



1

2

3

4

Document Identifier: DSP1016

Date: 2019-03-15

Version: 1.0.1

5

Telnet Service Profile

6

Supersedes: 1.0.0

7

Document Class: Normative

8

Document Status: Published

9

Document Language: en-US

10

11 Copyright Notice

12 Copyright © 2009, 2019 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 for uses consistent with this purpose, provided that correct attribution is given. As DMTF
16 specifications may be revised from time to time, the particular version and release date should always be
17 noted.

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28 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

34

35

CONTENTS

37	Foreword	6
38	Introduction.....	7
39	1 Scope	9
40	2 Normative references	9
41	3 Terms and definitions	10
42	4 Symbols and abbreviated terms.....	11
43	5 Synopsis	11
44	6 Description	12
45	6.1 Telnet session life cycle	13
46	7 Implementation requirements.....	13
47	7.1 Representing a telnet service	13
48	7.2 Representing the server's view of a telnet session.....	16
49	7.3 Relationship with IP interfaces (optional).....	17
50	8 Methods.....	17
51	8.1 CIM_ProtocolService.ListenOnPort() (optional)	17
52	8.2 CIM_ProtocolService.RequestStateChange()	18
53	8.3 Profile conventions for operations	19
54	8.4 CIM_BindsTo	20
55	8.5 CIM_ElementCapabilities	20
56	8.6 CIM_ElementSettingData	21
57	8.7 CIM_HostedAccessPoint	21
58	8.8 CIM_HostedService	22
59	8.9 CIM_ProtocolService	22
60	8.10 CIM_ProvidesEndpoint	22
61	8.11 CIM_ServiceAccessBySAP	23
62	8.12 CIM_TelnetCapabilities.....	23
63	8.13 CIM_TelnetSettingData.....	23
64	8.14 CIM_TelnetProtocolEndpoint.....	23
65	8.15 CIM_TCIPProtocolEndpoint.....	24
66	9 Use cases.....	25
67	9.1 Object diagrams.....	25
68	9.2 Configuring session default settings	29
69	9.3 Modifying active session settings	29
70	9.4 Disabling the telnet service	29
71	9.5 Determining the telnet service capabilities	29
72	9.6 Determining the listening ports of the telnet service.....	29
73	9.7 Adding a listening port for the telnet service.....	30
74	9.8 Stopping the telnet service from listening on a specific port.....	31
75	9.9 Determining whether ElementName can be modified	31
76	9.10 Determining whether state management is supported	32
77	10 CIM Elements	32
78	10.1 CIM_BindsTo TCPProtocolEndpoint.....	33
79	10.2 CIM_BindsTo IPProtocolEndpoint.....	33
80	10.3 CIM_ElementCapabilities	33
81	10.4 CIM_ElementSettingData telnet service	34
82	10.5 CIM_ElementSettingData telnet session.....	34
83	10.6 CIM_HostedAccessPoint	34
84	10.7 CIM_HostedService	35
85	10.8 CIM_ProtocolService	36
86	10.9 CIM_ProvidesEndpoint	36

87	10.10 CIM_RegisteredProfile.....	37
88	10.11 CIM_ServiceAccessBySAP	37
89	10.12 CIM_TelnetCapabilities.....	38
90	10.13 CIM_TelnetProtocolEndpoint.....	38
91	10.14 CIM_TelnetSettingData.....	39
92	10.15 CIM_TCPProtocolEndpoint.....	40
93	ANNEX A (Informative) Change log.....	41

94

95 Figures

96	Figure 1 Telnet Service Profile: Class diagram	12
97	Figure 2 Registered Profile	25
98	Figure 3 Telnet service listening for connections.....	26
99	Figure 4 One active session	27
100	Figure 5 Session changed	28
101	Figure 6 Listening on a single port on all interfaces	30
102	Figure 7 Port added bound to specific interface	31

103

104 Tables

105	Table 1 Referenced profiles.....	11
106	Table 2 CIM_ProtocolService.ListenOnPort() method: Return code values	17
107	Table 3 CIM_ProtocolService.ListenOnPort() method: Parameters	18
108	Table 4 CIM_ProtocolService.RequestStateChange() method: Return code values	19
109	Table 5 CIM_ProtocolService.RequestStateChange() method: Parameters	19
110	Table 6 Operations: CIM_BindsTo.....	20
111	Table 7 Operations: CIM_ElementCapabilities	20
112	Table 8 Operations: CIM_ElementSettingData.....	21
113	Table 9 Operations: CIM_HostedAccessPoint.....	21
114	Table 10 Operations: CIM_HostedService	22
115	Table 11 Operations: CIM_ProtocolService.....	22
116	Table 12 Operations: CIM_ProvidesEndpoint.....	22
117	Table 13 Operations: CIM_ServiceAccessBySAP.....	23
118	Table 14 Operations: CIM_TelnetSettingData	23
119	Table 15 Operations: CIM_TelnetProtocolEndpoint	23
120	Table 16 Operations: CIM_TCPProtocolEndpoint	24
121	Table 17 CIM Elements: Telnet Service Profile	32
122	Table 18 Class: CIM_BindsTo (TCPProtocolEndpoint)	33
123	Table 19 Class: CIM_BindsTo (IPProtocolEndpoint)	33
124	Table 20 Class: CIM_ElementCapabilities.....	33
125	Table 21 Class: CIM_ElementSettingData (telnet service).....	34
126	Table 22 Class: CIM_ElementSettingData (telnet session).....	34
127	Table 23 Class: CIM_HostedAccessPoint	34
128	Table 24 Class: CIM_HostedService	35
129	Table 25 Class: CIM_ProtocolService	36
130	Table 26 Class: CIM_ProvidesEndpoint	36

131 Table 27 Class: CIM_RegisteredProfile 37

132 Table 28 Class: CIM_ServiceAccessBySAP 37

133 Table 29 Class: CIM_TelnetCapabilities 38

134 Table 30 Class: CIM_TelnetProtocolEndpoint 38

135 Table 31 Class: CIM_TelnetSettingData 39

136 Table 32 Class: CIM_TCPProtocolEndpoint 40

137

138

Foreword

139 The *Telnet Service Profile* (DSP1016) was prepared by the Server Management Working Group.

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

142 Acknowledgments

143 The DMTF acknowledges the following individuals for their contributions to this document:

144 Editor:

- 145 • Aaron Merkin IBM

146 Contributors:

- 147 • Jon Hass Dell
- 148 • Jeff Hilland Hewlett Packard Enterprise
- 149 • John Leung Intel
- 150 • Khachatur Papanyan Dell
- 151 • Sivakumar Sathappan - AMD
- 152 • Christina Shaw Hewlett Packard Enterprise
- 153 • Enoch Suen Dell
- 154 • Perry Vincent Intel

155

Introduction

156 The information in this specification should be sufficient for a provider or consumer of this data to identify
157 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to
158 represent and manage a telnet service, its associated configuration information, and any active
159 connections.

160 The target audience for this specification is implementers who are writing CIM-based providers or
161 consumers of management interfaces that represent the component described in this document.
162

163

164

Telnet Service Profile

1 Scope

166 The *Telnet Service Profile* extends the management capability of referencing profiles by adding the
167 capability to represent a telnet service and its associated sessions.

2 Normative references

169 The following referenced documents are indispensable for the application of this document. For dated or
170 versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies.
171 For references without a date or version, the latest published edition of the referenced document
172 (including any corrigenda or DMTF update versions) applies.

173 DMTF DSP0004, *CIM Infrastructure Specification 3.0*,
174 http://www.dmtf.org/standards/published_documents/DSP0004_3.0.pdf

175 DMTF DSP0200, *CIM Operations over HTTP 1.2*,
176 <https://www.dmtf.org/sites/default/files/standards/documents/DSP200.html>

177 DMTF DSP0215, *Server Management Managed Element Addressing Specification 1.0*,
178 http://www.dmtf.org/standards/published_documents/DSP0215_1.0.pdf

179 DMTF DSP0223, *Generic Operations 1.0*,
180 http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf

181 DMTF DSP0228, *Message Registry XML Schema 1.0*,
182 http://schemas.dmtf.org/wbem/messageregistry/1/dsp0228_1.0.1.xsd

183 DMTF DSP1001, *Management Profile Specification Usage Guide 1.1*,
184 http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf

185 DMTF DSP1033, *Profile Registration Profile 1.0*,
186 https://www.dmtf.org/sites/default/files/standards/documents/DSP1033_1.0.pdf

187 DMTF DSP1053, *Base Metrics Profile 1.0*,
188 https://www.dmtf.org/sites/default/files/standards/documents/DSP1053_1.0.pdf

189 DMTF DSP1054, *Indications Profile 1.2*,
190 http://www.dmtf.org/standards/published_documents/DSP1054_1.2.pdf

191 DMTF DSP8016, *WBEM Operations Message Registry 1.0*,
192 http://schemas.dmtf.org/wbem/messageregistry/1/dsp8016_1.0.xml

193 DMTF DSP8020, *Message Registry XML Schema Specification 1.0*,
194 http://schemas.dmtf.org/wbem/metricregistry/1/dsp8020_1.0.xsd

195 IETF RFC1208, *A Glossary of Networking Terms, March 1991*,
196 <https://tools.ietf.org/html/rfc1208>

197 IETF RFC5234, *ABNF: Augmented BNF for Syntax Specifications, January 2008*,
198 <http://tools.ietf.org/html/rfc5234>

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

201 The Open Group, "Regular Expressions" in *The Single UNIX® Specification, Version 2*,
202 <http://www.opengroup.org/onlinepubs/7908799/xbd/re.html> Terms and Definitions

203 3 Terms and definitions

204 In this document, some terms have a specific meaning beyond the normal English meaning. Those terms
205 are defined in this clause.

206 The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"),
207 "may", "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
208 in [ISO/IEC Directives, Part 2](#), Clause 7. The terms in parentheses are alternatives for the preceding term,
209 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
210 [ISO/IEC Directives, Part 2](#), Clause 7 specifies additional alternatives. Occurrences of such additional
211 alternatives shall be interpreted in their normal English meaning.

212 The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as
213 described in [ISO/IEC Directives, Part 2](#), Clause 6.

214 The terms "normative" and "informative" in this document are to be interpreted as described in [ISO/IEC](#)
215 [Directives, Part 2](#), Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do
216 not contain normative content. Notes and examples are always informative elements.

217 The terms defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document. The following additional
218 terms are used in this document.

219 3.1

220 **conditional**

221 indicates requirements to be followed strictly to conform to the document when the specified conditions
222 are met

223 3.2

224 **mandatory**

225 indicates requirements to be followed strictly to conform to the document and from which no deviation is
226 permitted

227 3.3

228 **optional**

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

230 3.4

231 **referencing profile**

232 indicates a profile that owns the definition of this class and can include a reference to this profile in its
233

234 3.5

235 **unspecified**

236 indicates that this profile does not define any constraints for the referenced CIM element or operation

237 3.6

238 **Listening Port**

239 a TCP/IP port that the telnet service is bound to and listening for incoming connection requests

240 **4 Symbols and abbreviated terms**

241 The abbreviations defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document. The following
 242 additional abbreviations are used in this document.

243 **4.1**

244 **CIM**

245 Common Information Model

246 **4.2**

247 **IP**

248 Internet Protocol

249 **4.3**

250 **TCP**

251 Transmission Control Protocol

252 **5 Synopsis**

253 **Profile Name:** *Telnet Service Profile*

254 **Version:** 1.0.0a

255 **Organization:** DMTF

256 **CIM Schema Version:** [2.12](#)

257 **Central Class:** CIM_ProtocolService

258 **Scoping Class:** CIM_ComputerSystem

259 The *Telnet Service Profile* extends the management capability of referencing profiles by adding the
 260 capability to represent a telnet service in a managed system. This profile includes a specification of the
 261 telnet service, its associated configuration, and any active sessions.

262 The Central Class for the *Telnet Service Profile* shall be the CIM_ProtocolService class. The Central
 263 Instance of the *Telnet Service Profile* shall be an instance of CIM_ProtocolService. The Scoping Class for
 264 the *Telnet Service Profile* shall be CIM_ComputerSystem. The Scoping Instance of the *Telnet Service*
 265 *Profile* shall be the instance of CIM_ComputerSystem to which the Central Instance is associated through
 266 an instance of CIM_HostedService.

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

268 **Table 1 – Referenced profiles**

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration Profile	DMTF	1.0	Mandatory	
IP Interface Profile	DMTF	1.0	Optional	See clause 7.3.

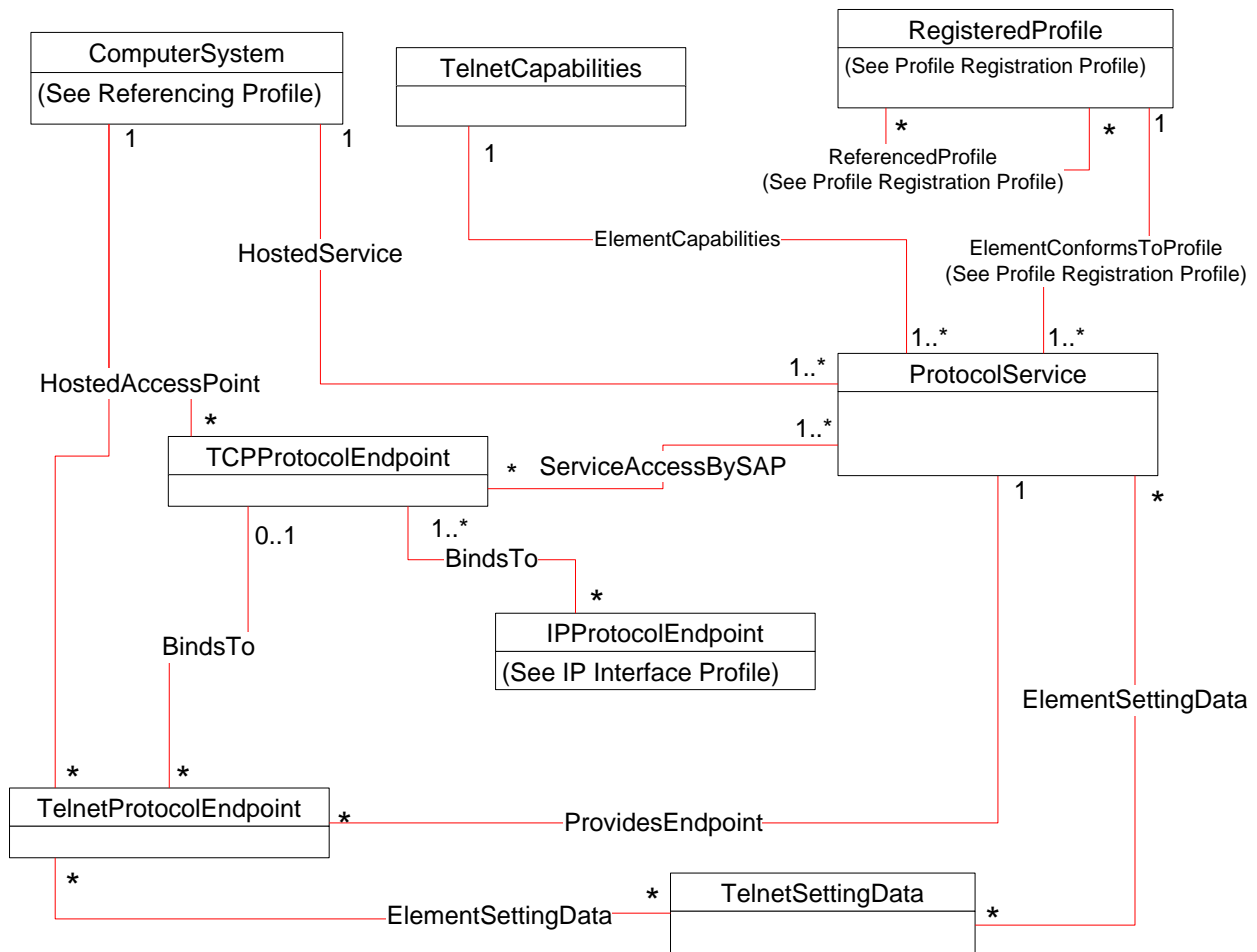
269 **6 Description**

270 The *Telnet Service Profile* extends the management capability of referencing profiles by adding the
 271 capability to represent a telnet service hosted on a managed system. Functionality within the scope of this
 272 profile includes:

- 273 • representation of the telnet service
- 274 • representation of the Telnet server's view of active sessions
- 275 • configuration of the telnet service
- 276 • configuration of the telnet sessions from the Telnet server

277 Functionality that is explicitly excluded from the scope of this profile is modeling of the telnet session at
 278 the Telnet client.

279 Figure 1 represents the class schema for the *Telnet Service Profile*. For simplicity, the prefix CIM_ has
 280 been removed from the names of the classes.



281

282

Figure 1 – Telnet Service Profile: Class diagram

283 This profile represents the capabilities of the telnet service, the current configuration of the telnet service,
284 the active sessions, and the default settings when new sessions are activated. The telnet service is
285 represented by an instance of CIM_ProtocolService, where the Protocol property has the value 3 (Telnet).
286 The capabilities of the telnet service are represented by an instance of CIM_TelnetCapabilities. The
287 current configuration of the telnet service is modeled with the properties from the instance of
288 CIM_ProtocolService. Each active session with the telnet service is represented by an instance of
289 CIM_TelnetProtocolEndpoint. The current configuration of an active session is reflected in the values of
290 the properties from the instance of CIM_TelnetProtocolEndpoint. CIM_TelnetSettingData represents a
291 complete configuration that a telnet session could have. For example, an instance of
292 CIM_TelnetSettingData contains the configuration that will be in effect for a telnet session when it is first
293 established. CIM_TCPProtocolEndpoint is an optional endpoint used to model the TCP/IP ports over
294 which a telnet service listens or a telnet session is active.

295 **6.1 Telnet session life cycle**

296 When a telnet session is established with the telnet service, an instance of CIM_TelnetProtocolEndpoint
297 is created. The CIM_TelnetProtocolEndpoint instance exists for the duration of the telnet session that it
298 represents. When the telnet session ends, the CIM_TelnetProtocolEndpoint is removed. When the
299 CIM_TelnetProtocolEndpoint is explicitly deleted through an intrinsic DeleteInstance operation, the telnet
300 session ends.

301 **7 Implementation requirements**

302 This clause details the requirements related to the arrangement of instances and their properties for
303 implementations of this profile.

304 **7.1 Representing a telnet service**

305 An instance of CIM_ProtocolService shall represent the telnet service being modeled.

306 **7.1.1 Telnet service state management is supported—conditional**

307 When management of the state of a telnet service is supported, exactly one instance of
308 CIM_TelnetCapabilities shall be associated with the CIM_ProtocolService instance through an instance of
309 CIM_ElementCapabilities.

310 Support for managing the state of the telnet service is conditional behavior. For information about how a
311 client can determine whether this behavior is supported, see clause 9.10. This behavior should be
312 implemented when the telnet service can be enabled or disabled in its entirety.

313 This clause describes the CIM elements and behaviors that shall be implemented when this behavior is
314 supported.

315 **7.1.1.1 CIM_TelnetCapabilities**

316 When state management is supported, exactly one instance of CIM_TelnetCapabilities shall be
317 associated with the CIM_ProtocolService instance through an instance of CIM_ElementCapabilities.

318 **7.1.1.1.1 CIM_TelnetCapabilities.RequestedStatesSupported**

319 The RequestedStatesSupported property may contain zero or more of the following values: 2 (Enabled),
320 3 (Disabled), or 11 (Reset).

321 7.1.1.2 CIM_ProtocolService.RequestedState

322 When the CIM_ProtocolService.RequestStateChange() method is successfully invoked, the value of the
323 RequestedState property shall be the value of the RequestedState parameter. If the method is not
324 successfully invoked, the value of the RequestedState property is indeterminate.

325 The CIM_ProtocolService.RequestedState property shall have one of the values specified in the
326 CIM_TelnetCapabilities.RequestedStatesSupported property or a value of 5 (No Change).

327 7.1.1.3 CIM_ProtocolService.EnabledState

328 When the RequestedState parameter has a value of 2 (Enabled) or 3 (Disabled) and the
329 CIM_ProtocolService.RequestStateChange() method completes successfully, the value of the
330 EnabledState property shall equal the value of the CIM_ProtocolService.RequestedState property.

331 If the method does not complete successfully, the value of the EnabledState property is indeterminate.

332 The EnabledState property shall have the value 2 (Enabled), 3 (Disabled), or 6 (Enabled but Offline).

333 7.1.2 Telnet service state management is not supported

334 This clause describes the CIM elements and behaviors that shall be implemented when management of
335 the telnet service state is not supported.

336 7.1.2.1 CIM_TelnetCapabilities

337 When state management is not supported, exactly one instance of CIM_TelnetCapabilities may be
338 associated with the CIM_ProtocolService instance through an instance of CIM_ElementCapabilities.

339 7.1.2.1.1 CIM_TelnetCapabilities.RequestedStatesSupported

340 The CIM_TelnetCapabilities.RequestedStatesSupported property shall not contain any values.

341 7.1.2.2 CIM_ProtocolService.RequestedState

342 The RequestedState property shall have the value 12 (Not Applicable).

343 7.1.2.3 CIM_ProtocolService.EnabledState

344 The EnabledState property shall have one of the following values: 2 (Enabled), 3 (Disabled), or 5 (Not
345 Applicable).

346 7.1.3 Modifying ElementName is supported—conditional

347 The CIM_ProtocolService.ElementName property may support being modified by the ModifyInstance
348 operation (see clause 8.9.1). This behavior is conditional. For information about how a client can
349 determine whether it is supported, see clause 9.8.

350 This clause describes the CIM elements and behavior requirements when an implementation supports
351 client modification of the CIM_ProtocolService.ElementName property.

352 7.1.3.1 CIM_TelnetCapabilities

353 An instance of CIM_TelnetCapabilities shall be associated with the CIM_ProtocolService instance
354 through an instance of CIM_ElementCapabilities.

355 7.1.3.1.1 CIM_TelnetCapabilities.ElementNameEditSupported

356 The ElementNameEditSupported property shall have a value of TRUE when the implementation supports
357 client modification of the CIM_ProtocolService.ElementName property.

358 7.1.3.1.2 CIM_EnabledLogicalElement.MaxElementNameLen

359 The MaxElementNameLen property shall be implemented.

360 7.1.4 Modifying ElementName is not supported

361 This clause describes the CIM elements and behaviors that shall be implemented when the
362 CIM_ProtocolService.ElementName does not support being modified by the ModifyInstance operation.

363 7.1.4.1 CIM_TelnetCapabilities

364 An instance of CIM_TelnetCapabilities may be associated with the CIM_ProtocolService instance through
365 an instance of CIM_ElementCapabilities.

366 7.1.4.1.1 CIM_TelnetCapabilities.ElementNameEditSupported

367 The ElementNameEditSupported property shall have a value of FALSE when the implementation does
368 not support client modification of the CIM_ProtocolService.ElementName property.

369 7.1.4.1.2 CIM_EnabledLogicalElement.MaxElementNameLen

370 The MaxElementNameLen property may be implemented. The MaxElementNameLen property is
371 irrelevant in this context.

372 7.1.5 Default configuration of the service

373 The default configuration is the configuration of the service when it was first installed on the managed
374 system. When an implementation exposes the default configuration, the default configuration shall be
375 represented by an instance of CIM_TelnetSettingData associated with the CIM_ProtocolService through
376 an instance of CIM_ElementSettingData, where the IsDefault property of the CIM_ElementSettingData
377 instance has a value of 1 (Is Default).

378 7.1.5.1 Listening port

379 An implementation may model one or more listening ports of the telnet service. When the implementation
380 models the listening ports, the following requirements apply.

381 7.1.5.1.1 CIM_TCPProtocolEndpoint

382 For each TCP/IP port to which the telnet service is bound, there shall be an instance of
383 CIM_TCPProtocolEndpoint in which the PortNumber property of the instance indicates the TCP/IP port
384 number to which the telnet service is listening.

385 7.1.5.1.2 Relationship to the telnet service

386 For each CIM_TCPProtocolEndpoint instance, an instance of CIM_ServiceAccessBySAP shall associate
387 the CIM_ProtocolService instance with the CIM_TCPProtocolEndpoint instance.

388 7.1.5.2 Managing listening ports

389 The implementation may support managing the ports on which the telnet service listens. This behavior is
390 optional. The ListenOnPort() method (see clause 8.1) of the CIM_ProtocolService class can be used to
391 add ports on which the telnet service will listen. Using the DeleteInstance intrinsic operation to delete an

392 instance of CIM_TCPProtocolEndpoint will stop the telnet service from listening on the represented port
393 (see clause 8.15.2).

394 **7.2 Representing the server's view of a telnet session**

395 Each active session with the telnet service shall be represented by an instance of
396 CIM_TelnetProtocolEndpoint.

397 **7.2.1 Relationship with service**

398 An instance of CIM_ProvidesEndpoint shall associate the CIM_ProtocolService with the
399 CIM_TelnetProtocolEndpoint.

400 **7.2.2 Port for session**

401 An implementation may model the TCP/IP port to which the telnet session is bound. This behavior is
402 optional. When the implementation models the TCP/IP port, the following requirements apply.

403 **7.2.2.1 CIM_TCPProtocolEndpoint**

404 When the TCP/IP port to which the telnet session is bound is modeled, the TCP/IP port shall be modeled
405 using an instance of CIM_TCPProtocolEndpoint.

406 **7.2.2.2 Relationship to session**

407 An instance of CIM_BindsTo shall associate the CIM_TelnetProtocolEndpoint instance with the
408 CIM_TCPProtocolEndpoint.

409 **7.2.3 Session default configuration**

410 When a telnet session is created, it will have an initial configuration. Implementations can indicate to
411 clients the configuration that will be assigned to a session. Implementations can also indicate to clients
412 the configuration that an active session had when the session was first established.

413 **7.2.3.1 Configuration that will be assigned**

414 An implementation may assign the same initial configuration for all telnet sessions that are spawned.
415 When the implementation assigns the same initial configuration for all telnet sessions, the configuration
416 that a session will have when it is established shall be represented by an instance of
417 CIM_TelnetSettingData that is associated with the instance of CIM_ProtocolService through an instance
418 of CIM_ElementSettingData, where the IsNext property of the CIM_ElementSettingData instance has a
419 value of 1 (Is Next).

420 **7.2.3.2 Initial configuration of a session**

421 The initial configuration of a session may be modeled. This behavior is optional. When the configuration
422 that a session had when it was established is modeled, it shall be represented by an instance of
423 CIM_TelnetSettingData that is associated with the instance of CIM_TelnetProtocolEndpoint through an
424 instance of CIM_ElementSettingData, where the IsCurrent property of the CIM_ElementSettingData
425 instance has a value of 1 (Is Current).

426 A discrete instance of CIM_TelnetSettingData is not required for each active session. The instance of
427 CIM_TelnetSettingData that is associated with the instance of CIM_TelnetProtocolEndpoint needs only to
428 accurately reflect the initial configuration of the session.

429 **7.3 Relationship with IP interfaces (optional)**

430 This clause details requirements for specifying the relationship between the telnet session or service and
 431 one or more IP interfaces of the system.

432 **7.3.1 Modeling the IP interface over which a session was established**

433 When the specific port for a telnet session is modeled, the specific IP interface over which the session is
 434 active may be modeled. This behavior is optional. When the implementation models the specific interface
 435 over which a telnet session is active, there shall be an instance of the CIM_BindsTo association where
 436 the value of the Antecedent property shall be a reference to the CIM_IPProtocolEndpoint instance and
 437 the value of the Dependent property shall be a reference to the CIM_TCPProtocolEndpoint instance.

438 **7.3.2 Modeling the IP interfaces for the service**

439 When the specific port for a telnet session or service is modeled, the specific IP interface over which the
 440 session is active may be modeled. This behavior is optional. When the implementation models the
 441 specific interface over which a telnet session is active, there shall be an instance of the CIM_BindsTo
 442 association where the value of the Antecedent property shall be a reference to the
 443 CIM_IPProtocolEndpoint instance and the value of the Dependent property shall be a reference to the
 444 CIM_TCPProtocolEndpoint instance. When the CIM_TCPProtocolEndpoint is not associated with one or
 445 more instances of CIM_IPProtocolEndpoint through an instance of CIM_BindsTo, the telnet service
 446 accepts connections over all the IP interfaces of the system.

447 **8 Methods**

448 This clause details the requirements for supporting intrinsic operations and extrinsic methods for the CIM
 449 elements defined by this profile.

450 **8.1 CIM_ProtocolService.ListenOnPort() (optional)**

451 The CIM_ProtocolService.ListenOnPort() method shall be supported when the ListeningPortManagement
 452 property of the associated instance of CIM_TelnetCapabilities has a value of TRUE. When the value of
 453 the ListeningPortManagement property is FALSE, the CIM_ProtocolService.ListenOnPort() method shall
 454 not be supported.

455 The CIM_ProtocolService.ListenOnPort() method is used to configure additional ports on which the
 456 CIM_ProtocolService instance will listen. Detailed requirements of the ListenOnPort() method are
 457 specified in Table 2 and Table 3.

458 No standard messages are defined.

459 **Table 2 – CIM_ProtocolService.ListenOnPort() 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

460

Table 3 – CIM_ProtocolService.ListenOnPort() method: Parameters

Qualifiers	Name	Type	Description/Values
IN	IPEndpoint	CIM_IPProtocolEndpoint REF	Optional reference to the specific CIM_IPProtocolEndpoint instance to which the created CIM_TCPProtocolEndpoint instance will be bound
OUT	Job	CIM_TCPProtocolEndpoint REF	CIM_TCPProtocolEndpoint instance that is created if the method is successful
IN, REQ	PortNumber	uint16	Desired port number for the service to listen on

461 When the method completes successfully, the implementation shall create an instance of
 462 CIM_TCPProtocolEndpoint. The value of the PortNumber property of the instance of
 463 CIM_TCPProtocolEndpoint shall be the value of the PortNumber parameter of the method invocation.
 464 The implementation shall create an instance of CIM_ServiceAccessBySAP that references the instance
 465 of CIM_TCPProtocolEndpoint and references the instance of CIM_ProtocolService on which the method
 466 was invoked.

467 The IPEndpoint parameter for the method is optional. The implementation shall perform the following
 468 actions when the IPEndpoint parameter is not specified:

- 469 • The implementation shall create an instance of CIM_HostedAccessPoint that references the
 470 newly created CIM_TCPProtocolEndpoint instance and the instance of CIM_ComputerSystem
 471 with which the CIM_ProtocolService instance is associated through an instance of
 472 CIM_HostedService (the scoping system).
- 473 • For each instance of CIM_IPProtocolEndpoint that is associated through an instance of
 474 CIM_HostedAccessPoint with the CIM_ComputerSystem instance with which the instance of
 475 CIM_ProtocolService on which this method was invoked is associated through an instance of
 476 CIM_HostedService, the implementation shall create an instance of the CIM_BindsTo
 477 association where the value of the Antecedent property shall be a reference to the
 478 CIM_IPProtocolEndpoint instance and the value of the Dependent property shall be a reference
 479 to the CIM_TCPProtocolEndpoint instance.

480 The implementation shall perform the following actions when the IPEndpoint parameter is specified:

- 481 • The implementation shall create an instance of CIM_HostedAccessPoint that references the
 482 newly created CIM_TCPProtocolEndpoint instance and the instance of CIM_ComputerSystem
 483 with which the CIM_IPProtocolEndpoint instance is associated through an instance of
 484 CIM_HostedAccessPoint.
- 485 • The implementation shall create an instance of the CIM_BindsTo association where the value of
 486 the Antecedent property shall be a reference to the CIM_IPProtocolEndpoint instance and the
 487 value of the Dependent property shall be a reference to the CIM_TCPProtocolEndpoint
 488 instance.

489 **8.2 CIM_ProtocolService.RequestStateChange()**

490 Invocation of the CIM_ProtocolService.RequestStateChange(
 491 the value specified in the RequestedState parameter. The Enabled and Disabled values of the
 492 RequestedState parameter correspond to enabling or disabling the functionality represented by the
 493 instance of CIM_ProtocolService. A value of 2 (Enabled) shall correspond to a request to enable the
 494 functionality. A value of 3 (Disabled) shall correspond to a request to disable the functionality. A value of
 495 11 (Reset) shall initiate a reset of the telnet service.

496 See clause 7.1.1.2 for information about the effect of this method on the RequestedState property.

497 The method shall be considered successful if the availability of the functionality upon completion of the
 498 method corresponds to the desired availability indicated by the RequestedState parameter. An actual
 499 change in state does not need to occur for the method to be considered successful. It is sufficient that the
 500 resultant state be equal to the requested state. Upon successful completion of the method, the Return
 501 Value shall be zero.

502 See clause 7.1.1.3 for information about the effect of this method on the EnabledState property.

503 Detailed requirements of the RequestStateChange() method are specified in Table 4 and Table 5.

504 No standard messages are defined.

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

507 **Table 4 – 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

508 **Table 5 – CIM_ProtocolService.RequestStateChange() method: Parameters**

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

509 **8.2.1 CIM_ProtocolService.RequestStateChange() ConditionalSupport**

510 Support for the RequestStateChange() method is conditional on the indication of support for at least one
 511 value for the RequestedState parameter as advertised through the RequestedStatesSupported property
 512 of an associated instance of CIM_EnabledLogicalElementCapabilities. When the
 513 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least one
 514 value, the CIM_ProtocolService.RequestStateChange() method shall be implemented and supported.
 515 The CIM_ProtocolService.RequestStateChange() method shall not return a value of 1 (Unsupported).

516 **8.3 Profile conventions for operations**

517 Support for operations for each profile class (including associations) is specified in the following
 518 ns in the default list in clause 8.3
 519 are supported as described by [DSP0200 v1.2](#)
 520 supported by this profile or where the profile requires behavior other than that described by
 521 [DSP0200 v1.2](#).

522 The default list of operations is as follows:

- 523 • GetInstance
- 524 • Associators
- 525 • AssociatorNames
- 526 • References
- 527 • ReferenceNames
- 528 • EnumerateInstances
- 529 • EnumerateInstanceNames

530 A compliant implementation shall support all of the operations in the default list for each class, unless the
531 *Mandatory*.

532 8.4 CIM_BindsTo

533 Table 6 lists operations that either have special requirements beyond those from [DSP0200 v1.2](#) or shall
534 not be supported.

535 **Table 6 – Operations: CIM_BindsTo**

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

536 8.5 CIM_ElementCapabilities

537 Table 7 lists operations that either have special requirements beyond those from [DSP0200 v1.2](#) or shall
538 not be supported.

539 **Table 7 – Operations: CIM_ElementCapabilities**

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

540 **8.6 CIM_ElementSettingData**

541 Table 8 lists operations that either have special requirements beyond those in [DSP0200 v1.2](#) or shall not
542 be supported.

543 **Table 8 – Operations: CIM_ElementSettingData**

Operation	Requirement	Messages
ModifyInstance	Optional	See clause 8.6.1.
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

544 **8.6.1 CIM_ElementSettingData – ModifyInstance**

545 When an instance of CIM_ElementSettingData associates an instance of CIM_TelnetSettingData with an
546 instance of CIM_TelnetProtocolEndpoint, the following rules shall govern the behavior of the
547 ModifyInstance operation:

- 548 • The ModifyInstance operation shall not allow the IsDefault property to be modified.
- 549 • The ModifyInstance operation shall not allow the IsCurrent property to be modified.
- 550 • When the ModifyInstance operation is used to modify the IsNext property to a value of 1 (Is
551 Next), the ModifyInstance operation shall implement the following behavior:

552 The ModifyInstance operation shall find all other instances of CIM_ElementSettingData
553 that associate a CIM_TelnetSettingData instance with the CIM_TelnetProtocolEndpoint
554 instance referenced by the target instance of CIM_ElementSettingData.

555 For each instance of CIM_ElementSettingData found, the ModifyInstance operation shall
556 modify the value of its IsNext property to a value of 2 (Is Not Next).

557 **8.7 CIM_HostedAccessPoint**

558 Table 9 lists operations that either have special requirements beyond those from [DSP0200 v1.2](#) or shall
559 not be supported.

560 **Table 9 – Operations: CIM_HostedAccessPoint**

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

561 **8.8 CIM_HostedService**

562 Table 10 lists operations that either have special requirements beyond those from [DSP0200 v1.2](#) or shall
563 not be supported.

564 **Table 10 – Operations: CIM_HostedService**

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

565 **8.9 CIM_ProtocolService**

566 Table 11 lists operations that either have special requirements beyond those from [DSP0200 v1.2](#) or shall
567 not be supported.

568 **Table 11 – Operations: CIM_ProtocolService**

Operation	Requirement	Messages
ModifyInstance	Optional. See clause 8.9.1.	None

569 **8.9.1 CIM_ProtocolService – ModifyInstance**

570 When the ElementNameEditSupported property of the CIM_TelnetCapabilities instance has a value of
571 TRUE, the ModifyInstance operation shall allow the value of the ElementName property of the
572 CIM_ProtocolService instance to be modified. The ModifyInstance operation shall enforce the length
573 restriction specified in the MaxElementNameLen property of the CIM_TelnetCapabilities instance.

574 When the ElementNameEditSupported property of the CIM_TelnetCapabilities has a value of FALSE, the
575 ModifyInstance operation shall not change the value of the ElementName property of the
576 CIM_ProtocolService instance.

577 **8.10 CIM_ProvidesEndpoint**

578 Table 12 lists operations that either have special requirements beyond those from [DSP0200 v1.2](#) or shall
579 not be supported.

580 **Table 12 – Operations: CIM_ProvidesEndpoint**

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

581 **8.11 CIM_ServiceAccessBySAP**

582 Table 13 lists operations that either have special requirements beyond those from [DSP0200 v1.2](#) or shall
583 not be supported.

584 **Table 13 – Operations: CIM_ServiceAccessBySAP**

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

585 **8.12 CIM_TelnetCapabilities**

586 All operations in the default list in clause 8.3 are supported as described by [DSP0200 v1.2](#).

587 **8.13 CIM_TelnetSettingData**

588 Table 14 lists operations that either have special requirements beyond those from [DSP0200 v1.2](#) or shall
589 not be supported.

590 **Table 14 – Operations: CIM_TelnetSettingData**

Operation	Requirement	Messages
ModifyInstance	Optional. See clause 8.13.1.	None

591 **8.13.1 CIM_TelnetSettingData – ModifyInstance**

592 When the CIM_TelnetSettingData instance is associated with the CIM_ProtocolService instance through
593 an instance of CIM_ElementSettingData and the value of the IsDefault property of the
594 CIM_ElementSettingData instance has a value of 1 (Is Default), the ModifyInstance operation shall not be
595 supported.

596 When the CIM_TelnetSettingData instance is not associated with an instance of CIM_ProtocolService
597 through an instance of CIM_ElementSettingData where the IsDefault property has a value of 1 (Is
598 Default), the ModifyInstance operation may be supported for the CIM_TelnetSettingData instance.

599 **8.14 CIM_TelnetProtocolEndpoint**

600 Table 15 lists operations that either have special requirements beyond those from [DSP0200 v1.2](#) or shall
601 not be supported.

602 **Table 15 – Operations: CIM_TelnetProtocolEndpoint**

Operation	Requirement	Messages
ModifyInstance	Optional. See clause 8.14.1.	None
DeleteInstance	Optional. See clause 8.14.2.	None

603 8.14.1 ModifyInstance

604 The ModifyInstance operation may be supported for an instance of CIM_TelnetProtocolEndpoint. When
 605 the ModifyInstance operation is supported for an instance of CIM_TelnetProtocolEndpoint, the
 606 ModifyInstance operation shall not modify the following properties:

- 607 • NameFormat
- 608 • ProtocolIFTType
- 609 • OtherTypeDescription

610 8.14.2 DeleteInstance

611 The DeleteInstance operation may be supported for instances of CIM_TelnetProtocolEndpoint. When the
 612 DeleteInstance operation is invoked against an instance, the corresponding telnet session shall be
 613 terminated prior to deleting the CIM_TelnetProtocolEndpoint instance. The implementation shall also
 614 remove any association instances that reference the CIM_TelnetProtocolEndpoint.

615 8.15 CIM_TCPProtocolEndpoint

616 Table 16 lists operations that either have special requirements beyond those in [DSP0200 v1.2](#) or shall not
 617 be supported.

618 **Table 16 – Operations: CIM_TCPProtocolEndpoint**

Operation	Requirement	Messages
ModifyInstance	Optional. See clause 8.15.1.	None
DeleteInstance	Optional. See clause 8.15.2.	None

619 8.15.1 ModifyInstance

620 The ModifyInstance operation may be supported for an instance of CIM_TCPProtocolEndpoint. When the
 621 ModifyInstance operation is supported for an instance of CIM_TCPProtocolEndpoint, the ModifyInstance
 622 operation shall not modify the following properties:

- 623 • NameFormat
- 624 • ProtocolIFTType
- 625 • PortNumber

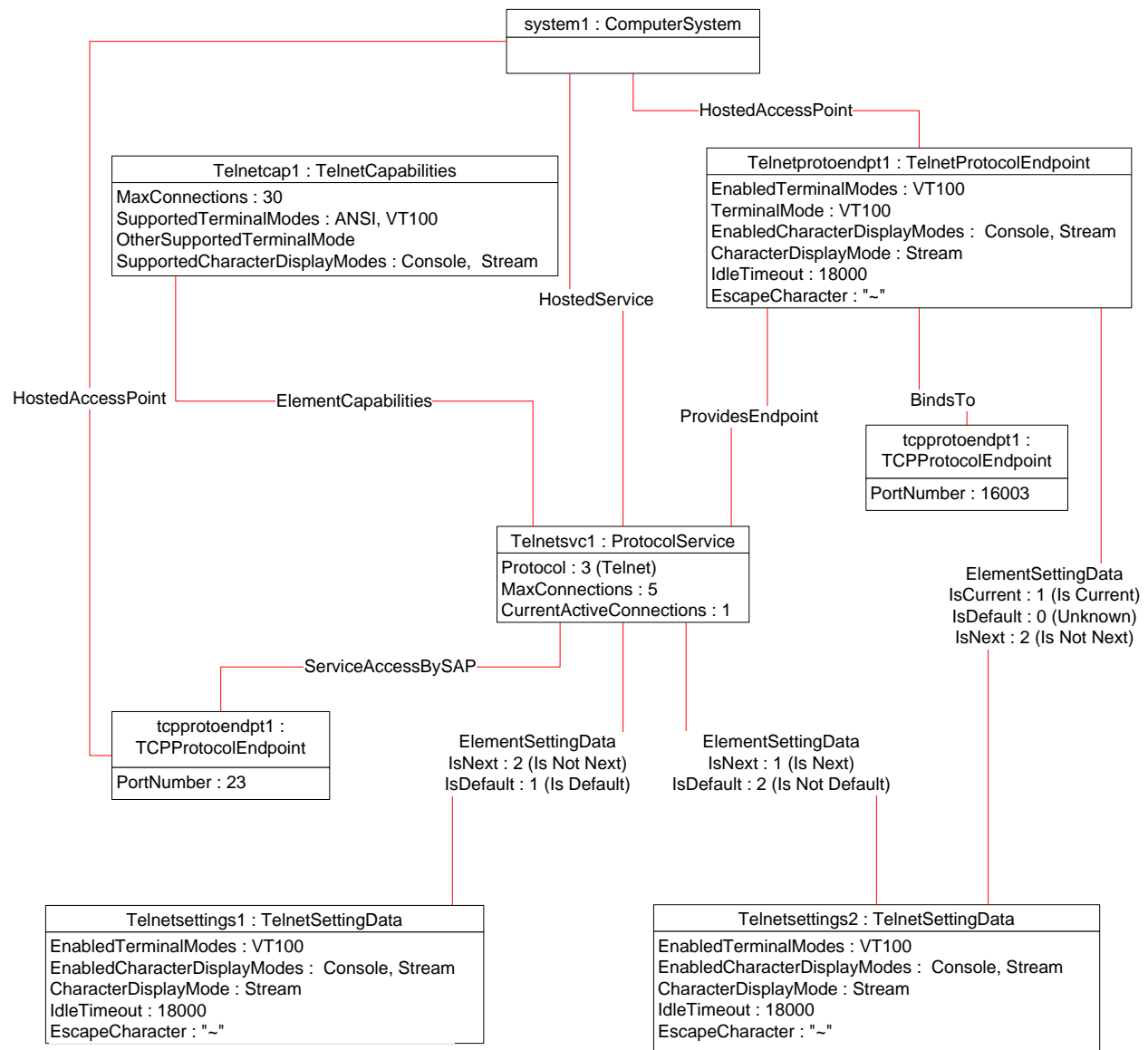
626 8.15.2 DeleteInstance

627 When the CIM_ProtocolService.ListenOnPort() method is supported for the instance of
 628 CIM_ProtocolService with which the CIM_TCPProtocolEndpoint is associated through an instance of
 629 CIM_ServiceAccessBySAP, the DeleteInstance operation shall be supported for the instance of
 630 CIM_TCPProtocolEndpoint. When the CIM_ProtocolService.ListenOnPort() method is not supported, the
 631 DeleteInstance operation shall not be supported.

632 When the DeleteInstance operation is successful for an instance of CIM_TCPProtocolEndpoint, the telnet
 633 service shall stop listening on the TCP/IP port indicated by the PortNumber property of the
 634 CIM_TCPProtocolEndpoint. The implementation shall also remove any association instances that
 635 reference the CIM_TCPProtocolEndpoint.

652 Figure 3 through Figure 5 illustrate the sequence of the telnet service listening for connections, a telnet
 653 session being established, and the configuration of the telnet session changing from the initial values.

654 Figure 3 is an object diagram that shows the telnet service enabled and listening for incoming
 655 connections. The instance of CIM_TelnetSettingData labeled Telnetsettings2 indicates the settings that
 656 will be applied to a telnet session when it is established. The instance of CIM_TelnetSettingData labeled
 657 Telnetsettings1 represents the default configuration for a session. The CIM_TelnetCapabilities instance
 658 indicates the capabilities of the telnet service and its associated sessions. In this example, the telnet
 659 service supports the ANSI and VT100 terminal modes, as indicated by the value of the
 660 SupportedTerminalModes property. However, the administrator has configured the service to enable
 661 sessions using only VT100. This configuration is indicated by the value of the EnabledTerminalModes
 662 property of the associated CIM_TelnetSettingData instances.

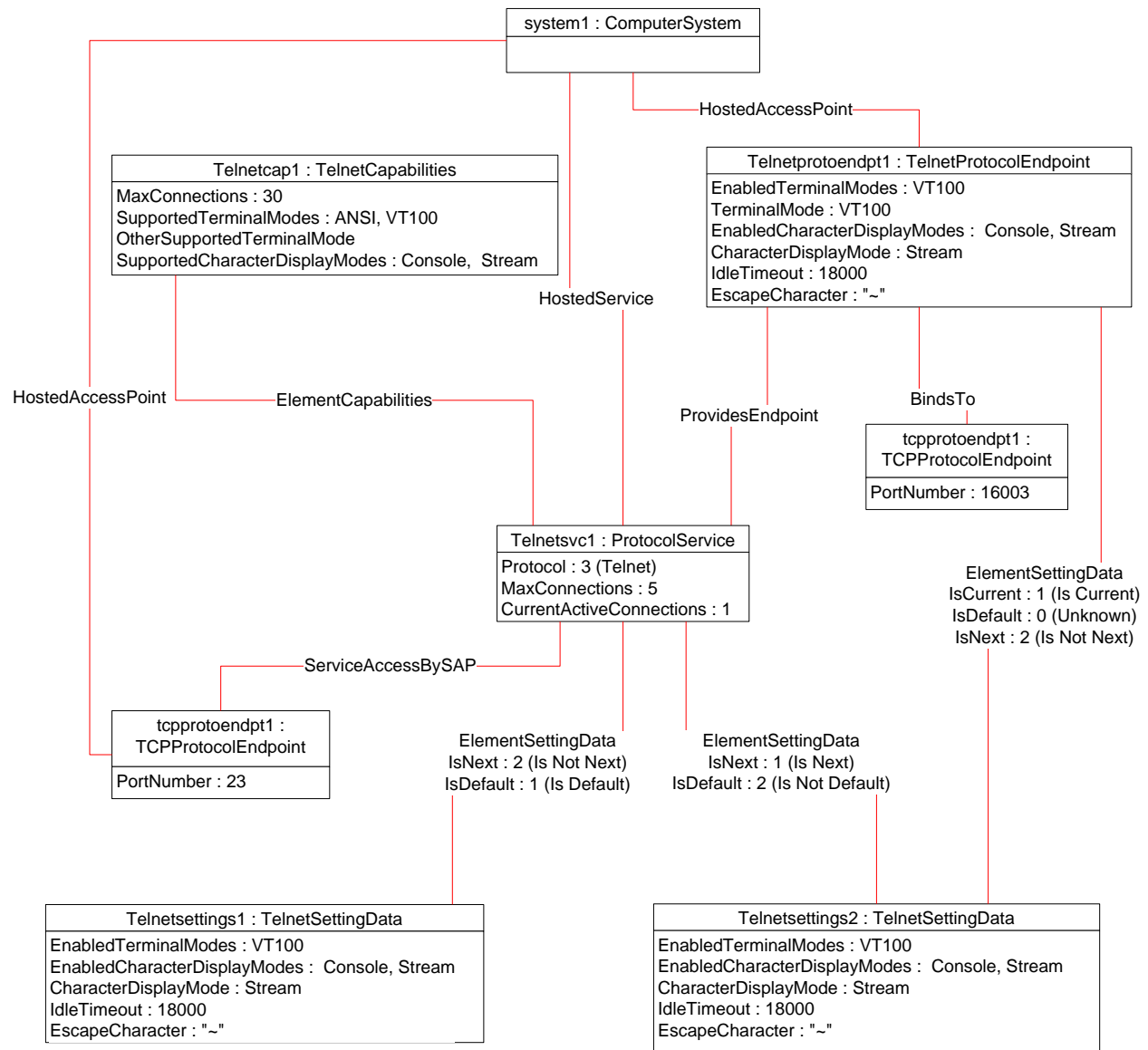


663

664

Figure 3 – Telnet service listening for connections

665 The object diagram in Figure 4 represents the same configuration as Figure 3 with the addition of an
 666 instance of CIM_TelnetProtocolEndpoint representing a newly established session. Notice that the value
 667 of the CurrentActiveConnections property of the CIM_ProtocolService instance (Telnetsvc1) has been
 668 incremented to reflect that a session is active. The values of the properties for the established session
 669 (Telnetprotoendpt1) correspond to the values of the instance of CIM_TelnetSettingData, where the value
 670 of the IsNext property on the CIM_ElementSettingData instance that associates the settings with the
 671 service has a value of 1 (Is Next).



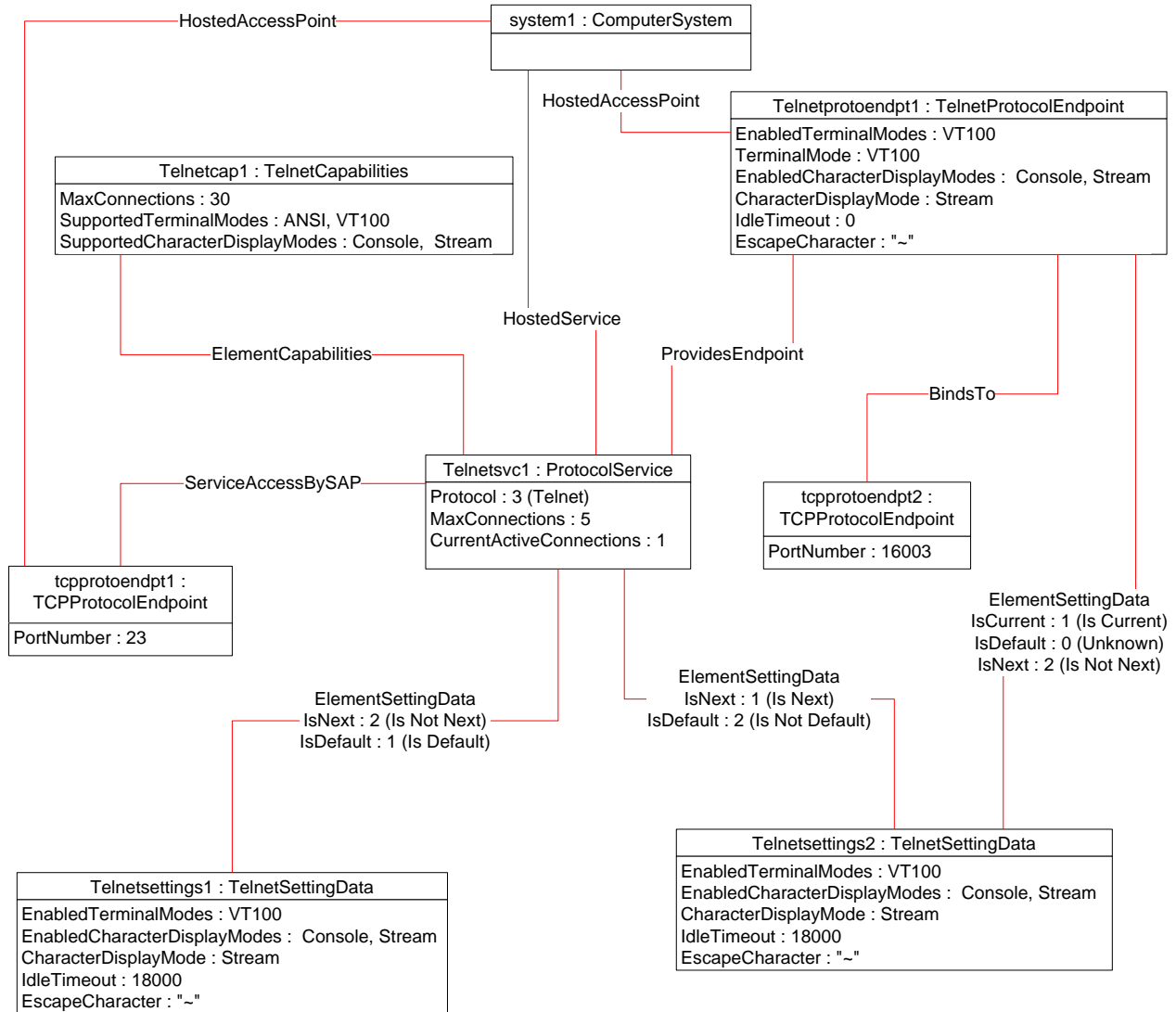
672

673

Figure 4 – One active session

674 The object diagram in Figure 5 represents the same configuration as in Figure 4 except that the user has
 675 changed session parameters from the values that were in effect when the session was initially
 676 established. The user has changed the character display mode from stream to console. This change is

677 reflected in the value of the CharacterDisplayMode property of Telnetprotoendpt1 because the
 678 CIM_TelnetProtocolEndpoint contains the actual values for the session. Notice that the value of the
 679 CharacterDisplayMode property of Telnetsettings2 remains unchanged.



680

681

Figure 5 – Session changed

682 9.2 Configuring session default settings

683 When a telnet session is established, session settings have default values. A client can change the
684 default values for subsequent sessions' settings as follows:

- 685 1) Find the instance of CIM_ElementSettingData that associates an instance of
686 CIM_TelnetSettingData with the CIM_ProtocolService, where the value of its IsNext property is
687 1 (Is Next) and the value of the IsDefault property is not 1 (Is Default).
- 688 2) Modify the properties of the referenced CIM_TelnetSettingData instance.

689 9.3 Modifying active session settings

690 A client can find the active sessions for a telnet service and modify their configuration as follows:

- 691 1) Find an instance of CIM_TelnetProtocolEndpoint that is associated with the
692 CIM_ProtocolService through an instance of CIM_ProvidesEndpoint.
- 693 2) Modify the properties of the CIM_TelnetProtocolEndpoint as needed.

694 9.4 Disabling the telnet service

695 If an implementation supports disabling the telnet service, a client can disable the telnet service by
696 invoking the RequestStateChange() method on CIM_ProtocolService instance with a value of Disabled
697 for the RequestedState parameter.

698 9.5 Determining the telnet service capabilities

699 A client can determine the capabilities of the telnet service as follows:

- 700 1) Find the instance of CIM_TelnetCapabilities associated with the CIM_ProtocolService through
701 an instance of CIM_ElementCapabilities.
- 702 2) View the properties of the CIM_TelnetCapabilities instance to see the supported function.

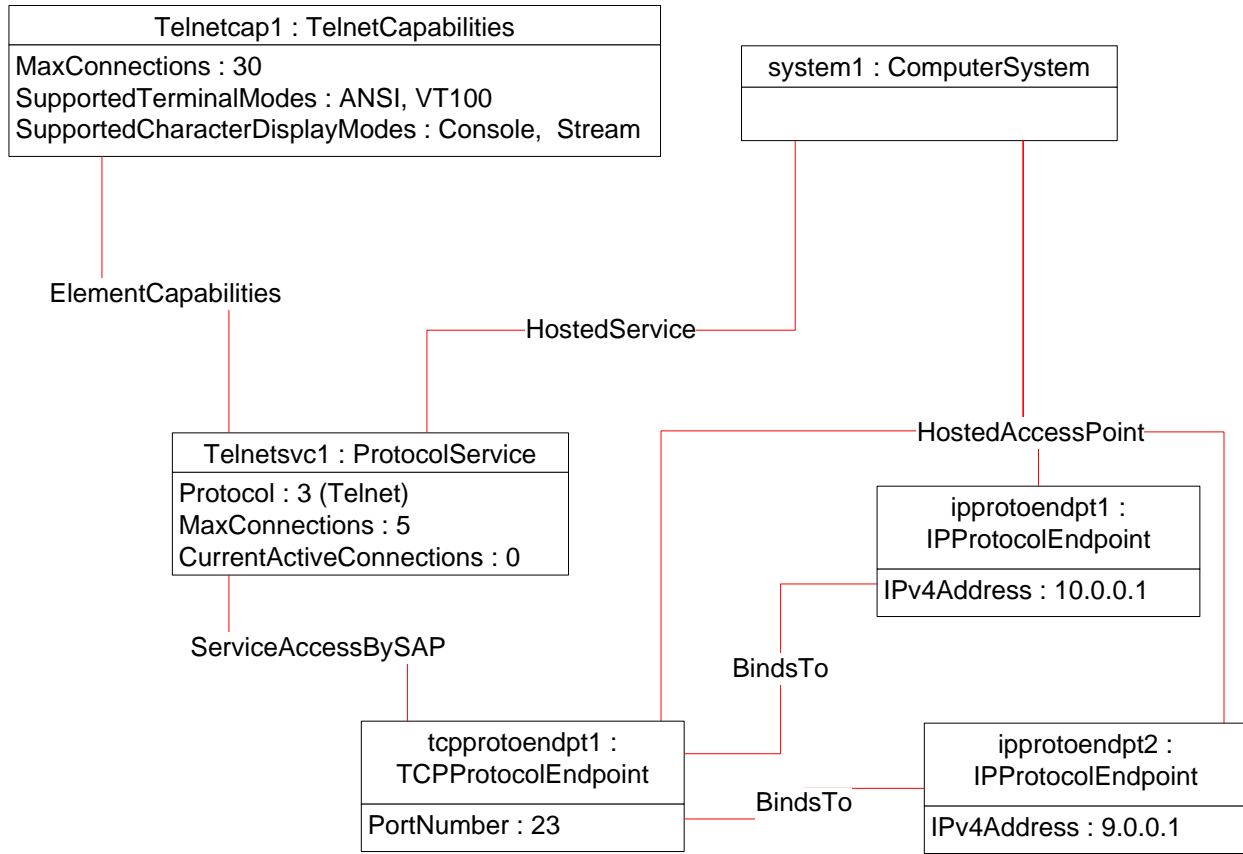
703 9.6 Determining the listening ports of the telnet service

704 An implementation can model the TCP/IP port upon which the telnet service listens for incoming
705 connection requests. When the implementation models the port, a client can determine the ports to which
706 the telnet service is bound as follows:

- 707 1) Find all instances of CIM_TCPProtocolEndpoint that are associated with the
708 CIM_ProtocolService through an instance of CIM_ServiceAccessBySAP.
- 709 2) For each instance of CIM_TCPProtocolEndpoint, query the PortNumber property.

710 Applying this query to Figure 6, the client would find a single instance of CIM_TCPProtocolEndpoint and
711 the value of the PortNumber property is 22.

712 Figure 6 is an object diagram for the telnet service listening on TCP/IP port 22 for incoming connection
713 requests across all of the IP interfaces of the host system. This is illustrated by the instances of
714 CIM_BindsTo that associate the instance of CIM_TCPProtocolEndpoint with the instances of
715 CIM_IPProtocolEndpoint.



716

717

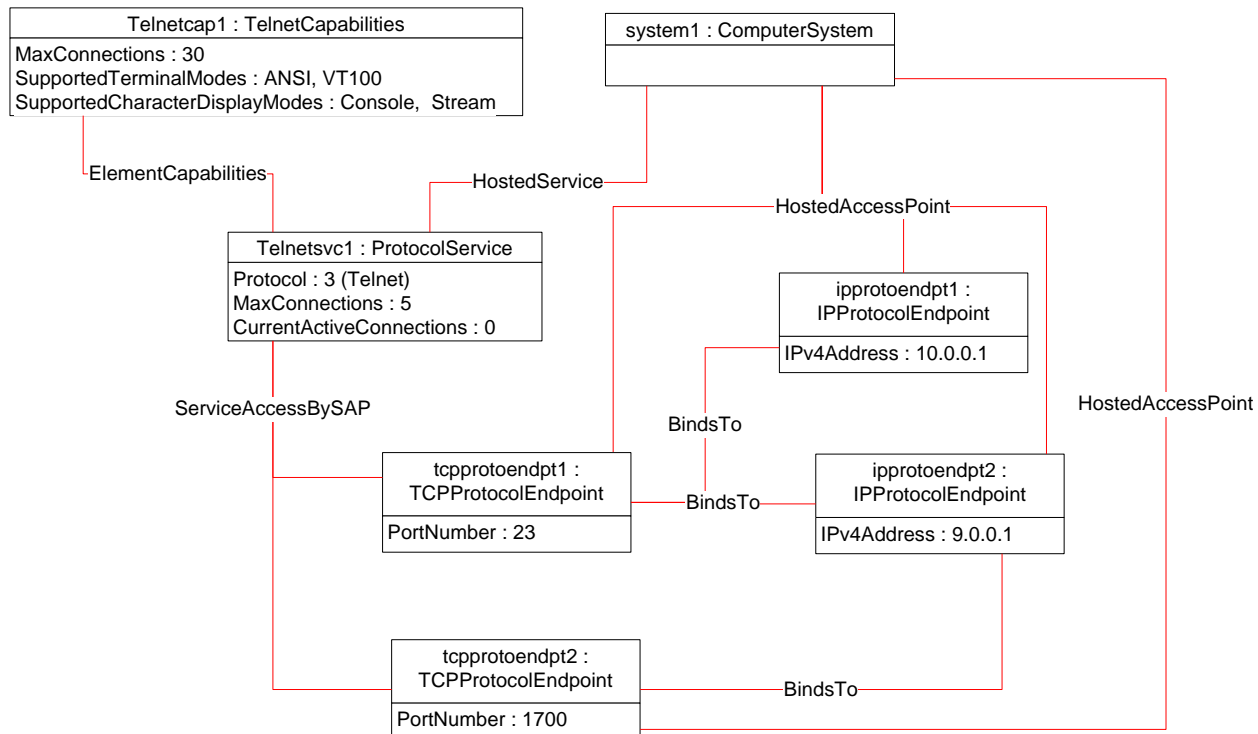
Figure 6 – Listening on a single port on all interfaces

9.7 Adding a listening port for the telnet service

719 An implementation can support adding and removing bindings between the telnet service and TCP/IP
 720 ports. When an implementation supports adding bindings, a client can configure the service to listen on all
 721 interfaces or a specific interface.

722 To have the telnet service listen on a port across all IP interfaces of the system, the client can invoke the
 723 ListenOnPort() method of the CIM_ProtocolService instance, specifying the appropriate PortNumber. To
 724 have the telnet service listen on a port for a specific interface, the client can invoke the ListenOnPort()
 725 method of the CIM_ProtocolService instance, specifying a reference to the CIM_IPProtocolEndpoint
 726 instance that represents the specific IP interface.

727 Figure 7 reflects the preceding algorithm applied to the configuration represented in Figure 6, where the
 728 ListenOnPort() method was invoked with the IPEndpoint parameter containing a reference to
 729 ipprotoendpt2 and a PortNumber parameter of 1700. The instance tcpprotoendpt2 is created and
 730 associated with ipprotoendpt2.



731

732

Figure 7 – Port added bound to specific interface

733 **9.8 Stopping the telnet service from listening on a specific port**

734 A client can stop the telnet service from listening on a specific port by invoking the intrinsic
 735 DeleteInstance operation against the instance of CIM_TCPProtocolEndpoint that represents the port.

736 Using the configuration shown in Figure 7 as an example, invoking the DeleteInstance operation against
 737 the instance tcpprotoendpt2 would cause the telnet service to no longer listen on port 1700.

738 **9.9 Determining whether ElementName can be modified**

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

- 741 1) Find the CIM_TelnetCapabilities instance that is associated with the target instance.
- 742 2) Query the value of the ElementNameEditSupported property of the CIM_TelnetCapabilities
- 743 instance. If the value is TRUE, the client can modify the ElementName property of the target
- 744 instance.

745 **9.10 Determining whether state management is supported**

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

- 748 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
 749 CIM_LANEndpoint instance.
- 750 2) Query the value of the RequestedStatesSupported property. If at least one value is specified,
 751 state management is supported.

752 **10 CIM Elements**

753 Table 17 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
 754 implemented as described in Table 17. Clauses 7 8
 755 additional requirements on these elements.

756 **Table 17 – CIM Elements: Telnet Service Profile**

Element Name	Requirement	Description
Classes		
CIM_BindsTo	Optional	See clauses 10.1 and 10.2.
CIM_ElementCapabilities	Mandatory	See clause 10.3.
CIM_ElementSettingData	Optional	See clauses 10.4 and 10.5.
CIM_HostedAccessPoint	Mandatory	See clause 10.6.
CIM_HostedService	Mandatory	See clause 10.7.
CIM_ProtocolService	Mandatory	See clause 10.8.
CIM_ProvidesEndpoint	Mandatory	See clause 10.9.
CIM_RegisteredProfile	Mandatory	See clause 10.10.
CIM_ServiceAccessBySAP	Conditional	See clause 10.11.
CIM_TelnetCapabilities	Mandatory	See clause 10.12.
CIM_TelnetProtocolEndpoint	Mandatory	See clause 10.13.
CIM_TelnetSettingData	Optional	See clause 10.14.
CIM_TCPProtocolEndpoint	Optional	See clause 10.15.
Indications		
None defined in this profile		

757 **10.1 CIM_BindsTo—TCPProtocolEndpoint**

758 When an instance of CIM_TCPProtocolEndpoint is instrumented, CIM_BindsTo is used to associate the
 759 CIM_TelnetProtocolEndpoint instance with the CIM_TCPProtocolEndpoint instance on which it depends.
 760 Table 18 provides information about the properties of CIM_BindsTo.

761 **Table 18 – Class: CIM_BindsTo (TCPProtocolEndpoint)**

Properties	Requirement	Notes
Antecedent	Mandatory	The value of this property shall be a reference to an instance of CIM_TCPProtocolEndpoint. Cardinality 0..1
Dependent	Mandatory	The value of this property shall be a reference to an instance of CIM_TelnetProtocolEndpoint. Cardinality *

762 **10.2 CIM_BindsTo—IPProtocolEndpoint**

763 When the relationship with an underlying IP interface is modeled according to clause 7.3, CIM_BindsTo is
 764 used to associate the CIM_TCPProtocolEndpoint instance with the CIM_IPProtocolEndpoint instance on
 765 which it depends. Table 19 provides information about the properties of CIM_BindsTo.

766 **Table 19 – Class: CIM_BindsTo (IPProtocolEndpoint)**

Properties	Requirement	Notes
Antecedent	Mandatory	The value of this property shall be a reference to an instance of CIM_IPProtocolEndpoint. Cardinality *
Dependent	Mandatory	The value of this property shall be a reference to an instance of CIM_TCPProtocolEndpoint. Cardinality 1..*

767 **10.3 CIM_ElementCapabilities**

768 CIM_ElementCapabilities is used to associate an instance of CIM_TelnetCapabilities with the
 769 CIM_ProtocolService. Table 20 provides information about the properties of CIM_ElementCapabilities.

770 **Table 20 – Class: CIM_ElementCapabilities**

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

771 **10.4 CIM_ElementSettingData—telnet service**

772 CIM_ElementSettingData is used to associate instances of CIM_TelnetSettingData with instances of
773 CIM_ProtocolService. Table 21 provides information about the properties of CIM_ElementSettingData.

774 **Table 21 – Class: CIM_ElementSettingData (telnet service)**

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

775 **10.5 CIM_ElementSettingData—telnet session**

776 CIM_ElementSettingData is used to associate instances of CIM_TelnetSettingData with instances of
777 CIM_TelnetProtocolEndpoint. Table 22 provides information about the properties of
778 CIM_ElementSettingData.

779 **Table 22 – Class: CIM_ElementSettingData (telnet session)**

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

780 **10.6 CIM_HostedAccessPoint**

781 CIM_HostedAccessPoint is used to associate the CIM_TelnetProtocolEndpoint and
782 CIM_TCIPProtocolEndpoint instances to their scoping CIM_ComputerSystem instance. Table 23 provides
783 information about the properties of CIM_HostedAccessPoint.

784 **Table 23 – Class: CIM_HostedAccessPoint**

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

785 **10.7 CIM_HostedService**

786 CIM_HostedService is used to associate the CIM_ProtocolService to its scoping CIM_ComputerSystem
 787 instance. Table 24 provides information about the properties of CIM_HostedService.

788 **Table 24 – Class: CIM_HostedService**

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

789 **10.8 CIM_ProtocolService**

790 CIM_ProtocolService represents the telnet service. Table 25 provides information about the properties of
791 CIM_ProtocolService.

792 **Table 25 – Class: CIM_ProtocolService**

Properties and Methods	Requirement	Notes
SystemCreationClassName	Mandatory	None
CreationClassName	Mandatory	None
SystemName	Mandatory	None
Name	Mandatory	None
Protocol	Mandatory	Matches 3 ("Telnet")
MaxConnections	Mandatory	A value of 0 (zero) shall indicate unknown.
RequestedState	Mandatory	See clauses 7.1.2.2 and 7.1.1.2.
EnabledState	Mandatory	See clauses 7.1.1.3 and 7.1.2.3.
HealthState	Mandatory	None
OperationalStatus	Mandatory	None
ElementName	Mandatory	See clauses 7.1.3 and 7.1.4.
RequestStateChange()	Mandatory	See clause 8.2.
ListenOnPort()	Mandatory	See clause 8.1.

793 **10.9 CIM_ProvidesEndpoint**

794 CIM_ProvidesEndpoint is used to associate the instance of CIM_ProtocolService with an instance of
795 CIM_TelnetProtocolEndpoint that represents a session with the service. Table 26 provides information
796 about the properties of CIM_ProvidesEndpoint.

797 **Table 26 – Class: CIM_ProvidesEndpoint**

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

798 **10.10 CIM_RegisteredProfile**

799 CIM_RegisteredProfile identifies the *Telnet Service Profile*. The CIM_RegisteredProfile class is defined by
 800 the *Profile Registration Profile*. With the exception of the mandatory values specified for the properties in
 801 Table 27, the behavior of the CIM_RegisteredProfile instance is in accordance with the constraints
 802 specified in the *Profile Registration Profile*.

803 **Table 27 – Class: CIM_RegisteredProfile**

Properties	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of " <i>Telnet Service Profile</i> ".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.1".
RegisteredOrganization	Mandatory	This property shall have a value of 2(DMTF).

804 **10.11 CIM_ServiceAccessBySAP**

805 CIM_ServiceAccessBySAP is used to associate the instance of CIM_ProtocolService with an instance of
 806 CIM_TCPEndpoint over which a session with the service can be established. Table 28 provides
 807 information about the properties of CIM_ServiceAccessBySAP.

808 **Table 28 – Class: CIM_ServiceAccessBySAP**

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

809 **10.12 CIM_TelnetCapabilities**

810 CIM_TelnetCapabilities represents the capabilities of a telnet service. Table 29 provides information
811 about the properties of CIM_TelnetCapabilities.

812 **Table 29 – Class: CIM_TelnetCapabilities**

Properties	Requirement	Notes
InstanceID	Mandatory	None
ElementName	Mandatory	None
RequestedStatesSupported	Mandatory	See clauses 7.1.1.1.1 and 7.1.2.1.1.
ElementNameEditSupported	Mandatory	See clauses 7.1.3.1.1 and 7.1.4.1.1.
MaxElementNameLen	Conditional	See clauses 7.1.3.1.2 and 7.1.4.1.2.
MaxConnections	Mandatory	A value of 0 (zero) shall indicate unknown.
SupportedTerminalModes	Mandatory	None
OtherSupportedTerminalMode	Conditional	This property shall have a value when the SupportedTerminalModes property has a value of 1 ("Other").
SupportedCharacterDisplayModes	Mandatory	None
OtherCharacterDisplayMode	Conditional	This property shall have a value when the SupportedCharacterDisplayModes property has a value of 1 ("Other").

813 **10.13 CIM_TelnetProtocolEndpoint**

814 CIM_TelnetProtocolEndpoint represents a session that is established with the telnet service. Table 30
815 provides information about the properties of CIM_TelnetProtocolEndpoint.

816 **Table 30 – Class: CIM_TelnetProtocolEndpoint**

Properties	Requirement	Notes
SystemCreationClassName	Mandatory	None
CreationClassName	Mandatory	None
SystemName	Mandatory	None
Name	Mandatory	None
NameFormat	Mandatory	None
ProtocolIFType	Mandatory	Matches 1 (Other)
OtherTypeDescription	Mandatory	Matches "Telnet"
ElementName	Mandatory	Pattern ".**"
RequestStateChange()	Mandatory	See clause 8.2.
EnabledTerminalModes	Mandatory	None
OtherEnabledTerminalModes	Conditional	This property shall have a value when the EnabledTerminalModes property has a value of 1 ("Other").
TerminalMode	Mandatory	None
OtherTerminalMode	Conditional	This property shall have a value when the TerminalMode property has a value of 1 ("Other").
EnabledCharacterDisplayModes	Mandatory	None

Properties	Requirement	Notes
OtherEnabledCharacterDisplayMode	Conditional	This property shall have a value when the EnabledCharacterDisplayModes property has a value of 1 ("Other").
CharacterDisplayMode	Mandatory	None
OtherCharacterDisplayMode	Conditional	This property shall have a value when the CharacterDisplayMode property has a value of 1 ("Other").
IdleTimeout	Mandatory	None
EscapeCharacter	Mandatory	None

817 **10.14 CIM_TelnetSettingData**

818 CIM_TelnetSettingData represents settings that can be applied to a telnet session. Table 31 provides
 819 information about the properties of CIM_TelnetSettingData.

820 **Table 31 – Class: CIM_TelnetSettingData**

Properties	Requirement	Notes
InstanceID	Mandatory	None
AddressOrigin	Mandatory	None
ElementName	Mandatory	Pattern ".*"
EnabledTerminalModes	Mandatory	None
OtherEnabledTerminalModes	Conditional	This property shall have a value when the EnabledTerminalModes property has a value of 1 ("Other").
TerminalMode	Mandatory	None
OtherTerminalMode	Conditional	This property shall have a value when the TerminalMode property has a value of 1 ("Other").
EnabledCharacterDisplayModes	Mandatory	None
OtherEnabledCharacterDisplayMode	Conditional	This property shall have a value when the EnabledCharacterDisplayModes property has a value of 1 ("Other").
CharacterDisplayMode	Mandatory	None
OtherCharacterDisplayMode	Conditional	This property shall have a value when the CharacterDisplayMode property has a value of "Other".
IdleTimeout	Mandatory	None
EscapeCharacter	Mandatory	None

821 **10.15 CIM_TCPProtocolEndpoint**

822 CIM_TCPProtocolEndpoint represents an IP port to which a telnet session or service can be bound.

823 Table 32 provides information about the properties of CIM_TCPProtocolEndpoint.

824 **Table 32 – Class: CIM_TCPProtocolEndpoint**

Properties	Requirement	Notes
SystemCreationClassName	Mandatory	None
CreationClassName	Mandatory	None
SystemName	Mandatory	None
Name	Mandatory	None
NameFormat	Mandatory	
ProtocolIFType	Mandatory	Matches 4111 ("TCP")
ElementName	Mandatory	
PortNumber	Mandatory	None

825

826
827
828
829

**ANNEX A
(Informative)**

Change log

Version	Date	Description
1.0.0	2009-06-22	
1.0.1	2019-03-15	This errata addresses these issues: <ul style="list-style-type: none"> • Updated RegisteredVersion to reflect errata version number in clause 10.10 • Updated RegisteredOrganization description to reflect correct value of 2 for DMTF in clause 10.10 and figure 2 in clause 9.1

830
831