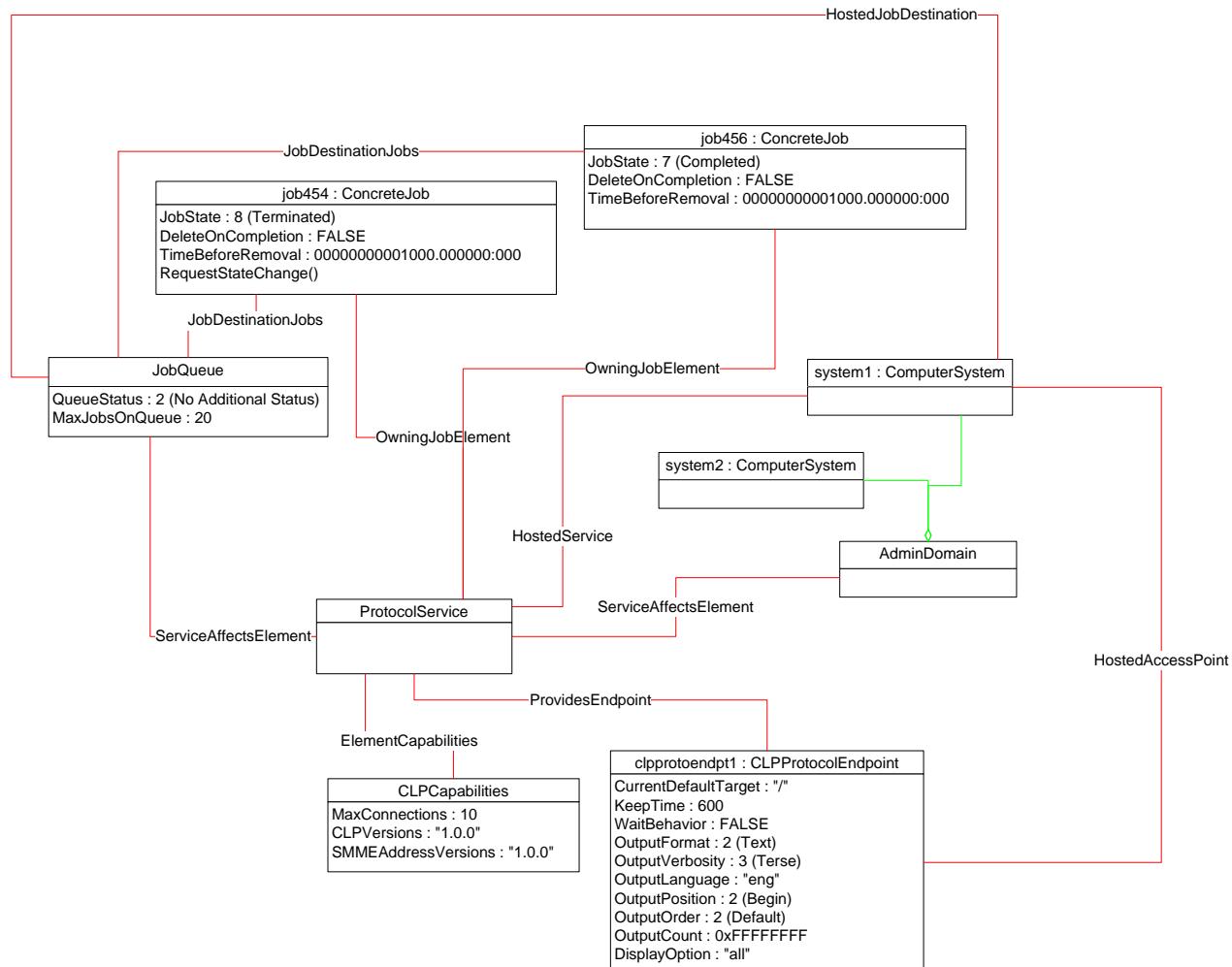


886

887

Figure 6 – Single Operation Executing

888 Figure 7 is an object diagram for the same CLP service as in Figure 6. Prior to the completion of the CLP
 889 operation represented by job454 another CLP command was issued to terminate the first operation. The
 890 latter CLP command resulted in the CLP operation modeled by job456. This operation successfully
 891 completed as indicated by the value of 7 (Completed) for its JobState property. The fact that the
 892 previously initiated CLP operation is reflected by a value of 8 (Terminated) for the JobState property of
 893 job454.

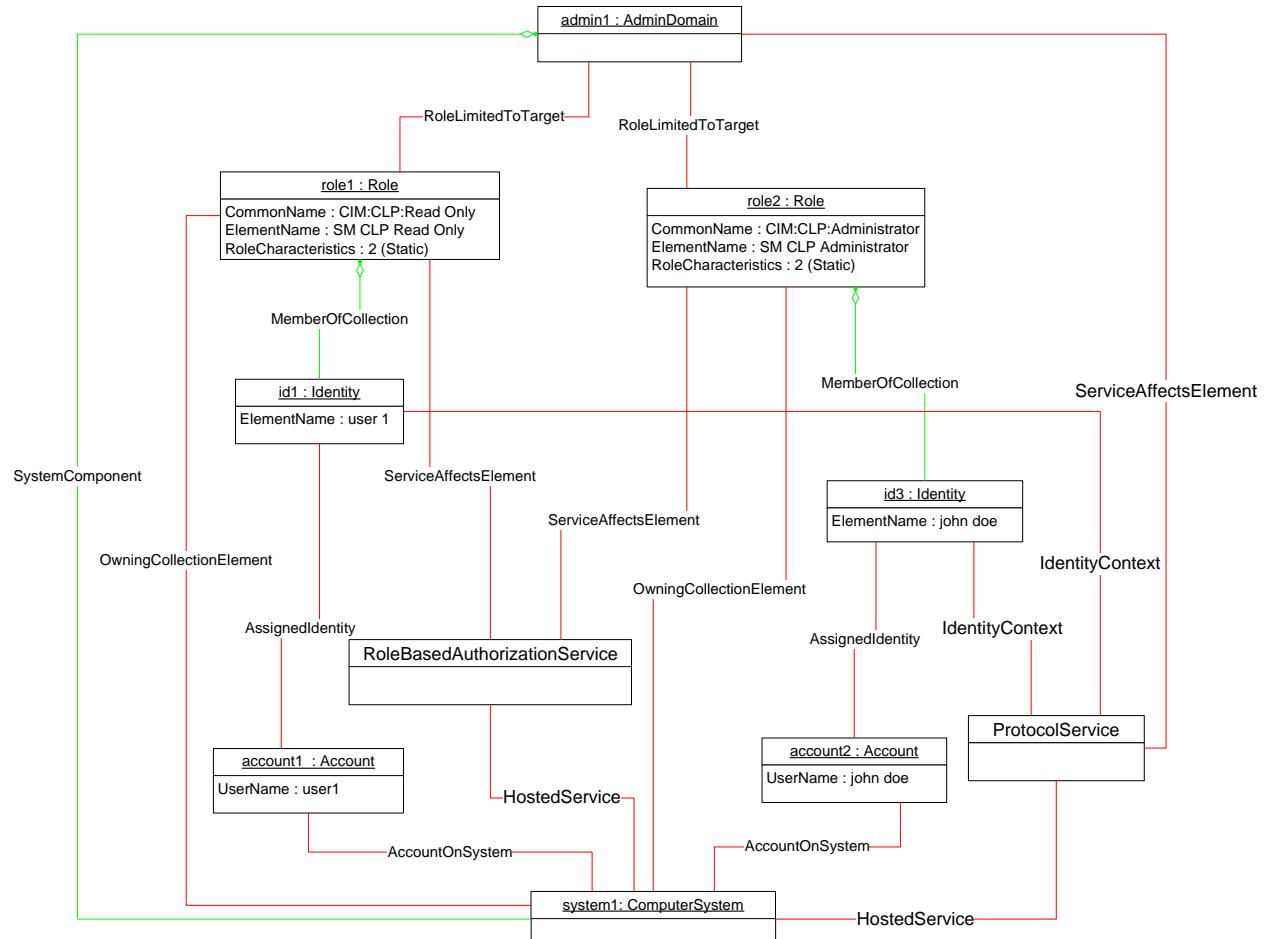


894

895

Figure 7 – CLP Command Used to Terminate Running CLP Operation

896 Figure 8 is an object diagram showing a subset of an implementation related to management of CLP
 897 roles. There are two instances of CIM_Role representing the Administrator and Read Only roles. There
 898 are two accounts on the system. Each account has a corresponding security principal represented by
 899 CIM_Identity whose context includes usage by the CLP service.

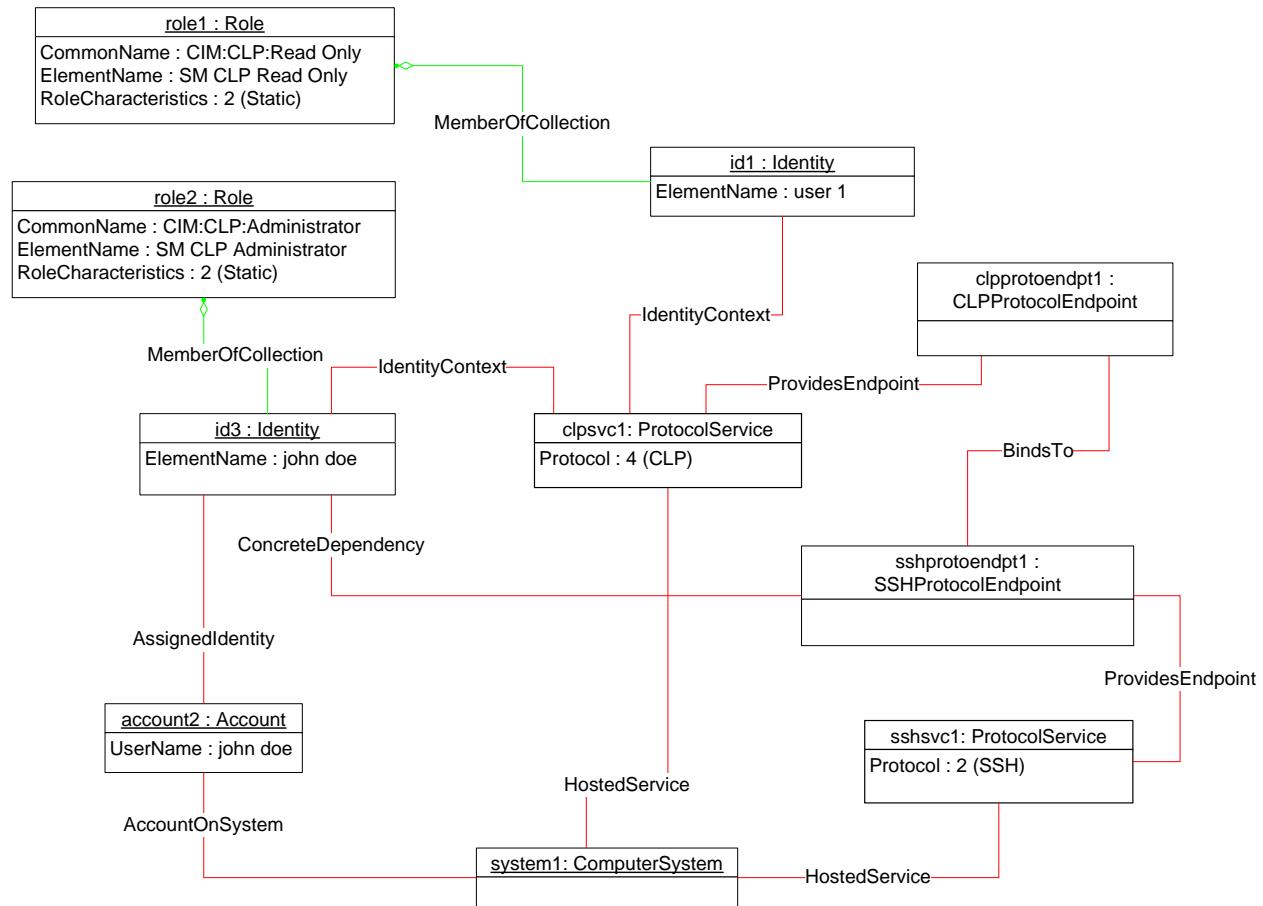


900

901

Figure 8 – CLP Roles

902 Figure 9 is an object diagram showing an active CLP session that has been established over SSH. id3
 903 is the corresponding security principal that resulted from the credentials provided when the SSH session
 904 was established. This is indicated by the CIM_ConcreteDependency association between the
 905 sshprotoendpt1 and id3. The credentials provided correspond to account2. This is indicated by the
 906 CIM_AssignedIdentity instance that associated id3 with account2. The user of the CLP session has
 907 Administrator rights. This is determined by association traversal from clpprotoendpt1 to
 908 sshprotoendpt1 through the CIM_BindsTo association, then to id3 across the
 909 CIM_ConcreteDependency association, then to role2 through the CIM_MemberOfCollection
 910 association. The account for id1 is not shown.



911

912 **Figure 9 – Active Session with Security Principal**

913 **9.2 Modifying Active Session Settings**

914 A user can find the active sessions for a CLP service and modify their configuration as follows:

- 915 1) Find an instance of CIM_CLPProtocolEndpoint associated with the CIM_ProtocolService
 916 through an instance of CIM_ProvidesEndpoint.
- 917 2) Modify the properties of the CIM_CLPProtocolEndpoint as desired.

918 **9.3 Disabling the CLP Service**

919 If an implementation supports disabling the CLP service, a user can disable the CLP service by invoking
 920 the RequestStateChange() method on CIM_ProtocolService instance with a value of Disabled for the
 921 RequestedState parameter.

922 **9.4 Determining the CLP Service Capabilities**

923 A user can determine the capabilities of the CLP service as follows:

- 924 1) Find the instance of CIM_CLPCapabilities associated with the CIM_ProtocolService through an
 925 instance of CIM_ElementCapabilities.
- 926 2) View the properties of the CIM_CLPCapabilities instance to see the supported function.

927 **9.5 Determining If ElementName Can Be Modified**

928 For a given instance of CIM_ProtocolService, a client can determine whether it can modify the
 929 ElementName as follows:

- 930 1) Find the CIM_CLPCapabilities instance that is associated with the target instance.
- 931 2) Query the value of the ElementNameEditSupported property of the CIM_CLPCapabilities
 932 instance. If the value is TRUE, the client can modify the ElementName property of the target
 933 instance.

934 **9.6 Determining If State Management Is Supported**

935 For a given instance of CIM_ProtocolService, a client can determine whether state management is
 936 supported as follows:

- 937 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
 938 CIM_LANEndpoint instance.
- 939 2) Query the value of the RequestedStatesSupported property. If at least one value is specified,
 940 state management is supported.

941 **10 CIM Elements**

942 Table 23 shows the instances of CIM Elements for this profile. Instances of these CIM Elements shall be
 943 implemented as described in Table 23. Section 7 may impose additional requirements on these elements.

944 **Table 23 – CIM Elements: Command Line Protocol Service Profile**

Element Name	Requirement	Notes
Classes		
CIM_BindsTo	Optional	See section 10.1.
CIM_CLPCapabilities	Mandatory	See section 10.2.
CIM_CLPProtocolEndpoint	Mandatory	See section 10.3
CIM_CLPSettingData	Optional	See sections 10.4 and 10.5.
CIM_ConcreteDependency	Optional	See section 10.6.
CIM_ConcreteJob	Mandatory	See section 10.7.
CIM_ElementCapabilities	Mandatory	See section 10.8.
CIM_ElementSettingData	Optional	See sections 10.9 and 10.10.
CIM_Error	Mandatory	See section 10.11.

Element Name	Requirement	Notes
CIM_HostedAccessPoint	Mandatory	See section 10.12.
CIM_HostedJobDestination	Mandatory	See section 10.13.
CIM_HostedService	Mandatory	See section 10.14.
CIM_JobDestinationJobs	Conditional	See section 10.15.
CIM_JobQueue	Mandatory	See section 10.17.
CIM_OwningCollectionElement	Mandatory	See section 10.18.
CIM_OwningJobElement	Conditional	See sections 10.19 and 10.20.
CIM_Privilege	Mandatory	See sections 10.21, 10.22, and 10.23.
CIM_ProtocolService	Mandatory	See section 10.24.
CIM_ProvidesEndpoint	Mandatory	See section 10.25.
CIM_RegisteredProfile	Mandatory	See section 10.26.
CIM_Role	Mandatory	See sections 10.27, 10.28, and 10.29.
CIM_RoleLimitedToTarget	Mandatory	See section 10.30.
CIM_ServiceAccessBySAP	Optional	See section 10.31.
CIM_ServiceAffectsElement	Mandatory	See sections 10.32 and 10.33.
Indications		
None defined in this profile		

945 **10.1 CIM_BindsTo**

946 CIM_BindsTo is used to relate the CIM_CLPProtocolEndpoint instance with the
 947 CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint instance on which it is dependent.

948 **Table 24 – Class: CIM_BindsTo**

Properties	Requirement	Notes
Antecedent	Mandatory	Key shall be a reference to an instance of CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint. Cardinality 0..1
Dependent	Mandatory	Key The value of this property shall be a reference to an instance of CIM_CLPProtocolEndpoint. Cardinality 1

949 **10.2 CIM_CLPCapabilities**

950 CIM_CLPCapabilities represents the capabilities of a CLP service.

951 **Table 25 – Class: CIM_CLPCapabilities**

Properties	Requirement	Notes
InstanceId	Mandatory	Key
ElementName	Mandatory	(pattern ".")
RequestedStatesSupported	Mandatory	See section 7.2.1.
ElementNameEditSupported	Mandatory	See section 7.4.1.
MaxElementNameLen	Conditional	See section 7.1.4.2.2.
MaxConnections	Mandatory	See section 7.1.2.1.
CLPVersions	Mandatory	Shall contain one or more occurrences pattern ([0123456789] "\." [0123456789] "\." [0123456789]?)
SMMEAddressVersions	Mandatory	Shall contain one or more occurrences pattern ([0123456789] "\." [0123456789] "\." [0123456789]?)

952 **10.3 CIM_CLPProtocolEndpoint**

953 CIM_CLPProtocolEndpoint represents a session established with the CLP service. There shall be an
954 instance of CLPProtocolEndpoint for each remote access point provided by the CLP Service to a Client.

955 **Table 26 – Class: CIM_CLPProtocolEndpoint**

Properties	Requirement	Notes
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
Name	Mandatory	Key
NameFormat	Mandatory	pattern ".+"
ProtocolIFTType	Mandatory	Matches 1 (Other)
OtherTypeDescription	Mandatory	Matches "CLP"
ElementName	Mandatory	pattern ".+"
CurrentDefaultTarget	Mandatory	pattern ".+"
KeepTime	Mandatory	None
WaitBehavior	Mandatory	None
OutputFormat	Mandatory	None
OutputVerbosity	Mandatory	None
OutputLanguage	Mandatory	None
OutputPosition	Mandatory	None
OutputOrder	Mandatory	None
OutputCount	Mandatory	None
DisplayOption	Mandatory	pattern ".+"

956 **10.4 CIM_CLPSettingData – Specification Default Configuration**

957 CIM_CLPSettingData represents settings which can be applied to a CLP session.

958 **Table 27 – Class: CIM_CLPSettingData**

Properties	Requirement	Notes
Instanceld	Mandatory	Key
ElementName	Mandatory	Matches "Specification Defaults"
CommandDefaultTarget	Mandatory	Matches "/"
KeepTime	Mandatory	Matches 300
WaitBehavior	Mandatory	Matches FALSE
OutputFormat	Mandatory	Matches 2 (Text)
OutputVerbosity	Mandatory	Matches 3 (Terse)
OutputLanguage	Mandatory	Matches "eng"
OutputPosition	Mandatory	Matches 2 (Begin)
OutputOrder	Mandatory	Matches 2 (Default)
OutputCount	Mandatory	Matches 0xFFFFFFFF
DisplayOption	Mandatory	Matches "all"

959 **10.5 CIM_CLPSettingData**

960 CIM_CLPSettingData represents settings which can be applied to a CLP session. There shall be one
 961 default CLPSettingData instance containing the default settings for the configurable properties of the CLP
 962 Service. There may be other CLPSettingData instances describing optional configurations.

963 **Table 28 – Class: CIM_CLPSettingData**

Properties	Requirement	Notes
Instanceld	Mandatory	Key
ElementName	Mandatory	pattern ".:"
CommandDefaultTarget	Mandatory	pattern ".+"
KeepTime	Mandatory	None
WaitBehavior	Mandatory	None
OutputFormat	Mandatory	None
OutputVerbosity	Mandatory	None
OutputLanguage	Mandatory	None
OutputPosition	Mandatory	None
OutputOrder	Mandatory	None
OutputCount	Mandatory	None
DisplayOption	Mandatory	pattern ".+"

964 **10.6 CIM_ConcreteDependency – (Access Ingress)**

965 Table 29 details the constraints for instances of CIM_ConcreteDependency beyond those specified in the
 966 DSP1034.

967 **Table 29 – Class: CIM_ConcreteDependency – (Access Ingress)**

Elements	Requirement	Notes
Antecedent	Mandatory	See section 7.5.5.2.

968 **10.7 CIM_ConcreteJob**

969 CIM_ConcreteJob represents a single CLP command job.

970 **Table 30 – Class: CIM_ConcreteJob**

Properties	Requirement	Notes
ElementName	Mandatory	See section 7.6.2.2.
Instanceld	Mandatory	Key
JobState	Mandatory	Matches 4 7 8 9 10 (Running Completed Terminated Killed Exception) See section 7.6.2.1.
DeleteOnCompletion	Mandatory	matches TRUE
TimeBeforeRemoval	Mandatory	See section 7.6.2.
RequestStateChange()	Mandatory	See section 8.2.
GetError()	Conditional	See section 8.3.
OperationalStatus	Mandatory	See section 7.6.3.

971 **10.8 CIM_ElementCapabilities**

972 CIM_ElementCapabilities is used to associate an instance of CIM_CLPCapabilities with the
 973 CIM_ProtocolService

974 **Table 31 – Class: CIM_ElementCapabilities**

Properties	Requirement	Notes
ManagedElement	Mandatory	Key This property shall be a reference to the Central Instance. Cardinality 1..*
Capabilities	Mandatory	Key This property shall be a reference to the CIM_CLPCapabilities instance. Cardinality 1

975 **10.9 CIM_ElementSettingData – CLP Service**

976 CIM_ElementSettingData is used to associate instances of CIM_CLPSettingData with instances of
 977 CIM_ProtocolService.

978 **Table 32 – Class: CIM_ElementSettingData (CLP Service)**

Properties	Requirement	Notes
ManagedElement	Mandatory	Key This property shall be a reference to the Central Instance. Cardinality *
Setting	Mandatory	Key This property shall be a reference to an instance of CIM_CLPSettingData. Cardinality *
IsDefault	Mandatory	Matches 1 (Is Default) or 2 (Is Not Default)
IsNext	Mandatory	Matches 1 (Is Next) or 2 (Is Not Next)

979 **10.10 CIM_ElementSettingData – CLP Session**

980 CIM_ElementSettingData is used to associate instances of CIM_CLPSettingData with instances of
 981 CIM_CLPProtocolEndpoint.

982 **Table 33 – Class: CIM_ElementSettingData (CLP Session)**

Properties	Requirement	Notes
ManagedElement	Mandatory	Key This property shall be a reference to an instance of CIM_CLPProtocolEndpoint. Cardinality *
Setting	Mandatory	Key This property shall be a reference to an instance of CIM_CLPSettingData. Cardinality *
IsCurrent	Mandatory	Matches 1 (Is Current) or 2 (Is Not Current)

983 **10.11 CIM_Error**

984 CIM_Error is returned by the CIM_ConcreteJob.GetError() method when a CLP operation fails.

985 **Table 34 – Class: CIM_Error**

Properties	Requirement	Notes
ErrorType	Mandatory	None
OwningEntity	Conditional	See section 7.6.3.1.
MessageID	Conditional	See section 7.6.3.1.
Message	Conditional	See section 7.6.3.1.
MessageArguments	Conditional	See section 7.6.3.1.
PerceivedSeverity	Mandatory	None
ProbableCause	Mandatory	None
RecommendedActions	Optional	None

Properties	Requirement	Notes
ErrorSource	Mandatory	None
ErrorSourceFormat	Mandatory	None
CIMStatusCodes	Mandatory	None

986 10.12 CIM_HostedAccessPoint

987 CIM_HostedAccessPoint is used to relate the CIM_CLPProtocolEndpoint and CIM_TCPIPProtocolEndpoint
 988 instances to their scoping CIM_ComputerSystem instance.

989 **Table 35 – Class: CIM_HostedAccessPoint**

Properties	Requirement	Notes
Antecedent	Mandatory	Key This property shall be a reference to an instance of CIM_ComputerSystem. Cardinality 1
Dependent	Mandatory	Key This property shall be a reference to an instance of CIM_CLPProtocolEndpoint or CIM_TCPIPProtocolEndpoint. Cardinality *

990 10.13 CIM_HostedJobDestination

991 CIM_HostedJobDestination is used to associate the single instance of CIM_JobQueue with hosting
 992 CIM_ComputerSystem instance.

993 **Table 36 – Class: CIM_HostedJobDestination**

Properties	Requirement	Notes
Antecedent	Mandatory	Key This property shall be a reference to the hosting CIM_ComputerSystem. Cardinality 1
Dependent	Mandatory	Key This property shall be a reference to the hosted CIM_JobQueue. Cardinality 1

994 10.14 CIM_HostedService

995 CIM_HostedService is used to relate the CIM_ProtocolService to its scoping CIM_ComputerSystem
 996 instance.

997 **Table 37 – Class: CIM_HostedService**

Properties	Requirement	Notes
Antecedent	Mandatory	Key This property shall be a reference to the Scoping Instance. Cardinality 1
Dependent	Mandatory	Key This property shall be a reference to the Central Instance. Cardinality 1..*

998 **10.15 CIM_IdentityContext**

999 Table 38 details the constraints for instances of CIM_IdentityContext beyond those specified in [DSP1034](#).

1000 **Table 38 – Class: CIM_IdentityContext**

Elements	Requirement	Notes
ElementProvidingContext	Mandatory	shall be a reference to the Central Instance Cardinality 1

1001 **10.16 CIM_JobDestinationJobs**

1002 CIM_JobDestinationJobs is used to associate instances of CIM_JobQueue with instances of
1003 CIM_ConcreteJob.

1004 **Table 39 – Class: CIM_JobDestinationJobs**

Properties	Requirement	Notes
Antecedent	Mandatory	Key This property shall be a reference to the CIM_JobQueue. Cardinality 1
Dependent	Mandatory	Key This property shall be a reference to the CIM_ConcreteJob. Cardinality *

1005 **10.17 CIM_JobQueue**

1006 CIM_JobQueue represents the CLP Service's operations queue.

1007 **Table 40 – Class: CIM_JobQueue**

Properties	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
Name	Mandatory	Key
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
ElementName	Mandatory	(pattern ".")
MaxJobsOnQueue	Mandatory	A value of 0 (zero) shall indicate the maximum number of jobs is unknown or unenforced.
QueueStatus	Mandatory	None
QueueStatusInfo	Mandatory	None

1008 **10.18 CIM_OwningCollectionElement**

1009 Table 41 details the constraints for properties of CIM_OwningCollectionElement beyond those specified
1010 in the [Role Based Authorization Profile](#).

1011

Table 41 – Class: CIM_OwningCollectionElement

Properties and Methods	Requirement	Description
OwningElement	Mandatory	See section 7.5.1.2.

10.19 CIM_OwningJobElement – CLP Service

1013 CIM_OwningJobElement is used to associate instances of CIM_ConcreteJob with the instance of
 1014 CIM_ProtocolService representing the CLP Service.

1015

Table 42 – Class: CIM_OwningJobElement

Properties	Requirement	Notes
Antecedent	Mandatory	Key This property shall be a reference to the hosting CIM_ProtocolService Cardinality 1
Dependent	Mandatory	Key This property shall be a reference to the hosted CIM_ConcreteJob. Cardinality *

10.20 CIM_OwningJobElement – CLP Session

1017 CIM_OwningJobElement is used to associate instances of CIM_ConcreteJob with the instance of
 1018 CIM_CLPProtocolEndpoint representing the CLP session where the CLP operation was initiated.

1019

Table 43 – Class: CIM_OwningJobElement

Properties	Requirement	Notes
Antecedent	Mandatory	Key This property shall be a reference to the hosting CIM_CLPProtocolEndpoint Cardinality 1
Dependent	Mandatory	Key This property shall be a reference to the hosted CIM_ConcreteJob. Cardinality *

10.21 CIM_Privilege (Administrator)

1021 An instance of CIM_Privilege is used to represent the privileges of the SM CLP Administrator Role.
 1022 Table 44 details the constraints for properties of CIM_Privilege beyond those specified in the [Role Based](#)
 1023 [Authorization Profile](#) when it used to model the privileges of the SM CLP Administrator Role.

1024

Table 44 – Class: CIM_Privilege (Administrator)

Properties and Methods	Requirement	Description
PrivilegeGranted	Mandatory	Matches TRUE
Activities	Mandatory	See section 7.5.2.2.1.
ActivityQualifiers	Mandatory	See section 7.5.2.2.2.
ElementName	Mandatory	Matches "SM CLP Administrator Privilege"

1025 **10.22 CIM_Privilege (Operator)**

1026 An instance of CIM_Privilege is used to represent the privileges of the SM CLP Operator Role. Table 45
 1027 details the constraints for properties of CIM_Privilege beyond those specified in the [Role Based](#)
 1028 [Authorization Profile](#) when it used to model the privileges of the SM CLP Operator Role.

1029 **Table 45 – Class: CIM_Privilege (Operator)**

Properties and Methods	Requirement	Description
PrivilegeGranted	Mandatory	Matches TRUE
Activities	Mandatory	See section 7.5.3.2.1.
ActivityQualifiers	Mandatory	See section 7.5.3.2.2.
ElementName	Mandatory	Matches "SM CLP Operator Privilege"

1030 **10.23 CIM_Privilege (Read Only)**

1031 An instance of CIM_Privilege is used to represent the privileges of the SM CLP Read Only Role. Table 46
 1032 details the constraints for properties of CIM_Privilege beyond those specified in the [Role Based](#)
 1033 [Authorization Profile](#) when it used to model the privileges of the SM CLP Read Only Role.

1034 **Table 46 – Class: CIM_Privilege (Read Only)**

Properties and Methods	Requirement	Description
PrivilegeGranted	Mandatory	Matches TRUE
Activities	Mandatory	See section 7.5.4.2.1.
ActivityQualifiers	Mandatory	See section 7.5.4.2.2.
ElementName	Mandatory	Matches "SM CLP Read Only Privilege"

1035 **10.24 CIM_ProtocolService**

1036 CIM_ProtocolService represents the CLP service.

1037 **Table 47 – Class: CIM_ProtocolService**

Properties	Requirement	Notes
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
Name	Mandatory	Key
Protocol	Mandatory	See section 7.1.1.
MaxConnections	Mandatory	A value of 0 (zero) shall indicate unknown.
RequestedState	Mandatory	See section 7.1.3.
EnabledState	Mandatory	See section 7.1.3.
ElementName	Mandatory	See section 7.1.4.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
RequestStateChange()	Conditional	See section 8.1.

1038 **10.25 CIM_ProvidesEndpoint**

1039 CIM_ProvidesEndpoint is used to associate the instance of CIM_ProtocolService with an instance of
 1040 CIM_CLPProtocolEndpoint representing a session with the service.

1041 **Table 48 – Class: CIM_ProvidesEndpoint**

Properties	Requirement	Notes
Antecedent	Mandatory	Key This property shall be a reference to the instance of CIM_ProtocolService. Cardinality 1
Dependent	Mandatory	Key This property shall be a reference to an instance of CIM_CLPProtocolEndpoint. Cardinality *

1042 **10.26 CIM_RegisteredProfile**

1043 CIM_RegisteredProfile identifies the *Command Line Protocol Service Profile*. The CIM_RegisteredProfile
 1044 class is defined by the [Profile Registration Profile](#). With the exception of the mandatory values specified
 1045 for the properties in Table 49, the behavior of the CIM_RegisteredProfile instance is in accordance with
 1046 the constraints specified in the [Profile Registration Profile](#).

1047 **Table 49 – Class: CIM_RegisteredProfile**

Properties	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Command Line Protocol Service".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of 2 ("DMTF").

1048 NOTE: Previous versions of this document included the suffix "Profile" for the RegisteredName value. If
 1049 implementations querying for the RegisteredName value find the suffix "Profile", they should ignore the suffix, with
 1050 any surrounding white spaces, before any comparison is done with the value as specified in this document.

1051 **10.27 CIM_Role (Administrator)**

1052 An instance of CIM_Role is used to model the CLP Administrator Role. Table 50 details the constraints
 1053 for properties of CIM_Role beyond those specified in the [Role Based Authorization Profile](#) when it is used
 1054 to model the CLP Administrator Role.

1055 **Table 50 – Class: CIM_Role (Administrator)**

Properties and Methods	Requirement	Notes
RoleCharacteristics	Mandatory	Shall contain 2 (Static)
CommonName	Mandatory	Matches "CIM:CLP:Administrator"
ElementName	Mandatory	Matches "SM CLP Administrator"

1056 **10.28 CIM_Role (Operator)**

1057 An instance of CIM_Role is used to model the CLP Operator Role. Table 51 details the constraints for
 1058 properties of CIM_Role beyond those specified in the [Role Based Authorization Profile](#) when it is used to
 1059 model the CLP Operator Role.

1060

Table 51 – Class: CIM_Role (Operator)

Properties and Methods	Requirement	Notes
RoleCharacteristics	Mandatory	Shall contain 2 (Static)
CommonName	Mandatory	Matches "CIM:CLP:Operator"
ElementName	Mandatory	Matches "SM CLP Operator"

10.29 CIM_Role (Read Only)

1062 An instance of CIM_Role is used to model the CLP Read Only Role. Table 52 details the constraints for
 1063 properties of CIM_Role beyond those specified in the [Role Based Authorization Profile](#) when it is used to
 1064 model the CLP Read Only Role.

1065

Table 52 – Class: CIM_Role (Read Only)

Properties and Methods	Requirement	Notes
RoleCharacteristics	Mandatory	Shall contain 2 (Static)
CommonName	Mandatory	Matches "CIM:CLP:Read Only"
ElementName	Mandatory	Matches "SM CLP Read Only"

10.30 CIM_RoleLimitedToTarget

1066 Table 53 details the constraints for properties of CIM_RoleLimitedToTarget beyond those specified in the
 1067 [Role Based Authorization Profile](#).

1069

Table 53 – Class: CIM_RoleLimitedToTarget

Properties and Methods	Requirement	Description
TargetElement	Mandatory	See section 7.5.1.1. Cardinality 1

10.31 CIM_ServiceAccessBySAP

1070 CIM_ServiceAccessBySAP is used to associate the instance of CIM_ProtocolService with an instance of
 1071 CIM_TCPIPProtocolEndpoint over which a session with the service can be established.

1073

Table 54 – Class: CIM_ServiceAccessBySAP

Properties	Requirement	Notes
Antecedent	Mandatory	Key This property shall be a reference to the instance of CIM_ProtocolService. Cardinality 1..*
Dependent	Mandatory	Key This property shall be a reference to an instance of CIM_TCPIPProtocolEndpoint. Cardinality *

1074 **10.32 CIM_ServiceAffectsElement – AdminDomain**

1075 CIM_ServiceAffectsElement associates an instance of CIM_ProtocolService with an instance of
 1076 CIM_AdminDomain representing the management domain of the CLP service.

1077 **Table 55 – Class: CIM_ServiceAffectsElement**

Properties	Requirement	Notes
ServiceProvided	Mandatory	Key This property shall be a reference to the Central Instance of the profile. Cardinality 1
UserOfService	Mandatory	Key shall be a reference to the Central Instance defined in <i>DSP1007</i> . Cardinality 1
ElementAffects	Mandatory	Matches 5 (Manages)

1078 **10.33 CIM_ServiceAffectsElement – Job Queue**

1079 CIM_ServiceAffectsElement associates an instance of CIM_ProtocolService with the instance of
 1080 CIM_JobQueue where CLP operations are executed.

1081 **Table 56 – Class: CIM_ServiceAffectsElement**

Properties	Requirement	Notes
ServiceProvided	Mandatory	Key This property shall be a reference to the Central Instance of the profile. Cardinality 1
UserOfService	Mandatory	Key shall be a reference to CIM_JobQueue. Cardinality 1
ElementAffects	Mandatory	Shall contain the values 5 (Manages) and 6 (Consumes)

1082
1083
1084
1085

ANNEX A (informative)

Change Log

Version	Date	Description
1.0.0a	10/10/2006	Preliminary Standard
1.0.0	06/17/2009	DMTF Standard Release

1086
1087