



1

2

3

4

Document Number: DSP1036

Date: 2012-02-23

Version: 1.0.2

5 **IP Interface Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: en-US**

9

10 Copyright Notice

11 Copyright © 2008, 2012 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

32

CONTENTS

| | | |
|----|---|----|
| 34 | Foreword | 6 |
| 35 | Introduction..... | 7 |
| 36 | Document conventions..... | 7 |
| 37 | 1 Scope | 9 |
| 38 | 2 Normative references..... | 9 |
| 39 | 3 Terms and definitions..... | 10 |
| 40 | 4 Symbols and abbreviated terms..... | 11 |
| 41 | 5 Synopsis..... | 11 |
| 42 | 6 Description | 12 |
| 43 | 6.1 Pending and alternate configuration management..... | 13 |
| 44 | 7 Implementation..... | 13 |
| 45 | 7.1 Basic IP configuration | 14 |
| 46 | 7.2 DHCP client is supported..... | 18 |
| 47 | 7.3 DNS client is supported | 18 |
| 48 | 7.4 Managing alternate configurations — Optional | 18 |
| 49 | 7.5 Applying an alternate configuration | 21 |
| 50 | 7.6 Relationship with a network interface | 23 |
| 51 | 8 Methods..... | 23 |
| 52 | 8.1 CIM_IPProtocolEndpoint.RequestStateChange() | 24 |
| 53 | 8.2 CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint() | 24 |
| 54 | 8.3 Profile conventions for operations | 25 |
| 55 | 8.4 CIM_BindsToLANEndpoint..... | 26 |
| 56 | 8.5 CIM_ElementSettingData | 26 |
| 57 | 8.6 CIM_HostedAccessPoint..... | 27 |
| 58 | 8.7 CIM_HostedService..... | 27 |
| 59 | 8.8 CIM_IPAssignmentSettingData | 28 |
| 60 | 8.9 CIM_IPConfigurationService | 28 |
| 61 | 8.10 CIM_IPProtocolEndpoint | 28 |
| 62 | 8.11 CIM_OrderedComponent | 29 |
| 63 | 8.12 CIM_RemoteAccessAvailableToElement..... | 29 |
| 64 | 8.13 CIM_RemoteServiceAccessPoint..... | 29 |
| 65 | 8.14 CIM_ServiceAffectsElement..... | 30 |
| 66 | 8.15 CIM_StaticIPAssignmentSettingData | 30 |
| 67 | 9 Use cases..... | 31 |
| 68 | 9.1 Miscellaneous object diagrams..... | 31 |
| 69 | 9.2 Determine supported configuration methods..... | 44 |
| 70 | 9.3 Determine gateway address | 44 |
| 71 | 9.4 Determine method used for current configuration | 44 |
| 72 | 9.5 Determine whether DHCP then static is supported | 45 |
| 73 | 9.6 View default configuration..... | 45 |
| 74 | 9.7 Configure the interface to use DHCP | 45 |
| 75 | 9.8 Establish a static IP configuration for an interface..... | 46 |
| 76 | 9.9 Apply a pending configuration — Synchronously | 46 |
| 77 | 9.10 Apply a pending configuration — Upon restart..... | 46 |
| 78 | 9.11 Determine whether DNS configuration was DHCP assigned | 47 |
| 79 | 9.12 Determine whether ElementName can be modified | 47 |
| 80 | 9.13 Determine whether state management is supported | 47 |
| 81 | 10 CIM Elements..... | 47 |
| 82 | 10.1 CIM_BindsToLANEndpoint..... | 48 |
| 83 | 10.2 CIM_ElementCapabilities | 48 |
| 84 | 10.3 CIM_ElementSettingData — CIM_IPAssignmentSettingData Reference | 49 |
| 85 | 10.4 CIM_ElementSettingData — CIM_StaticIPAssignmentSettingData Reference | 49 |

| | | | |
|----|-------|--|----|
| 86 | 10.5 | CIM_EnabledLogicalElementCapabilities | 49 |
| 87 | 10.6 | CIM_HostedAccessPoint — CIM_RemoteServiceAccessPoint Reference | 50 |
| 88 | 10.7 | CIM_HostedAccessPoint — CIM_IPProtocolEndpoint Reference | 50 |
| 89 | 10.8 | CIM_HostedService | 50 |
| 90 | 10.9 | CIM_IPAssignmentSettingData | 51 |
| 91 | 10.10 | CIM_IPConfigurationService | 51 |
| 92 | 10.11 | CIM_IPProtocolEndpoint | 51 |
| 93 | 10.12 | CIM_OrderedComponent | 52 |
| 94 | 10.13 | CIM_RegisteredProfile | 52 |
| 95 | 10.14 | CIM_RemoteAccessAvailableToElement | 53 |
| 96 | 10.15 | CIM_RemoteServiceAccessPoint | 53 |
| 97 | 10.16 | CIM_ServiceAffectsElement | 53 |
| 98 | 10.17 | CIM_StaticIPAssignmentSettingData | 54 |

99

100 Figures

| | | |
|-----|--|----|
| 101 | Figure 1 – IP Interface Profile: Class diagram | 12 |
| 102 | Figure 2 – Registered profile | 31 |
| 103 | Figure 3 – Basic configuration — IPv4 | 32 |
| 104 | Figure 4 – Basic configuration — IPv6 | 33 |
| 105 | Figure 5 – Basic configuration — IPv4 and IPv6 | 34 |
| 106 | Figure 6 – Static current and pending configuration | 35 |
| 107 | Figure 7 – Static and DHCP pending configurations | 36 |
| 108 | Figure 8 – DHCP timed out to a static configuration | 37 |
| 109 | Figure 9 – Service processor and server share an NIC | 38 |
| 110 | Figure 10 – Configuration choices | 39 |
| 111 | Figure 11 – DHCP assigned partial DNS | 40 |
| 112 | Figure 12 – DHCP with DNS statically configured | 41 |
| 113 | Figure 13 – Static without DNS configuration — One | 42 |
| 114 | Figure 14 – Static without DNS configuration — Two | 43 |
| 115 | Figure 15 – Static without DNS configuration — Three | 44 |

116

117 Tables

| | | |
|-----|---|----|
| 118 | Table 1 – Referenced profiles | 12 |
| 119 | Table 2 – CIM_IPProtocolEndpoint.RequestStateChange() Method: Return code values | 24 |
| 120 | Table 3 – CIM_IPProtocolEndpoint.RequestStateChange() Method: Parameters | 24 |
| 121 | Table 4 – CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint() Method: Return code | |
| 122 | values | 25 |
| 123 | Table 5 – CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint() Method: Parameters | 25 |
| 124 | Table 6 – Operations: CIM_BindsToLANEndpoint | 26 |
| 125 | Table 7 – Operations: CIM_ElementSettingData | 26 |
| 126 | Table 8 – Operations: CIM_HostedAccessPoint | 27 |
| 127 | Table 9 – Operations: CIM_HostedService | 27 |
| 128 | Table 10 – Operations: CIM_IPProtocolEndpoint | 28 |
| 129 | Table 11 – Operations: CIM_OrderedComponent | 29 |
| 130 | Table 12 – Operations: CIM_RemoteAccessAvailableToElement | 29 |
| 131 | Table 13 – Operations: CIM_ServiceAffectsElement | 30 |

132 Table 14 – Operations: CIM_StaticIPAssignmentSettingData..... 30

133 Table 15 – CIM Elements: IP Interface Profile..... 47

134 Table 16 – Class: CIM_BindsToLANEndpoint..... 48

135 Table 17 – Class: CIM_ElementCapabilities..... 48

136 Table 18 – Class: CIM_ElementSettingData — CIM_IPAssignmentSettingData..... 49

137 Table 19 – Class: CIM_ElementSettingData — CIM_StaticIPAssignmentSettingData..... 49

138 Table 20 – Class: CIM_EnabledLogicalElementCapabilities..... 49

139 Table 21 – Class: CIM_HostedAccessPoint — CIM_RemoteServiceAccessPoint..... 50

140 Table 22 – Class: CIM_HostedAccessPoint — CIM_IPProtocolEndpoint..... 50

141 Table 23 – Class: CIM_HostedService..... 50

142 Table 24 – Class: CIM_IPAssignmentSettingData..... 51

143 Table 25 – Class: CIM_IPConfigurationService..... 51

144 Table 26 – Class: CIM_IPProtocolEndpoint..... 51

145 Table 27 – Class: CIM_OrderedComponent..... 52

146 Table 28 – Class: CIM_RegisteredProfile..... 52

147 Table 29 – Class: CIM_RemoteAccessAvailableToElement..... 53

148 Table 30 – Class: CIM_RemoteServiceAccessPoint..... 53

149 Table 31 – Class: CIM_ServiceAffectsElement..... 53

150 Table 32 – Class: CIM_StaticIPAssignmentSettingData..... 54

151

152

Foreword

153 The *IP Interface Profile* (DSP1036) was prepared by the Server Management Working Group, the
154 Physical Platform Profiles Working Group and the Server Desktop Mobile Platforms Working Group of the
155 DMTF.

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

158 Acknowledgments

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

160 Editors:

- 161 • Jeff Hilland – Hewlett-Packard Company
- 162 • Aaron Merkin – IBM
- 163 • Satheesh Thomas – AMI

164 Contributors:

- 165 • RadhaKrishna Dasari – Dell
- 166 • Jon Hass – Dell
- 167 • John Leung – Intel
- 168 • Aaron Merkin – IBM
- 169 • Khachatur Papanyan – Dell
- 170 • Christina Shaw – Hewlett-Packard Company
- 171 • Enoch Suen – Dell
- 172 • Perry Vincent – Intel

173

174

Introduction

175 The information in this specification should be sufficient for a provider or consumer of this data to identify
176 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to
177 represent and manage an IP interface and its associated configuration information. The target audience
178 for this specification is implementers who are writing CIM-based providers or consumers of management
179 interfaces that represent the component described in this document.

180 Document conventions

181 Typographical conventions

182 The following typographical conventions are used in this document:

- 183 • Document titles are marked in *italics*.

184 Experimental material

185 Experimental material has yet to receive sufficient review to satisfy the adoption requirements set forth by
186 the DMTF. Experimental material is included in this document as an aid to implementers who are
187 interested in likely future developments. Experimental material may change as implementation
188 experience is gained. It is likely that experimental material will be included in an upcoming revision of the
189 document. Until that time, experimental material is purely informational.

190 The following typographical convention indicates experimental material:

191 **EXPERIMENTAL**

192 Experimental material appears here.

193 **EXPERIMENTAL**

194 In places where this typographical convention cannot be used (for example, tables or figures), the
195 "EXPERIMENTAL" label is used alone

197

IP Interface Profile

198 1 Scope

199 The *IP Interface Profile* extends the management capability of referencing profiles by adding the
200 capability to represent an IP interface of a managed system. This profile includes a specification of the IP
201 interface, its associated IP configuration, optional support for managing pending configurations, optional
202 support for the relationship with a DNS client, and optional support for the relationship with a DHCP client.

203 2 Normative references

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

208 DMTF DSP0004, *CIM Infrastructure Specification 2.6*,
209 http://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf

210 DMTF DSP0200, *CIM Operations over HTTP 1.3*,
211 http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

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

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

216 DMTF DSP1004, *Base Server Profile 1.0*,
217 http://www.dmtf.org/standards/published_documents/DSP1004_1.0.pdf

218 DMTF DSP1014, *Ethernet Port Profile 1.0*,
219 http://www.dmtf.org/standards/published_documents/DSP1014_1.0.pdf

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

222 DMTF DSP1035, *Host LAN Network Port Profile 1.0*,
223 http://www.dmtf.org/standards/published_documents/DSP1035_1.0.pdf

224 DMTF DSP1037, *DHCP Client Profile 1.0*,
225 http://www.dmtf.org/standards/published_documents/DSP1037_1.0.pdf

226 DMTF DSP1038, *DNS Client Profile 1.0*,
227 http://www.dmtf.org/standards/published_documents/DSP1038_1.0.pdf

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

229 IETF, RFC 2131, *Dynamic Host Configuration Protocol*, March 1997, <http://www.ietf.org/rfc/rfc2131.txt>

230 IETF, RFC 4291, *IP Version 6 Addressing Architecture*, February 2006, <http://www.ietf.org/rfc/rfc4291.txt>

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

233 3 Terms and definitions

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

236 The terms "shall" ("required"), "shall not," "should" ("recommended"), "should not" ("not recommended"),
237 "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
238 in [ISO/IEC Directives, Part 2](#), Annex H. The terms in parenthesis are alternatives for the preceding term,
239 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
240 [ISO/IEC Directives, Part 2](#), Annex H specifies additional alternatives. Occurrences of such additional
241 alternatives shall be interpreted in their normal English meaning.

242 The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as
243 described in [ISO/IEC Directives, Part 2](#), Clause 5.

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

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

249 3.1

250 **can**

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

252 3.2

253 **cannot**

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

255 3.3

256 **conditional**

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

259 3.4

260 **mandatory**

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

263 3.5

264 **may**

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

266 3.6

267 **need not**

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

269 3.7

270 **optional**

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

272 3.8

273 **pending configuration**

274 the configuration that will be applied to an IP interface the next time the interface accepts a configuration

275 3.9

276 **referencing profile**

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

279 **3.10**280 **shall**

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

283 **3.11**284 **shall not**

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

287 **3.12**288 **should**

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

291 **3.13**292 **should not**

293 indicates that a certain possibility or course of action is deprecated but not prohibited

294 **3.14**295 **unspecified**

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

297 **4 Symbols and abbreviated terms**

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

300 **4.1**301 **DHCP**

302 Dynamic Host Configuration Protocol

303 **4.2**304 **DNS**

305 Domain Name System

306 **4.3**307 **IP**

308 Internet Protocol

309 **5 Synopsis**

310 **Profile name:** IP Interface

311 **Version:** 1.0.2

312 **Organization:** DMTF

313 **CIM Schema version:** 2.19

314 **Central class:** CIM_IPProtocolEndpoint

315 **Scoping class:** CIM_ComputerSystem

316 The *IP Interface Profile* extends the management capability of referencing profiles by adding the
 317 capability to represent an IP interface of a managed system. This profile includes a specification of the IP
 318 interface, its associated IP configuration, optional support for managing pending configurations, optional
 319 support for the relationship with a DNS client, and optional support for the relationship with a DHCP client.

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

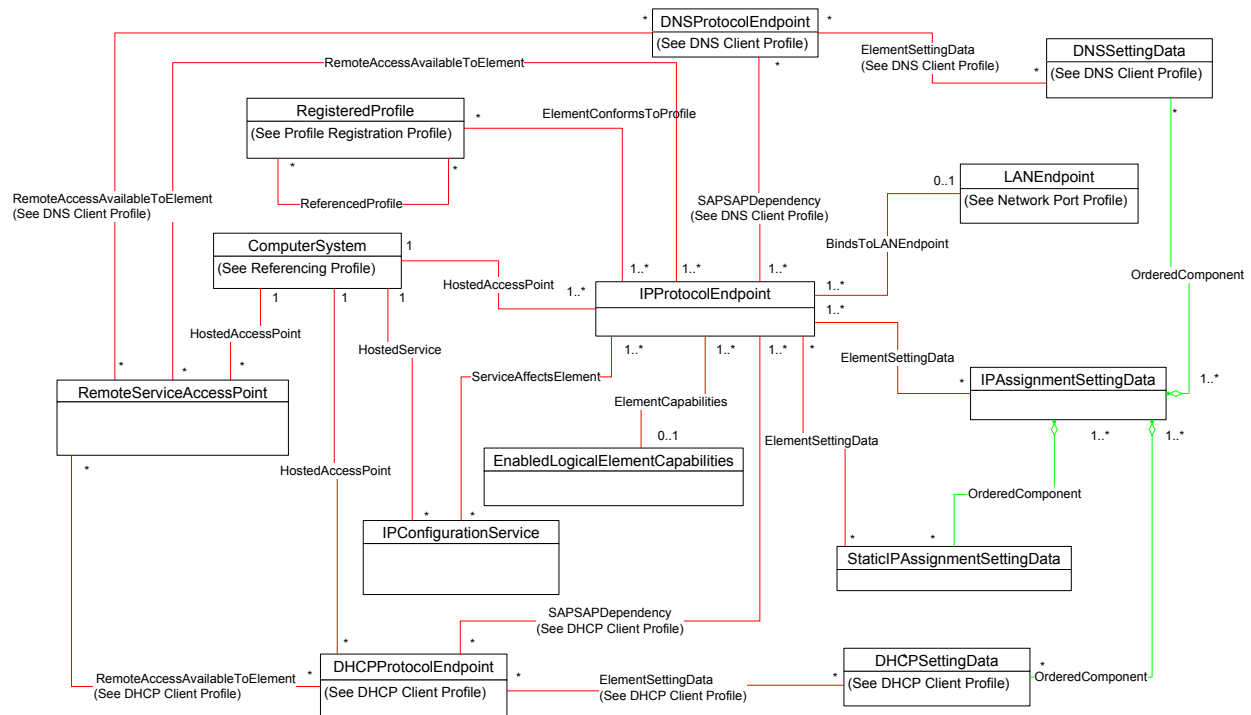
321 **Table 1 – Referenced profiles**

| Profile Name | Organization | Version | Requirement | Description |
|-----------------------|--------------|---------|-------------|-------------|
| Profile Registration | DMTF | 1.0 | Mandatory | None |
| DNS Client | DMTF | 1.0 | Optional | See 7.3. |
| DHCP Client | DMTF | 1.0 | Optional | See 7.2. |
| Host LAN Network Port | DMTF | 1.0 | Optional | See 7.6. |

322 6 Description

323 The *IP Interface Profile* describes an IP interface and associated IP configuration information in a
 324 managed system.

325 Figure 1 represents the class schema for the *IP Interface Profile*. For simplicity, the CIM_ prefix has been
 326 removed from the names of the classes. Note that this class diagram is meant to be used in conjunction
 327 with the class diagrams from the *DHCP Client Profile* ([DSP1037](#)) and the *DNS Client Profile* ([DSP1038](#)).



328

329 **Figure 1 – IP Interface Profile: Class diagram**

330 The *IP Interface Profile* extends the management capability of referencing profiles by adding the
331 capability to represent an IP interface in a managed system. Functionality within the scope of this profile
332 includes:

- 333 • IPv4 interface (optionally associated with a network interface)
- 334 • optional relationship with a DNS client
- 335 • optional relationship with a DHCP client
- 336 • current and pending configurations

337 Functionality explicitly excluded from the scope of this profile includes:

- 338 • modeling of the network gateway
- 339 • modeling of TCP/UDP ports

340 Any representation of network elements is purely from the perspective of the IP interface. That is, no
341 provisions are made for the modeling of network resources for the purposes of managing those
342 resources.

343 This profile represents the current configuration of an IP interface, associated configurations that could be
344 applied, the DNS client, and the DHCP client. Support for the DNS and DHCP clients is not required. In
345 general, the various subclasses of `CIM_ProtocolEndpoint` reflect the current configuration and status of
346 their respective elements.

347 Functionality provided by other systems (Gateway, DHCP server, and DNS server) is modeled from the
348 IP interface view and is therefore represented by instances of `CIM_RemoteServiceAccessPoint`.

349 6.1 Pending and alternate configuration management

350 Pending configurations, which are associated with the IP interface and could be applied in the future, are
351 represented by instances of `CIM_IPAssignmentSettingData` and its subclasses. Each pending
352 configuration can include multiple settings that will be applied to the different elements of the endpoint
353 configuration. Settings for a particular element of the configuration are represented with the appropriate
354 subclass of `CIM_IPAssignmentSettingData` and aggregated into one or more instances of
355 `CIM_IPAssignmentSettingData` that represent the configuration.

356 The management of DNS and DHCP clients as part of an alternate configuration is handled differently for
357 the two clients. DHCP and static IP configuration management are generally treated as alternatives to
358 each other. For the basic configuration of an IP interface, the information is assigned either statically or
359 through DHCP. DNS configuration occurs differently. When DNS and static configuration occur together,
360 there is no overlap. Thus the DNS settings that are part of the configuration are applied to the DNS client.
361 When DHCP and DNS settings are used together, portions of the DNS configuration can potentially be
362 assigned through DHCP.

363 The intended usage model for alternate configurations is that an implementation presents a finite set of
364 alternate configurations. It is expected that an alternate configuration will be instrumented for each unique
365 ordering of static and DHCP assignment supported by the implementation. An alternate configuration can
366 also be provided for each unique configuration persisted (either in the instrumentation layer or underlying
367 modeled component). DNS configuration is presented as an optional aspect of each unique alternate
368 configuration with which DNS usage is supported.

369 7 Implementation

370 This clause details the requirements related to the arrangement of instances and properties of instances
371 for implementations of this profile.

372 7.1 Basic IP configuration

373 The basic configuration of the IP interface consists of the IP address, subnet mask, and default gateway.

374 7.1.1 CIM_IPProtocolEndpoint

375 An instance of CIM_IPProtocolEndpoint shall represent the IP interface. The properties of the instance of
376 CIM_IPProtocolEndpoint shall reflect the current configuration of an IP interface.

377 7.1.1.1 CIM_IPProtocolEndpoint.AddressOrigin

378 The AddressOrigin property indicates the configuration method that resulted in the configuration being
379 assigned to the CIM_IPProtocolEndpoint.

380 7.1.1.1.1 AddressOrigin — Static

381 A value of 3 (Static) shall indicate that the configuration was assigned statically. The AddressOrigin
382 property shall have a value of 3 (Static) when the configuration is the result of an instance of
383 CIM_StaticIPAssignmentSettingData being successfully applied. Clause 7.5.3.3 explains what it means
384 for settings to be successfully applied.

385 7.1.1.1.2 AddressOrigin — DHCP

386 A value of 4 (DHCP) shall indicate that the configuration was obtained through an associated DHCP
387 client. The AddressOrigin property shall have a value of 4 (DHCP) when the configuration is the result of
388 an instance of CIM_DHCPSettingData being successfully applied.

389 7.1.1.2 CIM_IPProtocolEndpoint.ProtocolIFType

390 The ProtocolIFType property shall indicate the current IP address type.

391 If the value is 4096 (IPv4) the IPv4Address and SubnetMask properties shall be implemented.

392 The value of CIM_IPProtocolEndpoint.ProtocolIFType shall be 4096,

393 EXPERIMENTAL

394 4097, or 4098.

395 If the value is 4097 (Ipv6) the IPv6Address, IPv6AddressType, and IPv6SubnetPrefixLength properties
396 shall be implemented.

397 If the value is 4098 (Ipv4/Ipv6) the IPv6Address, IPv6AddressType, and IPv6SubnetPrefixLength
398 properties shall be implemented and the IPv6AddressType shall be 7 (Embedded IPv4 Address).

399 EXPERIMENTAL

400 7.1.1.3 CIM_IPProtocolEndpoint.IPv4Address

401 If the value of CIM_IPProtocolEndpoint.ProtocolIFType is 4096 (IPv4), the IPv4Address property shall
402 indicate the current IPv4 address assigned to this IP endpoint. The value of the property shall be
403 specified in dotted decimal notation as defined in IETF [RFC 1208](#). A value of 0.0.0.0 shall indicate that a
404 valid IP address is not assigned to this IP endpoint.

405 7.1.1.4 CIM_IPProtocolEndpoint.SubnetMask

406 If the value of CIM_IPProtocolEndpoint.ProtocolIFType is 4096 (IPv4), the SubnetMask property shall be
407 specified using dotted decimal notation as defined in IETF [RFC 1208](#). A value of 0.0.0.0 shall indicate
408 that a valid subnet mask is not assigned to this IP endpoint.

409 EXPERIMENTAL**410 7.1.1.5 CIM_IPProtocolEndpoint.Ipv6Address**

411 If the value of CIM_IPProtocolEndpoint.ProtocolIFTType is 4097 (IPv6) or 4098 (IPv4/IPv6), the
412 Ipv6Address property shall indicate the current IPv6 address assigned to this IP endpoint. The value of
413 the property shall be specified in the notation specified in IETF [RFC 4291](#), section 2.2.

414 EXPERIMENTAL

415 7.1.2 IP interface state management is supported — conditional

416 When management of the state of an IP interface is supported, exactly one instance of
417 CIM_EnabledLogicalElementCapabilities shall be associated with the CIM_IPProtocolEndpoint instance
418 through an instance of CIM_ElementCapabilities. The existence of the CIM_ElementCapabilities instance
419 is conditional on the existence of the CIM_EnabledLogicalElementCapabilities instance.

420 Support for managing the state of the IP interface is optional behavior. This clause describes the CIM
421 elements and behaviors that shall be implemented when this behavior is supported.

422 7.1.2.1 CIM_EnabledLogicalElementCapabilities

423 The instance of CIM_EnabledLogicalElementCapabilities is used to advertise the state management
424 supported for the IP interface.

425 7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

426 The RequestedStatesSupported property may contain zero or more of the following values: 2 (Enabled),
427 3 (Disabled), or 11 (Reset).

428 7.1.2.2 CIM_IPProtocolEndpoint.RequestedState

429 When the CIM_IPProtocolEndpoint.RequestStateChange() method is successfully invoked, the value of
430 the RequestedState property shall be the value of the RequestedState parameter. If the method is not
431 successfully invoked, the value of the RequestedState property is indeterminate.

432 The CIM_IPProtocolEndpoint.RequestedState property shall have one of the values specified in the
433 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property or a value of 5 (No
434 Change).

435 7.1.2.3 CIM_IPProtocolEndpoint.EnabledState

436 When the RequestedState parameter has a value of 2 (Enabled) or 3 (Disabled) and the
437 CIM_IPProtocolEndpoint.RequestStateChange() method completes successfully, the value of the
438 EnabledState property shall equal the value of the CIM_IPProtocolEndpoint.RequestedState property.

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

440 The EnabledState property shall have one of the following values: 2 (Enabled), 3 (Disabled), or 6
441 (Enabled but Offline).

442 7.1.3 IP interface state management is not supported

443 This clause describes the CIM elements and behaviors that shall be implemented when management of
444 the IP Interface state is not supported.

445 7.1.3.1 CIM_EnabledLogicalElementCapabilities

446 When state management is not supported, exactly one instance of
447 CIM_EnabledLogicalElementCapabilities may be associated with the CIM_IPProtocolEndpoint instance
448 through an instance of CIM_ElementCapabilities.

449 7.1.3.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

450 The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall not contain any
451 values.

452 7.1.3.2 CIM_IPProtocolEndpoint.RequestedState

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

454 7.1.3.3 CIM_IPProtocolEndpoint.EnabledState

455 The EnabledState property shall have one of the following values: 2 (Enabled), 3 (Disabled), 5 (Not
456 Applicable), or 6 (Enabled but Offline).

457 7.1.4 Modifying ElementName is supported — conditional

458 The CIM_IPProtocolEndpoint.ElementName property may support being modified by the ModifyInstance
459 operation. See 8.10.1.1.

460 This behavior is conditional. This clause describes the CIM elements and behavior requirements when an
461 implementation supports client modification of the CIM_IPProtocolEndpoint.ElementName property.

462 7.1.4.1 CIM_EnabledLogicalElementCapabilities

463 An instance of CIM_EnabledLogicalElementCapabilities shall be associated with the
464 CIM_IPProtocolEndpoint instance through an instance of CIM_ElementCapabilities.

465 7.1.4.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

466 The ElementNameEditSupported property shall have a value of TRUE.

467 7.1.4.1.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

468 The MaxElementNameLen property shall be implemented.

469 7.1.5 Modifying ElementName is not supported

470 This clause describes the CIM elements and behaviors that shall be implemented when the
471 CIM_IPProtocolEndpoint.ElementName property does not support being modified by the ModifyInstance
472 operation.

473 7.1.5.1 CIM_EnabledLogicalElementCapabilities

474 An instance of CIM_EnabledLogicalElementCapabilities may be associated with the
475 CIM_IPProtocolEndpoint instance through an instance of CIM_ElementCapabilities.

476 7.1.5.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

477 The ElementNameEditSupported property shall have a value of FALSE.

478 7.1.5.1.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

479 The MaxElementNameLen property may be implemented. The MaxElementNameLen property is
480 irrelevant in this context.

481 7.1.6 Default gateway

482 An IP interface can be configured with the address of a network gateway. Modeling of the default gateway
483 is optional. When the IP interface is configured with the address of a default gateway, an instance of
484 CIM_RemoteServiceAccessPoint shall represent the default gateway. The instance of
485 CIM_RemoteServiceAccessPoint shall be associated with the instance of CIM_IPProtocolEndpoint
486 through an instance of CIM_RemoteAccessAvailableToElement. An instance of
487 CIM_RemoteServiceAccessPoint may represent the default gateway even when a valid default gateway
488 has not been configured for the IP interface. It can be more convenient for an implementation to always
489 instantiate the instance of CIM_RemoteServiceAccessPoint even if a default gateway has not been
490 assigned to the IP interface rather than conditionally provide the relevant instances. For IPv4, this will
491 result in a single instance of CIM_RemoteServiceAccessPoint associated with the instance of
492 CIM_IPProtocolEndpoint.

493 EXPERIMENTAL

494 For IPv6 or IPv4/IPv6 there may be one or more instances of CIM_RemoteServiceAccessPoint
495 associated with the instance of CIM_IPProtocolEndpoint, because there may be more than one default
496 gateway. In this case, the use of CIM_RemoteAccessAvailableToElement.OrderOfAccess can be used to
497 represent the list of default gateways in priority order.

498 EXPERIMENTAL

499 For IPv6 or IPv4/IPv6 there may be one or more instances of CIM_RemoteServiceAccessPoint
500 associated with the instance of CIM_IPProtocolEndpoint, since there may be more than one default
501 gateway. In this case, the use of CIM_RemoteAccessAvailableToElement.OrderOfAccess can be used to
502 represent the list of default gateways in priority order.

503 7.1.6.1 CIM_RemoteServiceAccessPoint.AccessInfo

504 If the associated value of CIM_IPProtocolEndpoint.ProtocolIFType = 4096 (IPv4), then the value of the
505 AccessInfo property shall be the IPv4 address of the default gateway. The value shall be specified in
506 dotted decimal notation as defined in IETF [RFC 1208](#). A value of 0.0.0.0 shall indicate that a default
507 gateway has not been assigned to the associated IP interface.

508 EXPERIMENTAL

509 If the associated value of CIM_IPProtocolEndpoint.ProtocolIFType = 4097 (IPv6), then the value of the
510 AccessInfo property shall be the IPv6 address of the default gateway. The value shall be specified in the
511 IPv6 notation as defined in IETF [RFC 4291](#). An unspecified address, which has the value of “::/128”, shall
512 indicate that a default gateway has not been assigned to the associated IP interface.

513 If the associated value of CIM_IPProtocolEndpoint.ProtocolIFType = 4098 (IPv4/IPv6), then the value of
514 the AccessInfo property shall be the IPv6 address of the default gateway. The value shall be specified in
515 the IPv6 notation as defined in IETF [RFC 4291](#). An Unspecified Address, which has the value of “::/128”,
516 shall indicate that a default gateway has not been assigned to the associated IP interface.

517 EXPERIMENTAL

518 7.1.6.2 CIM_RemoteAccessAvailableToElement.Antecedent

519 The value of the Antecedent reference shall be the instance of CIM_RemoteServiceAccessPoint.
520 Cardinality *.

521 7.1.6.3 CIM_RemoteAccessAvailableToElement.Dependent

522 The value of the Dependent reference shall be the instance of CIM_IPProtocolEndpoint. Cardinality *.

523 7.1.6.4 CIM_RemoteAccessAvailableToElement.OrderOfAccess

524 If the associated value of CIM_IPProtocolEndpoint.ProtocolIFType = 4096 (IPv4), then the
525 OrderOfAccess property shall have a value of 0 (Zero).

526 7.2 DHCP client is supported

527 When a DHCP client is supported for the IP interface, the DHCP Client Profile ([DSP1037](#)) shall be
528 supported. This behavior is optional.

529 7.3 DNS client is supported

530 When a DNS client is supported for the IP interface, the DNS Client Profile ([DSP1038](#)) shall be
531 supported. This behavior is optional.

532 7.4 Managing alternate configurations — Optional

533 Implementations may support the management of alternate or pending configurations for an IP interface.
534 When an implementation supports the management of alternate configurations, the following behavior
535 shall be supported.

536 7.4.1 Configuration management is supported

537 The CIM_IPConfigurationService class provides management of alternate configurations and support for
538 configuring additional interfaces. When an implementation supports management of alternate
539 configurations, exactly one instance of CIM_IPConfigurationService shall be associated with the Central
540 Instance of the profile through an instance of CIM_ServiceAffectsElement. The existence of the
541 CIM_ServiceAffectsElement association is conditional on the existence of the
542 CIM_IPConfigurationService instance.

543 The CIM_IPConfigurationService instance shall be associated with a CIM_ComputerSystem instance
544 through an instance of CIM_HostedService. The existence of the CIM_HostedService association is
545 conditional on the existence of the CIM_IPConfigurationService instance.

546 7.4.2 Representing an alternate configuration using CIM_IPAssignmentSettingData

547 Each instance of CIM_IPAssignmentSettingData shall represent a possible configuration for an IP
548 interface. The detailed settings for the IP interface shall be contained in the instances of subclasses of
549 CIM_IPAssignmentSettingData, which are associated with the instance of CIM_IPAssignmentSettingData
550 through instances of CIM_OrderedComponent.

551 The existence of one or more instances of CIM_IPAssignmentSettingData is conditional on the existence
552 of the CIM_IPConfigurationService instance. The existence of one or more instances of
553 CIM_ElementSettingData is conditional on the existence of one or more instances of
554 CIM_IPAssignmentSettingData.

555 7.4.2.1 Associating an alternate configuration with an IP interface

556 The instance of CIM_IPAssignmentSettingData shall be associated with the instance of
557 CIM_IPProtocolEndpoint through an instance of CIM_ElementSettingData.

558 7.4.2.1.1 CIM_ElementSettingData.IsCurrent

559 When an instance of CIM_ElementSettingData associates an instance of CIM_IPAssignmentSettingData
560 with an instance of CIM_IPProtocolEndpoint, the CIM_ElementSettingData.IsCurrent property shall have
561 a value of 1 (Is Current) when the configuration represented by the referenced instance of
562 CIM_IPAssignmentSettingData is the last configuration applied to the IP interface represented by the
563 referenced instance of CIM_IPProtocolEndpoint.

564 When an instance of CIM_ElementSettingData associates an instance of CIM_IPAssignmentSettingData
565 with an instance of CIM_IPProtocolEndpoint, the CIM_ElementSettingData.IsCurrent property shall have
566 a value of 2 (Is Not Current) when the configuration represented by the referenced instance of
567 CIM_IPAssignmentSettingData is not the last configuration applied to the IP interface represented by the
568 referenced instance of CIM_IPProtocolEndpoint.

569 **7.4.3 Associating settings using CIM_OrderedComponent**

570 Instances of the subclasses of CIM_IPAssignmentSettingData contain the details of the IP configuration.
571 The CIM_OrderedComponent association aggregates these instances into instances of
572 CIM_IPAssignmentSettingData. An instance of CIM_IPAssignmentSettingData will have one or more
573 instances of its subclasses associated with it through an instance of CIM_OrderedComponent. An
574 instance of a subclass of CIM_IPAssignmentSettingData will be associated with one or more instances of
575 CIM_IPAssignmentSettingData.

576 **7.4.3.1 CIM_OrderedComponent.GroupComponent**

577 An instance of CIM_IPAssignmentSettingData shall be the value of the GroupComponent property of an
578 instance of CIM_OrderedComponent. Cardinality 1..*

579 **7.4.3.2 CIM_OrderedComponent.PartComponent**

580 An instance of a subclass of CIM_IPAssignmentSettingData shall be the value of the PartComponent
581 property of an instance of CIM_OrderedComponent. Cardinality *

582 **7.4.3.3 Interpretation of CIM_OrderedComponent.AssignedSequence**

583 The relative value of the CIM_OrderedComponent.AssignedSequence property shall indicate the order in
584 which aggregated instances of subclasses of CIM_IPAssignmentSettingData are applied to their
585 associated CIM_ProtocolEndpoint instances.

586 **7.4.3.3.1 Use of 0 (zero)**

587 When the CIM_OrderedComponent.AssignedSequence property has a value of 0 (zero), the instance of
588 CIM_SettingData referenced by the CIM_OrderedComponent.PartComponent property shall not be
589 applied when the configuration represented by the CIM_IPAssignmentSettingData instance that is the
590 value of the CIM_OrderedComponent.GroupComponent property is applied. The
591 CIM_OrderedComponent.AssignedSequence property may have the value 0 (zero) when the instance of
592 CIM_OrderedComponent references an instance of CIM_DNSSettingData or
593 CIM_DNSGeneralSettingData. The CIM_OrderedComponent.AssignedSequence property shall not have
594 the value 0 (zero) when the instance of CIM_OrderedComponent does not reference an instance of
595 CIM_DNSSettingData or CIM_DNSGeneralSettingData.

596 **7.4.3.3.2 Discreteness**

597 Two instances of CIM_OrderedComponent that reference the same instance of
598 CIM_IPAssignmentSettingData shall not have the same value for their AssignedSequence properties
599 unless the value is 0 (zero).

600 **7.4.4 Alternate static configuration**

601 When an implementation supports the manual assignment of an IP configuration to the IP endpoint, an
602 instance of CIM_StaticIPAssignmentSettingData shall be associated with the CIM_IPProtocolEndpoint
603 through an instance of CIM_ElementSettingData. This instance of CIM_StaticIPAssignmentSettingData
604 shall be associated with at least one instance of CIM_IPAssignmentSettingData through an instance of
605 CIM_OrderedComponent. When the aggregating IP configuration has been applied to the IP interface
606 and the IP interface is using the settings contained in the instance of
607 CIM_StaticIPAssignmentSettingData, the IsCurrent property of the CIM_ElementSettingData instance

608 has the value 1 (Is Current). Otherwise, the CIM_ElementSettingData.IsCurrent property shall have the
609 value 2 (Is Not Current).

610 **7.4.5 Alternate DHCP configuration**

611 When an alternate configuration includes the configuration of the DHCP client, the implementation will
612 follow the rules for representing a pending DHCP configuration defined in the DHCP Client Profile
613 ([DSP1037](#)).

614 **7.4.6 DNS client alternate configuration**

615 When an alternate configuration includes the configuration of the DNS client, the implementation will
616 follow the rules for representing a pending DNS configuration defined in the DNS Client Profile
617 ([DSP1038](#)).

618 **7.4.7 Relationship between DHCP and DNS configuration**

619 Some settings of the DNS configuration might be provided by the DHCP server.

620 An instance of CIM_IPAssignmentSettingData can have associated with it an instance of
621 CIM_DHCPSettingData and an instance of CIM_DNSSettingData. It is necessary to be able to
622 differentiate between a configuration in which the manual DNS settings take precedence and one in
623 which the DHCP assigned values take precedence. The DNS configuration is assigned according to the
624 principle of last applied. That is, within a given configuration, the last value applied for a property takes
625 precedence.

626 **7.4.7.1 Relationship between DHCP options and the DNS configuration**

627 This clause details the requirements for the relationship between DHCP options and CIM elements that
628 model the DNS configuration. For the requirements expressed in this clause, the following definitions
629 apply:

630 DHCPPE – the instance of CIM_DHCPProtocolEndpoint that represents the DHCP client for an IP
631 interface

632 DNSPE – the instance of CIM_DNSProtocolEndpoint that represents the DNS client that is associated
633 through an instance of CIM_SAPSAPDependency with the same instance of CIM_IPProtocolEndpoint
634 with which the DHCPPE is associated through an instance of CIM_SAPSAPDependency

635 DNS Pending – the instance of CIM_DNSSettingData that is associated through an instance of
636 CIM_OrderedComponent with the instance of CIM_IPAssignmentSettingData that is being applied to the
637 CIM_IPProtocolEndpoint instance

638 DHCP Pending – the instance of CIM_DHCPSettingData that is associated through an instance of
639 CIM_OrderedComponent with the instance of CIM_IPAssignmentSettingData that is being applied to the
640 CIM_IPProtocolEndpoint instance

641 The following requirements shall be met when the *DHCP Client Profile* ([DSP1037](#)) and the *DNS Client*
642 *Profile* ([DSP1038](#)) are implemented:

- 643 • When the OptionsReceived property of the DHCPPE instance and the DHCPOptionsToUse
644 property of the DNSPE instance both contain the value 8 (Domain Name Server), the DNS
645 Servers instrumented in accordance with the "DNS Server Representation" section of the *DNS*
646 *Client Profile* ([DSP1038](#)) shall identify the DNS server addresses specified by the DHCP server
647 as the data for the Domain Name Server DHCP option.
- 648 • When the OptionsReceived property of the DHCPPE instance and the DHCPOptionsToUse
649 property of the DNSPE instance both contain the value 14 (Host Name), the value of the

650 Hostname property of the DNSPE instance shall be the hostname specified by the DHCP server
651 as the data for the Host Name DHCP option.

652 • When the OptionsReceived property of the DHCPPE instance and the DHCPOptionsToUse
653 property of the DNSPE instance both contain the value 17 (Domain Name), the value of the
654 DomainName property of the DNSPE instance shall be the domain name specified by the DHCP
655 server as the data for the Domain Name DHCP option.

656 When the RequestedHostname property of the DNS Pending instance has a non-null value and the
657 RequestedOptions or RequiredOptions property of the DHCP Pending instance contains the value 14
658 (Host Name), the DHCP client shall use the value of the RequestedHostname property as the data for the
659 Host Name DHCP option.

660 **7.4.8 Representing a pending configuration**

661 When an implementation supports alternate configurations, exactly one instance of
662 CIM_IPAssignmentSettingData shall be associated with the Central Instance through an instance of
663 CIM_ElementSettingData whose IsNext property has the value 1 (Is Next) .

664 Exactly one instance of CIM_IPAssignmentSettingData may be associated with the Central Instance
665 through an instance of CIM_ElementSettingData whose IsNext property has the value 3 (Is Next For
666 Single Use).

667 If an instance of CIM_IPAssignmentSettingData is associated with the Central Instance through an
668 instance of CIM_ElementSettingData whose IsNext property has the value 3 (Is Next For Single Use), this
669 instance of CIM_IPAssignmentSettingData shall represent the pending configuration. If no instance of
670 CIM_IPAssignmentSettingData is associated with the Central Instance through an instance of
671 CIM_ElementSettingData whose IsNext property has the value 3 (Is Next For Single Use), the instance of
672 CIM_IPAssignmentSettingData that is associated with the Central Instance through an instance of
673 CIM_ElementSettingData whose IsNext property has the value 1 (Is Next) shall represent the pending
674 configuration.

675 **7.5 Applying an alternate configuration**

676 Two methods exist for applying an alternate configuration to an IP interface. The first method allows a
677 client to explicitly select an alternate configuration to apply to an IP interface. A client can use the
678 CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint() method described in 8.1.1.1 to apply a
679 specific alternate configuration to the IP interface. The second method implicitly applies the pending
680 configuration to the IP interface when the IP interface transitions through a state transition or into a state
681 such that it will accept the pending configuration.

682 **7.5.1 Applying the pending configuration upon transition to enabled**

683 When the value of the EnabledState property of the CIM_IPProtocolEndpoint instance has a value other
684 than 2 (Enabled) and the value of the EnabledState property transitions to 2 (Enabled), the
685 implementation shall apply the pending configuration.

686 **7.5.2 Determining the target CIM_ProtocolEndpoint instance**

687 An instance of CIM_IPAssignmentSettingData or its subclasses may be associated with more than one
688 instance of a subclass of CIM_ProtocolEndpoint through instances of CIM_ElementSettingData.
689 Instances of subclasses of CIM_IPAssignmentSettingData may be aggregated into one or more instances
690 of CIM_IPAssignmentSettingData where the aggregating CIM_IPAssignmentSettingData instances are
691 associated with different instances of CIM_IPProtocolEndpoint. This is allowed as a convenience for
692 instrumentation to reduce the number of instances required when multiple IP interfaces share a common
693 configuration.

694 The following rules unambiguously identify the instance of a subclass of CIM_ProtocolEndpoint that will
695 have an instance of a subclass of CIM_SettingData applied to it when a pending configuration is applied
696 to an instance of CIM_IPProtocolEndpoint. Note that the DNS and DHCP related classes are owned by
697 the *DNS Client Profile* ([DSP1038](#)) and *DHCP Client Profile* ([DSP1037](#)), respectively. The algorithm for
698 determining their use is provided here because it is part of the behavior of applying a configuration.

699 When a pending IP configuration is applied, each instance of CIM_StaticIPAssignmentSettingData that is
700 associated with the CIM_IPAssignmentSettingData instance through an instance of
701 CIM_OrderedComponent shall be applied to the CIM_IPProtocolEndpoint instance that is identified as
702 follows:

- 703 1) The CIM_IPProtocolEndpoint instance shall be associated with the
704 CIM_StaticIPAssignmentSettingData instance through an instance of CIM_ElementSettingData.
- 705 2) The CIM_IPProtocolEndpoint instance shall be the CIM_IPProtocolEndpoint instance to which
706 the aggregating CIM_IPAssignmentSettingData is being applied.

707 When a pending IP configuration is applied, each instance of CIM_DHCPSettingData that is associated
708 with the CIM_IPAssignmentSettingData instance through an instance of CIM_OrderedComponent shall
709 be applied to the CIM_DHCPProtocolEndpoint instance that is identified as follows:

- 710 1) The CIM_DHCPProtocolEndpoint instance shall be associated with the CIM_DHCPSettingData
711 instance through an instance of CIM_ElementSettingData.
- 712 2) The CIM_DHCPProtocolEndpoint instance shall be associated through an instance of
713 CIM_SAPSAPDependency with the CIM_IPProtocolEndpoint instance to which the aggregating
714 CIM_IPAssignmentSettingData is being applied.

715 When a pending IP configuration is applied, each instance of CIM_DNSSettingData that is associated
716 with the CIM_IPAssignmentSettingData instance through an instance of CIM_OrderedComponent shall
717 be applied to the CIM_DNSProtocolEndpoint instance that is identified as follows:

- 718 1) The CIM_DNSProtocolEndpoint instance shall be associated with the CIM_DNSSettingData
719 instance through an instance of CIM_ElementSettingData.
- 720 2) The CIM_DNSProtocolEndpoint instance shall be associated through an instance of
721 CIM_SAPSAPDependency with the CIM_IPProtocolEndpoint instance to which the aggregating
722 CIM_IPAssignmentSettingData is being applied.

723 7.5.3 Applying static IP settings

724 When an instance of CIM_StaticIPAssignmentSettingData is applied to the CIM_IPProtocolEndpoint
725 instance, the values of the properties of the CIM_IPProtocolEndpoint instance shall be the values of the
726 properties of the CIM_StaticIPAssignmentSettingData instance.

727 7.5.3.1 CIM_StaticIPAssignmentSettingData.GatewayIPv4Address

728 If the associated value of CIM_IPProtocolEndpoint.ProtocolIFType = 4096 (IPv4), then the value of the
729 AccessInfo property of the CIM_RemoteServiceAccessPoint that represents the default gateway shall be
730 the value of the CIM_StaticIPAssignmentSettingData.GatewayIPv4Address property.

731 EXPERIMENTAL

732 If the associated value of CIM_IPProtocolEndpoint.ProtocolIFType = 4098 (IPv4/IPv6), then the value of
733 the AccessInfo property of the CIM_RemoteServiceAccessPoint that represents the default IPv4 gateway
734 shall be the value of the CIM_StaticIPAssignmentSettingData.GatewayIPv4Address property.

735 7.5.3.2 CIM_StaticIPAssignmentSettingData.GatewayIPv6Address

736 If the associated value of CIM_IPProtocolEndpoint.ProtocolIFType = 4097 (IPv6), then the value of the
737 AccessInfo property of the CIM_RemoteServiceAccessPoint that represents the default IPv6 gateway
738 shall be the value of the CIM_StaticIPAssignmentSettingData.GatewayIPv6Address property.

739 EXPERIMENTAL

740 7.5.3.3 Successful application of settings

741 An instance of CIM_StaticIPAssignmentSettingData shall be considered successfully applied when the
742 properties of the associated instance of CIM_IPProtocolEndpoint to which the instance of
743 CIM_StaticIPAssignmentSettingData has been applied have the values of the relevant properties of the
744 CIM_StaticIPAssignmentSettingData instance.

745 7.5.4 Applying DHCP settings

746 When a pending configuration includes the configuration of the DHCP client, the DHCP configuration is
747 applied as defined in the *DHCP Client Profile* ([DSP1037](#)).

748 7.5.5 Applying DNS settings

749 When a pending configuration includes DNS client configuration, the DNS configuration is applied as
750 defined in the *DNS Client Profile* ([DSP1038](#)). When the AssignedSequence property of the
751 CIM_OrderedComponent association that references an instance of CIM_DNSSettingData or
752 CIM_DNSGeneralSettingData has a non-zero value, the referenced instance of CIM_DNSSettingData or
753 CIM_DNSGeneralSettingData shall be applied, regardless of whether the application of a preceding
754 CIM_SettingData instance was successful.

755 7.5.6 Resolving overlapped settings

756 When more than one instance of CIM_StaticIPAssignmentSettingData or CIM_DHCPSettingData is
757 associated with the same instance of CIM_IPAssignmentSettingData, each CIM_SettingData instance
758 shall be applied in order (as described in 7.4.3.3) until the implementation determines that the resultant
759 configuration is valid. The amount of time an implementation waits after applying an instance of
760 CIM_SettingData before deciding whether the resultant configuration is valid is implementation specific
761 and outside the scope of this specification. The criterion for determining whether a configuration that is
762 represented by a specific CIM_SettingData instance is valid is implementation specific and outside the
763 scope of this specification.

764 7.6 Relationship with a network interface

765 An IP interface is generally bound to an underlying network interface. The underlying network interface
766 might participate in a LAN and be modeled using the *Host LAN Network Port Profile* ([DSP1035](#)) or a
767 specialization thereof. When the underlying network interface is modeled with instrumentation compliant
768 with the *Host LAN Network Port Profile* ([DSP1035](#)), an instance of CIM_BindsToLANEndpoint shall
769 associate the Central Instance of this profile with an instance of CIM_LANEndpoint that is compliant with
770 the *Host LAN Network Port Profile* ([DSP1035](#)).

771 8 Methods

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

774 8.1 CIM_IPProtocolEndpoint.RequestStateChange()

775 Invocation of the RequestStateChange() method changes the element's state to the value specified in the
 776 RequestedState parameter. The 2 (Enabled) and 3 (Disabled) values of the RequestedState parameter
 777 shall correspond to enabling or disabling the IP network interface, respectively. A value of 11 (Reset)
 778 shall correspond to disabling and then enabling the IP interface.

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

780 No standard messages are defined.

781 Invoking the RequestStateChange() method multiple times could result in earlier requests being
 782 overwritten or lost.

783 **Table 2 – CIM_IPProtocolEndpoint.RequestStateChange() Method: Return code values**

| Value | Description |
|-------|---|
| 0 | Request was successfully executed. |
| 1 | Method is unsupported. |
| 2 | Error occurred. |
| 4096 | Job started: REF returned to started CIM_ConcreteJob. |

784 **Table 3 – CIM_IPProtocolEndpoint.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 the transition to a new state is supposed to take: 0 or NULL – No time requirements <interval> – Maximum time allowed |

785 8.1.1.1 CIM_IPProtocolEndpoint.RequestStateChange() — conditional support

786 When an instance of CIM_EnabledLogicalElementCapabilities is associated with the
 787 CIM_IPProtocolEndpoint instance and the
 788 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least one
 789 value, the CIM_IPProtocolEndpoint.RequestStateChange() method shall be implemented and supported.
 790 The CIM_IPProtocolEndpoint.RequestStateChange() method shall not return a value of 1 (Not
 791 Supported).

792 8.2 CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint()

793 The CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint() method is used to apply a
 794 configuration, as represented by an aggregating instance of CIM_IPAssignmentSettingData, to an IP
 795 interface, as represented by an instance of CIM_IPProtocolEndpoint. Implementation of this method is
 796 optional.

797 Detailed requirements of the ApplySettingToIPProtocolEndpoint() method are specified in Table 4 and
798 Table 5.

799 No standard messages are defined.

800 **Table 4 – CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint() Method: Return code**
801 **values**

| Value | Description |
|--------|--|
| 0 | Request was successfully executed. |
| 1 | Unsupported |
| 2 | Unknown/unspecified error |
| 4 | Failed |
| 0x1000 | Input parameters have been validated and a job started to apply the configuration. |

802 **Table 5 – CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint() Method: Parameters**

| Qualifiers | Name | Type | Description/Values |
|------------|---------------|---------------------------------|-------------------------------------|
| IN, REQ | Configuration | CIM_IPAssignmentSettingData REF | The settings to apply |
| IN, REQ | Endpoint | CIM_IPProtocolEndpoint REF | CIM_IPProtocolEndpoint to configure |
| OUT | Job | CIM_ConcreteJob REF | Returned if job started |

803 The CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint() method shall be implemented as
804 follows:

- 805 • The implementation shall validate that an instance of CIM_ServiceAffectsElement references
806 the CIM_IPConfigurationService instance and the CIM_IPProtocolEndpoint instance that is
807 identified by the Endpoint parameter to the method. If the association does not exist, the return
808 code of the method shall be 4 (Failed).
- 809 • The implementation shall validate that an instance of CIM_ElementSettingData associates the
810 instance of CIM_IPProtocolEndpoint that is identified by the Endpoint parameter with the
811 instance of CIM_IPAssignmentSettingData that is identified by the Configuration parameter. If
812 the association does not exist, the return code of the method shall be 4 (Failed).

813 When the parameters have been validated and the method is applying the settings, the method shall
814 apply the settings as documented in 7.5 and its subclauses.

815 **8.3 Profile conventions for operations**

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

818 The default list of operations is as follows:

- 819 • GetInstance
- 820 • EnumerateInstances
- 821 • EnumerateInstanceNames
- 822 • Associators
- 823 • AssociatorNames

- 824 • References
- 825 • ReferenceNames

826 **8.4 CIM_BindsToLANEndpoint**

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

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

831 **Table 6 – Operations: CIM_BindsToLANEndpoint**

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

832 **8.5 CIM_ElementSettingData**

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

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

837 **Table 7 – Operations: CIM_ElementSettingData**

| Operation | Requirement | Messages |
|-----------------|-------------------------|----------|
| ModifyInstance | Conditional. See 8.5.1. | None |
| Associators | Unspecified | None |
| AssociatorNames | Unspecified | None |
| References | Unspecified | None |
| ReferenceNames | Unspecified | None |

838 **8.5.1 CIM_ElementSettingData — ModifyInstance**

839 The behavior of the ModifyInstance operation varies depending on the property of the association
 840 modified and the instances that are referenced by the association instance.

841 **8.5.1.1 CIM_ElementSettingData Referencing CIM_IPAssignmentSettingData**

842 When an instance of CIM_ElementSettingData associates an instance of CIM_IPAssignmentSettingData
 843 with an instance of CIM_IPProtocolEndpoint, the following rules shall govern the behavior of the
 844 ModifyInstance operation:

- 845 • The ModifyInstance operation shall not allow the IsDefault property to be modified.
- 846 • The ModifyInstance operation shall not allow the IsCurrent property to be modified.
- 847 • When the ModifyInstance operation is used to set the IsNext property to a value of 1 (Is Next),
 848 the ModifyInstance operation shall implement the following behavior:

- 849 1) The ModifyInstance operation shall find all other instances of CIM_ElementSettingData
 850 that associate an instance of CIM_IPAssignmentSettingData with the instance of
 851 CIM_IPProtocolEndpoint referenced by the target instance of CIM_ElementSettingData
 852 where the IsNext property has a value of 1 (Is Next).
- 853 2) For each instance of CIM_ElementSettingData found, the ModifyInstance operation shall
 854 modify the value of its IsNext property to have a value of 2 (Is Not Next).
- 855 • When the IsNext property has a value of 1 (Is Next), the ModifyInstance operation shall not be
 856 supported.
 - 857 • When the ModifyInstance operation is used to set the IsNext property to a value of 3 (Is Next for
 858 Single Use), the ModifyInstance operation shall implement the following behavior:
 - 859 1) The ModifyInstance operation shall find all other instances of CIM_ElementSettingData
 860 that associate an instance of CIM_IPAssignmentSettingData with the instance of
 861 CIM_IPProtocolEndpoint referenced by the target instance of CIM_ElementSettingData
 862 where the IsNext property has a value of 3 (Is Next For Single Use).
 - 863 2) For each instance of CIM_ElementSettingData found, the ModifyInstance operation shall
 864 modify the value of its IsNext property to have a value of 2 (Is Not Next).

865 **8.5.1.2 CIM_ElementSettingData Referencing CIM_StaticIPAssignmentSettingData**

866 When an instance of CIM_ElementSettingData associates an instance of
 867 CIM_StaticIPAssignmentSettingData with an instance of CIM_IPProtocolEndpoint, the ModifyInstance
 868 operation shall not be supported.

869 **8.6 CIM_HostedAccessPoint**

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

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

874 **Table 8 – Operations: CIM_HostedAccessPoint**

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

875 **8.7 CIM_HostedService**

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

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

880 **Table 9 – Operations: CIM_HostedService**

| Operation | Requirement | Messages |
|-------------|-------------|----------|
| Associators | Unspecified | None |

| | | |
|-----------------|-------------|------|
| AssociatorNames | Unspecified | None |
| References | Unspecified | None |
| ReferenceNames | Unspecified | None |

881 **8.8 CIM_IPAssignmentSettingData**

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

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

884 **8.9 CIM_IPConfigurationService**

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

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

887 **8.10 CIM_IPProtocolEndpoint**

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

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

892 **Table 10 – Operations: CIM_IPProtocolEndpoint**

| Operation | Requirement | Messages |
|----------------|--------------------------|----------|
| ModifyInstance | Conditional. See 8.10.1. | None |

893 **8.10.1 CIM_IPProtocolEndpoint — ModifyInstance operation**

894 This clause details the specific requirements for the ModifyInstance operation applied to an instance of
895 CIM_IPProtocolEndpoint.

896 **8.10.1.1 CIM_IPProtocolEndpoint.ElementName property**

897 When an instance of CIM_EnabledLogicalElementCapabilities is associated with the
898 CIM_IPProtocolEndpoint instance and the
899 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported property has a value of TRUE, the
900 implementation shall allow the ModifyInstance operation to change the value of the ElementName
901 property of the CIM_IPProtocolEndpoint instance. The ModifyInstance operation shall enforce the length
902 restriction specified in the MaxElementNameLen property of the instance of
903 CIM_EnabledLogicalElementCapabilities.

904 When no instance of CIM_EnabledLogicalElementCapabilities is associated with the
905 CIM_IPProtocolEndpoint instance, or the ElementNameEditSupported property of the
906 CIM_EnabledLogicalElementCapabilities instance has a value of FALSE, the implementation shall not
907 allow the ModifyInstance operation to change the value of the ElementName property of the
908 CIM_IPProtocolEndpoint instance.

909 8.11 CIM_OrderedComponent

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

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

914 **Table 11 – Operations: CIM_OrderedComponent**

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

915 8.11.1 CIM_OrderedComponent — ModifyInstance

916 The ModifyInstance operation may be supported for CIM_OrderedComponent. When an instance of
 917 CIM_OrderedComponent references an instance of CIM_DNSSettingData or an instance of
 918 CIM_DNSGeneralSettingData, the AssignedSequence property may be modified. When an instance of
 919 CIM_OrderedComponent references an instance of CIM_StaticIPAssignmentSettingData or an instance
 920 of CIM_DHCPSettingData, the AssignedSequence property shall not be modified.

921 8.12 CIM_RemoteAccessAvailableToElement

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

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

926 **Table 12 – Operations: CIM_RemoteAccessAvailableToElement**

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

927 8.13 CIM_RemoteServiceAccessPoint

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

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

930 **8.14 CIM_ServiceAffectsElement**

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

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

935 **Table 13 – Operations: CIM_ServiceAffectsElement**

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

936 **8.15 CIM_StaticIPAssignmentSettingData**

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

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

941 **Table 14 – Operations: CIM_StaticIPAssignmentSettingData**

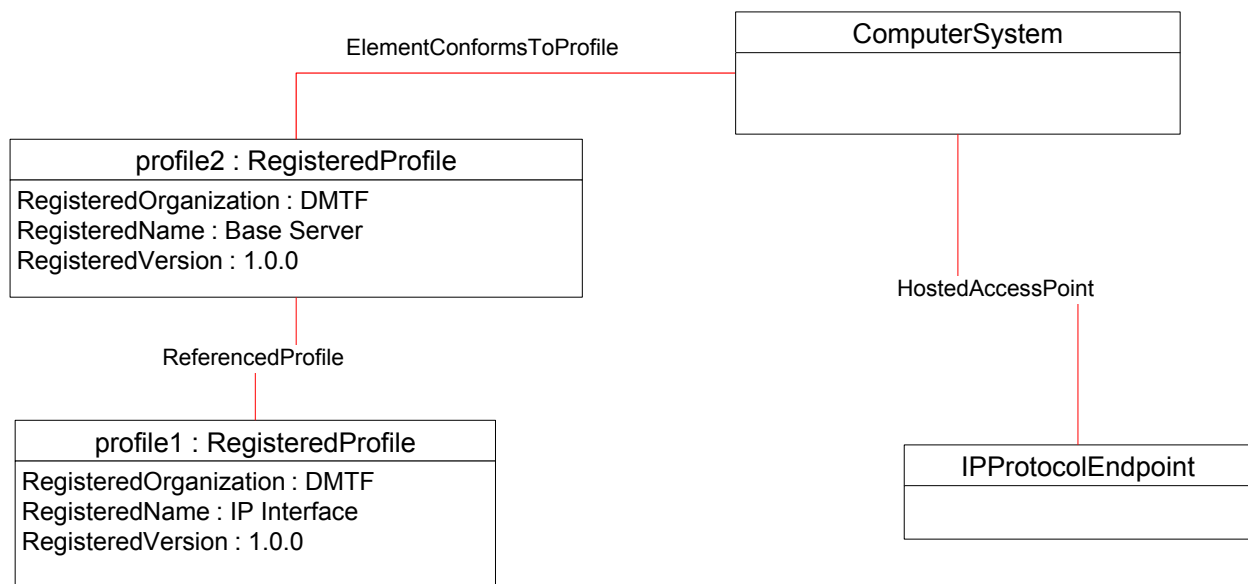
| Operation | Requirement | Messages |
|----------------|-------------|----------|
| ModifyInstance | Optional | None |

943 9 Use cases

944 This clause contains object diagrams and use cases for the *IP Interface Profile*.

945 9.1 Miscellaneous object diagrams

946 The object diagram in Figure 2 shows one possible method for advertising profile conformance. The
 947 instances of CIM_RegisteredProfile are used to identify the version of the *IP Interface Profile* with which
 948 an instance of CIM_IPProtocolEndpoint and its associated instances are conformant. An instance of
 949 CIM_RegisteredProfile exists for each profile that is instrumented in the system. One instance of
 950 CIM_RegisteredProfile identifies the “DMTF Base Server Profile version 1.0.0”. The other instance
 951 identifies the “DMTF IP Interface Profile version 1.0.0”. The CIM_IPProtocolEndpoint instance is scoped
 952 to an instance of CIM_ComputerSystem. This instance of CIM_ComputerSystem is conformant with the
 953 DMTF [Base Server Profile](#) version 1.0.0 as indicated by the CIM_ElementConformsToProfile association
 954 to the CIM_RegisteredProfile instance.



955

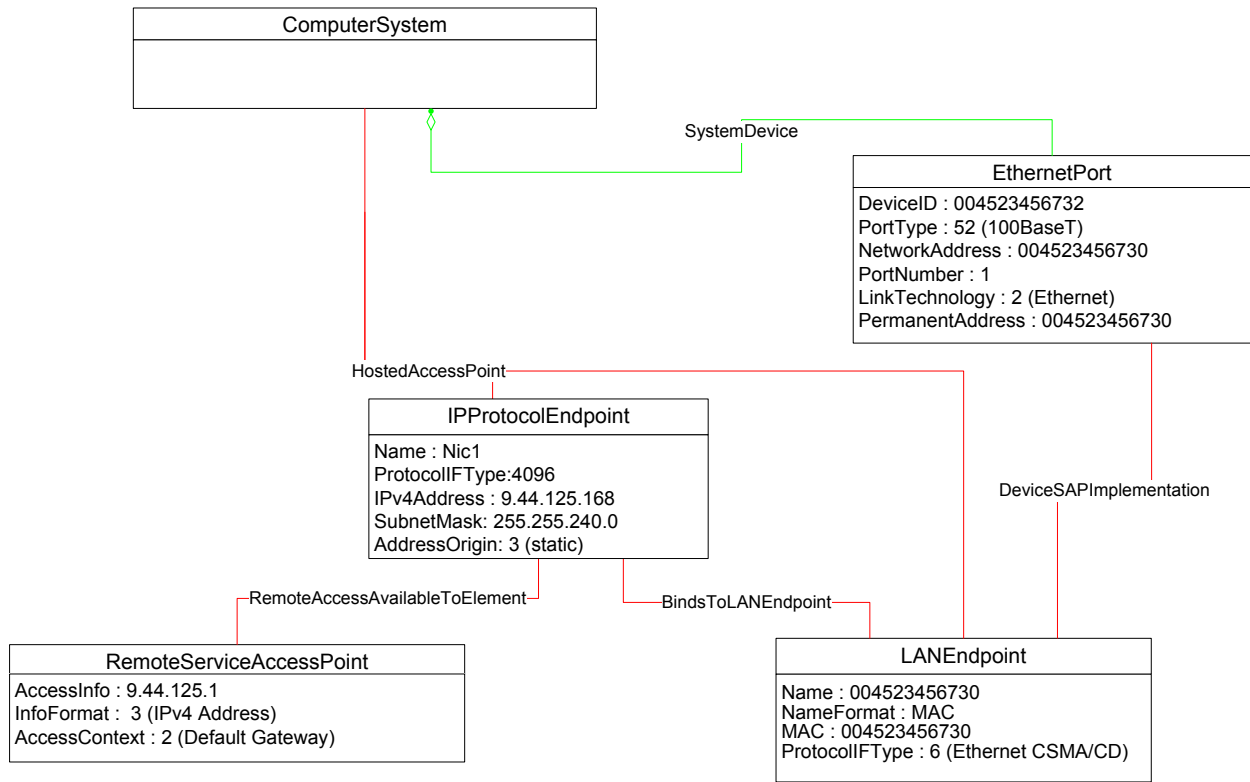
956

Figure 2 – Registered profile

957 The object diagram shown in Figure 3 contains the basic elements used to model the current
 958 configuration of an IP interface when the CIM_IPProtocolEndpoint.ProtocolIFType is 4096 (Ipv4). The IP
 959 interface is bound to an Ethernet NIC, as illustrated by the CIM_BindsToLANEndpoint association
 960 between the CIM_IPProtocolEndpoint instance and the CIM_LANEndpoint instance. The AddressOrigin
 961 property of the CIM_IPProtocolEndpoint has a value of "static", indicating that the configuration was
 962 statically assigned. In this diagram, the [Ethernet Port Profile](#) and *IP Interface Profile* have been
 963 implemented.

964 The default gateway used by the IP interface is represented by the instance of
 965 CIM_RemoteServiceAccessPoint that is associated with the CIM_IPProtocolEndpoint instance through an
 966 instance of CIM_RemoteAccessAvailableToElement.

967



968

969

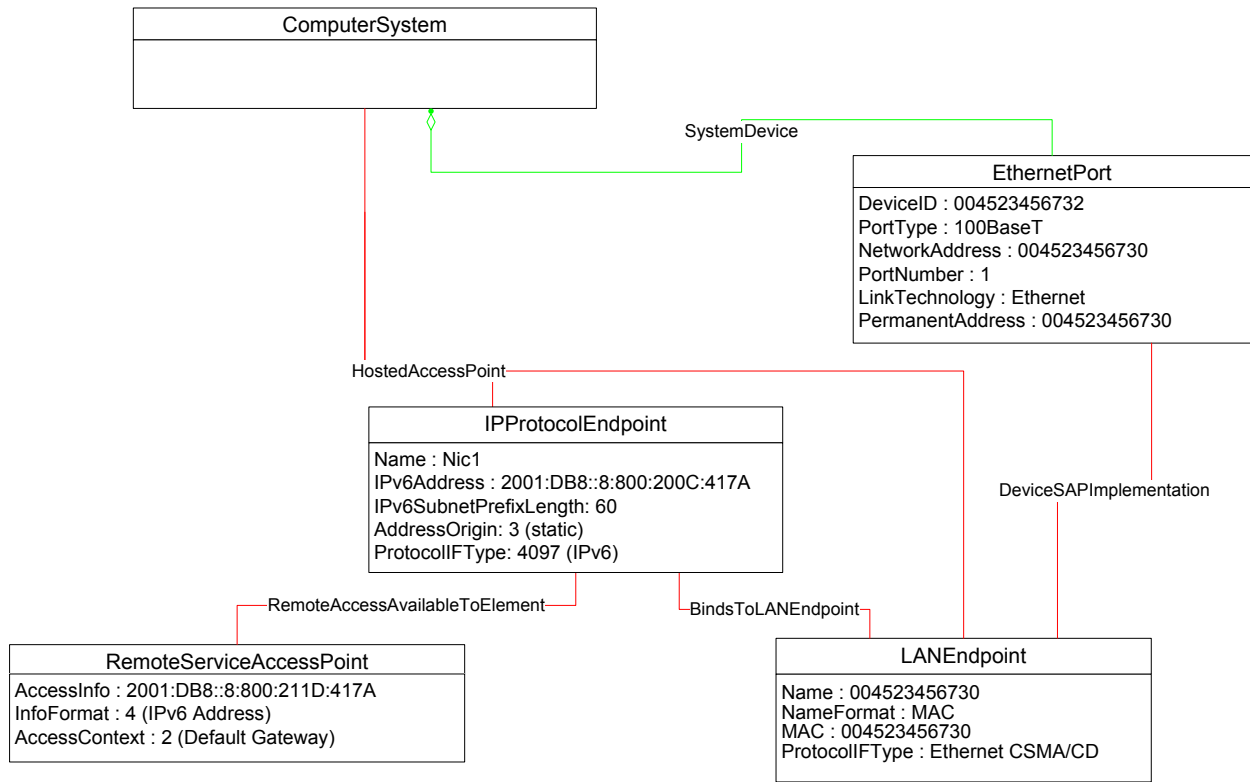
Figure 3 – Basic configuration — IPv4

970 **EXPERIMENTAL**

971 The object diagram shown in Figure 4 contains the basic elements used to model the current
 972 configuration of an IP interface when the CIM_IPProtocolEndpoint.ProtocolIFType is 4097 (IPv6). Note
 973 the similarities between this figure and the previous diagram. In this diagram, the [Ethernet Port Profile](#)
 974 and *IP Interface Profile* have been implemented.

975

976



977

978

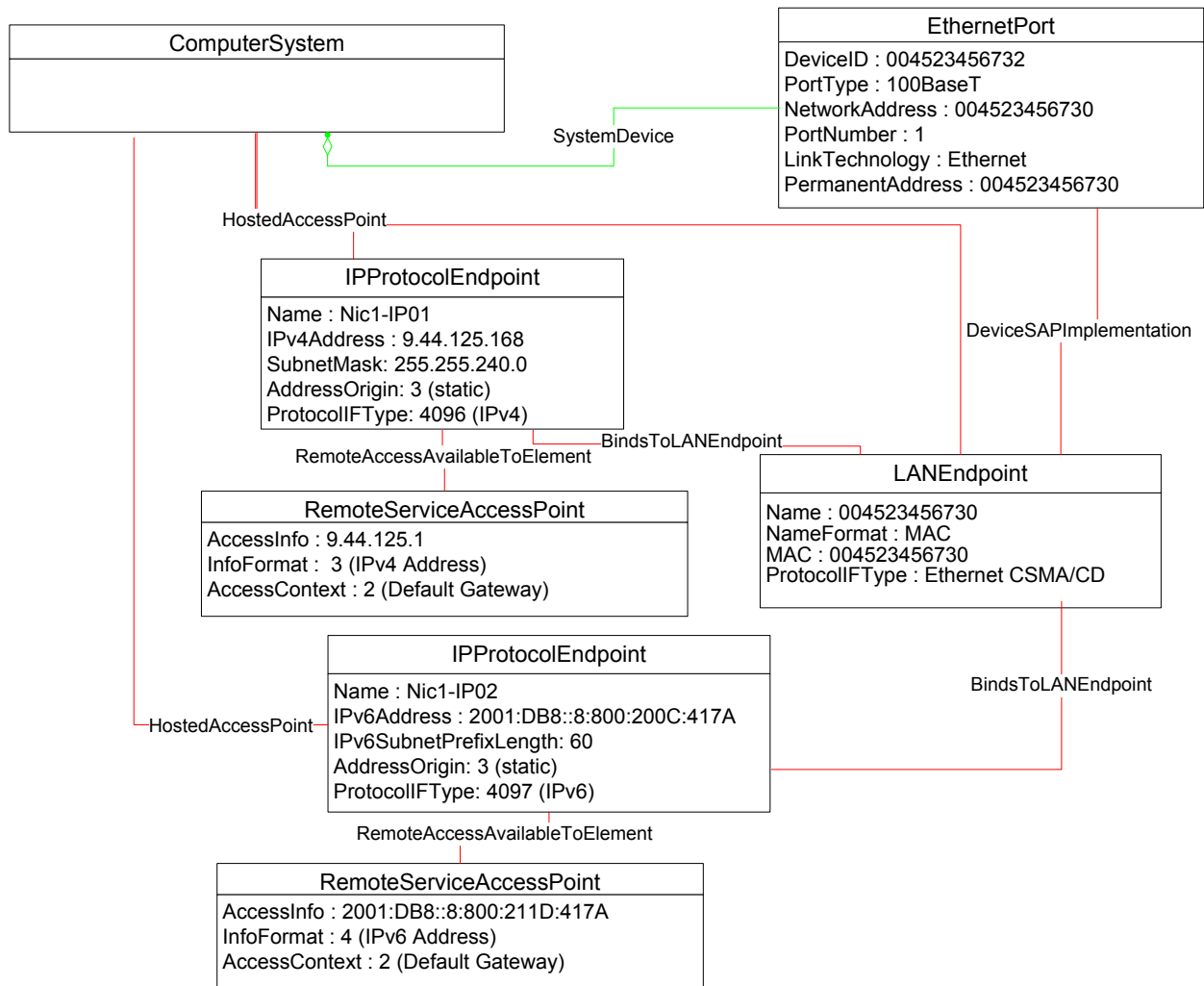
Figure 4 – Basic configuration — IPv6

979

The object diagram shown in Figure 5 contains the basic elements used to model the current configuration of two IP interfaces on a single EthernetPort — one that has an IPv4 address and one that has an IPv6 address. In this diagram, the [Ethernet Port Profile](#) and *IP Interface Profile* have been implemented.

982

983



984
985

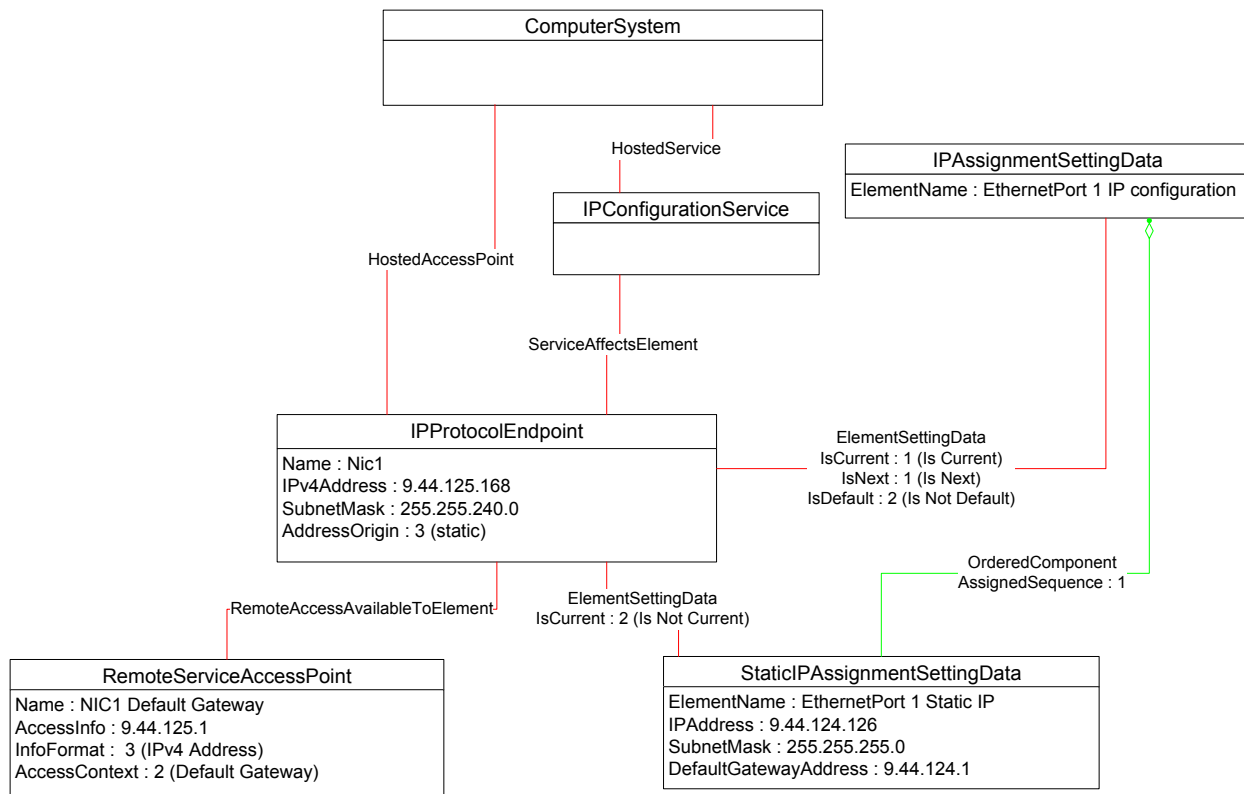
986

Figure 5 – Basic configuration — IPv4 and IPv6

987 **EXPERIMENTAL**

988 Figure 6 illustrates the elements and properties of an IP interface that supports static configuration. The
 989 IP interface currently has a single, alternate configuration associated with it. The optional IP configuration
 990 management behavior is depicted in this object diagram. Note that the pending configuration has been
 991 modified after it was applied to the CIM_IPProtocolEndpoint. Hence the values for properties of
 992 CIM_IPProtocolEndpoint do not align with the values of properties of the
 993 CIM_StaticIPAssignmentSettingData instance.

994



995

996

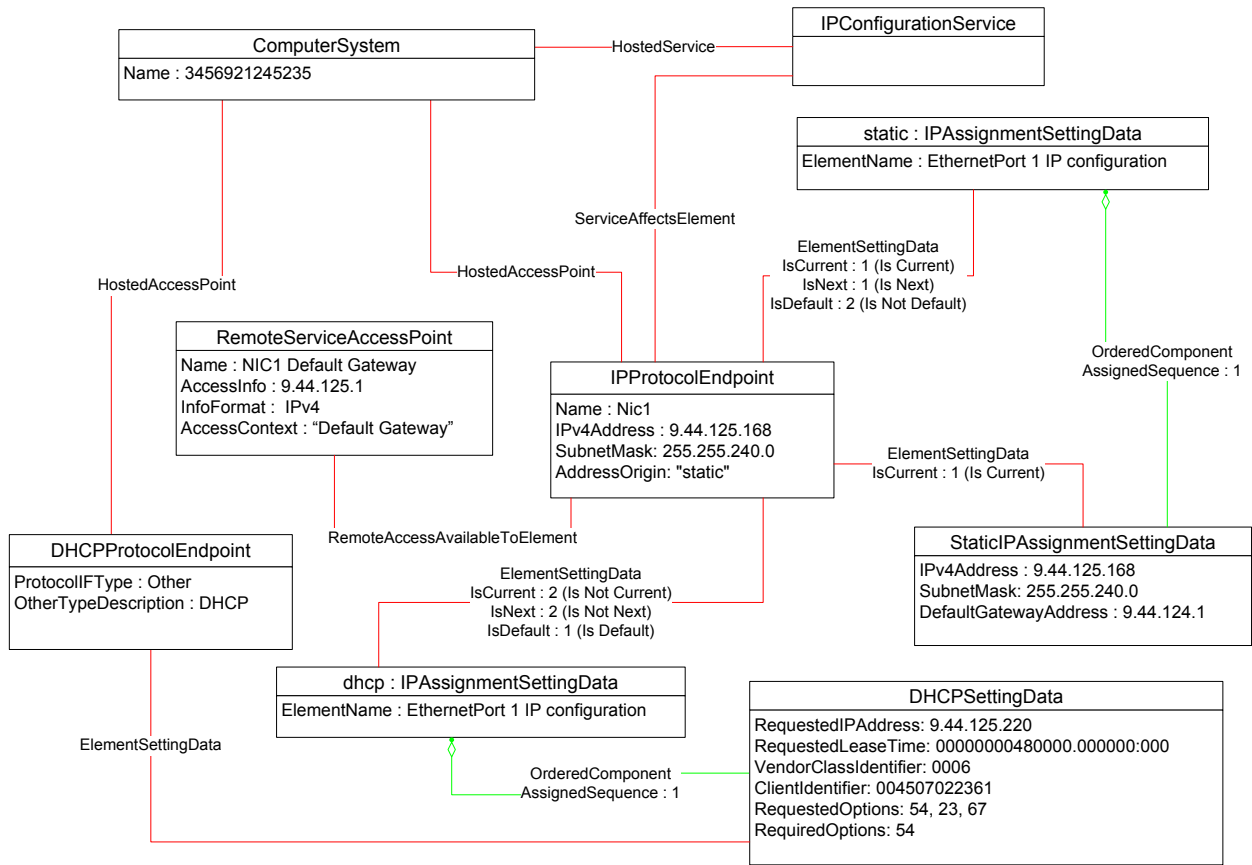
Figure 6 – Static current and pending configuration

997 The object diagram in Figure 7 illustrates an IP interface with two supported alternate configurations. Two
 998 discrete IP configuration options are available for the IP interface. Each option is represented by an
 999 instance of `CIM_IPAssignmentSettingData`. One configuration option represents the ability to statically
 1000 assign the IP configuration. This option is indicated by the instance of `CIM_OrderedComponent` that
 1001 associates the `CIM_IPAssignmentSettingData` instance with an instance of
 1002 `CIM_StaticIPAssignmentSettingData`. The other configuration option is to obtain the configuration through
 1003 a DHCP client. This option is indicated by the instance of `CIM_OrderedComponent` that associates the
 1004 `CIM_IPAssignmentSettingData` with an instance of `CIM_DHCPSettingData`.

1005 In this example, each configuration option consists of a single instance of a subclass of
 1006 `CIM_IPAssignmentSettingData`. Therefore, the value of the `AssignedSequence` property of the
 1007 `CIM_OrderedComponent` instances is irrelevant.

1008 The default configuration is to attempt to obtain a configuration through DHCP. This default is indicated
 1009 by the `IsDefault` property having a value of 1 (Is Default) on the `CIM_ElementSettingData` instance that
 1010 associates the `CIM_IPAssignmentSettingData` instance with the `CIM_IPProtocolEndpoint` instance.
 1011 However, the current configuration of the IP interface was statically assigned using the configuration
 1012 identified by the `CIM_IPAssignmentSettingData` instance *static*. This configuration is indicated by the
 1013 value of the `IsCurrent` property on the instance of `CIM_ElementSettingData` that associates the
 1014 `CIM_IPAssignmentSettingData` instance *static* with the `CIM_IPProtocolEndpoint` instance, and by the
 1015 value of the `AddressOrigin` property on the `CIM_IPProtocolEndpoint` instance. When the interface is
 1016 restarted, the static configuration will be used again for the IP interface. This behavior is indicated by the
 1017 value of the `IsNext` property on the instance of `CIM_ElementSettingData` that associates the
 1018 `CIM_IPAssignmentSettingData` instance *static* to the `CIM_IPProtocolEndpoint` instance.

1019



1020

1021

Figure 7 – Static and DHCP pending configurations

1022
1023

The object diagram in Figure 8 provides an example of an IP interface that was configured to default to a statically assigned IP configuration if the DHCP client failed to obtain a configuration from a DHCP server.

1024
1025
1026
1027

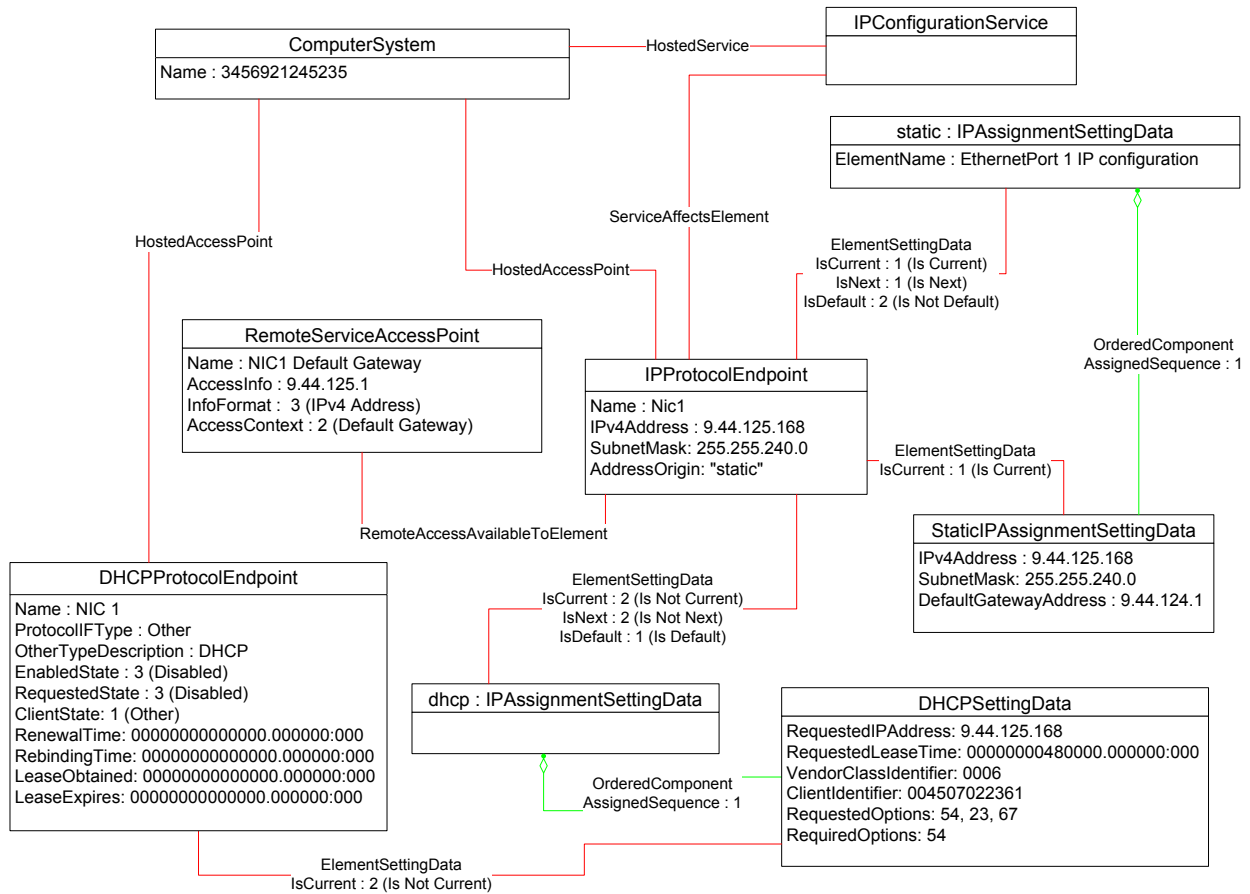
The EnabledState and ClientState properties of the CIM_DHCPProtocolEndpoint instance indicate that the DHCP client is not disabled but neither is it actively attempting to obtain a configuration any longer. No instance of CIM_RemoteServiceAccessPoint is associated with the CIM_DHCPProtocolEndpoint instance because the DHCP client failed to communicate with a DHCP server.

1028
1029

The AddressOrigin property of the CIM_IPProtocolEndpoint instance reflects that the address was assigned statically.

1030

1031



1032

1033

Figure 8 – DHCP timed out to a static configuration

1034

1035

1036

1037

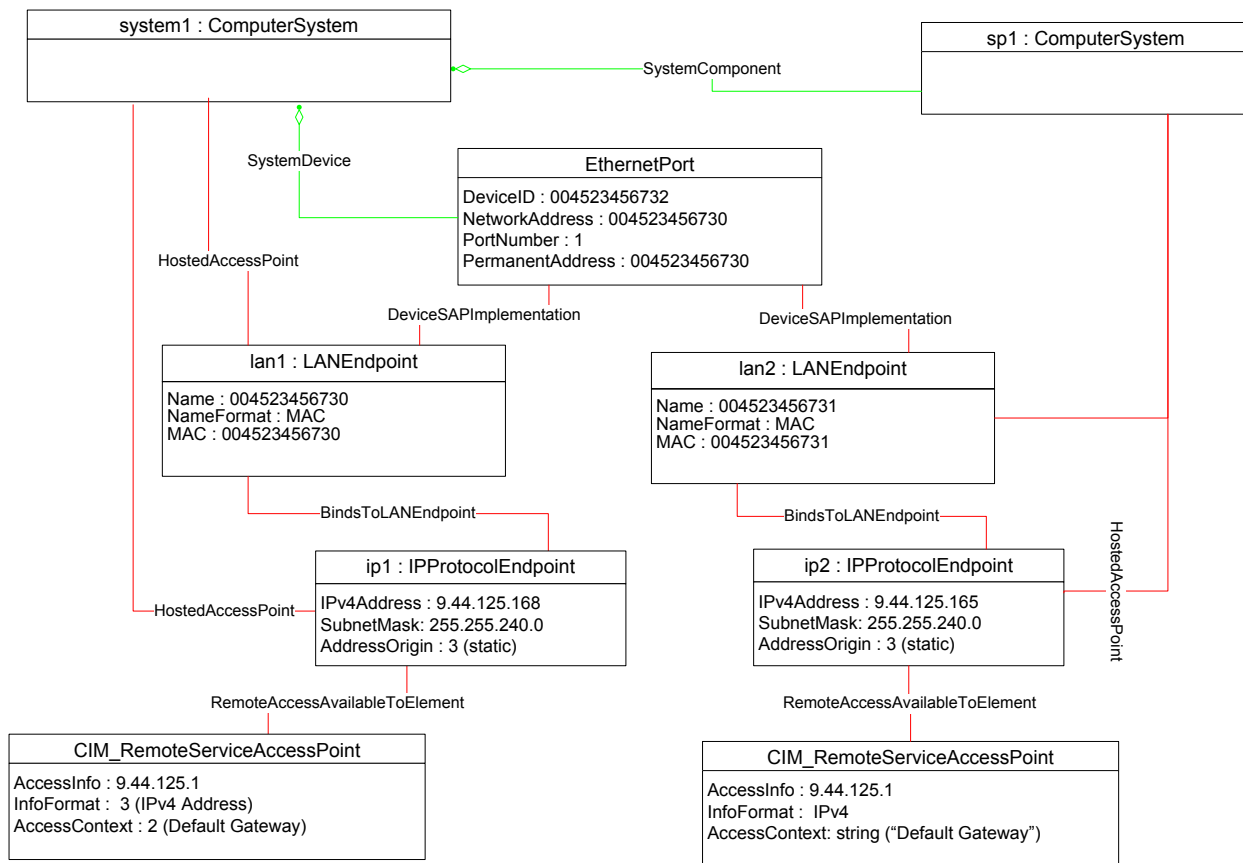
1038

1039

1040

The object diagram in Figure 9 illustrates a configuration in which a system contains an integrated service processor and they share the network interface of the system. The CIM_EthernetPort instance is associated with the system1 instance, which indicates that the network device is owned by the server. The MAC property of the lan1 instance matches the PermanentAddress property of the CIM_EthernetPort instance, which indicates that the server is using the hardware MAC. The MAC property of the lan2 instance is different, which indicates that the service processor has been assigned a logical MAC. The system and service processor each have a unique IP interface that has been statically configured.

1041



1042

1043

Figure 9 – Service processor and server share an NIC

1044 The object diagrams in Figure 10 through Figure 14 show different aspects of a single system. The
 1045 system has support for the DNS and DHCP clients. For configurations using DHCP, the DNS
 1046 configuration can be statically assigned or partially assigned through DHCP. The system itself does not
 1047 support the persistence of alternate configurations. Rather the instrumentation layer presents the different
 1048 configuration possibilities as distinct alternate configurations.

1049 Note that in the following figures extraneous classes that are not relevant to the point being illustrated are
 1050 not shown. For example, the CIM_HostedAccessPoint associations are never included.

1051 The object diagram in Figure 10 outlines the alternate configurations presented by the instrumentation
 1052 layer for the system. Three alternate configurations are shown: static_only, dhcp_only, and dhcp_static.

1053 The system persists a single underlying static IP configuration, which is represented by static1. When the
 1054 configuration selected is static only or DHCP and then static, the same client static IP configuration is
 1055 used.

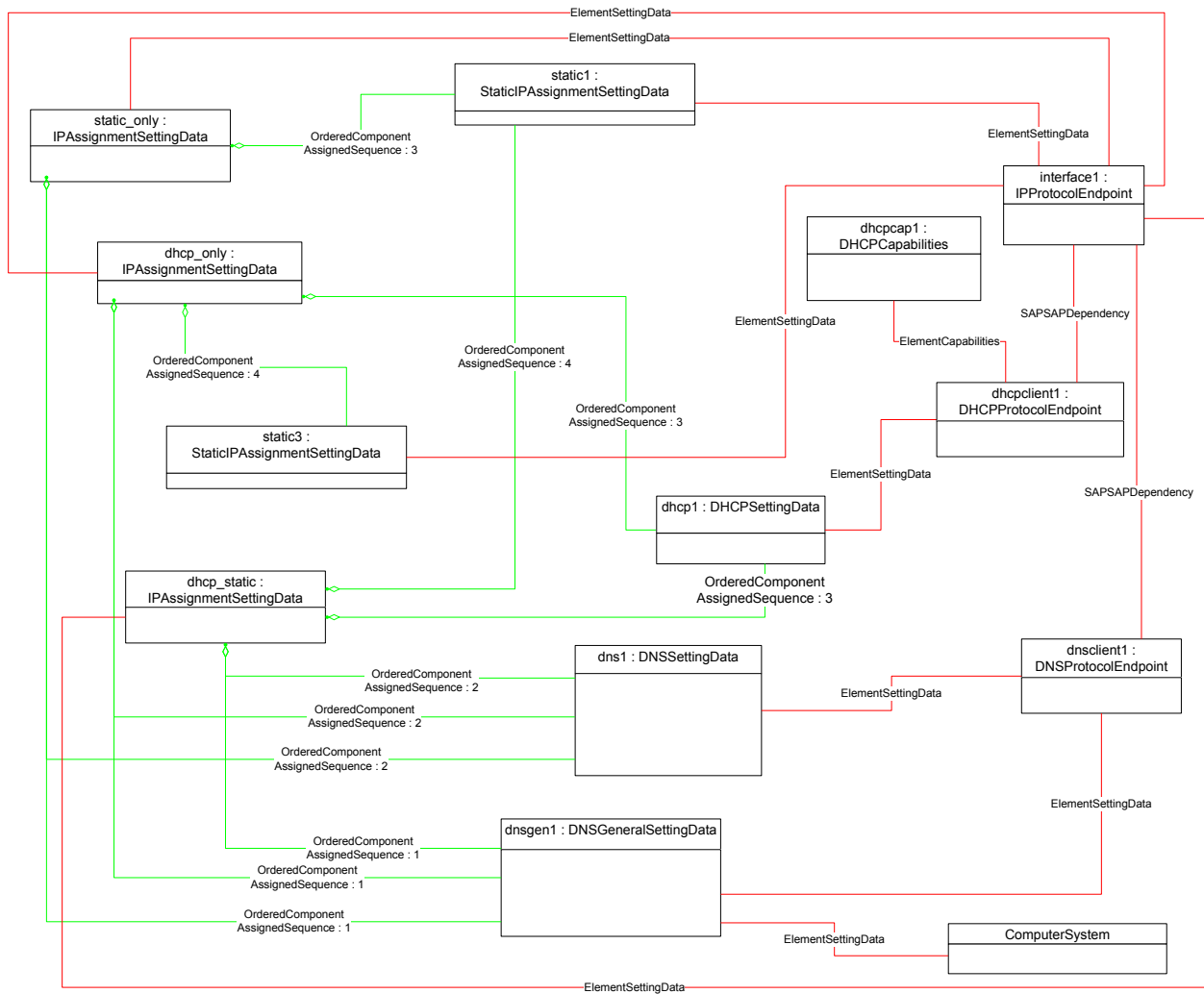
1056 The system persists a single underlying DNS configuration represented by dns1 and dnsgen1.

1057 static_only represents a configuration that uses static assignment of the IP configuration, including
 1058 support for static configuration of the DNS client. This behavior is indicated by the aggregated instances:
 1059 static1, dns1, and dnsgen1.

1060 dhcp_only represents a configuration that uses DHCP to obtain the IP configuration. This behavior is
 1061 indicated by the aggregated instance dhcp1. The DNS configuration can be assigned through DHCP or
 1062 statically assigned. This behavior is indicated by the aggregated instances dns1 and dnsgen1. In the

1063 event the DHCP client is unable to obtain a configuration, the system is implemented to default to a hard-
 1064 coded, well-known default static IP configuration. The existence of a default configuration is indicated by
 1065 the aggregated instance static3. Note that no advertisement mechanism is specified in the profile to
 1066 indicate that static3 represents hard-coded values that cannot be modified by the client. If the system
 1067 were implemented such that the DHCP client would be continually in use without a timeout to a static
 1068 configuration, the aggregated instance static3 would not exist.

1069 dhcp_static represents a configuration that attempts to use DHCP to obtain an IP configuration. In the
 1070 event the DHCP client fails to obtain a configuration, the system defaults to a client-assigned static IP
 1071 configuration. This behavior is indicated by the instances dhcp1 and static1 and the relative values of the
 1072 AssignedSequence property of the instances of CIM_OrderedComponent, which aggregate them into
 1073 dhcp_static.



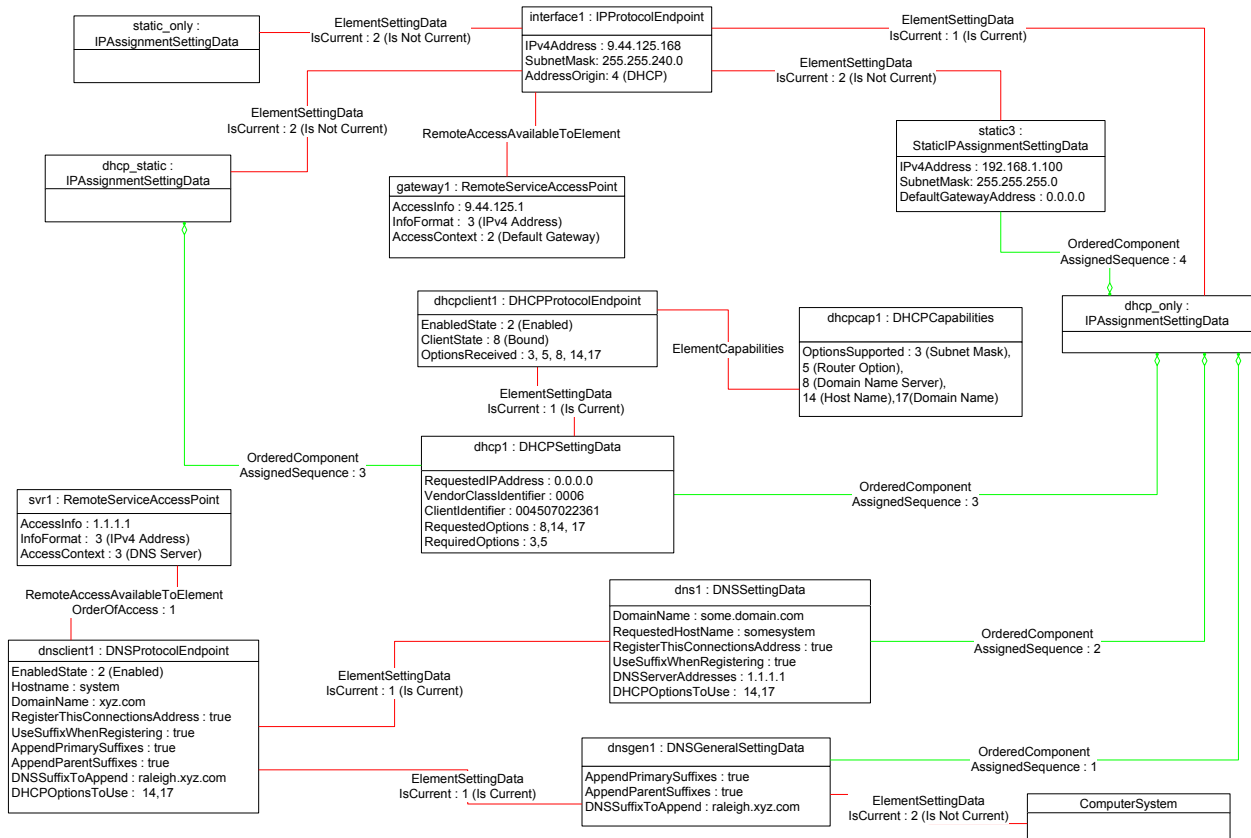
1074

1075

Figure 10 – Configuration choices

1076 The object diagram in Figure 11 reflects the system when the DHCP configuration method has been used
 1077 and the DNS configuration has partially been assigned through DHCP and partially statically configured.

1078 The use of the DHCP-only configuration is indicated by the IsCurrent property of the instance of
 1079 CIM_ElementSettingData that associates dhcp_only to interface1 having the value 1 (Is Current). The
 1080 DHCP configuration includes DHCP options that affect the DNS configuration. The DHCP options 8, 14,
 1081 and 17 are requested as indicated by the RequestedOptions property of dhcp1. Each of these options
 1082 was in turn received by the DHCP client, which is indicated by the value of the OptionsReceived property
 1083 of dhcpclient1. The DNS client has been configured to use the values received for options 14 and 17 as
 1084 indicated by the presence of these values in the DHCPOptionsToUse property of dnsclient1. The
 1085 properties on dnsclient1 reflect the current DNS client configuration. Note that the actual current
 1086 configuration does not directly reflect the configuration indicated by dns1 and dnsgen1. The two
 1087 properties for which values were supplied by the DHCP options instead reflect the values assigned by the
 1088 DHCP server.



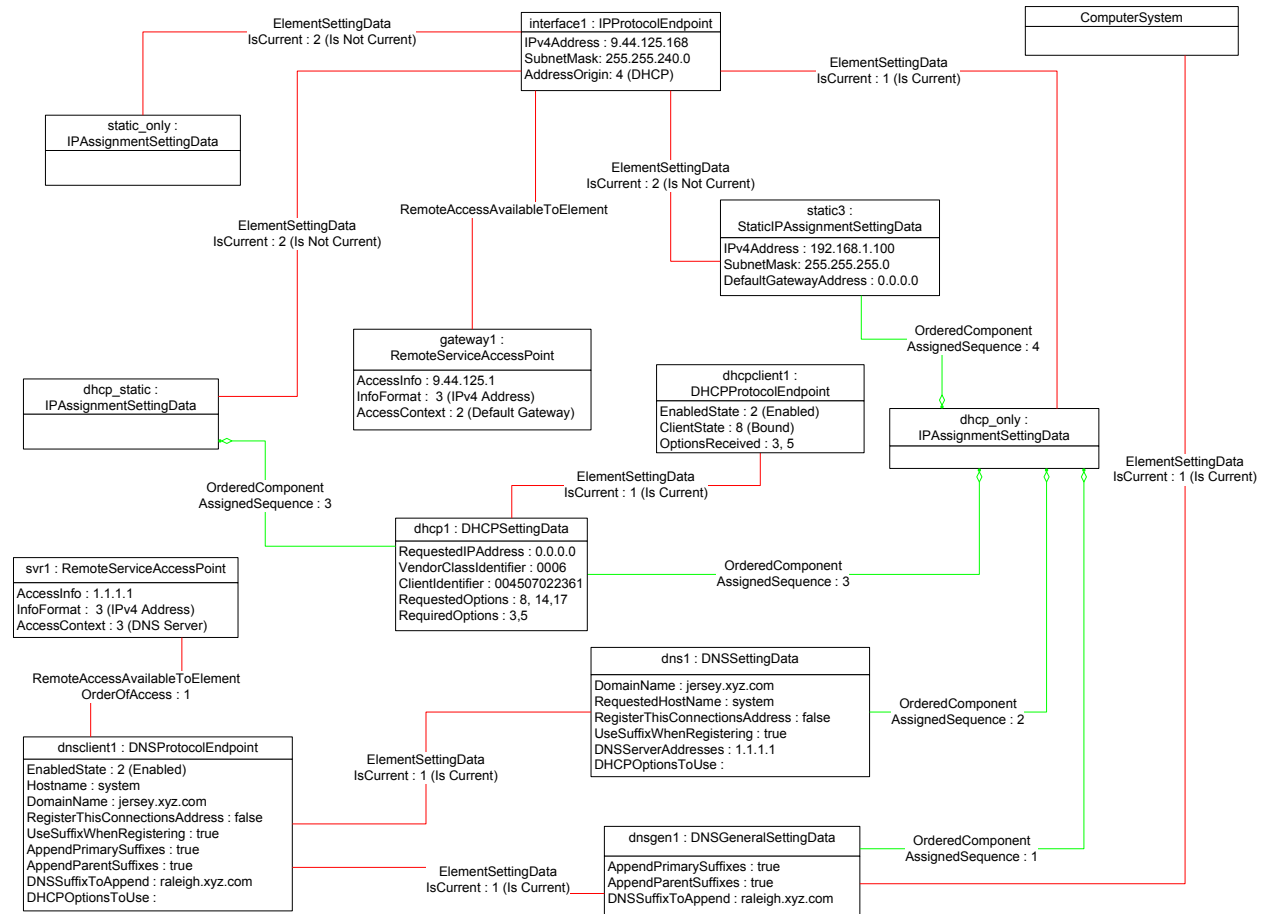
1089

Figure 11 – DHCP assigned partial DNS

1090

1091 The object diagram in Figure 12 reflects the system when the DHCP configuration method has been used
 1092 and the DNS configuration has been statically configured.

1093 The use of the DHCP-only configuration is indicated by the IsCurrent property of the instance of
 1094 CIM_ElementSettingData that associates dhcp_only to interface1 having the value 1 (Is Current).
 1095 Although the DHCP configuration includes DHCP options that affect the DNS configuration, the values
 1096 returned are not being used by the DNS client. This behavior is indicated by the absence of any values in
 1097 the DHCPOptionsToUse property of dnsclient1. The actual current configuration directly reflects the
 1098 configuration indicated by dns1 and dnsgen1 because no DHCP options are selected for use.



1099

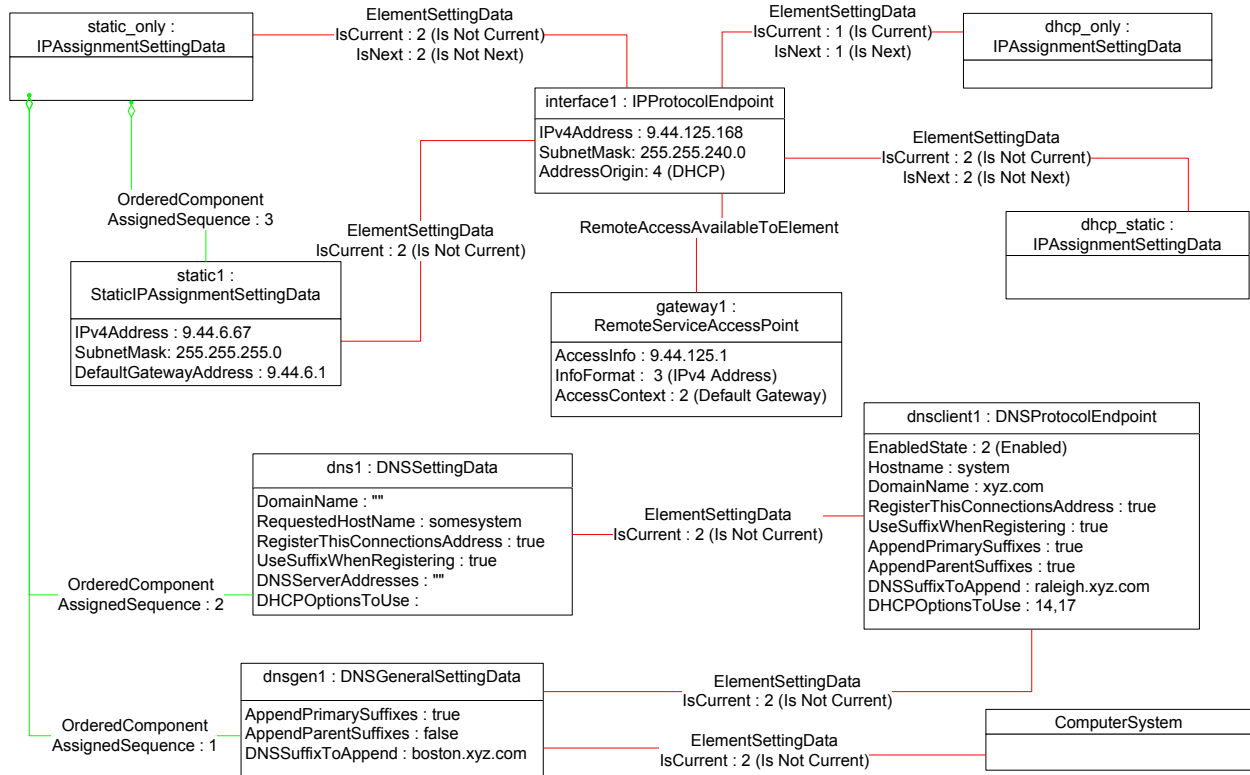
1100

Figure 12 – DHCP with DNS statically configured

1101 **9.1.1 Sequence for disabled DNS client**

1102 The following three object diagrams illustrate the system when a client is configuring it to use a static IP
 1103 configuration with the DNS client disabled. The client first modifies the pending static configuration so that
 1104 the DNS settings will not be applied. Then it disables the DNS client directly. Finally, it applies the static
 1105 configuration.

1106 The object diagram in Figure 13 illustrates the state of the system before the client begins modifying it to
 1107 use a static IP configuration with DNS disabled. The last configuration applied was the DHCP-only
 1108 configuration, which is indicated by the value of the IsCurrent property of the CIM_ElementSettingData
 1109 instance that references dhcp_only and interface1. The static_only configuration has not yet been
 1110 modified by the client. As shown, the alternate DNS configuration represented by dns1 and dnsgen1
 1111 would be applied if static_only were applied to interface1.

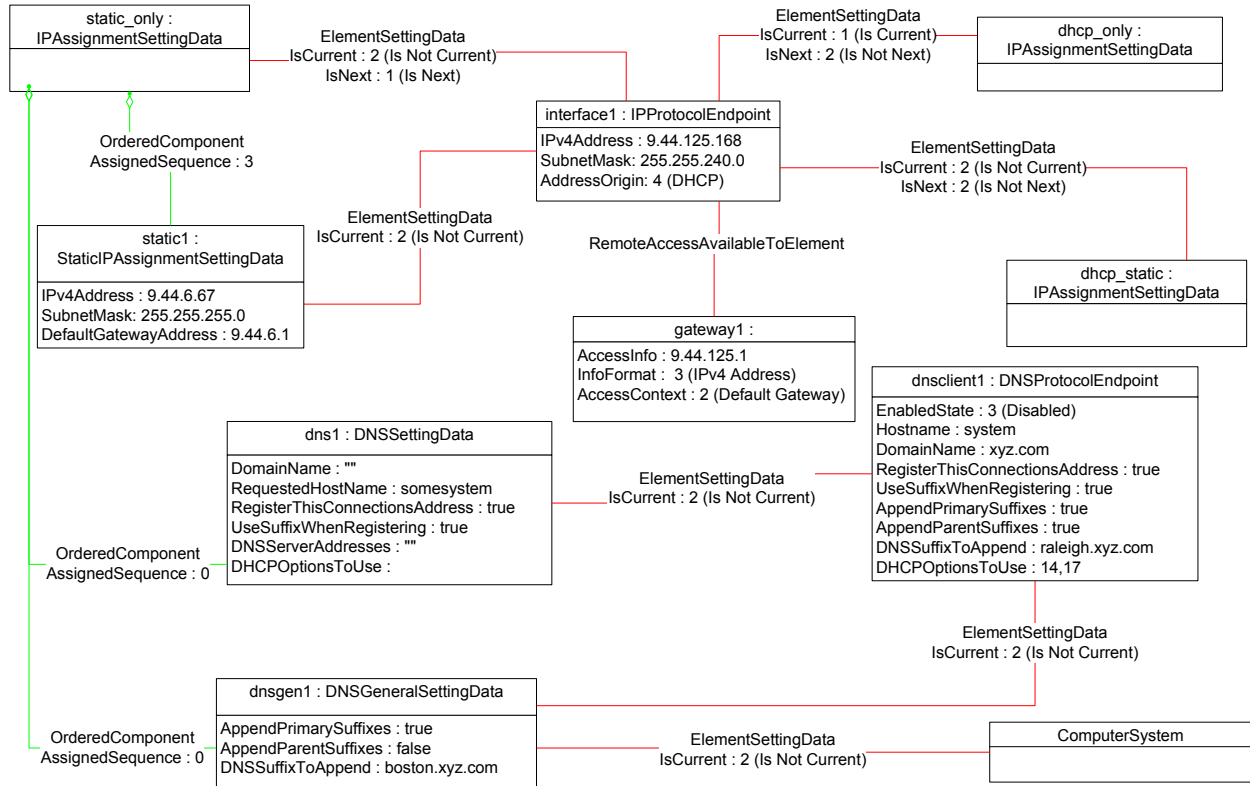


1112

1113

Figure 13 – Static without DNS configuration — One

1114 In Figure 14, static_only has been made the pending configuration for interface1. This behavior is
 1115 indicated by the value of the IsNext property of the instance of CIM_ElementSettingData that references
 1116 static_only and interface1. static_only has been modified such that the DNS configuration will not be
 1117 applied. This behavior is indicated by the AssignedSequence property having a value of 0 (zero) for each
 1118 of the CIM_OrderedComponent instances that reference static_only and dns1 or dnsgen1. Separately,
 1119 the DNS client has been disabled, which is indicated by the value of the EnabledState property of
 1120 dnsclient1.



1121

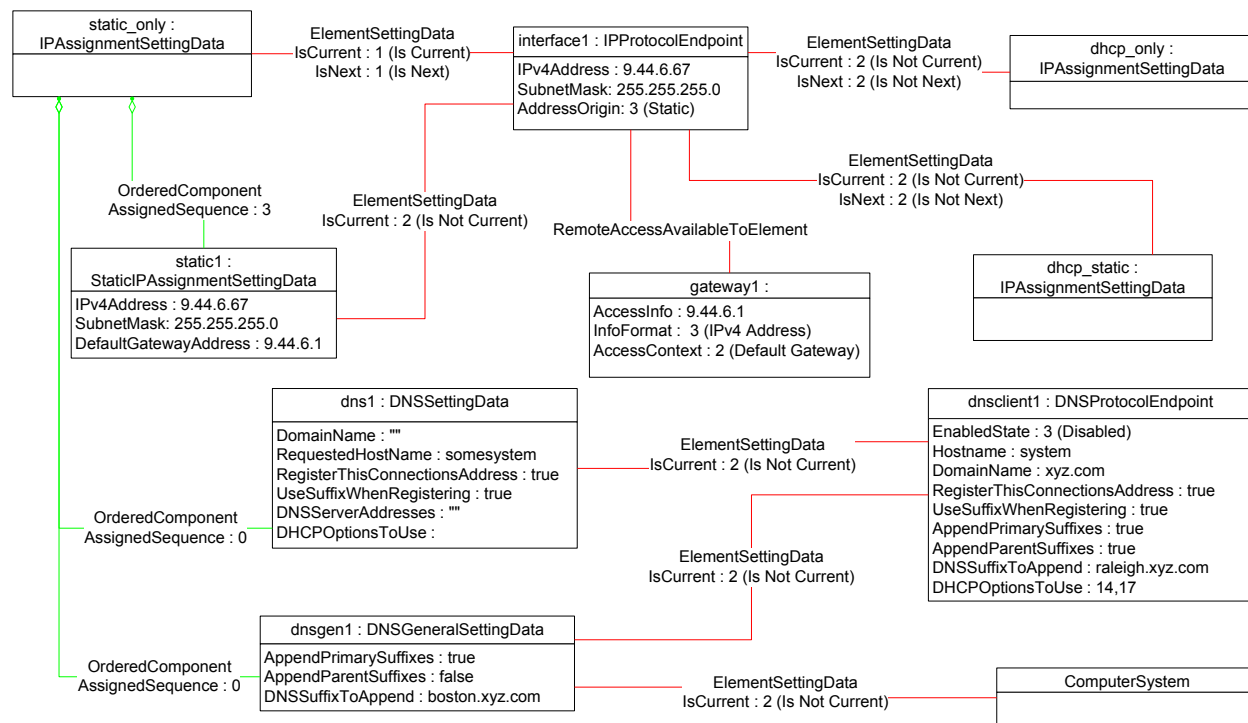
1122

Figure 14 – Static without DNS configuration — Two

1123

Figure 15 shows the system after static_only has been applied to interface1. Note that the current DNS configuration has not changed as a result of applying static_only to interface1.

1124



1125

1126

Figure 15 – Static without DNS configuration — Three

1127 9.2 Determine supported configuration methods

1128 A client can determine which configuration methods are supported for a given interface as follows:

- 1129 1) Find all instances of CIM_IPAssignmentSettingData that are associated with the
- 1130 CIM_IPProtocolEndpoint instance.
- 1131 2) For each instance of CIM_IPAssignmentSettingData:
 - 1132 a) Find all instances of subclasses of CIM_IPAssignmentSettingData that are associated with
 - 1133 the CIM_IPAssignmentSettingData instance through an instance of
 - 1134 CIM_OrderedComponent.
 - 1135 b) Query the value of the AddressOrigin property to determine the supported identified
 - 1136 configuration method.

1137 9.3 Determine gateway address

1138 A client can find the default gateway in use for an IP interface as follows:

- 1139 1) Find all instances of CIM_RemoteServiceAccessPoint that are associated with the
- 1140 CIM_IPProtocolEndpoint instance through an instance of
- 1141 CIM_RemoteAccessAvailableToElement.
- 1142 2) For each instance of CIM_RemoteServiceAccessPoint, determine if the value of the
- 1143 AccessContext property is "Default Gateway". If so, query the value of the AccessInfo property.

1144 9.4 Determine method used for current configuration

1145 A client can determine the method by which the IP configuration was assigned by querying the

1146 AddressOrigin property of the CIM_IPProtocolEndpoint instance.

1147 9.5 Determine whether DHCP then static is supported

1148 An implementation may support attempting to acquire its IP configuration through a DHCP client and
1149 defaulting to a static configuration if the client fails to acquire a configuration from a DHCP server. A client
1150 can determine whether this functionality is supported as follows:

- 1151 1) Find all instances of CIM_IPAssignmentSettingData (the parent class and not subclasses) that
1152 are associated with the CIM_IPProtocolEndpoint instance.
- 1153 2) For each instance of CIM_IPAssignmentSettingData:
 - 1154 a) Find all instances of CIM_DHCPSettingData that are associated through an instance of
1155 CIM_OrderedComponent.
 - 1156 b) Find all instances of CIM_StaticIPAssignmentSettingData that are associated through an
1157 instance of CIM_OrderedComponent.
- 1158 3) Determine if there is an instance of CIM_DHCPSettingData such that the value of the
1159 AssignedSequence property of the CIM_OrderedComponent that associates the instance of
1160 CIM_DHCPSettingData with the instance of CIM_IPAssignmentSettingData is less than the
1161 value of the AssignedSequence property of an instance of CIM_OrderedComponent that
1162 associates the CIM_StaticIPAssignmentSettingData with the instance of
1163 CIM_IPAssignmentSettingData. If so, DHCP then static is supported.

1164 9.6 View default configuration

1165 A client can view the default configuration for an IP interface as follows:

- 1166 1) Find all instances of CIM_ElementSettingData that associate an instance of
1167 CIM_IPAssignmentSettingData (the parent class and not subclasses) with the
1168 CIM_IPProtocolEndpoint instance.
- 1169 2) For each instance of CIM_ElementSettingData, see if the value of the IsDefault property is 1 (Is
1170 Default).

1171 9.7 Configure the interface to use DHCP

1172 An implementation may support attempting to acquire its IP configuration through a DHCP client. A client
1173 can determine whether this functionality is supported and configure the interface to use it as follows:

- 1174 1) Find all instances of CIM_IPAssignmentSettingData (the parent class and not subclasses) that
1175 are associated with the CIM_IPProtocolEndpoint instance.
- 1176 2) For each instance of CIM_IPAssignmentSettingData:
 - 1177 a) Find an instance of CIM_DHCPSettingData that is associated through an instance of
1178 CIM_OrderedComponent.
 - 1179 b) Verify that no instances of CIM_StaticIPAssignmentSettingData are associated with the
1180 instance of CIM_IPAssignmentSettingData.

1181 This instance of CIM_IPAssignmentSettingData represents a DHCP configuration.

- 1182 3) Find an instance of CIM_IPConfigurationService that is associated with the
1183 CIM_IPProtocolEndpoint instance through an instance of CIM_ServiceAffectsElement.
- 1184 4) Invoke the ApplySettingToIPProtocolEndpoint() method of the CIM_IPConfigurationService
1185 instance, specifying the instances of CIM_IPProtocolEndpoint and
1186 CIM_IPAssignmentSettingData.

1187 **9.8 Establish a static IP configuration for an interface**

1188 A client can manually assign an IP configuration to an interface as follows:

- 1189 1) Find all instances of CIM_IPAssignmentSettingData (the parent class and not subclasses) that
1190 are associated with the CIM_IPProtocolEndpoint instance.
- 1191 2) For each instance of CIM_IPAssignmentSettingData:
 - 1192 a) Find an instance of CIM_StaticIPAssignmentSettingData that is associated through an
1193 instance of CIM_OrderedComponent.
 - 1194 b) Verify that no other instances of CIM_StaticIPAssignmentSettingData or instances of
1195 CIM_DHCPSettingData are associated with the instance of CIM_IPAssignmentSettingData
1196 through an instance of CIM_OrderedComponent.
 - 1197 c) For the instance of CIM_ElementSettingData that associates the
1198 CIM_IPAssignmentSettingData instance with the instance of CIM_IPProtocolEndpoint,
1199 verify that the value of the IsDefault property is 2 (Is Not Default).

1200 This instance of CIM_IPAssignmentSettingData represents a modifiable, static configuration for
1201 the IP interface.
- 1202 3) Modify the properties of the CIM_StaticIPAssignmentSettingData instance to contain the
1203 appropriate configuration for the IP interface.
- 1204 4) Apply the pending configuration using the steps in 9.9 or 9.10.

1205 **9.9 Apply a pending configuration — Synchronously**

1206 Some implementations may support modifying the configuration of an IP interface without requiring a
1207 restart of the underlying network interface. If this behavior is supported by the implementation, then given
1208 an instance of CIM_IPProtocolEndpoint for which the configuration should be modified and an instance of
1209 CIM_IPAssignmentSettingData that represents the new configuration, a client can:

- 1210 1) Find an instance of CIM_IPConfigurationService that is associated with the
1211 CIM_IPProtocolEndpoint instance through an instance of CIM_ServiceAffectsElement.
- 1212 2) Invoke the ApplySettingToIPProtocolEndpoint() method of the CIM_IPConfigurationService,
1213 specifying the instances of CIM_IPProtocolEndpoint and CIM_IPAssignmentSettingData.

1214 **9.10 Apply a pending configuration — Upon restart**

1215 Some implementations may require that the IP interface be restarted in order for a new configuration that
1216 is bound to the interface to take effect. If an implementation requires that the IP interface be restarted,
1217 then given an instance of CIM_IPProtocolEndpoint for which the configuration should be modified and an
1218 instance of CIM_IPAssignmentSettingData that represents the new configuration, a client can:

- 1219 1) Find an instance of CIM_ElementSettingData that associates the CIM_IPAssignmentSettingData
1220 instance with the CIM_IPProtocolEndpoint instance.
- 1221 2) Set the IsNext property of the CIM_ElementSettingData instance to a value of 1 (Is Next).
- 1222 3) Invoke the RequestStateChange() method of the CIM_IPProtocolEndpoint instance, with a
1223 RequestedState of 11 (Reset).

1224 9.11 Determine whether DNS configuration was DHCP assigned

1225 Starting at the CIM_DNSProtocolEndpoint instance, a client can determine if any elements of the DNS
1226 configuration were assigned through DHCP as follows:

- 1227 1) Find the instance of CIM_IPProtocolEndpoint that is associated through an instance of
1228 CIM_SAPSAPDependency.
- 1229 2) Find the instance of CIM_DHCPProtocolEndpoint that is associated with the
1230 CIM_IPProtocolEndpoint instance through an instance of CIM_SAPSAPDependency.
- 1231 3) Query the EnabledState property of the CIM_DHCPProtocolEndpoint instance for the value 2
1232 (Enabled) to ensure that the DHCP client was used.
- 1233 4) Query the OptionsReceived property of the CIM_DHCPProtocolEndpoint instance to determine
1234 if one of the DNS-related options (8, 14, or 17) was received.

1235 9.12 Determine whether ElementName can be modified

1236 A client can determine whether it can modify the ElementName property of an instance of
1237 CIM_IPProtocolEndpoint as follows:

- 1238 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
1239 CIM_IPProtocolEndpoint instance.
- 1240 2) Query the value of the ElementNameEditSupported property of the
1241 CIM_EnabledLogicalElementCapabilities instance. If the value is TRUE, the client can modify
1242 the ElementName property of the target instance.

1243 9.13 Determine whether state management is supported

1244 A client can determine whether state management is supported for an instance of
1245 CIM_IPProtocolEndpoint as follows:

- 1246 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
1247 CIM_IPProtocolEndpoint instance.
- 1248 2) Query the value of the RequestedStatesSupported property. If at least one value is specified,
1249 state management is supported.

1250 10 CIM Elements

1251 Table 15 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
1252 implemented as described in Table 15. Clauses 7 (“Implementation”) and 8 (“Methods”) may impose
1253 additional requirements on these elements.

1254 **Table 15 – CIM Elements: IP Interface Profile**

| Element Name | Requirement | Description |
|---------------------------------------|-------------|--------------------------|
| Classes | | |
| CIM_BindsToLANEndpoint | Optional | See 7.6 and 10.1. |
| CIM_ElementCapabilities | Conditional | See 7.1.2 and 10.2. |
| CIM_EnabledLogicalElementCapabilities | Optional | See 7.1.2 and 10.5. |
| CIM_ElementSettingData | Conditional | See 7.4, 10.3, and 10.4. |
| CIM_HostedAccessPoint | Mandatory | See 10.6 and 10.7. |
| CIM_HostedService | Conditional | See 7.4.1 and 10.8. |

| Element Name | Requirement | Description |
|------------------------------------|-------------|--------------------|
| CIM_IPAssignmentSettingData | Conditional | See 7.4 and 10.9. |
| CIM_IPConfigurationService | Optional | See 7.4 and 10.10. |
| CIM_IPProtocolEndpoint | Mandatory | See 10.11. |
| CIM_OrderedComponent | Conditional | See 10.12. |
| CIM_RegisteredProfile | Mandatory | See 10.13. |
| CIM_RemoteAccessAvailableToElement | Conditional | See 10.14. |
| CIM_RemoteServiceAccessPoint | Optional | See 10.15. |
| CIM_ServiceAffectsElement | Conditional | See 7.4 and 10.16. |
| CIM_StaticIPAssignmentSettingData | Conditional | See 10.17. |
| Indications | | |
| None defined in this profile | | |

1255 10.1 CIM_BindsToLANEndpoint

1256 CIM_BindsToLANEndpoint relates the CIM_IPProtocolEndpoint instance with the CIM_LANEndpoint
 1257 instance on which it depends. Table 16 provides information about the properties of
 1258 CIM_BindsToLANEndpoint.

1259 **Table 16 – Class: CIM_BindsToLANEndpoint**

| Elements | Requirement | Description |
|------------|-------------|--|
| Antecedent | Mandatory | Key: This shall be a reference to an instance of CIM_LANEndpoint. Cardinality 0..1 |
| Dependent | Mandatory | Key: This shall be a reference to the Central Instance. Cardinality 1 |

1260 10.2 CIM_ElementCapabilities

1261 CIM_ElementCapabilities associates an instance of CIM_EnabledLogicalElementCapabilities with the
 1262 CIM_IPProtocolEndpoint instance. Table 17 provides information about the properties of
 1263 CIM_ElementCapabilities.

1264 **Table 17 – Class: CIM_ElementCapabilities**

| Elements | Requirement | Description |
|----------------|-------------|---|
| ManagedElement | Mandatory | Key: This shall be a reference to the Central Instance. Cardinality 1..* |
| Capabilities | Mandatory | Key: This shall be a reference to the instance of CIM_EnabledLogicalElementCapabilities. Cardinality 0..1 |

1265 **10.3 CIM_ElementSettingData — CIM_IPAssignmentSettingData Reference**

1266 CIM_ElementSettingData associates instances of CIM_IPAssignmentSettingData with the
 1267 CIM_IPProtocolEndpoint instance. Table 18 provides information about the properties of
 1268 CIM_ElementSettingData.

1269 **Table 18 – Class: CIM_ElementSettingData — CIM_IPAssignmentSettingData**

| Elements | Requirement | Description |
|----------------|-------------|---|
| ManagedElement | Mandatory | Key: This shall be a reference to the Central Instance. Cardinality 1..* |
| SettingData | Mandatory | Key: This shall be a reference to an instance of CIM_IPAssignmentSettingData. Cardinality * |
| IsDefault | Mandatory | Matches 1 (Is Default) or 2 (Is Not Default) |
| IsCurrent | Mandatory | Matches 1 (Is Current) or 2 (Is Not Current) |
| IsNext | Mandatory | Matches 1 (Is Next), 2 (Is Not Next), or 3 (Is Next For Single Use) |

1270 **10.4 CIM_ElementSettingData — CIM_StaticIPAssignmentSettingData Reference**

1271 CIM_ElementSettingData associates instances of CIM_StaticIPAssignmentSettingData with the
 1272 CIM_IPProtocolEndpoint instance. Table 19 provides information about the properties of
 1273 CIM_ElementSettingData.

1274 **Table 19 – Class: CIM_ElementSettingData — CIM_StaticIPAssignmentSettingData**

| Elements | Requirement | Description |
|----------------|-------------|---|
| ManagedElement | Mandatory | Key: This shall be a reference to the Central Instance. Cardinality 1..* |
| SettingData | Mandatory | Key: This shall be a reference to an instance of CIM_StaticIPAssignmentSettingData. Cardinality * |
| IsCurrent | Mandatory | Matches 1 (Is Current) or 2 (Is Not Current) |

1275 **10.5 CIM_EnabledLogicalElementCapabilities**

1276 CIM_EnabledLogicalElementCapabilities indicates support for managing the IP interface. Table 20
 1277 provides information about the properties of CIM_EnabledLogicalElementCapabilities.

1278 **Table 20 – Class: CIM_EnabledLogicalElementCapabilities**

| Elements | Requirement | Description |
|--------------------------|-------------|------------------------------|
| InstanceID | Mandatory | Key |
| RequestedStatesSupported | Mandatory | See 7.1.2.1.1 and 7.1.3.1.1. |
| ElementNameEditSupported | Mandatory | See 7.1.4.1.1 and 7.1.5.1.1. |
| MaxElementNameLen | Conditional | See 7.1.4.1.2 and 7.1.5.1.2. |

1279 10.6 CIM_HostedAccessPoint — CIM_RemoteServiceAccessPoint Reference

1280 An instance of CIM_HostedAccessPoint Association between an instance of CIM_ProtocolEndpoint and
 1281 CIM_RemoteServiceAccessPoint shall only be instantiated if CIM_RemoteServiceAccessPoint is
 1282 supported.

1283 CIM_HostedAccessPoint relates the CIM_RemoteServiceAccessPoint instance that represents the
 1284 default gateway with its scoping CIM_ComputerSystem instance. Table 21 provides information about the
 1285 properties of CIM_HostedAccessPoint.

1286 **Table 21 – Class: CIM_HostedAccessPoint — CIM_RemoteServiceAccessPoint**

| Elements | Requirement | Description |
|------------|-------------|--|
| Antecedent | Mandatory | Key: This shall be a reference to the Scoping Instance. Cardinality 1 |
| Dependent | Mandatory | Key: This shall be a reference to an instance of CIM_RemoteServiceAccessPoint. Cardinality * |

1287 10.7 CIM_HostedAccessPoint — CIM_IPProtocolEndpoint Reference

1288 CIM_HostedAccessPoint relates the Central Instance with its Scoping Instance. Table 22 provides
 1289 information about the properties of CIM_HostedAccessPoint.

1290 **Table 22 – Class: CIM_HostedAccessPoint — CIM_IPProtocolEndpoint**

| Elements | Requirement | Description |
|------------|-------------|---|
| Antecedent | Mandatory | Key: This shall be a reference to the Scoping Instance. Cardinality 1 |
| Dependent | Mandatory | Key: This shall be a reference to an instance of CIM_IPProtocolEndpoint. Cardinality 1..* |

1291 10.8 CIM_HostedService

1292 CIM_HostedService relates the CIM_IPConfigurationService instance to its scoping
 1293 CIM_ComputerSystem instance. Table 23 provides information about the properties of
 1294 CIM_HostedService.

1295 **Table 23 – Class: CIM_HostedService**

| Elements | Requirement | Description |
|------------|-------------|--|
| Antecedent | Mandatory | Key: This shall be a reference to the Central Instance. Cardinality 1 |
| Dependent | Mandatory | Key: This shall be a reference to an instance of CIM_IPConfigurationService. Cardinality * |

1296 **10.9 CIM_IPAssignmentSettingData**

1297 CIM_IPAssignmentSettingData is the aggregation point for the SettingData instances that define a
 1298 configuration that can be applied to an IP interface. Table 24 provides information about the properties of
 1299 CIM_IPAssignmentSettingData.

1300 **Table 24 – Class: CIM_IPAssignmentSettingData**

| Elements | Requirement | Description |
|---------------|-------------|----------------------------|
| InstanceID | Mandatory | Key |
| AddressOrigin | Mandatory | Matches 2 (Not Applicable) |
| ElementName | Mandatory | Pattern ".*" |

1301 **10.10 CIM_IPConfigurationService**

1302 CIM_IPConfigurationService represents the ability to configure an IP interface. Table 25 provides
 1303 information about the properties of CIM_IPConfigurationService.

1304 **Table 25 – Class: CIM_IPConfigurationService**

| Elements | Requirement | Description |
|-------------------------------------|-------------|--------------|
| SystemCreationClassName | Mandatory | Key |
| CreationClassName | Mandatory | Key |
| SystemName | Mandatory | Key |
| Name | Mandatory | Key |
| ElementName | Mandatory | Pattern ".*" |
| ApplySettingToIPProtocolEndpoint() | Optional | See 8.1.1.1. |

1305 **10.11 CIM_IPProtocolEndpoint**

1306 CIM_IPProtocolEndpoint represents an IP interface that is associated with an Ethernet interface. Table 26
 1307 provides information about the properties of CIM_IPProtocolEndpoint.

1308 **Table 26 – Class: CIM_IPProtocolEndpoint**

| Elements | Requirement | Description |
|-------------------------|-------------|--------------------------|
| SystemCreationClassName | Mandatory | Key |
| CreationClassName | Mandatory | Key |
| SystemName | Mandatory | Key |
| Name | Mandatory | Key |
| NameFormat | Mandatory | Pattern ".*" |
| ProtocolIFType | Mandatory | See 7.1.1.2. |
| RequestedState | Mandatory | See 7.1.2.2 and 7.1.3.2. |
| EnabledState | Mandatory | See 7.1.2.3 and 7.1.3.3. |
| ElementName | Mandatory | Pattern ".*" |
| RequestStateChange() | Conditional | See 8.1. |

| Elements | Requirement | Description |
|------------------------|-------------|--|
| IPv4Address | Conditional | See 7.1.1.2. |
| SubnetMask | Conditional | See 7.1.1.2 and 7.1.1.4. |
| AddressOrigin | Mandatory | See 7.1.1.1. |
| IPv6Address | Conditional | See 7.1.1.2 and 7.1.1.5 – EXPERIMENTAL |
| IPv6AddressType | Conditional | See 7.1.1.2 – EXPERIMENTAL |
| IPv6SubnetPrefixLength | Conditional | See 7.1.1.2 – EXPERIMENTAL |

1309 10.12 CIM_OrderedComponent

1310 CIM_OrderedComponent associates an instance of CIM_IPAssignmentSettingData to the instances of
 1311 CIM_StaticIPAssignmentSettingData, CIM_DHCPSettingData, CIM_DNSSettingData, and
 1312 CIM_DNSGeneralSettingData that compose a configuration. Table 27 provides information about the
 1313 properties of CIM_OrderedComponent.

1314 **Table 27 – Class: CIM_OrderedComponent**

| Elements | Requirement | Description |
|------------------|-------------|--------------------------|
| GroupComponent | Mandatory | Key: See 7.4.3.1. |
| PartComponent | Mandatory | Key: See 7.4.3.2. |
| AssignedSequence | Mandatory | See 7.4.3.3. |

1315 10.13 CIM_RegisteredProfile

1316 CIM_RegisteredProfile identifies the *IP Interface Profile* in order for a client to determine whether an
 1317 instance of CIM_IPProtocolEndpoint is conformant with this profile. The CIM_RegisteredProfile class is
 1318 defined by the *Profile Registration Profile* ([DSP1033](#)). With the exception of the mandatory values
 1319 specified for the properties in Table 28, the behavior of the CIM_RegisteredProfile instance is in
 1320 accordance with the *Profile Registration Profile* ([DSP1033](#)).

1321 **Table 28 – Class: CIM_RegisteredProfile**

| Elements | Requirement | Description |
|------------------------|-------------|---|
| RegisteredName | Mandatory | This property shall have a value of "IP Interface". |
| RegisteredVersion | Mandatory | This property shall have a value of "1.0.2". |
| RegisteredOrganization | Mandatory | This property shall have a value of "DMTF". |

1322 NOTE Previous versions of this document included the suffix "Profile" for the RegisteredName value. If
 1323 implementations querying for the RegisteredName value find the suffix "Profile", they should ignore the suffix, with
 1324 any surrounding white spaces, before any comparison is done with the value as specified in this document.

1325 **10.14 CIM_RemoteAccessAvailableToElement**

1326 CIM_RemoteAccessAvailableToElement associates the CIM_IPProtocolEndpoint instance with the
 1327 CIM_RemoteServiceAccessPoint instance that represents the network gateway. Table 29 provides
 1328 information about the properties of CIM_RemoteAccessAvailableToElement.

1329 **Table 29 – Class: CIM_RemoteAccessAvailableToElement**

| Elements | Requirement | Description |
|---------------|-------------|--------------------------|
| Antecedent | Mandatory | Key: See 7.1.6.2. |
| Dependent | Mandatory | Key: See 7.1.6.3. |
| OrderOfAccess | Mandatory | See 7.1.6.4. |

1330 **10.15 CIM_RemoteServiceAccessPoint**

1331 CIM_RemoteServiceAccessPoint represents the managed system's view of the default gateway. Table
 1332 30 provides information about the properties of CIM_RemoteServiceAccessPoint.

1333 **Table 30 – Class: CIM_RemoteServiceAccessPoint**

| Elements | Requirement | Description |
|-------------------------|-------------|-----------------------------|
| SystemCreationClassName | Mandatory | Key |
| CreationClassName | Mandatory | Key |
| SystemName | Mandatory | Key |
| Name | Mandatory | Key |
| AccessContext | Mandatory | Matches 2 (Default Gateway) |
| AccessInfo | Mandatory | See 7.1.6.1. |
| InfoFormat | Mandatory | Matches 3 (IPv4 Address) |
| ElementName | Mandatory | Pattern ".*" |

1334 **10.16 CIM_ServiceAffectsElement**

1335 CIM_ServiceAffectsElement associates an instance of CIM_IPConfigurationService with an instance of
 1336 CIM_IPProtocolEndpoint that the service is able to configure. Table 31 provides information about the
 1337 properties of CIM_ServiceAffectsElement.

1338 **Table 31 – Class: CIM_ServiceAffectsElement**

| Elements | Requirement | Description |
|------------------|-------------|---|
| AffectingElement | Mandatory | Key: This shall be a reference to the instance of CIM_IPConfigurationService. Cardinality * |
| AffectedElement | Mandatory | Key: This shall be a reference to the Central Instance. Cardinality 1..* |
| ElementEffects | Mandatory | Matches 5 (Manages) |

1339 **10.17 CIM_StaticIPAssignmentSettingData**

1340 CIM_StaticIPAssignmentSettingData represents a static configuration that can be applied to an instance
 1341 of CIM_IPProtocolEndpoint. Table 32 provides information about the properties of
 1342 CIM_StaticIPAssignmentSettingData.

1343 **Table 32 – Class: CIM_StaticIPAssignmentSettingData**

| Elements | Requirement | Description |
|------------------------|-------------|--------------------|
| InstanceID | Mandatory | Key |
| AddressOrigin | Mandatory | Matches 3 (Static) |
| ElementName | Mandatory | Pattern ".*" |
| IPv4Address | Mandatory | |
| SubnetMask | Mandatory | |
| GatewayIPv4Address | Conditional | See 7.5.3.1. |
| IPv6Address | Optional | EXPERIMENTAL |
| IPv6AddressType | Optional | EXPERIMENTAL |
| IPv6SubnetPrefixLength | Optional | EXPERIMENTAL |
| GatewayIPv6Address | Optional | EXPERIMENTAL |

1344

**ANNEX A
(informative)****Change log**1345
1346
1347
1348

| Version | Date | Description |
|---------|------------|--|
| 1.0.0a | 2006-07-11 | Preliminary Standard |
| 1.0.0 | 2008-07-27 | Final Standard & addition of IPv6 support as Experimental |
| 1.0.1 | 2009-09-24 | Errata 1.0.1 |
| 1.0.1 | 2010-07-29 | Version 1.0.1 of the Final Standard formatted for DMTF Standard release |
| 1.0.2 | 2012-02-23 | Errata 1.0.2 Section 8 - Removed CIM_SystemDevice. Section 9 - Correction in association for CIM_RemoteServiceAccessPoint. Section 10 - Spelling for CIM_ServiceAffectsElement.ElementEffects |

1349
1350