



1

2

3

4

**Document Number: DSP1016**

**Date: 2009-06-22**

**Version: 1.0.0**

5 **Telnet Service Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

## 9 Copyright Notice

10 Copyright © 2006, 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 to  
14 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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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 .....	9
39	4 Symbols and Abbreviated Terms.....	11
40	5 Synopsis.....	11
41	6 Description .....	11
42	6.1 Telnet Session Lifecycle .....	13
43	7 Implementation Requirements .....	13
44	7.1 Representing a Telnet Service .....	13
45	7.2 Representing the Server's View of a Telnet Session.....	15
46	7.3 Relationship with IP Interfaces (Optional).....	16
47	8 Methods.....	17
48	8.1 CIM_ProtocolService.ListenOnPort( ) (Optional).....	17
49	8.2 CIM_ProtocolService.RequestStateChange( ) .....	18
50	8.3 Profile Conventions for Operations.....	19
51	8.4 CIM_BindsTo .....	19
52	8.5 CIM_ElementCapabilities .....	20
53	8.6 CIM_ElementSettingData .....	20
54	8.7 CIM_HostedAccessPoint.....	21
55	8.8 CIM_HostedService .....	21
56	8.9 CIM_ProtocolService .....	21
57	8.10 CIM_ProvidesEndpoint.....	22
58	8.11 CIM_ServiceAccessBySAP .....	22
59	8.12 CIM_TelnetCapabilities.....	22
60	8.13 CIM_TelnetSettingData .....	23
61	8.14 CIM_TelnetProtocolEndpoint.....	23
62	8.15 CIM_TCPProtocolEndpoint.....	24
63	9 Use Cases.....	24
64	9.1 Object Diagrams .....	24
65	9.2 Configuring Session Default Settings .....	29
66	9.3 Modifying Active Session Settings.....	29
67	9.4 Disabling the Telnet Service .....	29
68	9.5 Determining the Telnet Service Capabilities.....	29
69	9.6 Determining the Listening Ports of the Telnet Service .....	29
70	9.7 Adding a Listening Port for the Telnet Service .....	30
71	9.8 Stopping the Telnet Service from Listening on a Specific Port .....	31
72	9.9 Determining Whether ElementName Can Be Modified .....	31
73	9.10 Determining Whether State Management Is Supported.....	32
74	10 CIM Elements.....	32
75	10.1 CIM_BindsTo — TCPProtocolEndpoint.....	33
76	10.2 CIM_BindsTo — IPProtocolEndpoint.....	33
77	10.3 CIM_ElementCapabilities .....	33
78	10.4 CIM_ElementSettingData — Telnet Service .....	34
79	10.5 CIM_ElementSettingData — m ,Telnet Session.....	34
80	10.6 CIM_HostedAccessPoint.....	34
81	10.7 CIM_HostedService .....	35
82	10.8 CIM_ProtocolService .....	35
83	10.9 CIM_ProvidesEndpoint.....	36

84	10.10 CIM_RegisteredProfile.....	36
85	10.11 CIM_ServiceAccessBySAP .....	36
86	10.12 CIM_TelnetCapabilities.....	37
87	10.13 CIM_TelnetProtocolEndpoint.....	37
88	10.14 CIM_TelnetSettingData .....	38
89	10.15 CIM_TCPProtocolEndpoint.....	39
90	ANNEX A (Informative) Change Log .....	40

91

## 92 Figures

93	Figure 1 – Telnet Service Profile: Class Diagram.....	12
94	Figure 2 – Registered Profile .....	25
95	Figure 3 – Telnet Service Listening for Connections.....	26
96	Figure 4 – One Active Session .....	27
97	Figure 5 – Session Changed .....	28
98	Figure 6 – Listening on a Single Port on All Interfaces.....	30
99	Figure 7 – Port Added Bound to Specific Interface.....	31

100

## 101 Tables

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

128 Table 27 – Class: CIM\_RegisteredProfile..... 36

129 Table 28 – Class: CIM\_ServiceAccessBySAP ..... 36

130 Table 29 – Class: CIM\_TelnetCapabilities..... 37

131 Table 30 – Class: CIM\_TelnetProtocolEndpoint..... 37

132 Table 31 – Class: CIM\_TelnetSettingData..... 38

133 Table 32 – Class: CIM\_TCIPProtocolEndpoint..... 39

134



136

## Foreword

137 The *Telnet Service Profile* (DSP1016) was prepared by the Server Management Working Group and the  
138 Physical Platform Profiles Working Group of the DMTF.

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

## 141 Acknowledgments

142 The authors wish to acknowledge the following people.

### 143 Editor:

- 144 • Aaron Merkin – IBM

### 145 Contributors:

- 146 • Jon Hass – Dell
- 147 • Khachatur Papanyan – Dell
- 148 • Enoch Suen – Dell
- 149 • Jeff Hilland – HP
- 150 • Christina Shaw – HP
- 151 • Perry Vincent – Intel
- 152 • John Leung – Intel

153

## Introduction

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

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



160

# Telnet Service Profile

## 161 1 Scope

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

## 164 2 Normative References

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

### 168 2.1 Approved References

169 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,  
170 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.5.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf)

171 DMTF DSP0200, *CIM Operations over HTTP 1.3*,  
172 [http://www.dmtf.org/standards/published\\_documents/DSP0200\\_1.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf)

173 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,  
174 [http://www.dmtf.org/standards/published\\_documents/DSP1001\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf)

175 DMTF DSP1033, *Profile Registration Profile 1.0*,  
176 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf)

177 DMTF DSP1036, *IP Interface Profile 1.0*,  
178 [http://www.dmtf.org/standards/published\\_documents/DSP1036\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1036_1.0.pdf)

### 179 2.2 Other References

180 IETF RFC 1208, *A Glossary of Networking Terms*, March 1991, <http://www.ietf.org/rfc/rfc1208.txt>

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

## 183 3 Terms and Definitions

184 For the purposes of this document, the terms and definitions given in [DSP1033](#) and [DSP1001](#) and the  
185 following apply.

### 186 3.1

#### 187 **can**

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

### 189 3.2

#### 190 **cannot**

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

- 192 **3.3**  
193 **conditional**  
194 indicates requirements to be followed strictly to conform to the document when the specified conditions  
195 are met
- 196 **3.4**  
197 **mandatory**  
198 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
199 permitted
- 200 **3.5**  
201 **may**  
202 indicates a course of action permissible within the limits of the document
- 203 **3.6**  
204 **need not**  
205 indicates a course of action permissible within the limits of the document
- 206 **3.7**  
207 **optional**  
208 indicates a course of action permissible within the limits of the document
- 209 **3.8**  
210 **referencing profile**  
211 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
212 "Referenced Profiles" table
- 213 **3.9**  
214 **shall**  
215 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
216 permitted
- 217 **3.10**  
218 **shall not**  
219 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
220 permitted
- 221 **3.11**  
222 **should**  
223 indicates that among several possibilities, one is recommended as particularly suitable, without  
224 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 225 **3.12**  
226 **should not**  
227 indicates that a certain possibility or course of action is deprecated but not prohibited
- 228 **3.13**  
229 **unspecified**  
230 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 231 **3.14**  
232 **Listening Port**  
233 a TCP/IP port that the telnet service is bound to and listening for incoming connection requests

234 **4 Symbols and Abbreviated Terms**

235 The following abbreviations are used in this document.

236 **4.1**

237 **CIM**

238 Common Information Model

239 **4.2**

240 **IP**

241 Internet Protocol

242 **4.3**

243 **TCP**

244 Transmission Control Protocol

245 **5 Synopsis**

246 **Profile Name:** Telnet Service

247 **Version:** 1.0.0

248 **Organization:** DMTF

249 **CIM Schema Version:** 2.22

250 **Central Class:** CIM\_ProtocolService

251 **Scoping Class:** CIM\_ComputerSystem

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

255 The Central Class for the *Telnet Service Profile* shall be the CIM\_ProtocolService class. The Central  
 256 Instance of the *Telnet Service Profile* shall be an instance of CIM\_ProtocolService. The Scoping Class for  
 257 the *Telnet Service Profile* shall be CIM\_ComputerSystem. The Scoping Instance of the *Telnet Service*  
 258 *Profile* shall be the instance of CIM\_ComputerSystem to which the Central Instance is associated through  
 259 an instance of CIM\_HostedService.

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

261 **Table 1 – Referenced Profiles**

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

262 **6 Description**

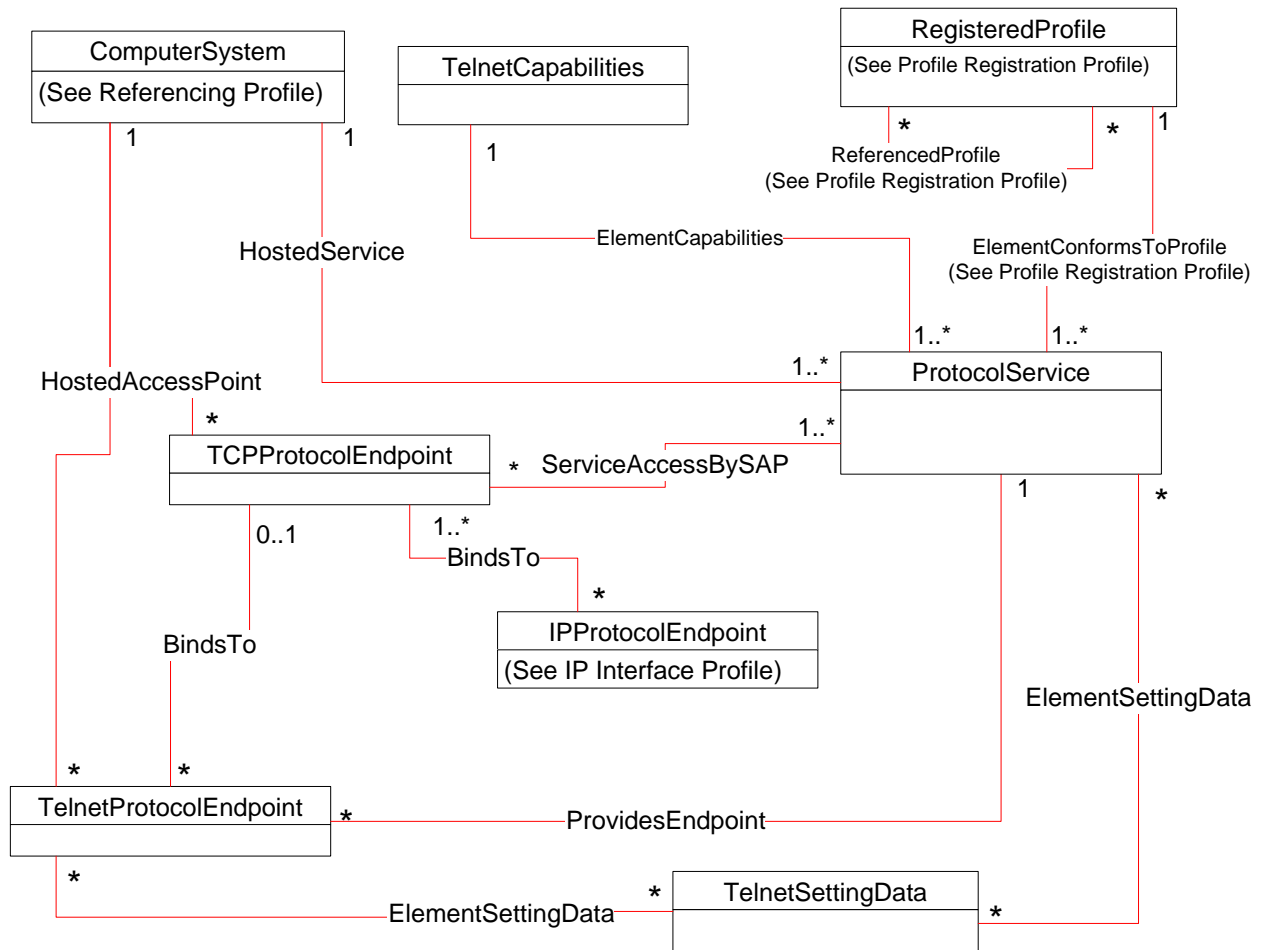
263 The *Telnet Service Profile* extends the management capability of referencing profiles by adding the  
 264 capability to represent a telnet service hosted on a managed system. Functionality within the scope of this  
 265 profile includes:

- 266 • representation of the telnet service
- 267 • representation of the Telnet server's view of active sessions

- 268 • configuration of the telnet service
- 269 • configuration of the telnet sessions from the Telnet server

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

272 Figure 1 represents the class schema for the *Telnet Service Profile*. For simplicity, the prefix CIM\_ has  
 273 been removed from the names of the classes.



274

275

**Figure 1 – Telnet Service Profile: Class Diagram**

276 This profile represents the capabilities of the telnet service, the current configuration of the telnet service,  
 277 the active sessions, and the default settings when new sessions are activated. The telnet service is  
 278 represented by an instance of CIM\_ProtocolService, where the Protocol property has the value 3 (Telnet).  
 279 The capabilities of the telnet service are represented by an instance of CIM\_TelnetCapabilities. The  
 280 current configuration of the telnet service is modeled with the properties from the instance of  
 281 CIM\_ProtocolService. Each active session with the telnet service is represented by an instance of  
 282 CIM\_TelnetProtocolEndpoint. The current configuration of an active session is reflected in the values of  
 283 the properties from the instance of CIM\_TelnetProtocolEndpoint. CIM\_TelnetSettingData represents a  
 284 complete configuration that a telnet session could have. For example, an instance of  
 285 CIM\_TelnetSettingData contains the configuration that will be in effect for a telnet session when it is first  
 286 established. CIM\_TCPProtocolEndpoint is an optional endpoint used to model the TCP/IP ports over  
 287 which a telnet service listens or a telnet session is active.

## 288 **6.1 Telnet Session Lifecycle**

289 When a telnet session is established with the telnet service, an instance of CIM\_TelnetProtocolEndpoint  
290 is created. The CIM\_TelnetProtocolEndpoint instance exists for the duration of the telnet session that it  
291 represents. When the telnet session ends, the CIM\_TelnetProtocolEndpoint is removed. When the  
292 CIM\_TelnetProtocolEndpoint is explicitly deleted through an intrinsic DeleteInstance operation, the telnet  
293 session ends.

## 294 **7 Implementation Requirements**

295 This section details the requirements related to the arrangement of instances and their properties for  
296 implementations of this profile.

### 297 **7.1 Representing a Telnet Service**

298 An instance of CIM\_ProtocolService shall represent the telnet service being modeled.

#### 299 **7.1.1 Telnet Service State Management Is Supported — Conditional**

300 This section describes the CIM elements and behaviors that shall be implemented when this behavior is  
301 supported.

##### 302 **7.1.1.1 General**

303 When management of the state of a telnet service is supported, exactly one instance of  
304 CIM\_TelnetCapabilities shall be associated with the CIM\_ProtocolService instance through an instance of  
305 CIM\_ElementCapabilities.

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

##### 309 **7.1.1.2 CIM\_TelnetCapabilities**

310 When state management is supported, exactly one instance of CIM\_TelnetCapabilities shall be  
311 associated with the CIM\_ProtocolService instance through an instance of CIM\_ElementCapabilities.

##### 312 **7.1.1.2.1 CIM\_TelnetCapabilities.RequestedStatesSupported**

313 The RequestedStatesSupported property may contain zero or more of the following values: 2 (Enabled),  
314 3 (Disabled), or 11 (Reset).

##### 315 **7.1.1.3 CIM\_ProtocolService.RequestedState**

316 When the CIM\_ProtocolService.RequestStateChange() method is successfully invoked, the value of the  
317 RequestedState property shall be the value of the RequestedState parameter. If the method is not  
318 successfully invoked, the value of the RequestedState property is indeterminate.

319 The CIM\_ProtocolService.RequestedState property shall have one of the values specified in the  
320 CIM\_TelnetCapabilities.RequestedStatesSupported property or a value of 5 (No Change).

##### 321 **7.1.1.4 CIM\_ProtocolService.EnabledState**

322 When the RequestedState parameter has a value of 2 (Enabled) or 3 (Disabled) and the  
323 CIM\_ProtocolService.RequestStateChange() method completes successfully, the value of the  
324 EnabledState property shall equal the value of the CIM\_ProtocolService.RequestedState property.

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

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

## 327 **7.1.2 Telnet Service State Management Is Not Supported**

328 This section describes the CIM elements and behaviors that shall be implemented when management of  
329 the telnet service state is not supported.

### 330 **7.1.2.1 CIM\_TelnetCapabilities**

331 When state management is not supported, exactly one instance of CIM\_TelnetCapabilities may be  
332 associated with the CIM\_ProtocolService instance through an instance of CIM\_ElementCapabilities.

#### 333 **7.1.2.1.1 CIM\_TelnetCapabilities.RequestedStatesSupported**

334 The CIM\_TelnetCapabilities.RequestedStatesSupported property shall not contain any values.

#### 335 **7.1.2.2 CIM\_ProtocolService.RequestedState**

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

#### 337 **7.1.2.3 CIM\_ProtocolService.EnabledState**

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

## 340 **7.1.3 Modifying ElementName Is Supported—Conditional**

341 The CIM\_ProtocolService.ElementName property may support being modified by the ModifyInstance  
342 operation (see 8.9.1). This behavior is conditional. For information about how a client can determine  
343 whether it is supported, see 9.8.

344 This section describes the CIM elements and behavior requirements when an implementation supports  
345 client modification of the CIM\_ProtocolService.ElementName property.

### 346 **7.1.3.1 CIM\_TelnetCapabilities**

347 An instance of CIM\_TelnetCapabilities shall be associated with the CIM\_ProtocolService instance  
348 through an instance of CIM\_ElementCapabilities.

#### 349 **7.1.3.1.1 CIM\_TelnetCapabilities.ElementNameEditSupported**

350 The ElementNameEditSupported property shall have a value of TRUE when the implementation supports  
351 client modification of the CIM\_ProtocolService.ElementName property.

#### 352 **7.1.3.1.2 CIM\_EnabledLogicalElement.MaxElementNameLen**

353 The MaxElementNameLen property shall be implemented.

## 354 **7.1.4 Modifying ElementName Is Not Supported**

355 This section describes the CIM elements and behaviors that shall be implemented when the  
356 CIM\_ProtocolService.ElementName does not support being modified by the ModifyInstance operation.

### 357 **7.1.4.1 CIM\_TelnetCapabilities**

358 An instance of CIM\_TelnetCapabilities may be associated with the CIM\_ProtocolService instance through  
359 an instance of CIM\_ElementCapabilities.

#### 360 **7.1.4.1.1 CIM\_TelnetCapabilities.ElementNameEditSupported**

361 The ElementNameEditSupported property shall have a value of FALSE when the implementation does  
362 not support client modification of the CIM\_ProtocolService.ElementName property.

#### 363 **7.1.4.1.2 CIM\_EnabledLogicalElement.MaxElementNameLen**

364 The MaxElementNameLen property may be implemented. The MaxElementNameLen property is  
365 irrelevant in this context.

### 366 **7.1.5 Default Configuration of the Service**

367 The default configuration is the configuration of the service when it was first installed on the managed  
368 system. When an implementation exposes the default configuration, the default configuration shall be  
369 represented by an instance of CIM\_TelnetSettingData associated with the CIM\_ProtocolService through  
370 an instance of CIM\_ElementSettingData, where the IsDefault property of the CIM\_ElementSettingData  
371 instance has a value of 1 (Is Default).

#### 372 **7.1.5.1 Listening Port**

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

##### 375 **7.1.5.1.1 CIM\_TCPProtocolEndpoint**

376 For each TCP/IP port to which the telnet service is bound, there shall be an instance of  
377 CIM\_TCPProtocolEndpoint in which the PortNumber property of the instance indicates the TCP/IP port  
378 number to which the telnet service is listening.

##### 379 **7.1.5.1.2 Relationship to the Telnet Service**

380 For each CIM\_TCPProtocolEndpoint instance, an instance of CIM\_ServiceAccessBySAP shall associate  
381 the CIM\_ProtocolService instance with the CIM\_TCPProtocolEndpoint instance.

#### 382 **7.1.5.2 Managing Listening Ports**

383 The implementation may support managing the ports on which the telnet service listens. This behavior is  
384 optional. The ListenOnPort() method (see 8.1) of the CIM\_ProtocolService class can be used to add  
385 ports on which the telnet service will listen. Using the DeleteInstance intrinsic operation to delete an  
386 instance of CIM\_TCPProtocolEndpoint will stop the telnet service from listening on the represented port  
387 (see 8.15.2).

## 388 **7.2 Representing the Server's View of a Telnet Session**

389 Each active session with the telnet service shall be represented by an instance of  
390 CIM\_TelnetProtocolEndpoint.

### 391 **7.2.1 Relationship with Service**

392 An instance of CIM\_ProvidesEndpoint shall associate the CIM\_ProtocolService with the  
393 CIM\_TelnetProtocolEndpoint.

### 394 **7.2.2 Port for Session**

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

### 397 **7.2.2.1 CIM\_TCPProtocolEndpoint**

398 When the TCP/IP port to which the telnet session is bound is modeled, the TCP/IP port shall be modeled  
399 using an instance of CIM\_TCPProtocolEndpoint.

### 400 **7.2.2.2 Relationship to Session**

401 An instance of CIM\_BindsTo shall associate the CIM\_TelnetProtocolEndpoint instance with the  
402 CIM\_TCPProtocolEndpoint.

## 403 **7.2.3 Session Default Configuration**

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

### 407 **7.2.3.1 Configuration That Will Be Assigned**

408 An implementation may assign the same initial configuration for all telnet sessions that are spawned.  
409 When the implementation assigns the same initial configuration for all telnet sessions, the configuration  
410 that a session will have when it is established shall be represented by an instance of  
411 CIM\_TelnetSettingData that is associated with the instance of CIM\_ProtocolService through an instance  
412 of CIM\_ElementSettingData, where the IsNext property of the CIM\_ElementSettingData instance has a  
413 value of 1 (Is Next).

### 414 **7.2.3.2 Initial Configuration of a Session**

415 The initial configuration of a session may be modeled. This behavior is optional. When the configuration  
416 that a session had when it was established is modeled, it shall be represented by an instance of  
417 CIM\_TelnetSettingData that is associated with the instance of CIM\_TelnetProtocolEndpoint through an  
418 instance of CIM\_ElementSettingData, where the IsCurrent property of the CIM\_ElementSettingData  
419 instance has a value of 1 (Is Current).

420 A discrete instance of CIM\_TelnetSettingData is not required for each active session. The instance of  
421 CIM\_TelnetSettingData that is associated with the instance of CIM\_TelnetProtocolEndpoint needs only to  
422 accurately reflect the initial configuration of the session.

## 423 **7.3 Relationship with IP Interfaces (Optional)**

424 This section details requirements for specifying the relationship between the telnet session or service and  
425 one or more IP interfaces of the system.

### 426 **7.3.1 Modeling the IP Interface over Which a Session Was Established**

427 When the specific port for a telnet session is modeled, the specific IP interface over which the session is  
428 active may be modeled. This behavior is optional. When the implementation models the specific interface  
429 over which a telnet session is active, there shall be an instance of the CIM\_BindsTo association where  
430 the value of the Antecedent property shall be a reference to the CIM\_IPProtocolEndpoint instance and  
431 the value of the Dependent property shall be a reference to the CIM\_TCPProtocolEndpoint instance.

### 432 **7.3.2 Modeling the IP Interfaces for the Service**

433 When the specific port for a telnet session or service is modeled, the specific IP interface over which the  
434 session is active may be modeled. This behavior is optional. When the implementation models the  
435 specific interface over which a telnet session is active, there shall be an instance of the CIM\_BindsTo  
436 association where the value of the Antecedent property shall be a reference to the  
437 CIM\_IPProtocolEndpoint instance and the value of the Dependent property shall be a reference to the  
438 CIM\_TCPProtocolEndpoint instance. When the CIM\_TCPProtocolEndpoint is not associated with one or



439 more instances of CIM\_IPProtocolEndpoint through an instance of CIM\_BindsTo, the telnet service  
 440 accepts connections over all the IP interfaces of the system.

441 **8 Methods**

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

444 **8.1 CIM\_ProtocolService.ListenOnPort() (Optional)**

445 The CIM\_ProtocolService.ListenOnPort() method shall be supported when the ListeningPortManagement  
 446 property of the associated instance of CIM\_TelnetCapabilities has a value of TRUE. When the value of  
 447 the ListeningPortManagement property is FALSE, the CIM\_ProtocolService.ListenOnPort() method shall  
 448 not be supported.

449 The CIM\_ProtocolService.ListenOnPort() method is used to configure additional ports on which the  
 450 CIM\_ProtocolService instance will listen. Detailed requirements of the ListenOnPort() method are  
 451 specified in Table 2 and Table 3.

452 No standard messages are defined.

453 **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

454 **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

455 When the method completes successfully, the implementation shall create an instance of  
 456 CIM\_TCPProtocolEndpoint. The value of the PortNumber property of the instance of  
 457 CIM\_TCPProtocolEndpoint shall be the value of the PortNumber parameter of the method invocation.  
 458 The implementation shall create an instance of CIM\_ServiceAccessBySAP that references the instance  
 459 of CIM\_TCPProtocolEndpoint and references the instance of CIM\_ProtocolService on which the method  
 460 was invoked.

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

- 463 • The implementation shall create an instance of CIM\_HostedAccessPoint that references the  
 464 newly created CIM\_TCPProtocolEndpoint instance and the instance of CIM\_ComputerSystem

465 with which the CIM\_ProtocolService instance is associated through an instance of  
 466 CIM\_HostedService (the scoping system).

- 467 • For each instance of CIM\_IPProtocolEndpoint that is associated through an instance of  
 468 CIM\_HostedAccessPoint with the CIM\_ComputerSystem instance with which the instance of  
 469 CIM\_ProtocolService on which this method was invoked is associated through an instance of  
 470 CIM\_HostedService, the implementation shall create an instance of the CIM\_BindsTo  
 471 association where the value of the Antecedent property shall be a reference to the  
 472 CIM\_IPProtocolEndpoint instance and the value of the Dependent property shall be a reference  
 473 to the CIM\_TCPProtocolEndpoint instance.

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

- 475 • The implementation shall create an instance of CIM\_HostedAccessPoint that references the  
 476 newly created CIM\_TCPProtocolEndpoint instance and the instance of CIM\_ComputerSystem  
 477 with which the CIM\_IPProtocolEndpoint instance is associated through an instance of  
 478 CIM\_HostedAccessPoint.
- 479 • The implementation shall create an instance of the CIM\_BindsTo association where the value of  
 480 the Antecedent property shall be a reference to the CIM\_IPProtocolEndpoint instance and the  
 481 value of the Dependent property shall be a reference to the CIM\_TCPProtocolEndpoint  
 482 instance.

## 483 8.2 CIM\_ProtocolService.RequestStateChange()

484 Invocation of the CIM\_ProtocolService.RequestStateChange() method will change the element's state to  
 485 the value specified in the RequestedState parameter. The Enabled and Disabled values of the  
 486 RequestedState parameter correspond to enabling or disabling the functionality represented by the  
 487 instance of CIM\_ProtocolService. A value of 2 (Enabled) shall correspond to a request to enable the  
 488 functionality. A value of 3 (Disabled) shall correspond to a request to disable the functionality. A value of  
 489 11 (Reset) shall initiate a reset of the telnet service.

490 See 7.1.1.3 for information about the effect of this method on the RequestedState property.

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

496 See 7.1.1.4 for information about the effect of this method on the EnabledState property.

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

498 No standard messages are defined.

499 Invoking the CIM\_ProtocolService.RequestStateChange() method multiple times could result in earlier  
 500 requests being overwritten or lost.

501 **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

502

**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

### 503 8.2.1 CIM\_ProtocolService.RequestStateChange() ConditionalSupport

504 Support for the RequestStateChange() method is conditional on the indication of support for at least one  
505 value for the RequestedState parameter as advertised through the RequestedStatesSupported property  
506 of an associated instance of CIM\_EnabledLogicalElementCapabilities. When the  
507 CIM\_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least one  
508 value, the CIM\_ProtocolService.RequestStateChange() method shall be implemented and supported.  
509 The CIM\_ProtocolService.RequestStateChange() method shall not return a value of 1 (Unsupported).

### 510 8.3 Profile Conventions for Operations

511 For each profile class (including associations), the implementation requirements for operations, including  
512 those in the following default list, are specified in class-specific subclauses of this clause.

513 The default list of operations is as follows:

- 514 • GetInstance
- 515 • Associators
- 516 • AssociatorNames
- 517 • References
- 518 • ReferenceNames
- 519 • EnumerateInstances
- 520 • EnumerateInstanceNames

### 521 8.4 CIM\_BindsTo

522 Table 6 lists implementation requirements for operations. If implemented, these operations shall be  
523 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 6, all operations in  
524 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

526

**Table 6 – Operations: CIM\_BindsTo**

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

## 527 8.5 CIM\_ElementCapabilities

528 Table 7 lists implementation requirements for operations. If implemented, these operations shall be  
 529 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 7, all operations in  
 530 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

532

**Table 7 – Operations: CIM\_ElementCapabilities**

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

## 533 8.6 CIM\_ElementSettingData

534 Table 8 lists implementation requirements for operations. If implemented, these operations shall be  
 535 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 8, all operations in  
 536 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

538

**Table 8 – Operations: CIM\_ElementSettingData**

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

### 539 8.6.1 CIM\_ElementSettingData — ModifyInstance

540 When an instance of CIM\_ElementSettingData associates an instance of CIM\_TelnetSettingData with an  
 541 instance of CIM\_TelnetProtocolEndpoint, the following rules shall govern the behavior of the  
 542 ModifyInstance operation:

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

- 547           – The ModifyInstance operation shall find all other instances of CIM\_ElementSettingData
- 548           that associate a CIM\_TelnetSettingData instance with the CIM\_TelnetProtocolEndpoint
- 549           instance referenced by the target instance of CIM\_ElementSettingData.
- 550           – For each instance of CIM\_ElementSettingData found, the ModifyInstance operation shall
- 551           modify the value of its IsNext property to a value of 2 (Is Not Next).

552   **8.7 CIM\_HostedAccessPoint**

553   Table 9 lists implementation requirements for operations. If implemented, these operations shall be  
 554   implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 9, all operations in  
 555   the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

557                                   **Table 9 – Operations: CIM\_HostedAccessPoint**

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

558   **8.8 CIM\_HostedService**

559   Table 10 lists implementation requirements for operations. If implemented, these operations shall be  
 560   implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 10, all operations  
 561   in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

563                                   **Table 10 – Operations: CIM\_HostedService**

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

564   **8.9 CIM\_ProtocolService**

565   Table 11 lists implementation requirements for operations. If implemented, these operations shall be  
 566   implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 11, all operations  
 567   in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

569

**Table 11 – Operations: CIM\_ProtocolService**

Operation	Requirement	Messages
ModifyInstance	Optional. See 8.9.1.	None

### 570 8.9.1 CIM\_ProtocolService — ModifyInstance

571 When the ElementNameEditSupported property of the CIM\_TelnetCapabilities instance has a value of  
 572 TRUE, the ModifyInstance operation shall allow the value of the ElementName property of the  
 573 CIM\_ProtocolService instance to be modified. The ModifyInstance operation shall enforce the length  
 574 restriction specified in the MaxElementNameLen property of the CIM\_TelnetCapabilities instance.

575 When the ElementNameEditSupported property of the CIM\_TelnetCapabilities has a value of FALSE, the  
 576 ModifyInstance operation shall not change the value of the ElementName property of the  
 577 CIM\_ProtocolService instance.

### 578 8.10 CIM\_ProvidesEndpoint

579 Table 12 lists implementation requirements for operations. If implemented, these operations shall be  
 580 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 12, all operations  
 581 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

583

**Table 12 – Operations: CIM\_ProvidesEndpoint**

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

### 584 8.11 CIM\_ServiceAccessBySAP

585 Table 13 lists implementation requirements for operations. If implemented, these operations shall be  
 586 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 13, all operations  
 587 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

589

**Table 13 – Operations: CIM\_ServiceAccessBySAP**

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

### 590 8.12 CIM\_TelnetCapabilities

591 All operations in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

### 593 8.13 CIM\_TelnetSettingData

594 Table 14 lists implementation requirements for operations. If implemented, these operations shall be  
 595 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 14, all operations  
 596 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

598 **Table 14 – Operations: CIM\_TelnetSettingData**

Operation	Requirement	Messages
ModifyInstance	Optional. See 8.13.1.	None

#### 599 8.13.1 CIM\_TelnetSettingData — ModifyInstance

600 When the CIM\_TelnetSettingData instance is associated with the CIM\_ProtocolService instance through  
 601 an instance of CIM\_ElementSettingData and the value of the IsDefault property of the  
 602 CIM\_ElementSettingData instance has a value of 1 (Is Default), the ModifyInstance operation shall not be  
 603 supported.

604 When the CIM\_TelnetSettingData instance is not associated with an instance of CIM\_ProtocolService  
 605 through an instance of CIM\_ElementSettingData where the IsDefault property has a value of 1 (Is  
 606 Default), the ModifyInstance operation may be supported for the CIM\_TelnetSettingData instance.

### 607 8.14 CIM\_TelnetProtocolEndpoint

608 Table 15 lists implementation requirements for operations. If implemented, these operations shall be  
 609 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 15, all operations  
 610 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

612 **Table 15 – Operations: CIM\_TelnetProtocolEndpoint**

Operation	Requirement	Messages
ModifyInstance	Optional. See 8.14.1.	None
DeletelInstance	Optional. See 8.14.2.	None

#### 613 8.14.1 ModifyInstance

614 The ModifyInstance operation may be supported for an instance of CIM\_TelnetProtocolEndpoint. When  
 615 the ModifyInstance operation is supported for an instance of CIM\_TelnetProtocolEndpoint, the  
 616 ModifyInstance operation shall not modify the following properties:

- 617 • NameFormat
- 618 • ProtocolIFType
- 619 • OtherTypeDescription

#### 620 8.14.2 DeletelInstance

621 The DeletelInstance operation may be supported for instances of CIM\_TelnetProtocolEndpoint. When the  
 622 DeletelInstance operation is invoked against an instance, the corresponding telnet session shall be  
 623 terminated prior to deleting the CIM\_TelnetProtocolEndpoint instance. The implementation shall also  
 624 remove any association instances that reference the CIM\_TelnetProtocolEndpoint.

## 625 8.15 CIM\_TCPProtocolEndpoint

626 Table 16 lists implementation requirements for operations. If implemented, these operations shall be  
 627 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 16, all operations  
 628 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

630 **Table 16 – Operations: CIM\_TCPProtocolEndpoint**

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

### 631 8.15.1 ModifyInstance

632 The ModifyInstance operation may be supported for an instance of CIM\_TCPProtocolEndpoint. When the  
 633 ModifyInstance operation is supported for an instance of CIM\_TCPProtocolEndpoint, the ModifyInstance  
 634 operation shall not modify the following properties:

- 635 • NameFormat
- 636 • ProtocolIFType
- 637 • PortNumber

### 638 8.15.2 DeleteInstance

639 When the CIM\_ProtocolService.ListenOnPort() method is supported for the instance of  
 640 CIM\_ProtocolService with which the CIM\_TCPProtocolEndpoint is associated through an instance of  
 641 CIM\_ServiceAccessBySAP, the DeleteInstance operation shall be supported for the instance of  
 642 CIM\_TCPProtocolEndpoint. When the CIM\_ProtocolService.ListenOnPort() method is not supported, the  
 643 DeleteInstance operation shall not be supported.

644 When the DeleteInstance operation is successful for an instance of CIM\_TCPProtocolEndpoint, the telnet  
 645 service shall stop listening on the TCP/IP port indicated by the PortNumber property of the  
 646 CIM\_TCPProtocolEndpoint. The implementation shall also remove any association instances that  
 647 reference the CIM\_TCPProtocolEndpoint.

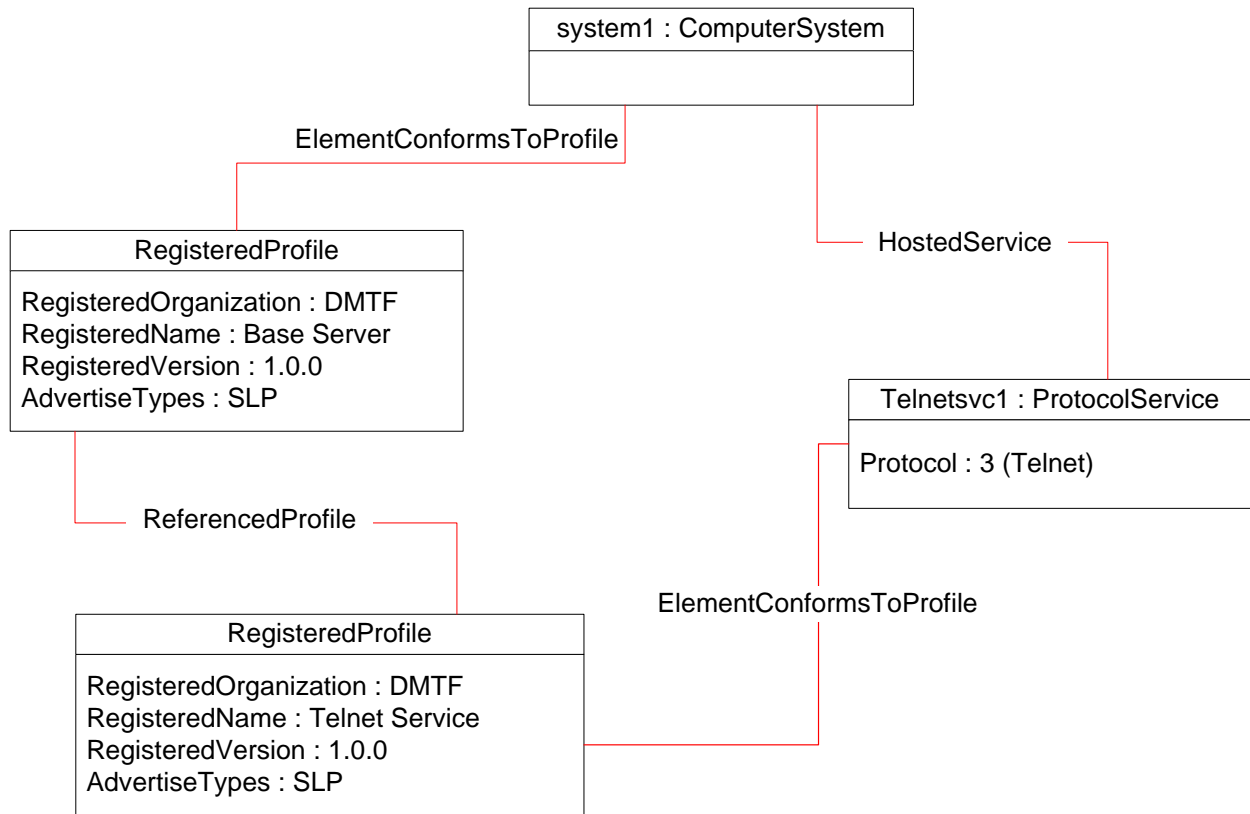
## 648 9 Use Cases

649 This section contains object diagrams and use cases for the *Telnet Service Profile*.

### 650 9.1 Object Diagrams

651 The object diagram in Figure 2 shows how instances of CIM\_RegisteredProfile are used to identify the  
 652 version of the *Telnet Service Profile* with which an instance of CIM\_ProtocolService and its associated  
 653 instances are conformant. An instance of CIM\_RegisteredProfile exists for each profile that is  
 654 instrumented in the system. One instance of CIM\_RegisteredProfile identifies the “DMTF Base Server  
 655 Profile version 1.0.0”. The other instance identifies the “DMTF Telnet Service Profile version 1.0.0”. The  
 656 CIM\_ProtocolService instance is scoped to an instance of CIM\_ComputerSystem. This instance of  
 657 CIM\_ComputerSystem is conformant with the DMTF Base Server Profile version 1.0.0 as indicated by the  
 658 CIM\_ElementConformsToProfile association to the CIM\_RegisteredProfile instance. The  
 659 CIM\_ProtocolService instance is conformant with this profile as indicated by the  
 660 CIM\_ElementConformsToProfile association between the instance and the instance of  
 661 CIM\_RegisteredProfile that identifies this profile.





662

663

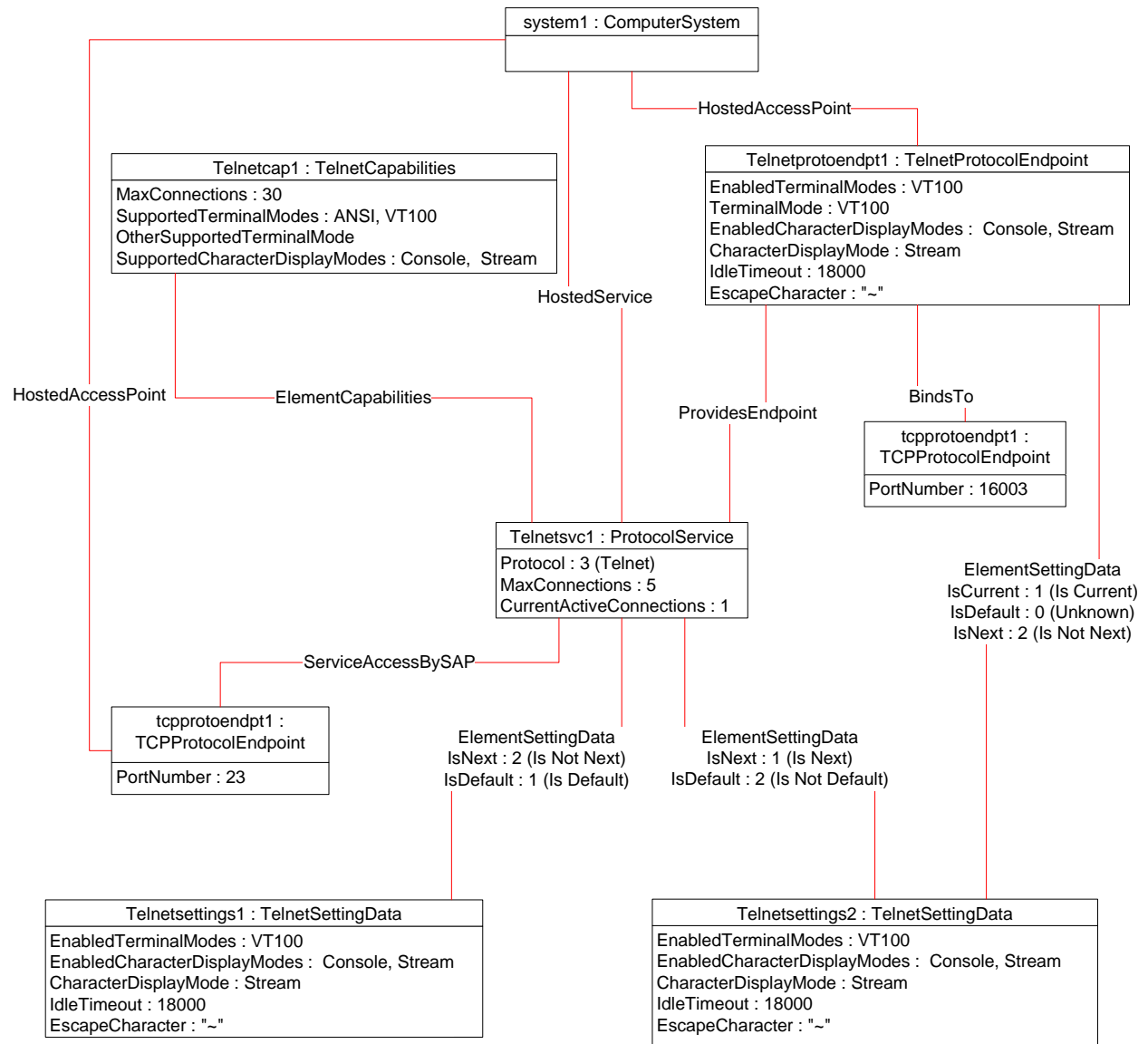
**Figure 2 – Registered Profile**

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

666 Figure 3 is an object diagram that shows the telnet service enabled and listening for incoming  
 667 connections. The instance of CIM\_TelnetSettingData labeled Telnetsettings2 indicates the settings that  
 668 will be applied to a telnet session when it is established. The instance of CIM\_TelnetSettingData labeled  
 669 Telnetsettings1 represents the default configuration for a session. The CIM\_TelnetCapabilities instance  
 670 indicates the capabilities of the telnet service and its associated sessions. In this example, the telnet  
 671 service supports the ANSI and VT100 terminal modes, as indicated by the value of the  
 672 SupportedTerminalModes property. However, the administrator has configured the service to enable  
 673 sessions using only VT100. This configuration is indicated by the value of the EnabledTerminalModes  
 674 property of the associated CIM\_TelnetSettingData instances.



677 The object diagram in Figure 4 represents the same configuration as Figure 3 with the addition of an  
 678 instance of CIM\_TelnetProtocolEndpoint representing a newly established session. Notice that the value  
 679 of the CurrentActiveConnections property of the CIM\_ProtocolService instance (Telnetsvc1) has been  
 680 incremented to reflect that a session is active. The values of the properties for the established session  
 681 (Telnetprotoendpt1) correspond to the values of the instance of CIM\_TelnetSettingData, where the value  
 682 of the IsNext property on the CIM\_ElementSettingData instance that associates the settings with the  
 683 service has a value of 1 (Is Next).

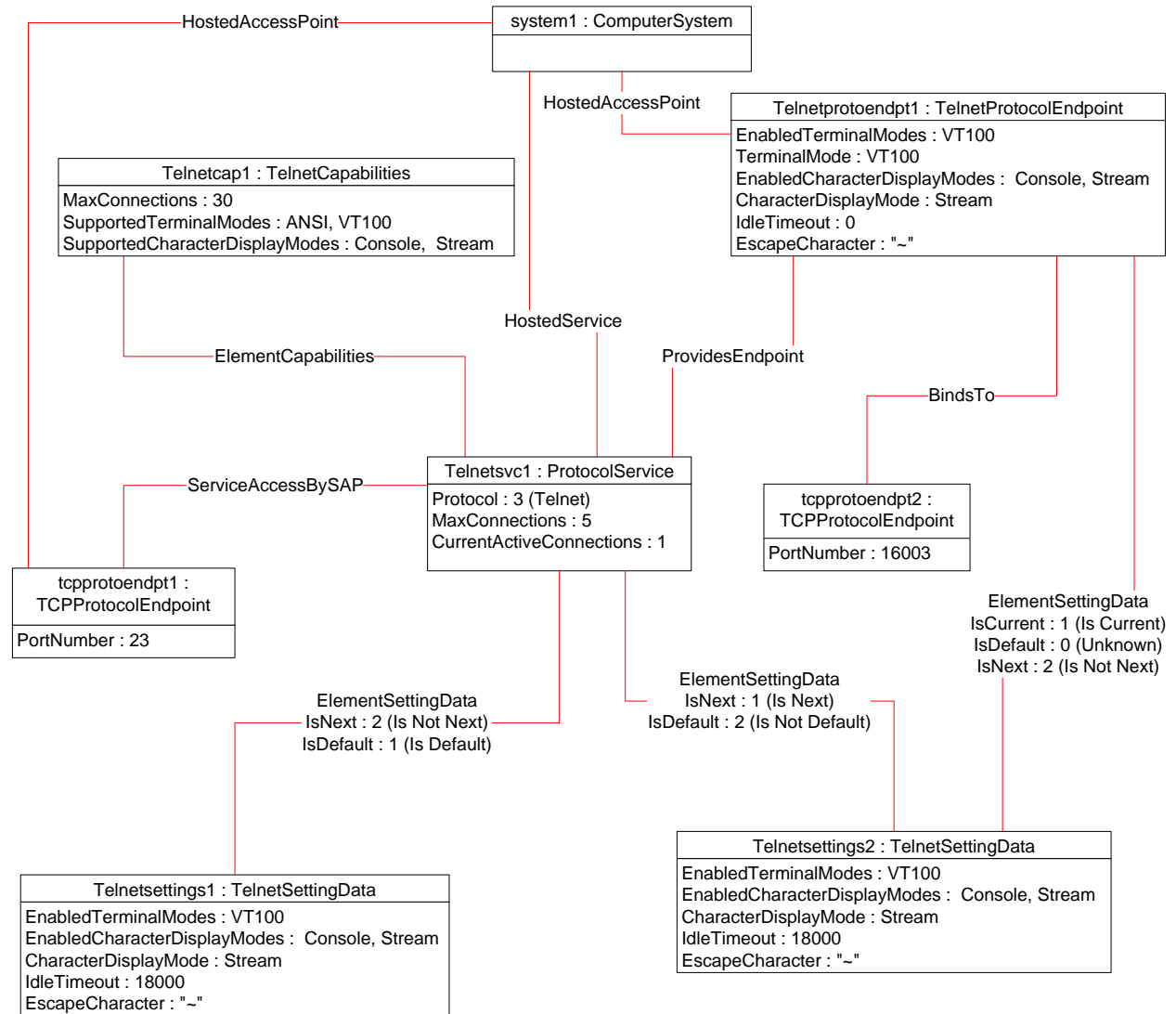


684

685

Figure 4 – One Active Session

686 The object diagram in Figure 5 represents the same configuration as in Figure 4 except that the user has  
 687 changed session parameters from the values that were in effect when the session was initially  
 688 established. The user has changed the character display mode from stream to console. This change is  
 689 reflected in the value of the CharacterDisplayMode property of Telnetprotoendpt1 because the  
 690 CIM\_TelnetProtocolEndpoint contains the actual values for the session. Notice that the value of the  
 691 CharacterDisplayMode property of Telnetsettings2 remains unchanged.



692

693

**Figure 5 – Session Changed**

## 694 9.2 Configuring Session Default Settings

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

- 697 1) Find the instance of CIM\_ElementSettingData that associates an instance of  
698 CIM\_TelnetSettingData with the CIM\_ProtocolService, where the value of its IsNext property is  
699 1 (Is Next) and the value of the IsDefault property is not 1 (Is Default).
- 700 2) Modify the properties of the referenced CIM\_TelnetSettingData instance.

## 701 9.3 Modifying Active Session Settings

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

- 703 1) Find an instance of CIM\_TelnetProtocolEndpoint that is associated with the  
704 CIM\_ProtocolService through an instance of CIM\_ProvidesEndpoint.
- 705 2) Modify the properties of the CIM\_TelnetProtocolEndpoint as needed.

## 706 9.4 Disabling the Telnet Service

707 If an implementation supports disabling the telnet service, a client can disable the telnet service by  
708 invoking the RequestStateChange() method on CIM\_ProtocolService instance with a value of Disabled  
709 for the RequestedState parameter.

## 710 9.5 Determining the Telnet Service Capabilities

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

- 712 1) Find the instance of CIM\_TelnetCapabilities associated with the CIM\_ProtocolService through  
713 an instance of CIM\_ElementCapabilities.
- 714 2) View the properties of the CIM\_TelnetCapabilities instance to see the supported function.

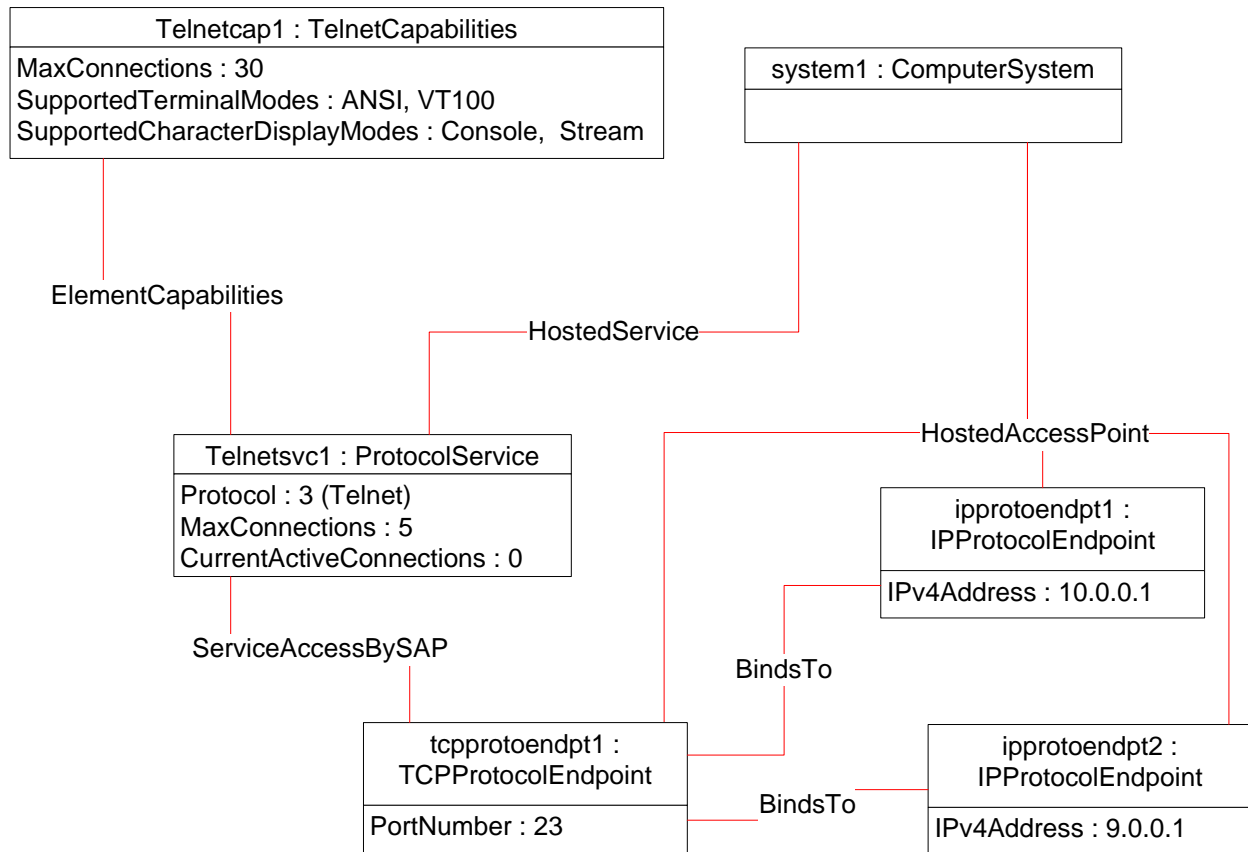
## 715 9.6 Determining the Listening Ports of the Telnet Service

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

- 719 1) Find all instances of CIM\_TCPProtocolEndpoint that are associated with the  
720 CIM\_ProtocolService through an instance of CIM\_ServiceAccessBySAP.
- 721 2) For each instance of CIM\_TCPProtocolEndpoint, query the PortNumber property.

722 Applying this query to Figure 6, the client would find a single instance of CIM\_TCPProtocolEndpoint and  
723 the value of the PortNumber property is 22.

724 Figure 6 is an object diagram for the telnet service listening on TCP/IP port 22 for incoming connection  
725 requests across all of the IP interfaces of the host system. This is illustrated by the instances of  
726 CIM\_BindsTo that associate the instance of CIM\_TCPProtocolEndpoint with the instances of  
727 CIM\_IPProtocolEndpoint.



728

729

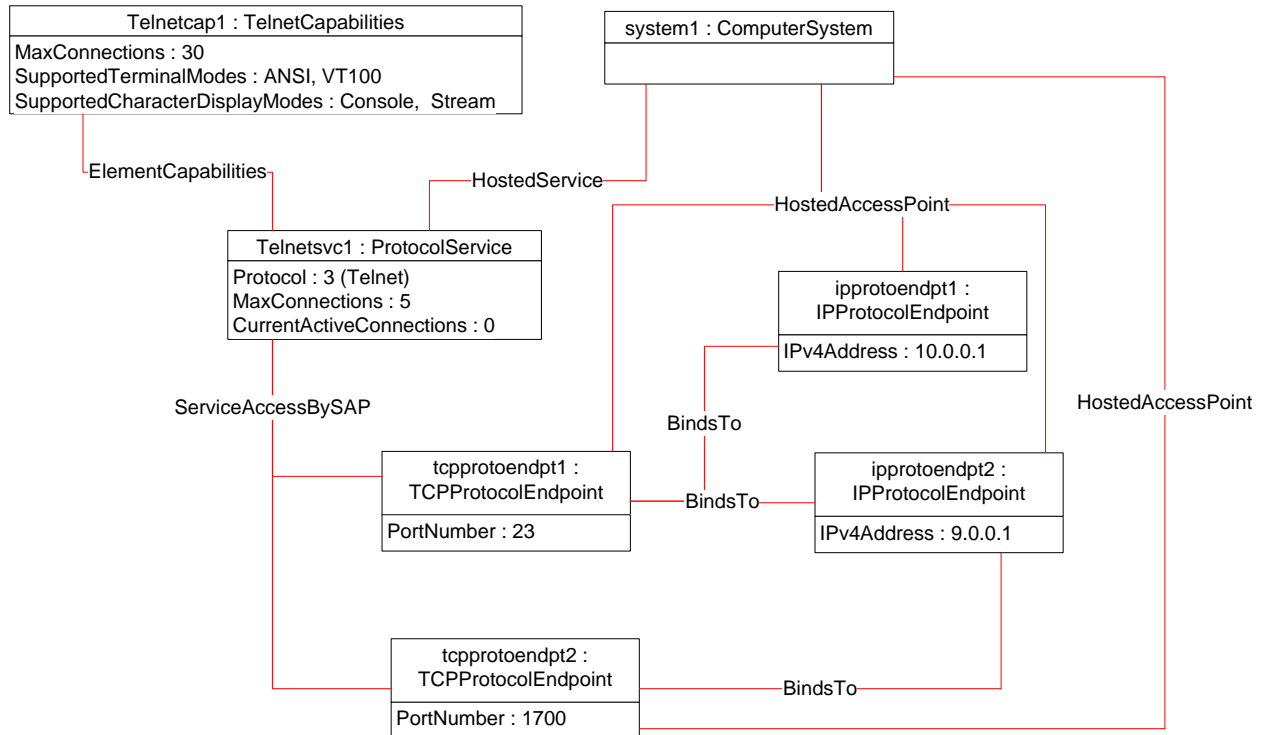
**Figure 6 – Listening on a Single Port on All Interfaces**

## 730 9.7 Adding a Listening Port for the Telnet Service

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

734 To have the telnet service listen on a port across all IP interfaces of the system, the client can invoke the  
 735 `ListenOnPort()` method of the `CIM_ProtocolService` instance, specifying the appropriate `PortNumber`. To  
 736 have the telnet service listen on a port for a specific interface, the client can invoke the `ListenOnPort()`  
 737 method of the `CIM_ProtocolService` instance, specifying a reference to the `CIM_IPProtocolEndpoint`  
 738 instance that represents the specific IP interface.

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



743

744

**Figure 7 – Port Added Bound to Specific Interface**

745 **9.8 Stopping the Telnet Service from Listening on a Specific Port**

746 A client can stop the telnet service from listening on a specific port by invoking the intrinsic  
 747 DeleteInstance operation against the instance of CIM\_TCPProtocolEndpoint that represents the port.

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

750 **9.9 Determining Whether ElementName Can Be Modified**

751 For a given instance of CIM\_ProtocolService, a client can determine whether it can modify the  
 752 ElementName as follows:

- 753 1) Find the CIM\_TelnetCapabilities instance that is associated with the target instance.
- 754 2) Query the value of the ElementNameEditSupported property of the CIM\_TelnetCapabilities  
 755 instance. If the value is TRUE, the client can modify the ElementName property of the target  
 756 instance.

## 757 9.10 Determining Whether State Management Is Supported

758 For a given instance of CIM\_ProtocolService, a client can determine whether state management is  
759 supported as follows:

- 760 1) Find the CIM\_EnabledLogicalElementCapabilities instance that is associated with the  
761 CIM\_LANEndpoint instance.
- 762 2) Query the value of the RequestedStatesSupported property. If at least one value is specified,  
763 state management is supported.

## 764 10 CIM Elements

765 Table 17 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be  
766 implemented as described in Table 17. Sections 7 (“Implementation”) and 8 (“Methods”) may impose  
767 additional requirements on these elements.

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

Element Name	Requirement	Description
<b>Classes</b>		
CIM_BindsTo	Optional	See 10.1 and 10.2.
CIM_ElementCapabilities	Mandatory	See 10.3.
CIM_ElementSettingData	Optional	See 10.4 and 10.5.
CIM_HostedAccessPoint	Mandatory	See 10.6.
CIM_HostedService	Mandatory	See 10.7.
CIM_ProtocolService	Mandatory	See 10.8.
CIM_ProvidesEndpoint	Mandatory	See 10.9.
CIM_RegisteredProfile	Mandatory	See 10.10.
CIM_ServiceAccessBySAP	Conditional	See 0.
CIM_TelnetCapabilities	Mandatory	See 10.12.
CIM_TelnetProtocolEndpoint	Mandatory	See 10.13.
CIM_TelnetSettingData	Optional	See 10.14.
CIM_TCPProtocolEndpoint	Optional	See 10.15.
<b>Indications</b>		
None defined in this profile		



769 **10.1 CIM\_BindsTo — TCPProtocolEndpoint**

770 When an instance of CIM\_TCPProtocolEndpoint is instrumented, CIM\_BindsTo is used to associate the  
 771 CIM\_TelnetProtocolEndpoint instance with the CIM\_TCPProtocolEndpoint instance on which it depends.  
 772 Table 18 provides information about the properties of CIM\_BindsTo.

773 **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 *

774 **10.2 CIM\_BindsTo — IPProtocolEndpoint**

775 When the relationship with an underlying IP interface is modeled according to 7.3, CIM\_BindsTo is used  
 776 to associate the CIM\_TCPProtocolEndpoint instance with the CIM\_IPProtocolEndpoint instance on which  
 777 it depends. Table 19 provides information about the properties of CIM\_BindsTo.

778 **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..*

779 **10.3 CIM\_ElementCapabilities**

780 CIM\_ElementCapabilities is used to associate an instance of CIM\_TelnetCapabilities with the  
 781 CIM\_ProtocolService. Table 20 provides information about the properties of CIM\_ElementCapabilities.

782 **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

783 **10.4 CIM\_ElementSettingData — Telnet Service**

784 CIM\_ElementSettingData is used to associate instances of CIM\_TelnetSettingData with instances of  
 785 CIM\_ProtocolService. Table 21 provides information about the properties of CIM\_ElementSettingData.

786 **Table 21 – Class: CIM\_ElementSettingData (Telnet Service)**

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

787 **10.5 CIM\_ElementSettingData — m ,Telnet Session**

788 CIM\_ElementSettingData is used to associate instances of CIM\_TelnetSettingData with instances of  
 789 CIM\_TelnetProtocolEndpoint. Table 22 provides information about the properties of  
 790 CIM\_ElementSettingData.

791 **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 *
SettingData	Mandatory	This property shall be a reference to an instance of CIM_TelnetSettingData. Cardinality *
IsCurrent	Mandatory	Matches 1 (Is Current) or 2 (Is Not Current)

792 **10.6 CIM\_HostedAccessPoint**

793 CIM\_HostedAccessPoint is used to associate the CIM\_TelnetProtocolEndpoint and  
 794 CIM\_TCPEndpoint instances to their scoping CIM\_ComputerSystem instance. Table 23 provides  
 795 information about the properties of CIM\_HostedAccessPoint.

796 **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_TCPEndpoint. Cardinality *

797 **10.7 CIM\_HostedService**

798 CIM\_HostedService is used to associate the CIM\_ProtocolService to its scoping CIM\_ComputerSystem  
 799 instance. Table 24 provides information about the properties of CIM\_HostedService.

800 **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..*

801 **10.8 CIM\_ProtocolService**

802 CIM\_ProtocolService represents the telnet service. Table 25 provides information about the properties of  
 803 CIM\_ProtocolService.

804 **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 7.1.2.2 and 7.1.1.3.
EnabledState	Mandatory	See 7.1.1.4 and 7.1.2.3.
HealthState	Mandatory	None
OperationalStatus	Mandatory	None
ElementName	Mandatory	See 7.1.3 and 7.1.4.
RequestStateChange( )	Mandatory	See 8.2.
ListenOnPort( )	Mandatory	See 8.1.

## 805 10.9 CIM\_ProvidesEndpoint

806 CIM\_ProvidesEndpoint is used to associate the instance of CIM\_ProtocolService with an instance of  
 807 CIM\_TelnetProtocolEndpoint that represents a session with the service. Table 26 provides information  
 808 about the properties of CIM\_ProvidesEndpoint.

809 **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 *

## 810 10.10 CIM\_RegisteredProfile

811 CIM\_RegisteredProfile identifies the *Telnet Service Profile*. The CIM\_RegisteredProfile class is defined by  
 812 the [Profile Registration Profile](#). With the exception of the mandatory values specified for the properties in  
 813 Table 27, the behavior of the CIM\_RegisteredProfile instance is in accordance with the constraints  
 814 specified in the [Profile Registration Profile](#).

815 **Table 27 – Class: CIM\_RegisteredProfile**

Properties	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Telnet Service".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of "DMTF".

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

## 819 10.11 CIM\_ServiceAccessBySAP

820 CIM\_ServiceAccessBySAP is used to associate the instance of CIM\_ProtocolService with an instance of  
 821 CIM\_TCPPProtocolEndpoint over which a session with the service can be established. Table 28 provides  
 822 information about the properties of CIM\_ServiceAccessBySAP.

823 **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 *

824 **10.12 CIM\_TelnetCapabilities**

825 CIM\_TelnetCapabilities represents the capabilities of a telnet service. Table 29 provides information  
 826 about the properties of CIM\_TelnetCapabilities.

827 **Table 29 – Class: CIM\_TelnetCapabilities**

Properties	Requirement	Notes
InstanceID	Mandatory	None
ElementName	Mandatory	None
RequestedStatesSupported	Mandatory	See 7.1.1.2.1 and 7.1.2.1.1.
ElementNameEditSupported	Mandatory	See 7.1.3.1.1 and 7.1.4.1.1.
MaxElementNameLen	Conditional	See 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
OtherSupportedCharacterDisplayMode	Conditional	This property shall have a value when the SupportedCharacterDisplayModes property has a value of 1 ("Other").

828 **10.13 CIM\_TelnetProtocolEndpoint**

829 CIM\_TelnetProtocolEndpoint represents a session that is established with the telnet service. Table 30  
 830 provides information about the properties of CIM\_TelnetProtocolEndpoint.

831 **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 8.2.
EnabledTerminalModes	Mandatory	None
OtherEnabledTerminalMode	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

## 832 10.14 CIM\_TelnetSettingData

833 CIM\_TelnetSettingData represents settings that can be applied to a telnet session. Table 31 provides  
834 information about the properties of CIM\_TelnetSettingData.

835 **Table 31 – Class: CIM\_TelnetSettingData**

Properties	Requirement	Notes
InstanceID	Mandatory	None
ElementName	Mandatory	Pattern ".*"
EnabledTerminalModes	Mandatory	None
OtherEnabledTerminalMode	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

836 **10.15 CIM\_TCPProtocolEndpoint**

837 CIM\_TCPProtocolEndpoint represents an IP port to which a telnet session or service can be bound.

838 Table 32 provides information about the properties of CIM\_TCPProtocolEndpoint.

839 **Table 32 – Class: CIM\_TCPProtocolEndpoint**

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

840

841  
842  
843  
844

## **ANNEX A (Informative)**

### **Change Log**

<b>Version</b>	<b>Date</b>	<b>Description</b>
1.0.0a	06/16/2006	Released as Preliminary Standard
1.0.0	6/22/2009	DMTF Standard Release

845  
846