1

5 # IP Interface Profile

10   Copyright Notice

11   Copyright © 2008, 2009, 2010, 2012, 2018 DMTF. All rights reserved.

33

34 CONTENTS

101 # Figures

118 # Tables

153 # Foreword

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

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

159 ## Acknowledgments

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

161 - RadhaKrishna Dasari – Dell Inc.

162 - Jon Hass – Dell Inc.

163 - Jeff Hilland – Hewlett Packard Enterprise

164 - John Leung – Intel Corporation

165 - Aaron Merkin – IBM

166 - Khachatur Papanyan – Dell Inc.

167 - Sivakumar Sathappan – Advanced Micro Devices

168 - Hemal Shah – Broadcom Inc.

169 - Christina Shaw – Hewlett Packard Enterprise

170 - Enoch Suen – Dell Inc.

171 - Satheesh Thomas – AMI

172 - Perry Vincent – Intel Corporation

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

181

182   # IP Interface Profile

183   1   Scope

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

188   2   Normative references

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

193   DMTF DSP0004, *CIM Infrastructure Specification 2.6*,
194   https://www.dmtf.org/sites/default/files/standards/documents/DSP0004_2.6.pdf

195   DMTF DSP0200, *CIM Operations over HTTP 1.3,*
196   https://www.dmtf.org/sites/default/files/standards/documents/DSP0200_1.3.pdf

197   DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
198   https://www.dmtf.org/sites/default/files/standards/documents/DSP1001_1.0.pdf

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

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

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

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

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

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

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

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

212   3   Terms and definitions

213   For the purposes of this document, the terms and definitions in DSP1033 and DSP1001 as well as the
214   following definitions apply.

215   **3.1**
216   **can**
217   used for statements of possibility and capability, whether material, physical, or causal

218 **3.2**
219 **cannot**
220 used for statements of possibility and capability, whether material, physical, or causal

221 **3.3**
222 **conditional**
223 indicates requirements to be followed strictly to conform to the document when the specified conditions
224 are met

225 **3.4**
226 **mandatory**
227 indicates requirements to be followed strictly to conform to the document and from which no deviation is
228 permitted

229 **3.5**
230 **may**
231 indicates a course of action permissible within the limits of the document

232 **3.6**
233 **need not**
234 indicates a course of action permissible within the limits of the document

235 **3.7**
236 **optional**
237 indicates a course of action permissible within the limits of the document

238 **3.8**
239 **pending configuration**
240 the configuration that will be applied to an IP interface the next time the interface accepts a configuration

241 **3.9**
242 **referencing profile**
243 indicates a profile that owns the definition of this class and can include a reference to this profile in its
244 "Referenced Profiles" table

245 **3.10**
246 **shall**
247 indicates requirements to be followed strictly to conform to the document and from which no deviation is
248 permitted

249 **3.11**
250 **shall not**
251 indicates requirements to be followed strictly to conform to the document and from which no deviation is
252 permitted

253 **3.12**
254 **should**
255 indicates that among several possibilities, one is recommended as particularly suitable, without
256 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

257 **3.13**
258 **should not**
259 indicates that a certain possibility or course of action is deprecated but not prohibited

260   **3.14**
261   **unspecified**
262   indicates that this profile does not define any constraints for the referenced CIM element or operation

263   **4   Symbols and abbreviated terms**

264   The following abbreviations are used in this document.

265   **4.1**
266   **DHCP**
267   Dynamic Host Configuration Protocol

268   **4.2**
269   **DNS**
270   Domain Name System

271   **4.3**
272   **IP**
273   Internet Protocol

274   **5   Synopsis**

275   **Profile Name:** IP Interface

276   **Version:** 1.1.1

277   **Organization:** DMTF

278   **CIM Schema Version:** 2.26

279   **Central Class:** CIM_IPProtocolEndpoint

280   **Scoping Class:** CIM_ComputerSystem

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

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

286   **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.

287 # 6 Description

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

290 Figure 1 represents the class schema for the *IP Interface Profile.* For simplicity, the CIM_ prefix has been
291 removed from the names of the classes. Note that this class diagram is meant to be used in conjunction
292 with the class diagrams from the *DHCP Client Profile* ([DSP1037](#)) and the *DNS Client Profile* ([DSP1038](#)).
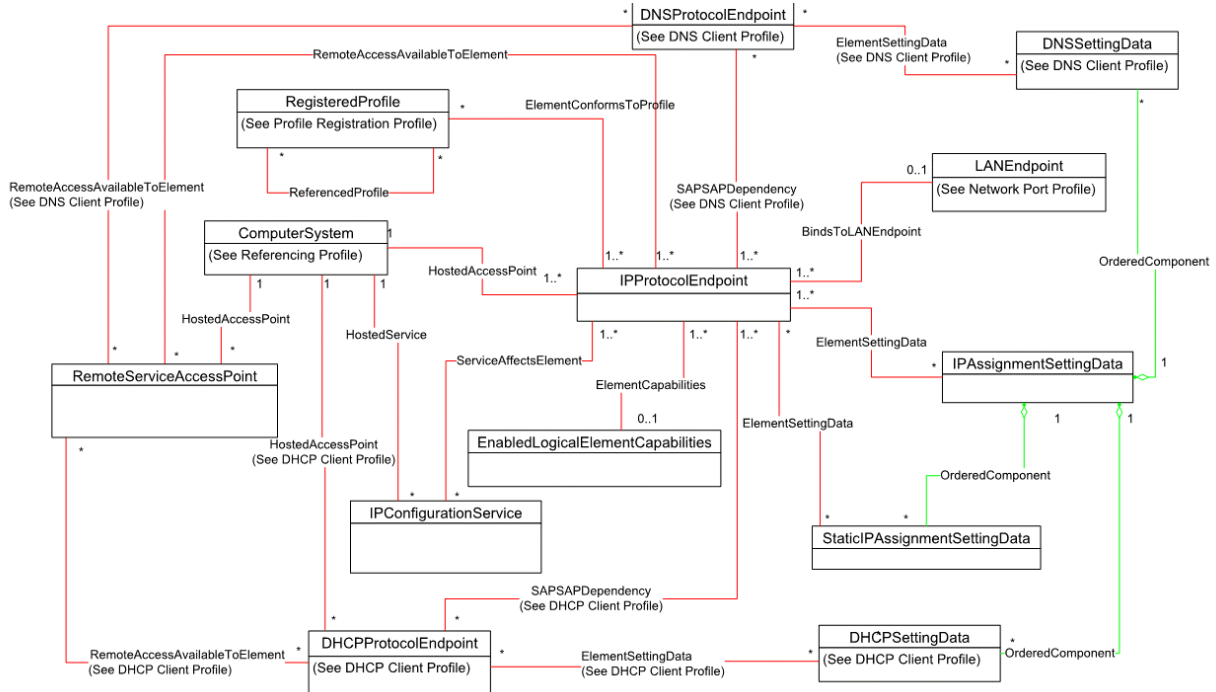
293

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

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

298 • IPv4 interface (optionally associated with a network interface)

299 • optional relationship with a DNS client

300 • optional relationship with a DHCP client

301 • current and pending configurations

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

303 • modeling of the network gateway

304 • modeling of TCP/UDP ports

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

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

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

6.1 Pending and alternate configuration management

315 Pending configurations, which are associated with the IP interface and could be applied in the future, are
316 represented by instances of CIM_IPAssignmentSettingData and its subclasses. Each pending
317 configuration can include multiple settings that will be applied to the different elements of the endpoint
318 configuration. Settings for a particular element of the configuration are represented with the appropriate
319 subclass of CIM_IPAssignmentSettingData and aggregated into one or more instances of
320 CIM_IPAssignmentSettingData that represent the configuration.

321 The management of DNS and DHCP clients as part of an alternate configuration is handled differently for
322 the two clients. DHCP and static IP configuration management are generally treated as alternatives to
323 each other. For the basic configuration of an IP interface, the information is assigned either statically or
324 through DHCP. DNS configuration occurs differently. When DNS and static configuration occur together,
325 there is no overlap. Thus the DNS settings that are part of the configuration are applied to the DNS client.
326 When DHCP and DNS settings are used together, portions of the DNS configuration can potentially be
327 assigned through DHCP.

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

7 Implementation

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

7.1 Basic IP configuration

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

7.1.1 CIM_IPProtocolEndpoint

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

7.1.1.1 CIM_IPProtocolEndpoint.AddressOrigin

343 The AddressOrigin property indicates the configuration method that resulted in the configuration being
344 assigned to the CIM_IPProtocolEndpoint.

7.1.1.1.1 AddressOrigin — Static

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

350 **7.1.1.1.2 AddressOrigin — DHCP**

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

354 **7.1.1.2 CIM_IPProtocolEndpoint.ProtocolIFType**

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

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

357 The value of CIM_IPProtocolEndpoint.ProtocolIFType shall be 4096, 4097, or 4098.

358 If the value is 4097 (Ipv6) the IPv6Address and IPv6SubnetPrefixLength properties shall be implemented
359 and IPv6AddressType may be implemented.

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

363 **7.1.1.3 CIM_IPProtocolEndpoint.IPv4Address**

364 If the value of CIM_IPProtocolEndpoint.ProtocolIFType is 4096 (IPv4), the IPv4Address property shall
365 indicate the current IPv4 address assigned to this IP endpoint. The value of the property shall be
366 specified in dotted decimal notation as defined in IETF RFC 1208. A value of 0.0.0.0 shall indicate that a
367 valid IP address is not assigned to this IP endpoint.

368 **7.1.1.4 CIM_IPProtocolEndpoint.SubnetMask**

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

372 **7.1.1.5 CIM_IPProtocolEndpoint.IPv6Address**

373 If the value of CIM_IPProtocolEndpoint.ProtocolIFType is 4097 (IPv6) or 4098 (IPv4/IPv6), the
374 IPv6Address property shall indicate the current IPv6 address assigned to this IP endpoint. The value of
375 the property shall be specified in the notation specified in IETF RFC 4291, section 2.2.

376 **7.1.2 IP interface state management is supported — conditional**

377 When management of the state of an IP interface is supported, exactly one instance of
378 CIM_EnabledLogicalElementCapabilities shall be associated with the CIM_IPProtocolEndpoint instance
379 through an instance of CIM_ElementCapabilities. The existence of the CIM_ElementCapabilities instance
380 is conditional on the existence of the CIM_EnabledLogicalElementCapabilities instance.

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

383 **7.1.2.1 CIM_EnabledLogicalElementCapabilities**

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

386 **7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported**

387 The RequestedStatesSupported property may contain zero or more of the following values: 2 (Enabled),
388 3 (Disabled), or 11 (Reset).

389 **7.1.2.2 CIM_IPProtocolEndpoint.RequestedState**

390 When the CIM_IPProtocolEndpoint.RequestStateChange( ) method is successfully invoked, the value of
391 the RequestedState property shall be the value of the RequestedState parameter. If the method is not
392 successfully invoked, the value of the RequestedState property is indeterminate.

393 The CIM_IPProtocolEndpoint.RequestedState property shall have one of the values specified in the
394 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property or a value of 5 (No
395 Change).

396 **7.1.2.3 CIM_IPProtocolEndpoint.EnabledState**

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

400 If the method is not completed successfully, the value of the EnabledState property is indeterminate.

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

403 **7.1.3 IP interface state management is not supported**

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

406 **7.1.3.1 CIM_EnabledLogicalElementCapabilities**

407 When state management is not supported, exactly one instance of
408 CIM_EnabledLogicalElementCapabilities may be associated with the CIM_IPProtocolEndpoint instance
409 through an instance of CIM_ElementCapabilities.

410 **7.1.3.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported**

411 The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall not contain any
412 values.

413 **7.1.3.2 CIM_IPProtocolEndpoint.RequestedState**

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

415 **7.1.3.3 CIM_IPProtocolEndpoint.EnabledState**

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

418 **7.1.4 Modifying ElementName is supported — conditional**

419 The CIM_IPProtocolEndpoint.ElementName property may support being modified by the ModifyInstance
420 operation. See 8.10.1.1.

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

423 **7.1.4.1 CIM_EnabledLogicalElementCapabilities**

424 An instance of CIM_EnabledLogicalElementCapabilities shall be associated with the
425 CIM_IPProtocolEndpoint instance through an instance of CIM_ElementCapabilities.

426 **7.1.4.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported**

427 The ElementNameEditSupported property shall have a value of TRUE.

428 **7.1.4.1.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen**

429 The MaxElementNameLen property shall be implemented.

430 ## 7.1.5 Modifying ElementName is not supported

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

434 **7.1.5.1 CIM_EnabledLogicalElementCapabilities**

435 An instance of CIM_EnabledLogicalElementCapabilities may be associated with the
436 CIM_IPProtocolEndpoint instance through an instance of CIM_ElementCapabilities.

437 **7.1.5.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported**

438 The ElementNameEditSupported property shall have a value of FALSE.

439 **7.1.5.1.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen**

440 The MaxElementNameLen property may be implemented. The MaxElementNameLen property is
441 irrelevant in this context.

442 ## 7.1.6 Default gateway

443 An IP interface can be configured with the address of a network gateway. Modeling of the default gateway
444 is optional. When the IP interface is configured with the address of a default gateway, an instance of
445 CIM_RemoteServiceAccessPoint shall represent the default gateway. The instance of
446 CIM_RemoteServiceAccessPoint shall be associated with the instance of CIM_IPProtocolEndpoint
447 through an instance of CIM_RemoteAccessAvailableToElement. An instance of
448 CIM_RemoteServiceAccessPoint may represent the default gateway even when a valid default gateway
449 has not been configured for the IP interface. It can be more convenient for an implementation to always
450 instantiate the instance of CIM_RemoteServiceAccessPoint even if a default gateway has not been
451 assigned to the IP interface rather than conditionally provide the relevant instances. For IPv4, this will
452 result in a single instance of CIM_RemoteServiceAccessPoint associated with the instance of
453 CIM_IPProtocolEndpoint.

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

458 **7.1.6.1 CIM_RemoteServiceAccessPoint.AccessInfo**

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

463 If the associated value of CIM_IPProtocolEndpoint.ProtocolIFType = 4097 (IPv6), then the value of the
464 AccessInfo property shall be the IPv6 address of the default gateway. The value shall be specified in the
465 IPv6 notation as defined in IETF RFC 4291. An unspecified address, which has the value of ":::/128", shall
466 indicate that a default gateway has not been assigned to the associated IP interface.

467 If the associated value of CIM_IPProtocolEndpoint.ProtocolIFType = 4098 (IPv4/IPv6), then the value of
468 the AccessInfo property shall be the IPv6 address of the default gateway. The value shall be specified in
469 the IPv6 notation as defined in IETF RFC 4291. An Unspecified Address, which has the value of ":/128",
470 shall indicate that a default gateway has not been assigned to the associated IP interface.

471 **7.1.6.2   CIM_RemoteAccessAvailableToElement.Antecedent**

472 The value of the Antecedent reference shall be the instance of CIM_RemoteServiceAccessPoint.
473 Cardinality *.

474 **7.1.6.3   CIM_RemoteAccessAvailableToElement.Dependent**

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

476 **7.1.6.4   CIM_RemoteAccessAvailableToElement.OrderOfAccess**

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

7.2   DHCP client is supported

480 When a DHCP client is supported for the IP interface, the *DHCP Client Profile* shall be supported. This
481 behavior is optional.

7.3   DNS client is supported

483 When a DNS client is supported for the IP interface, the *DNS Client Profile* shall be supported. This
484 behavior is optional.

7.4   Managing alternate configurations — optional

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

7.4.1   Configuration management is supported

490 The CIM_IPConfigurationService class provides management of alternate configurations and support for
491 configuring additional interfaces. When an implementation supports management of alternate
492 configurations, exactly one instance of CIM_IPConfigurationService shall be associated with the Central
493 Instance of the profile through an instance of CIM_ServiceAffectsElement. The existence of the
494 CIM_ServiceAffectsElement association is conditional on the existence of the
495 CIM_IPConfigurationService instance.

496 The CIM_IPConfigurationService instance shall be associated with a CIM_ComputerSystem instance
497 through an instance of CIM_HostedService. The existence of the CIM_HostedService association is
498 conditional on the existence of the CIM_IPConfigurationService instance.

7.4.2   Representing an alternate configuration using CIM_IPAssignmentSettingData

500 Each instance of CIM_IPAssignmentSettingData shall represent a possible configuration for an IP
501 interface. The detailed settings for the IP interface shall be contained in the instances of subclasses of
502 CIM_IPAssignmentSettingData, which are associated with the instance of CIM_IPAssignmentSettingData
503 through instances of CIM_OrderedComponent.

504 The existence of one or more instances of CIM_IPAssignmentSettingData is conditional on the existence
505 of the CIM_IPConfigurationService instance. The existence of one or more instances of

506   CIM_ElementSettingData is conditional on the existence of one or more instances of
507   CIM_IPAssignmentSettingData.

508   7.4.2.1   Associating an alternate configuration with an IP Interface

509   The instance of CIM_IPAssignmentSettingData shall be associated with the instance of
510   CIM_IPProtocolEndpoint through an instance of CIM_ElementSettingData.

511   7.4.2.1.1   CIM_ElementSettingData.IsCurrent

512   When an instance of CIM_ElementSettingData associates an instance of CIM_IPAssignmentSettingData
513   with an instance of CIM_IPProtocolEndpoint, the CIM_ElementSettingData.IsCurrent property shall have
514   a value of 1 (Is Current) when the configuration represented by the referenced instance of
515   CIM_IPAssignmentSettingData is the last configuration applied to the IP interface represented by the
516   referenced instance of CIM_IPProtocolEndpoint.

517   When an instance of CIM_ElementSettingData associates an instance of CIM_IPAssignmentSettingData
518   with an instance of CIM_IPProtocolEndpoint, the CIM_ElementSettingData.IsCurrent property shall have
519   a value of 2 (Is Not Current) when the configuration represented by the referenced instance of
520   CIM_IPAssignmentSettingData is not the last configuration applied to the IP interface represented by the
521   referenced instance of CIM_IPProtocolEndpoint.

522   7.4.3   Associating settings using CIM_OrderedComponent

523   Instances of the subclasses of CIM_IPAssignmentSettingData contain the details of the IP configuration.
524   The CIM_OrderedComponent association aggregates these instances into instances of
525   CIM_IPAssignmentSettingData. An instance of CIM_IPAssignmentSettingData will have one or more
526   instances of its subclasses associated with it through an instance of CIM_OrderedComponent. An
527   instance of a subclass of CIM_IPAssignmentSettingData will be associated with one or more instances of
528   CIM_IPAssignmentSettingData.

529   7.4.3.1   CIM_OrderedComponent.GroupComponent

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

532   7.4.3.2   CIM_OrderedComponent.PartComponent

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

535   7.4.3.3   Interpretation of CIM_OrderedComponent.AssignedSequence

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

539   7.4.3.3.1   Use of 0 (zero)

540   When the CIM_OrderedComponent.AssignedSequence property has a value of 0 (zero), the instance of
541   CIM_SettingData referenced by the CIM_OrderedComponent.PartComponent property shall not be
542   applied when the configuration represented by the CIM_IPAssignmentSettingData instance that is the
543   value of the CIM_OrderedComponent.GroupComponent property is applied. The
544   CIM_OrderedComponent.AssignedSequence property may have the value 0 (zero) when the instance of
545   CIM_OrderedComponent references an instance of CIM_DNSSettingData or
546   CIM_DNSGeneralSettingData. The CIM_OrderedComponent.AssignedSequence property shall not have
547   the value 0 (zero) when the instance of CIM_OrderedComponent does not reference an instance of
548   CIM_DNSSettingData or CIM_DNSGeneralSettingData.

549   **7.4.3.3.2   Discreteness**

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

553   **7.4.4   Alternate static configuration**

554   When an implementation supports the manual assignment of an IP configuration to the IP endpoint, an
555   instance of CIM_StaticIPAssignmentSettingData shall be associated with the CIM_IPProtocolEndpoint
556   through an instance of CIM_ElementSettingData.

557   CIM_ElementSettingData is conditional on the existence of one or more instances of
558   CIM_StaticIPAssignmentSettingData

559   This instance of CIM_StaticIPAssignmentSettingData shall be associated with at least one instance of
560   CIM_IPAssignmentSettingData through an instance of CIM_OrderedComponent. When the aggregating
561   IP configuration has been applied to the IP interface and the IP interface is using the settings contained in
562   the instance of CIM_StaticIPAssignmentSettingData, the IsCurrent property of the
563   CIM_ElementSettingData instance has the value 1 (Is Current). Otherwise, the
564   CIM_ElementSettingData.IsCurrent property shall have the value 2 (Is Not Current).

565   **7.4.5   Alternate DHCP configuration**

566   When an alternate configuration includes the configuration of the DHCP client, the implementation will
567   follow the rules for representing a pending DHCP configuration defined in the *DHCP Client Profile*.

568   **7.4.6   DNS client alternate configuration**

569   When an alternate configuration includes the configuration of the DNS client, the implementation will
570   follow the rules for representing a pending DNS configuration defined in the *DNS Client Profile*.

571   **7.4.7   Relationship between DHCP and DNS configuration**

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

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

579   **7.4.7.1   Relationship between DHCP options and the DNS configuration**

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

583   DHCPPE – the instance of CIM_DHCPProtocolEndpoint that represents the DHCP client for an IP
584   interface

585   DNSPE – the instance of CIM_DNSProtocolEndpoint that represents the DNS client that is associated
586   through an instance of CIM_SAPSAPDependency with the same instance of CIM_IPProtocolEndpoint
587   with which the DHCPPE is associated through an instance of CIM_SAPSAPDependency

588   DNS Pending – the instance of CIM_DNSSettingData that is associated through an instance of
589   CIM_OrderedComponent with the instance of CIM_IPAssignmentSettingData that is being applied to the
590   CIM_IPProtocolEndpoint instance

591 DHCP Pending – the instance of CIM_DHCPSettingData that is associated through an instance of
592 CIM_OrderedComponent with the instance of CIM_IPAssignmentSettingData that is being applied to the
593 CIM_IPProtocolEndpoint instance

594 The following requirements shall be met when the *DHCP Client Profile* and the *DNS Client Profile* are
595 implemented:

596 • When the OptionsReceived property of the DHCPPE instance and the DHCPOptionsToUse
597 property of the DNSPE instance both contain the value 8 (Domain Name Server), the DNS
598 Servers instrumented in accordance with the "DNS Server Representation" clause of the *DNS*
599 *Client Profile* shall identify the DNS server addresses specified by the DHCP server as the data
600 for the Domain Name Server DHCP option.

601 • When the OptionsReceived property of the DHCPPE instance and the DHCPOptionsToUse
602 property of the DNSPE instance both contain the value 14 (Host Name), the value of the
603 Hostname property of the DNSPE instance shall be the hostname specified by the DHCP server
604 as the data for the Host Name DHCP option.

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

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

613 ### 7.4.8   Representing a pending configuration

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

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

620 If an instance of CIM_IPAssignmentSettingData is associated with the Central Instance through an
621 instance of CIM_ElementSettingData whose IsNext property has the value 3 (Is Next For Single Use), this
622 instance of CIM_IPAssignmentSettingData shall represent the pending configuration. If no instance of
623 CIM_IPAssignmentSettingData is associated with the Central Instance through an instance of
624 CIM_ElementSettingData whose IsNext property has the value 3 (Is Next For Single Use), the instance of
625 CIM_IPAssignmentSettingData that is associated with the Central Instance through an instance of
626 CIM_ElementSettingData whose IsNext property has the value 1 (Is Next) shall represent the pending
627 configuration.

628 ## 7.5   Applying an alternate configuration

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

7.5.1 Applying the pending configuration upon transition to enabled

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

7.5.2 Determining the target CIM_ProtocolEndpoint instance

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

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

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

    1)   The CIM_IPProtocolEndpoint instance shall be associated with the CIM_StaticIPAssignmentSettingData instance through an instance of CIM_ElementSettingData.

    2)   The CIM_IPProtocolEndpoint instance shall be the CIM_IPProtocolEndpoint instance to which the aggregating CIM_IPAssignmentSettingData is being applied.

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

    1)   The CIM_DHCPProtocolEndpoint instance shall be associated with the CIM_DHCPSettingData instance through an instance of CIM_ElementSettingData.

    2)   The CIM_DHCPProtocolEndpoint instance shall be associated through an instance of CIM_SAPSAPDependency with the CIM_IPProtocolEndpoint instance to which the aggregating CIM_IPAssignmentSettingData is being applied.

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

    1)   The CIM_DNSProtocolEndpoint instance shall be associated with the CIM_DNSSettingData instance through an instance of CIM_ElementSettingData.

    2)   The CIM_DNSProtocolEndpoint instance shall be associated through an instance of CIM_SAPSAPDependency with the CIM_IPProtocolEndpoint instance to which the aggregating CIM_IPAssignmentSettingData is being applied.

7.5.3 Applying static IP settings

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

680 **7.5.3.1 CIM_StaticIPAssignmentSettingData.GatewayIPv4Address**

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

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

687 **7.5.3.2 CIM_StaticIPAssignmentSettingData.GatewayIPv6Address**

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

691 **7.5.3.3 Successful application of settings**

692 An instance of CIM_StaticIPAssignmentSettingData shall be considered successfully applied when the
693 properties of the associated instance of CIM_IPProtocolEndpoint to which the instance of
694 CIM_StaticIPAssignmentSettingData has been applied have the values of the relevant properties of the
695 CIM_StaticIPAssignmentSettingData instance.

696 ## 7.5.4 Applying DHCP settings

697 When a pending configuration includes the configuration of the DHCP client, the DHCP configuration is
698 applied as defined in the *DHCP Client Profile*.

699 ## 7.5.5 Applying DNS settings

700 When a pending configuration includes DNS client configuration, the DNS configuration is applied as
701 defined in the *DNS Client Profile*. When the AssignedSequence property of the CIM_OrderedComponent
702 association that references an instance of CIM_DNSSettingData or CIM_DNSGeneralSettingData has a
703 non-zero value, the referenced instance of CIM_DNSSettingData or CIM_DNSGeneralSettingData shall
704 be applied, regardless of whether the application of a preceding CIM_SettingData instance was
705 successful.

706 ## 7.5.6 Resolving overlapped settings

707 When more than one instance of CIM_StaticIPAssignmentSettingData or CIM_DHCPSettingData is
708 associated with the same instance of CIM_IPAssignmentSettingData, each CIM_SettingData instance
709 shall be applied in order (as described in 7.4.3.3) until the implementation determines that the resultant
710 configuration is valid. The amount of time an implementation waits after applying an instance of
711 CIM_SettingData before deciding whether the resultant configuration is valid is implementation specific
712 and outside the scope of this specification. The criterion for determining whether a configuration that is
713 represented by a specific CIM_SettingData instance is valid is implementation specific and outside the
714 scope of this specification.

715 ## 7.6 Relationship with a network interface

716 An IP interface is generally bound to an underlying network interface. The underlying network interface
717 might participate in a LAN and be modeled using the *Host LAN Network Port Profile* or a specialization
718 thereof. When the underlying network interface is modeled with instrumentation compliant with the *Host*
719 *LAN Network Port Profile*, an instance of CIM_BindsToLANEndpoint shall associate the Central Instance
720 of this profile with an instance of CIM_LANEndpoint that is compliant with the *Host LAN Network Port*
721 *Profile*.

722   8   Methods

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

725   8.1   CIM_IPProtocolEndpoint.RequestStateChange( )

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

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

731   No standard messages are defined.

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

734   **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.

735   **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

736   8.1.1.1   CIM_IPProtocolEndpoint.RequestStateChange( ) — Conditional support

737   When an instance of CIM_EnabledLogicalElementCapabilities is associated with the
738   CIM_IPProtocolEndpoint instance and the
739   CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least one
740   value, the CIM_IPProtocolEndpoint.RequestStateChange( ) method shall be implemented and supported.
741   The CIM_IPProtocolEndpoint.RequestStateChange( ) method shall not return a value of 1 (Not
742   Supported).

743   ## 8.2   CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint( )

744   The CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint( ) method is used to apply a
745   configuration, as represented by an aggregating instance of CIM_IPAssignmentSettingData, to an IP
746   interface, as represented by an instance of CIM_IPProtocolEndpoint. Implementation of this method is
747   optional.

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

750   No standard messages are defined.

751   **Table 4 – CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint( ) method: Return code**
752   **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.

753   **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

754   The CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint( ) method shall be implemented as
755   follows:

756   •   The implementation shall validate that an instance of CIM_ServiceAffectsElement references
757   the CIM_IPConfigurationService instance and the CIM_IPProtocolEndpoint instance that is
758   identified by the Endpoint parameter to the method. If the association does not exist, the return
759   code of the method shall be 4 (Failed).

760   •   The implementation shall validate that an instance of CIM_ElementSettingData associates the
761   instance of CIM_IPProtocolEndpoint that is identified by the Endpoint parameter with the
762   instance of CIM_IPAssignmentSettingData that is identified by the Configuration parameter. If
763   the association does not exist, the return code of the method shall be 4 (Failed).

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

766   ## 8.3   Profile conventions for operations

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

769   The default list of operations is as follows:

770   • GetInstance

771   • EnumerateInstances

772   • EnumerateInstanceNames

773   • Associators

774   • AssociatorNames

775   • References

776   • ReferenceNames

777   8.4   CIM_BindsToLANEndpoint

778   Table 6 lists implementation requirements for operations. If implemented, these operations shall be
779   implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 6, all operations in
780   the default list in 8.3 shall be implemented as defined in DSP0200.

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

782   **Table 6 – Operations: CIM_BindsToLANEndpoint**

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

783   8.5   CIM_ElementSettingData

784   Table 7 lists implementation requirements for operations. If implemented, these operations shall be
785   implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 7, all operations in
786   the default list in 8.3 shall be implemented as defined in DSP0200.

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

788   **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

789   8.5.1   CIM_ElementSettingData — ModifyInstance

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

792 **8.5.1.1 CIM_ElementSettingData Referencing CIM_IPAssignmentSettingData**

793 When an instance of CIM_ElementSettingData associates an instance of CIM_IPAssignmentSettingData
794 with an instance of CIM_IPProtocolEndpoint, the following rules shall govern the behavior of the
795 ModifyInstance operation:

796 • The ModifyInstance operation shall not allow the IsDefault property to be modified.

797 • The ModifyInstance operation shall not allow the IsCurrent property to be modified.

798 • When the ModifyInstance operation is used to set the IsNext property to a value of 1 (Is Next),
799 the ModifyInstance operation shall implement the following behavior:

800 1) The ModifyInstance operation shall find all other instances of CIM_ElementSettingData
801 that associate an instance of CIM_IPAssignmentSettingData with the instance of
802 CIM_IPProtocolEndpoint referenced by the target instance of CIM_ElementSettingData
803 where the IsNext property has a value of 1 (Is Next).

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

806 • When the IsNext property has a value of 1 (Is Next), the ModifyInstance operation shall not be
807 supported.

808 • When the ModifyInstance operation is used to set the IsNext property to a value of 3 (Is Next for
809 Single Use), the ModifyInstance operation shall implement the following behavior:

810 1) The ModifyInstance operation shall find all other instances of CIM_ElementSettingData
811 that associate an instance of CIM_IPAssignmentSettingData with the instance of
812 CIM_IPProtocolEndpoint referenced by the target instance of CIM_ElementSettingData
813 where the IsNext property has a value of 3 (Is Next For Single Use).

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

816 **8.5.1.2 CIM_ElementSettingData Referencing CIM_StaticIPAssignmentSettingData**

817 When an instance of CIM_ElementSettingData associates an instance of
818 CIM_StaticIPAssignmentSettingData with an instance of CIM_IPProtocolEndpoint, the ModifyInstance
819 operation shall not be supported.

820 ## 8.6 CIM_HostedAccessPoint

821 Table 8 lists implementation requirements for operations. If implemented, these operations shall be
822 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 8, all operations in
823 the default list in 8.3 shall be implemented as defined in DSP0200.

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

825   **Table 8 – Operations: CIM_HostedAccessPoint**

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

826 8.7 CIM_HostedService

827 Table 9 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 9, 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 9 – Operations: CIM_HostedService**

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

832 8.8 CIM_IPAssignmentSettingData

833 All operations in the default list in 8.3 shall be implemented as defined in DSP0200.

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

835 8.9 CIM_IPConfigurationService

836 All operations in the default list in 8.3 shall be implemented as defined in DSP0200.

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

838 8.10 CIM_IPProtocolEndpoint

839 Table 10 lists implementation requirements for operations. If implemented, these operations shall be
840 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 10, all operations
841 in the default list in 8.3 shall be implemented as defined in DSP0200.

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

843   **Table 10 – Operations: CIM_IPProtocolEndpoint**

Operation	Requirement	Messages
ModifyInstance	Conditional. See 8.10.1.	None

844 8.10.1 CIM_IPProtocolEndpoint — ModifyInstance operation

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

847 8.10.1.1 CIM_IPProtocolEndpoint.ElementName property

848 When an instance of CIM_EnabledLogicalElementCapabilities is associated with the
849 CIM_IPProtocolEndpoint instance and the
850 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported property has a value of TRUE, the
851 implementation shall allow the ModifyInstance operation to change the value of the ElementName
852 property of the CIM_IPProtocolEndpoint instance. The ModifyInstance operation shall enforce the length
853 restriction specified in the MaxElementNameLen property of the instance of
854 CIM_EnabledLogicalElementCapabilities.

855 When no instance of CIM_EnabledLogicalElementCapabilities is associated with the
856 CIM_IPProtocolEndpoint instance, or the ElementNameEditSupported property of the
857 CIM_EnabledLogicalElementCapabilities instance has a value of FALSE, the implementation shall not
858 allow the ModifyInstance operation to change the value of the ElementName property of the
859 CIM_IPProtocolEndpoint instance.

860 ## 8.11 CIM_OrderedComponent

861 Table 11 lists implementation requirements for operations. If implemented, these operations shall be
862 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 11, all operations
863 in the default list in 8.3 shall be implemented as defined in DSP0200.

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

865 **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

866 ### 8.11.1 CIM_OrderedComponent — ModifyInstance

867 The ModifyInstance operation may be supported for CIM_OrderedComponent. When an instance of
868 CIM_OrderedComponent references an instance of CIM_DNSSettingData or an instance of
869 CIM_DNSGeneralSettingData, the AssignedSequence property may be modified. When an instance of
870 CIM_OrderedComponent references an instance of CIM_StaticIPAssignmentSettingData or an instance
871 of CIM_DHCPSettingData, the AssignedSequence property shall not be modified.

872 ## 8.12 CIM_RemoteAccessAvailableToElement

873 Table 12 lists implementation requirements for operations. If implemented, these operations shall be
874 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 12, all operations
875 in the default list in 8.3 shall be implemented as defined in DSP0200.

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

877 **Table 12 – Operations: CIM_RemoteAccessAvailableToElement**

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

878 ## 8.13 CIM_RemoteServiceAccessPoint

879 All operations in the default list in 8.3 shall be implemented as defined in DSP0200.

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

881 **8.14 CIM_ServiceAffectsElement**

882 Table 13 lists implementation requirements for operations. If implemented, these operations shall be
883 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 13, all operations
884 in the default list in 8.3 shall be implemented as defined in DSP0200.

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

886 **Table 13 – Operations: CIM_ServiceAffectsElement**

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

887 **8.15 CIM_StaticIPAssignmentSettingData**

888 Table 14 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 14, 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 14 – Operations: CIM_StaticIPAssignmentSettingData**

Operation	Requirement	Messages
ModifyInstance	Optional	None

893   # 9   Use cases

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

895   ## 9.1   Miscellaneous object diagrams

896   The object diagram in Figure 2 shows one possible method for advertising profile conformance. The
897   instances of CIM_RegisteredProfile are used to identify the version of the *IP Interface Profile* with which
898   an instance of CIM_IPProtocolEndpoint and its associated instances are conformant. An instance of
899   CIM_RegisteredProfile exists for each profile that is instrumented in the system. One instance of
900   CIM_RegisteredProfile identifies the "DMTF Base Server Profile version 1.0.0". The other instance
901   identifies the "DMTF IP Interface Profile version 1.1.0". The CIM_IPProtocolEndpoint instance is scoped
902   to an instance of CIM_ComputerSystem. This instance of CIM_ComputerSystem is conformant with the
903   DMTF *Base Server Profile* version 1.0.0 as indicated by the CIM_ElementConformsToProfile association
904   to the CIM_RegisteredProfile instance.
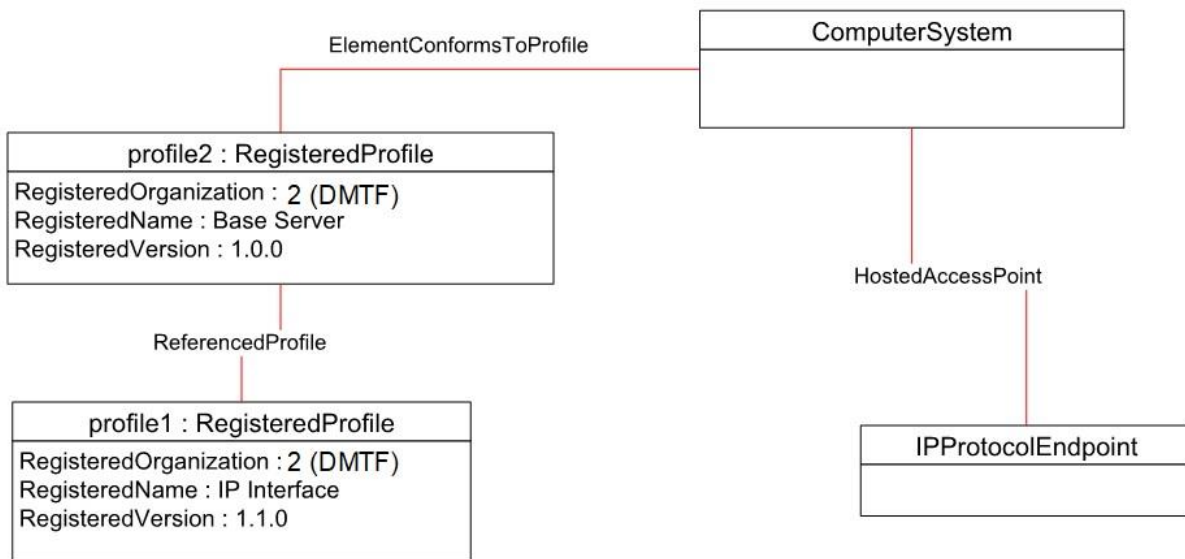
905

906   **Figure 2 – Registered Profile**

907   The object diagram shown in Figure 3 contains the basic elements used to model the current
908   configuration of an IP interface when the CIM_IPProtocolEndpoint.ProtocolIFType is 4096 (Ipv4). The IP
909   interface is bound to an Ethernet NIC, as illustrated by the CIM_BindsToLANEndpoint association
910   between the CIM_IPProtocolEndpoint instance and the CIM_LANEndpoint instance. The AddressOrigin
911   property of the CIM_IPProtocolEndpoint has a value of "static", indicating that the configuration was
912   statically assigned. In this diagram, the *Ethernet Port Profile* and *IP Interface Profile* have been
913   implemented.

914   The default gateway used by the IP interface is represented by the instance of
915   CIM_RemoteServiceAccessPoint that is associated with the CIM_IPProtocolEndpoint instance through an
916   instance of CIM_RemoteAccessAvailableToElement.

917
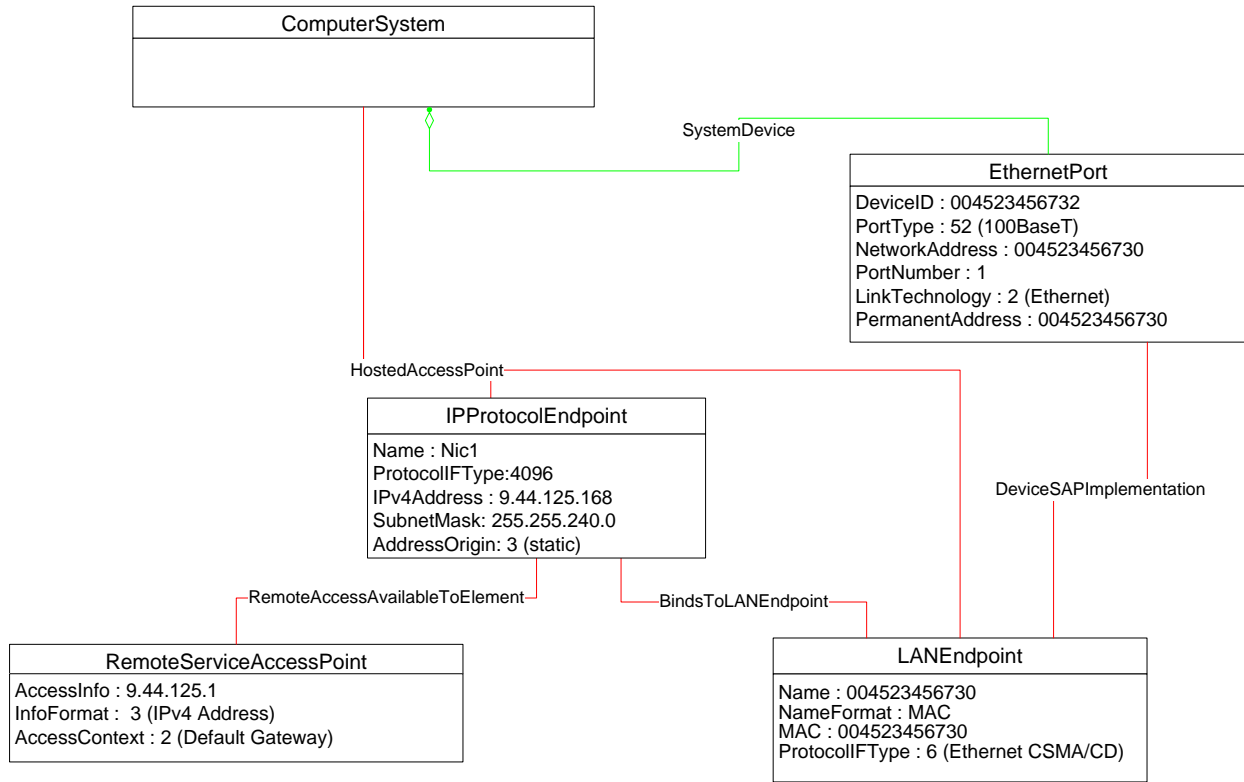
Figure 3 – Basic configuration — IPv4

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

924

925
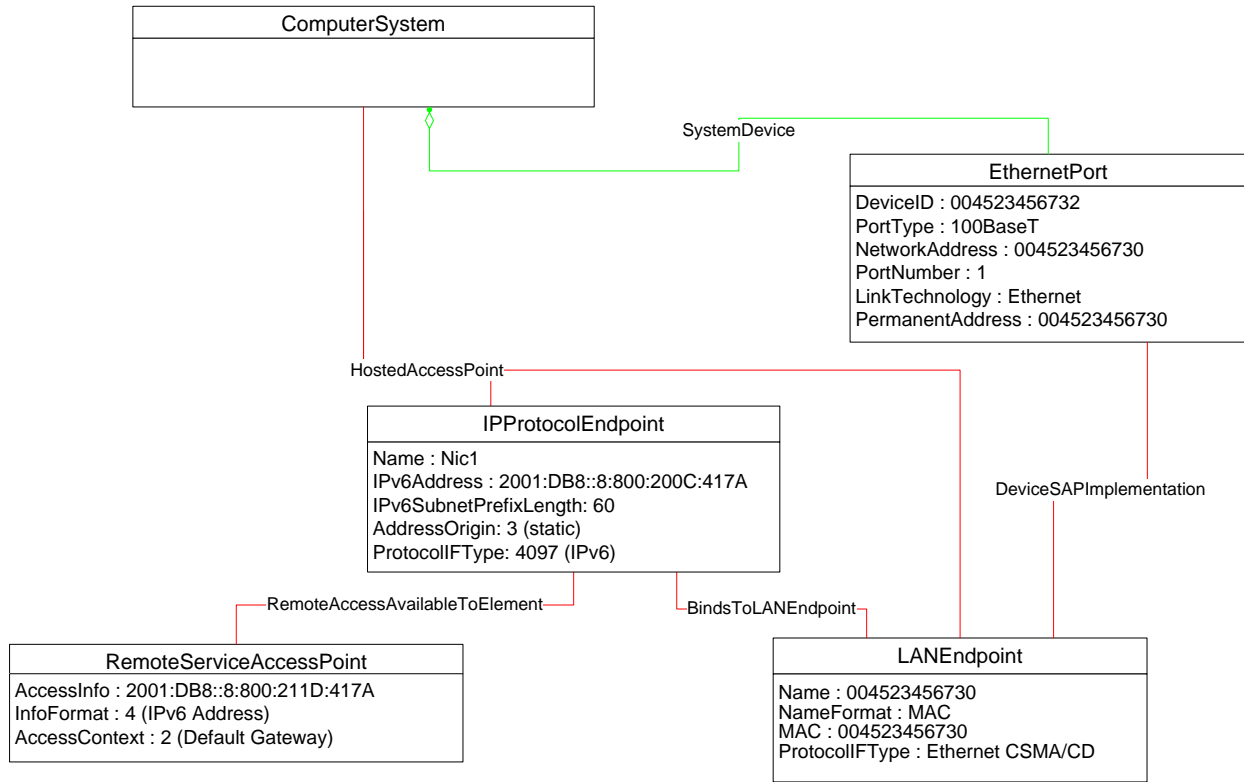
Figure 4 – Basic configuration — IPv6

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

932
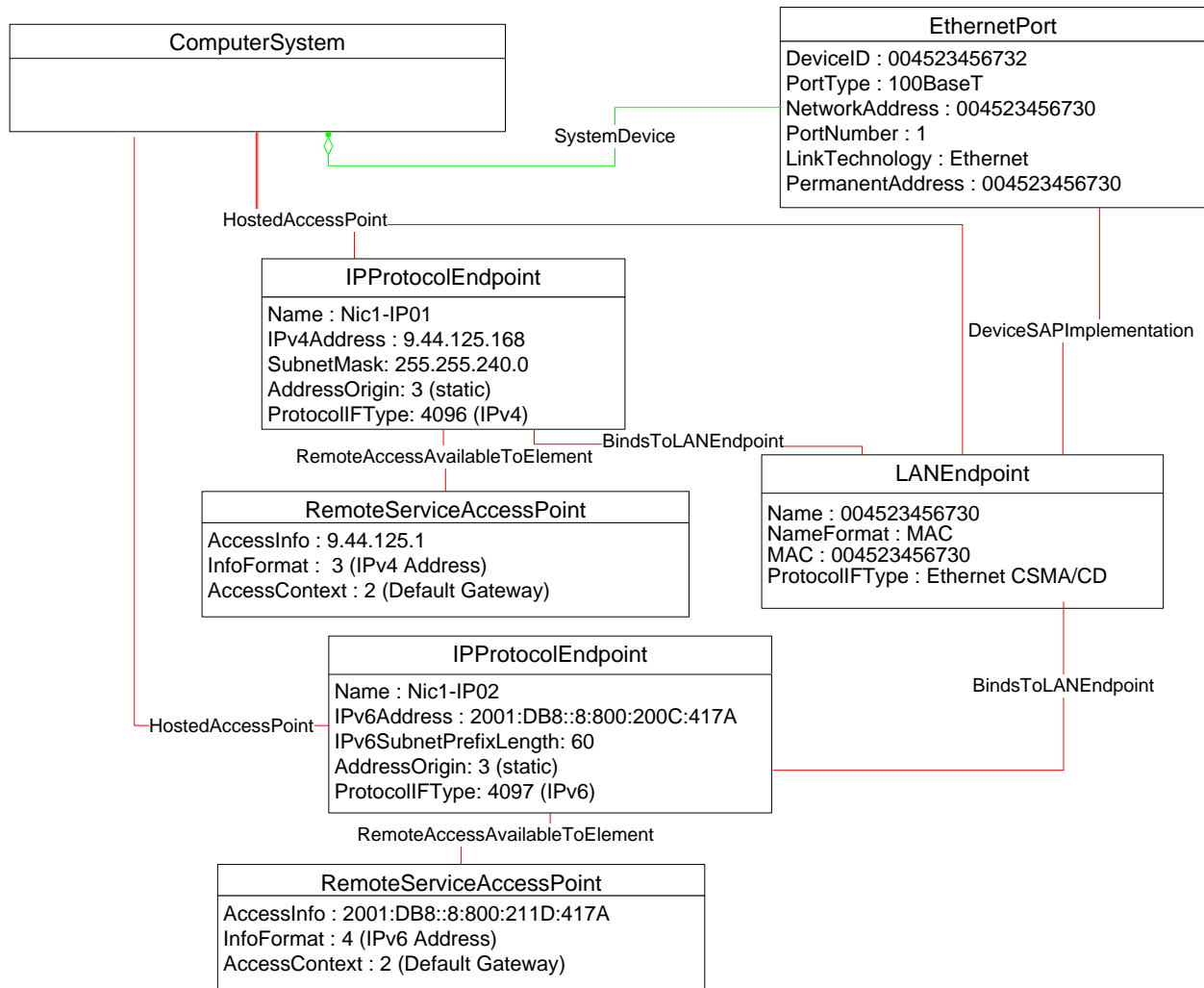
933

934   **Figure 5 – Basic configuration — IPv4 and IPv6**

935   Figure 6 illustrates the elements and properties of an IP interface that supports static configuration. The
936   IP interface currently has a single, alternate configuration associated with it. The optional IP configuration
937   management behavior is depicted in this object diagram. Note that the pending configuration has been
938   modified after it was applied to the CIM_IPProtocolEndpoint. Hence the values for properties of
939   CIM_IPProtocolEndpoint do not align with the values of properties of the
940   CIM_StaticIPAssignmentSettingData instance.
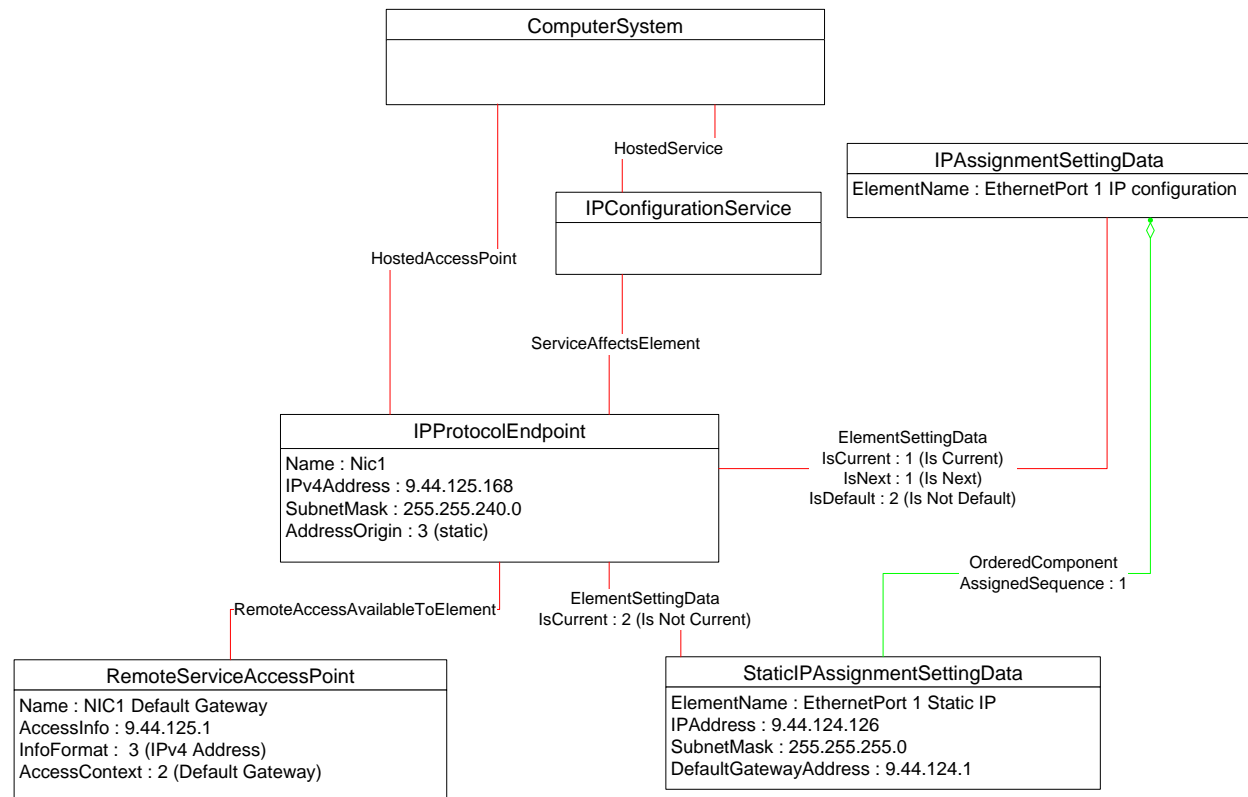
941

942   **Figure 6 – Static current and pending configuration**

943   The object diagram in Figure 7 illustrates an IP interface with two supported alternate configurations. Two
944   discrete IP configuration options are available for the IP interface. Each option is represented by an
945   instance of CIM_IPAssignmentSettingData. One configuration option represents the ability to statically
946   assign the IP configuration. This option is indicated by the instance of CIM_OrderedComponent that
947   associates the CIM_IPAssignmentSettingData instance with an instance of
948   CIM_StaticIPAssignmentSettingData. The other configuration option is to obtain the configuration through
949   a DHCP client. This option is indicated by the instance of CIM_OrderedComponent that associates the
950   CIM_IPAssignmentSettingData with an instance of CIM_DHCPSettingData.

951   In this example, each configuration option consists of a single instance of a subclass of
952   CIM_IPAssignmentSettingData. Therefore, the value of the AssignedSequence property of the
953   CIM_OrderedComponent instances is irrelevant.

954   The default configuration is to attempt to obtain a configuration through DHCP. This default is indicated
955   by the IsDefault property having a value of 1 (Is Default) on the CIM_ElementSettingData instance that
956   associates the CIM_IPAssignmentSettingData instance with the CIM_IPProtocolEndpoint instance.
957   However, the current configuration of the IP interface was statically assigned using the configuration
958   identified by the CIM_IPAssignmentSettingData instance *static*. This configuration is indicated by the
959   value of the IsCurrent property on the instance of CIM_ElementSettingData that associates the
960   CIM_IPAssignmentSettingData instance *static* with the CIM_IPProtocolEndpoint instance, and by the
961   value of the AddressOrigin property on the CIM_IPProtocolEndpoint instance. When the interface is
962   restarted, the static configuration will be used again for the IP interface. This behavior is indicated by the
963   value of the IsNext property on the instance of CIM_ElementSettingData that associates the
964   CIM_IPAssignmentSettingData instance *static* to the CIM_IPProtocolEndpoint instance.
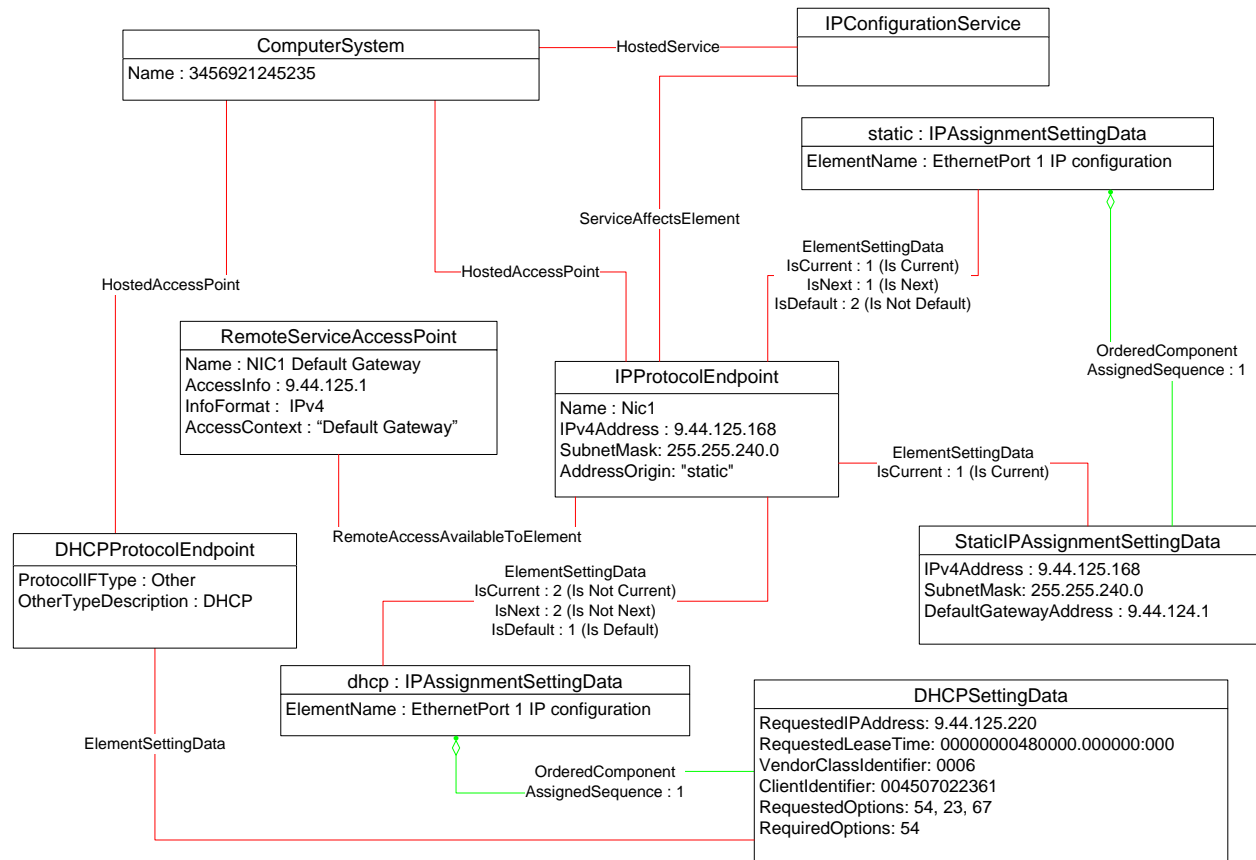
965

966 **Figure 7 – Static and DHCP pending configurations**

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

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

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

975
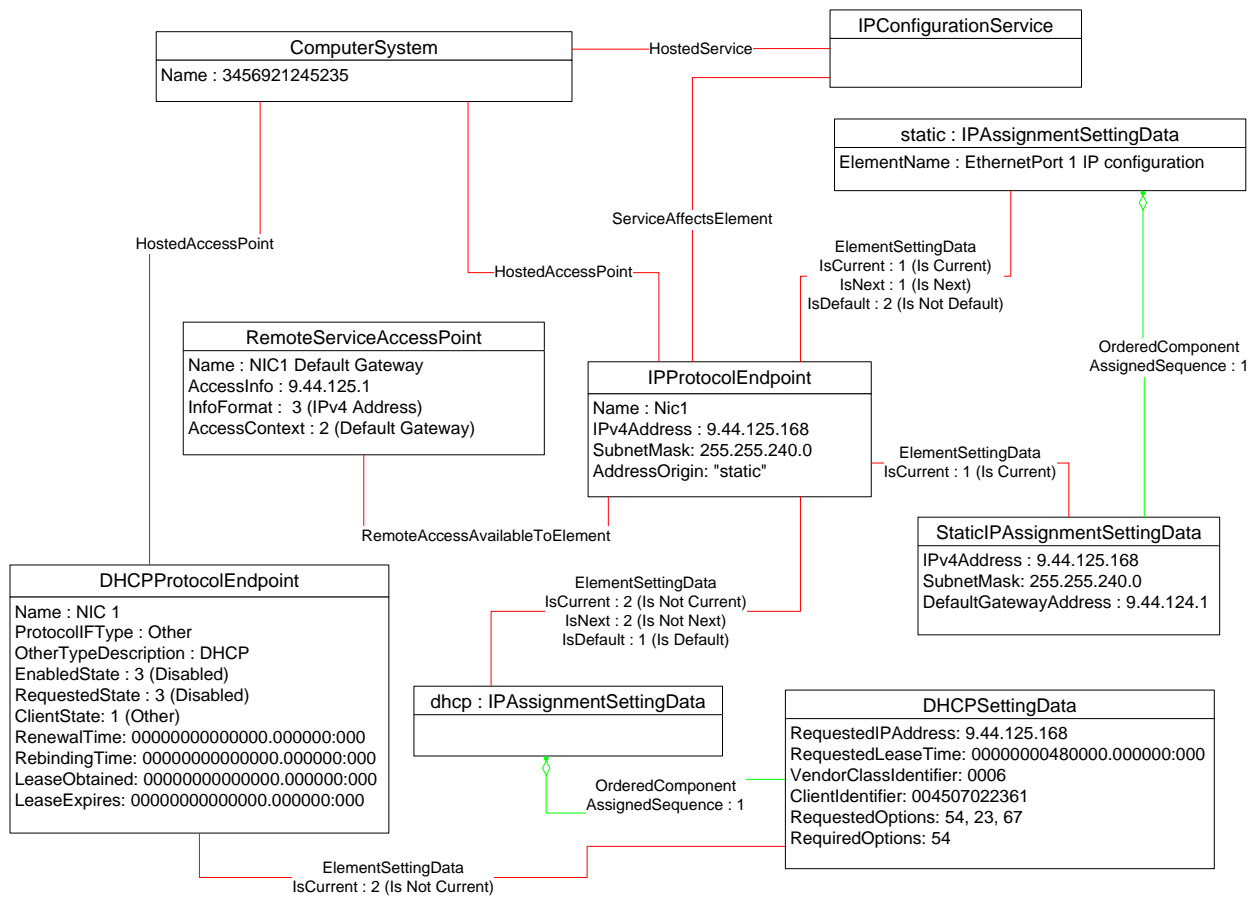
976

977   **Figure 8 – DHCP timed out to a static configuration**

978   The object diagram in Figure 9 illustrates a configuration in which a system contains an integrated service
979   processor and they share the network interface of the system. The CIM_EthernetPort instance is
980   associated with the system1 instance, which indicates that the network device is owned by the server.
981   The MAC property of the lan1 instance matches the PermanentAddress property of the CIM_EthernetPort
982   instance, which indicates that the server is using the hardware MAC. The MAC property of the lan2
983   instance is different, which indicates that the service processor has been assigned a logical MAC. The
984   system and service processor each have a unique IP interface that has been statically configured.
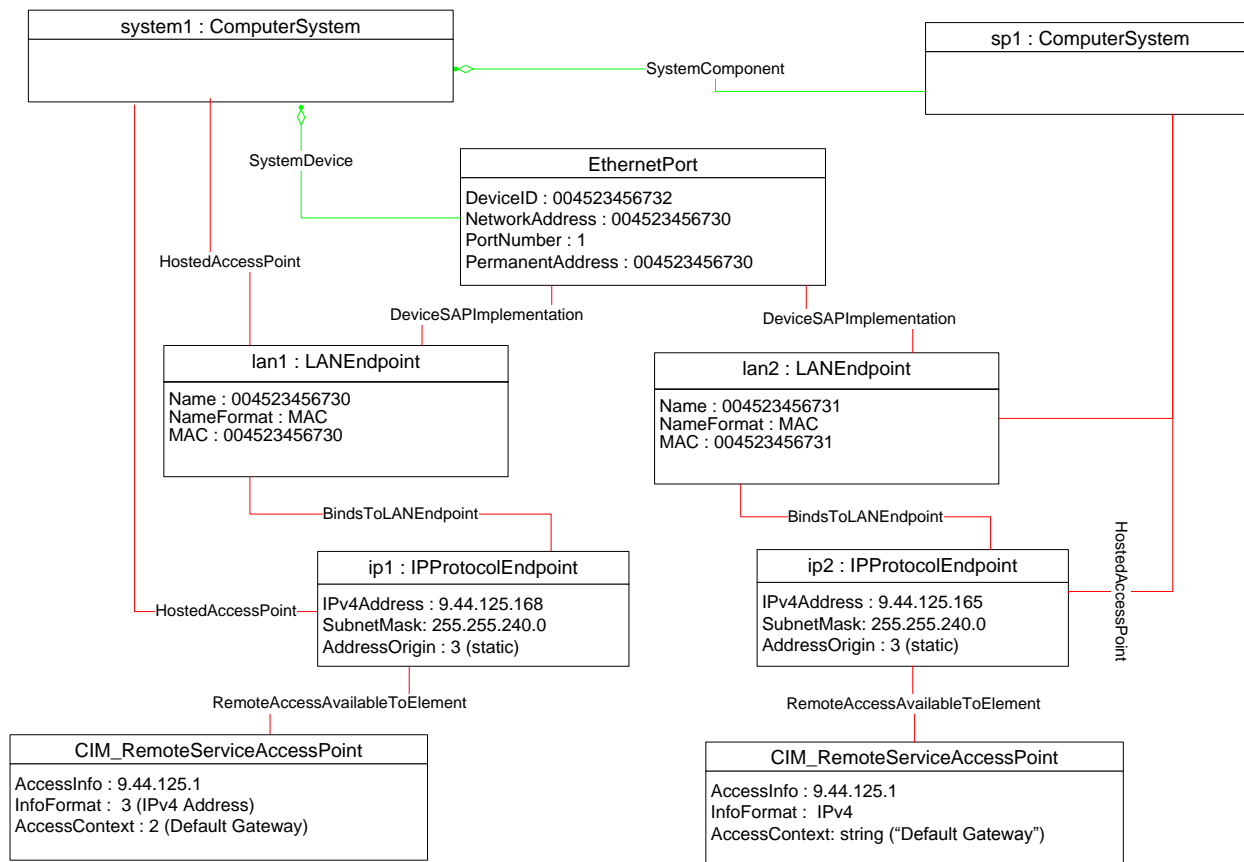
985

986 **Figure 9 – Service processor and server share an NIC**

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

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

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

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

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

1000 static_only represents a configuration that uses static assignment of the IP configuration, including
1001 support for static configuration of the DNS client. This behavior is indicated by the aggregated instances:
1002 static1, dns1, and dnsgen1.

1003 dhcp_only represents a configuration that uses DHCP to obtain the IP configuration. This behavior is
1004 indicated by the aggregated instance dhcp1. The DNS configuration can be assigned through DHCP or
1005 statically assigned. This behavior is indicated by the aggregated instances dns1 and dnsgen1. In the

1006 event the DHCP client is unable to obtain a configuration, the system is implemented to default to a hard-
1007 coded, well-known default static IP configuration. The existence of a default configuration is indicated by
1008 the aggregated instance static3. Note that no advertisement mechanism is specified in the profile to
1009 indicate that static3 represents hard-coded values that cannot be modified by the client. If the system
1010 were implemented such that the DHCP client would be continually in use without a timeout to a static
1011 configuration, the aggregated instance static3 would not exist.

1012 dhcp_static represents a configuration that attempts to use DHCP to obtain an IP configuration. In the
1013 event the DHCP client fails to obtain a configuration, the system defaults to a client-assigned static IP
1014 configuration. This behavior is indicated by the instances dhcp1 and static1 and the relative values of the
1015 AssignedSequence property of the instances of CIM_OrderedComponent, which aggregate them into
1016 dhcp_static.
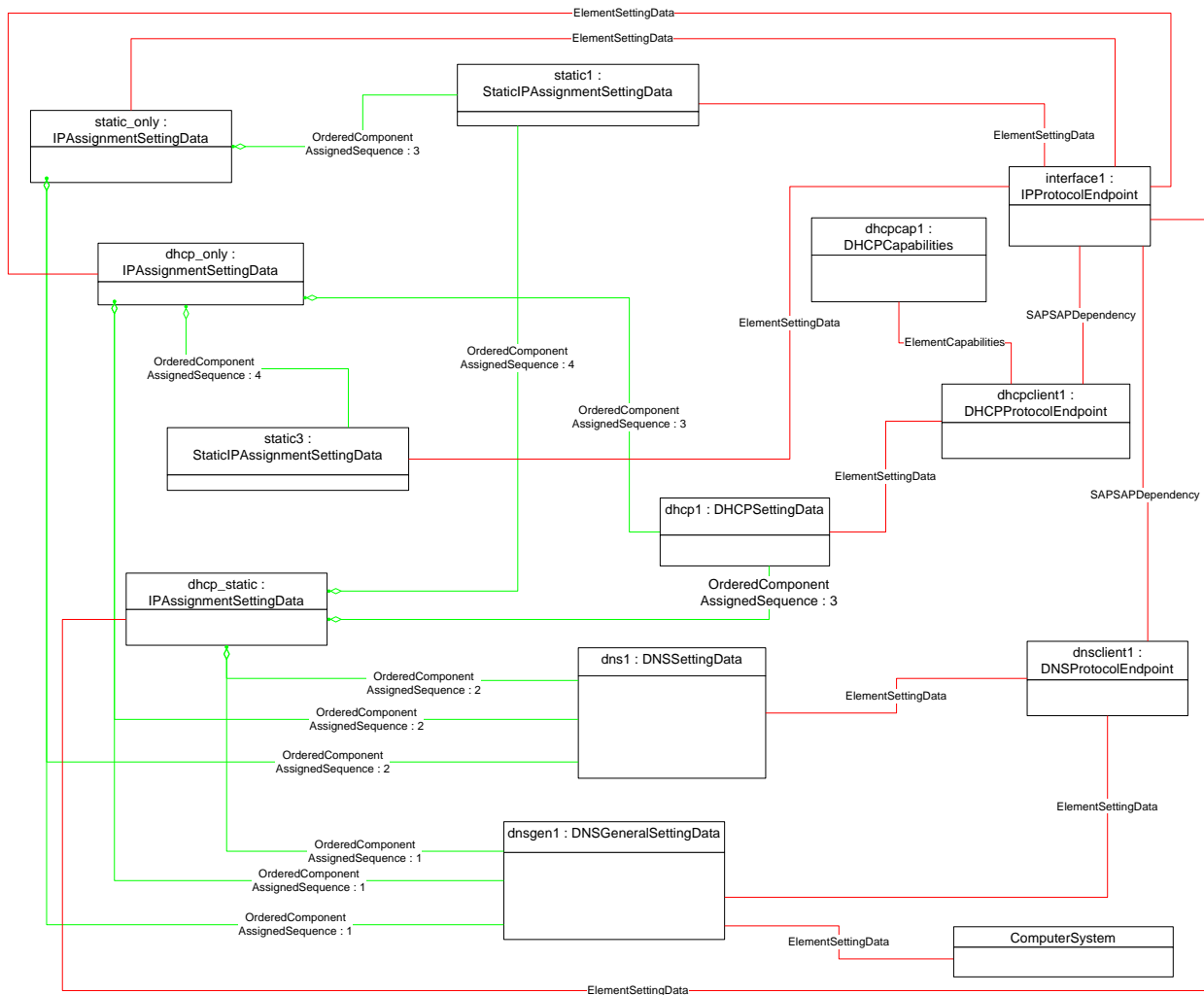
1017

1018   **Figure 10 – Configuration choices**

1019 The object diagram in Figure 11 reflects the system when the DHCP configuration method has been used
1020 and the DNS configuration has partially been assigned through DHCP and partially statically configured.

1021 The use of the DHCP-only configuration is indicated by the IsCurrent property of the instance of
1022 CIM_ElementSettingData that associates dhcp_only to interface1 having the value 1 (Is Current). The
1023 DHCP configuration includes DHCP options that affect the DNS configuration. The DHCP options 8, 14,
1024 and 17 are requested as indicated by the RequestedOptions property of dhcp1. Each of these options
1025 was in turn received by the DHCP client, which is indicated by the value of the OptionsReceived property
1026 of dhcpclient1. The DNS client has been configured to use the values received for options 14 and 17 as
1027 indicated by the presence of these values in the DHCPOptionsToUse property of dnsclient1. The
1028 properties on dnsclient1 reflect the current DNS client configuration. Note that the actual current
1029 configuration does not directly reflect the configuration indicated by dns1 and dnsgen1. The two
1030 properties for which values were supplied by the DHCP options instead reflect the values assigned by the
1031 DHCP server.
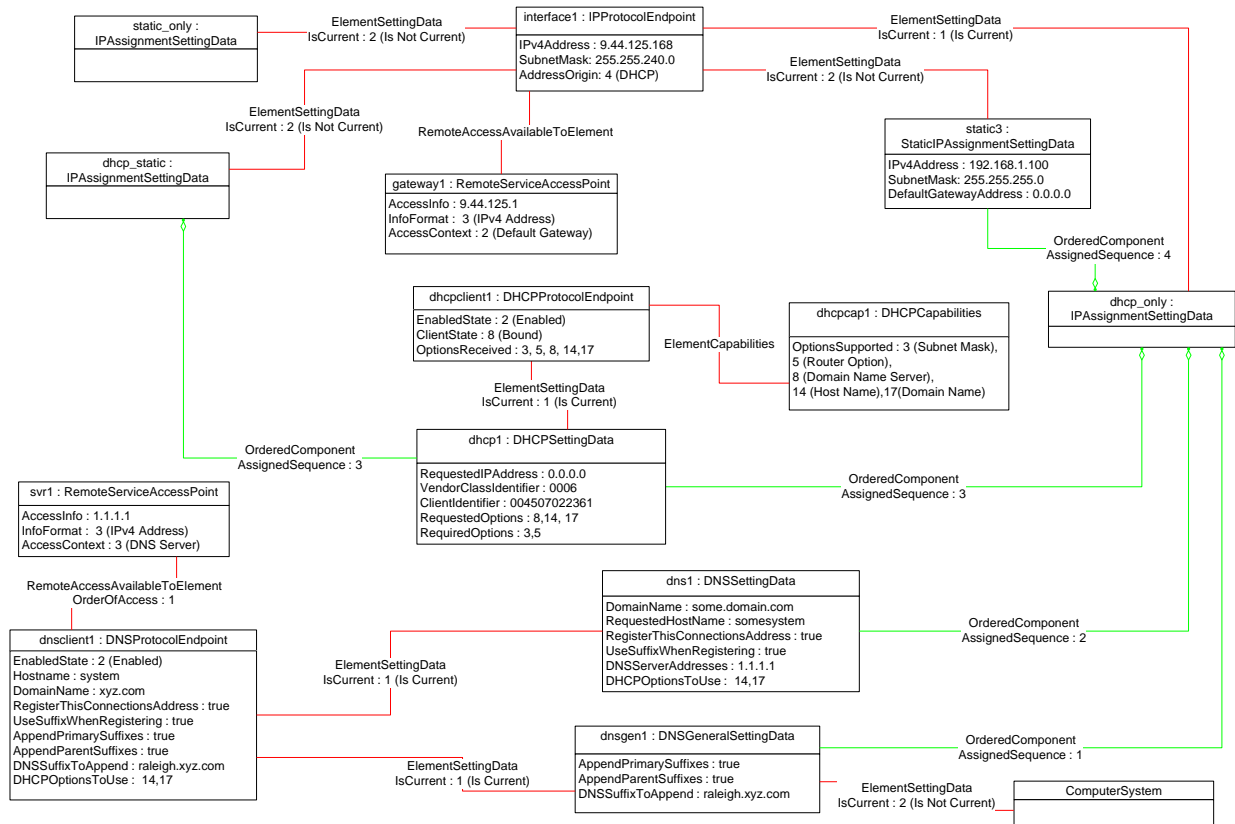
1032

1033 **Figure 11 – DHCP assigned partial DNS**

1034 The object diagram in Figure 12 reflects the system when the DHCP configuration method has been used
1035 and the DNS configuration has been statically configured.

1036 The use of the DHCP-only configuration is indicated by the IsCurrent property of the instance of
1037 CIM_ElementSettingData that associates dhcp_only to interface1 having the value 1 (Is Current).
1038 Although the DHCP configuration includes DHCP options that affect the DNS configuration, the values
1039 returned are not being used by the DNS client. This behavior is indicated by the absence of any values in

1040 the DHCPOptionsToUse property of dnsclient1. The actual current configuration directly reflects the
1041 configuration indicated by dns1 and dnsgen1 because no DHCP options are selected for use.

1042

```
ElementSettingData                    interface1 : IPProtocolEndpoint           ComputerSystem
IsCurrent : 2 (Is Not Current)        IPv4Address : 9.44.125.168
                                      SubnetMask: 255.255.240.0
                                      AddressOrigin: 4 (DHCP)        ElementSettingData
                                                                    IsCurrent : 1 (Is Current)
static_only :
IPAssignmentSettingData                            ElementSettingData
                                                   IsCurrent : 2 (Is Not Current)
                                                                     static3 :
                              ElementSettingData                     StaticIPAssignmentSettingData
                              IsCurrent : 2 (Is Not Current)
                                                                     IPv4Address : 192.168.1.100
          RemoteAccessAvailableToElement                            SubnetMask: 255.255.255.0
                                                                    DefaultGatewayAddress : 0.0.0.0
                              gateway1 :                                       OrderedComponent
dhcp_static :                 RemoteServiceAccessPoint         dhcpclient1 :    AssignedSequence : 4
IPAssignmentSettingData                                        DHCPProtocolEndpoint
                              AccessInfo : 9.44.125.1          EnabledState : 2 (Enabled)    dhcp_only :
                              InfoFormat :  3 (IPv4 Address)   ClientState : 8 (Bound)       IPAssignmentSettingData
                              AccessContext : 2 (Default Gateway)  OptionsReceived : 3, 5
                                                                                              ElementSettingData
          OrderedComponent                                                                    IsCurrent : 1 (Is Current)
          AssignedSequence : 3                ElementSettingData
                                              IsCurrent : 1 (Is Current)
svr1 : RemoteServiceAccessPoint
                                    dhcp1 : DHCPSettingData
AccessInfo : 1.1.1.1                                                      OrderedComponent
InfoFormat :  3 (IPv4 Address)      RequestedIPAddress : 0.0.0.0          AssignedSequence : 3
AccessContext : 3 (DNS Server)      VendorClassIdentifier : 0006
                                    ClientIdentifier : 004507022361
                                    RequestedOptions : 8, 14,17
                                    RequiredOptions : 3,5             dns1 : DNSSettingData
RemoteAccessAvailableToElement
OrderOfAccess : 1                                                     DomainName : jersey.xyz.com
                                                                     RequestedHostName : system
dnsclient1 : DNSProtocolEndpoint                                     RegisterThisConnectionsAddress : FALSE
                                                                     UseSuffixWhenRegistering : TRUE
EnabledState : 2 (Enabled)                                           DNSServerAddresses : 1.1.1.1
Hostname : system                          ElementSettingData        DHCPOptionsToUse :
DomainName : jersey.xyz.com                IsCurrent : 1 (Is Current)          OrderedComponent
RegisterThisConnectionsAddress : FALSE                                          AssignedSequence : 2
UseSuffixWhenRegistering : TRUE
AppendPrimarySuffixes : TRUE
AppendParentSuffixes : TRUE         dnsgen1 : DNSGeneralSettingData              OrderedComponent
DNSSuffixToAppend : raleigh.xyz.com                                             AssignedSequence : 1
DHCPOptionsToUse :                  AppendPrimarySuffixes : TRUE
                            ElementSettingData  AppendParentSuffixes : TRUE
                            IsCurrent : 1 (Is Current)  DNSSuffixToAppend : raleigh.xyz.com
```

1043 **Figure 12 – DHCP with DNS statically configured**

1044 ### 9.1.1 Sequence for disabled DNS client

1045 The following three object diagrams illustrate the system when a client is configuring it to use a static IP
1046 configuration with the DNS client disabled. The client first modifies the pending static configuration so that
1047 the DNS settings will not be applied. Then it disables the DNS client directly. Finally, it applies the static
1048 configuration.

1049 The object diagram in Figure 13 illustrates the state of the system before the client begins modifying it to
1050 use a static IP configuration with DNS disabled. The last configuration applied was the DHCP-only
1051 configuration, which is indicated by the value of the IsCurrent property of the CIM_ElementSettingData
1052 instance that references dhcp_only and interface1. The static_only configuration has not yet been
1053 modified by the client. As shown, the alternate DNS configuration represented by dns1 and dnsgen1
1054 would be applied if static_only were applied to interface1.
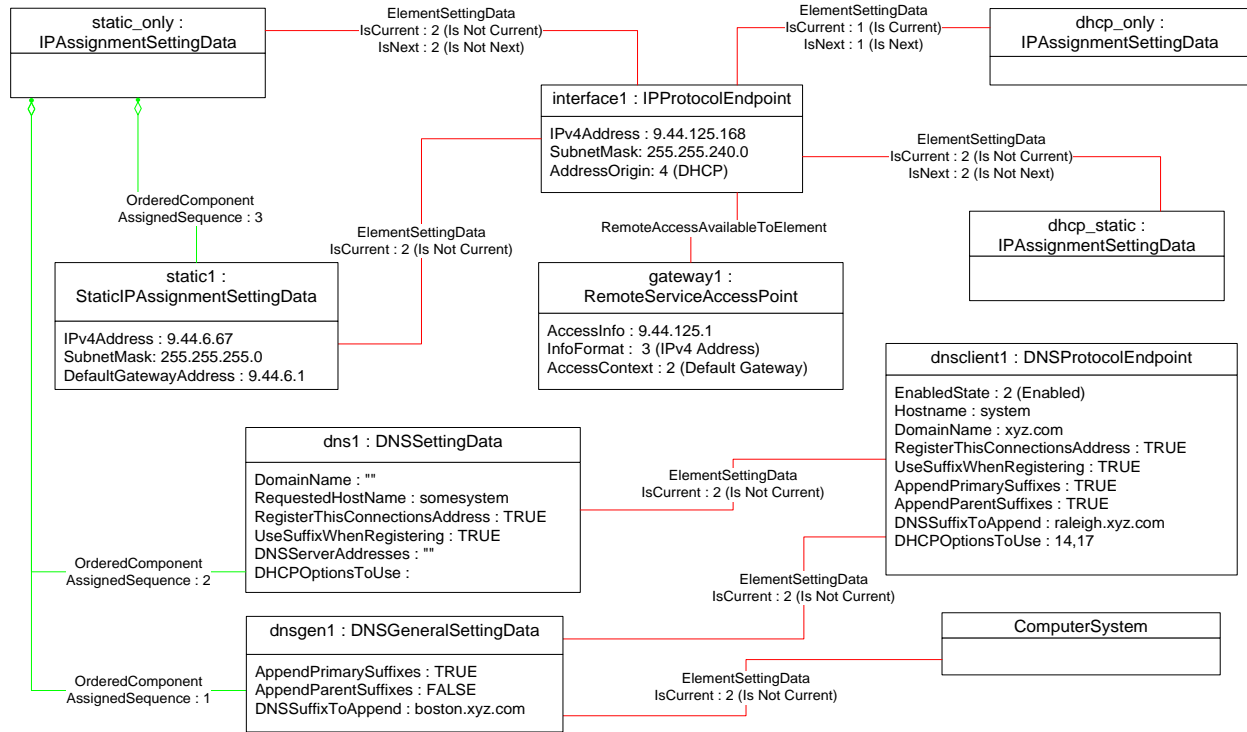
1055

Figure 13 – Static without DNS configuration — One

1057 In Figure 14, static_only has been made the pending configuration for interface1. This behavior is
1058 indicated by the value of the IsNext property of the instance of CIM_ElementSettingData that references
1059 static_only and interface1. static_only has been modified such that the DNS configuration will not be
1060 applied. This behavior is indicated by the AssignedSequence property having a value of 0 (zero) for each
1061 of the CIM_OrderedComponent instances that reference static_only and dns1 or dnsgen1. Separately,
1062 the DNS client has been disabled, which is indicated by the value of the EnabledState property of
1063 dnsclient1.
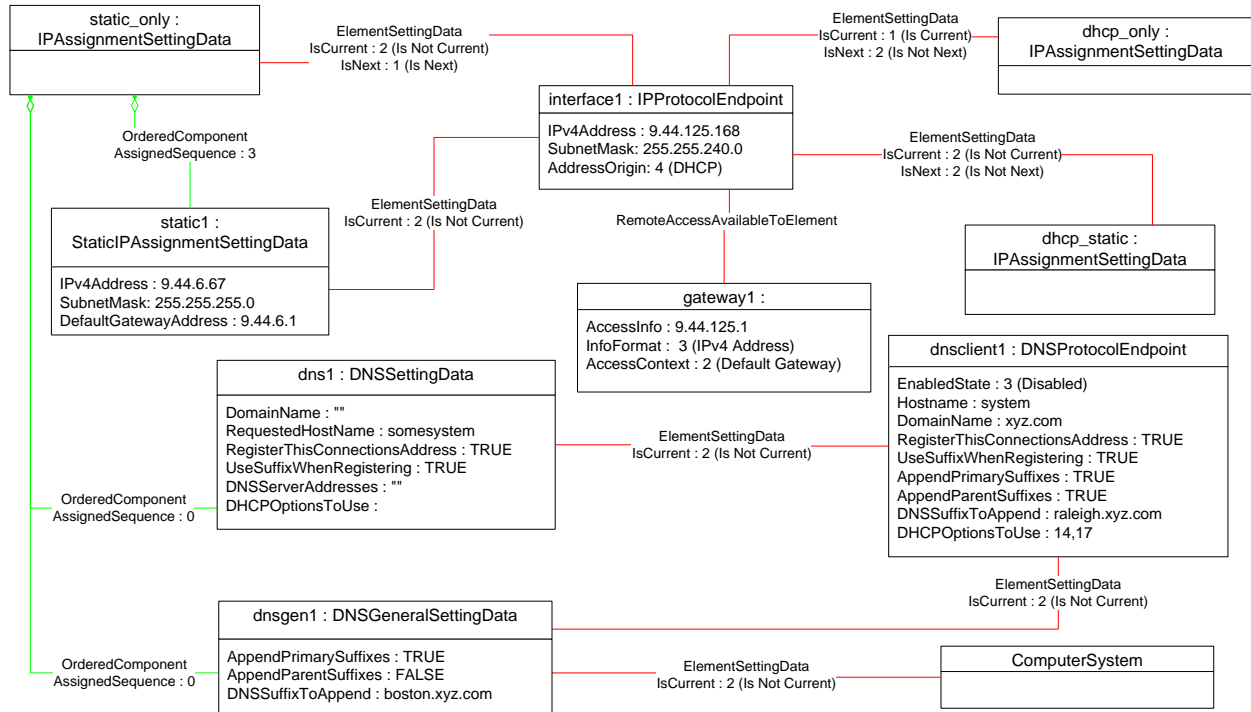
1064

1065   **Figure 14 – Static without DNS configuration — Two**

1066   Figure 15 shows the system after static_only has been applied to interface1. Note that the current DNS
1067   configuration has not changed as a result of applying static_only to interface1.
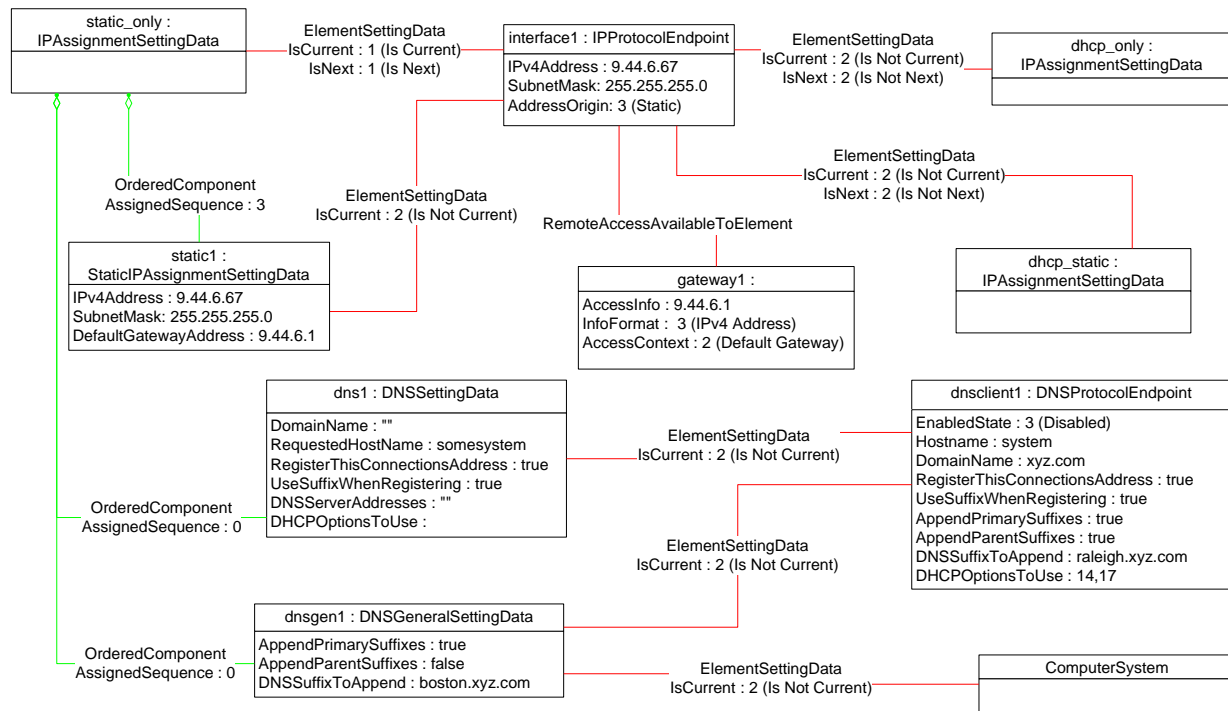
1068

1069 **Figure 15 – Static without DNS configuration — Three**

9.2 Determine supported configuration methods

1071 A client can determine which configuration methods are supported for a given interface as follows:

1072   1)   Find all instances of CIM_IPAssignmentSettingData that are associated with the
1073   CIM_IPProtocolEndpoint instance.

1074   2)   For each instance of CIM_IPAssignmentSettingData:

1075   a)   Find all instances of subclasses of CIM_IPAssignmentSettingData that are associated with
1076   the CIM_IPAssignmentSettingData instance through an instance of
1077   CIM_OrderedComponent.

1078   b)   Query the value of the AddressOrigin property to determine the supported identified
1079   configuration method.

9.3 Determine gateway address

1081 A client can find the default gateway in use for an IP interface as follows:

1082   1)   Find all instances of CIM_RemoteServiceAccessPoint that are associated with the
1083   CIM_IPProtocolEndpoint instance through an instance of
1084   CIM_RemoteAccessAvailableToElement.

1085   2)   For each instance of CIM_RemoteServiceAccessPoint, determine if the value of the
1086   AccessContext property is "Default Gateway". If so, query the value of the AccessInfo property.

9.4 Determine method used for current configuration

1088 A client can determine the method by which the IP configuration was assigned by querying the
1089 AddressOrigin property of the CIM_IPProtocolEndpoint instance.

9.5 Determine whether DHCP then static is supported

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

1) Find all instances of CIM_IPAssignmentSettingData (the parent class and not subclasses) that are associated with the CIM_IPProtocolEndpoint instance.

2) For each instance of CIM_IPAssignmentSettingData:

   a) Find all instances of CIM_DHCPSettingData that are associated through an instance of CIM_OrderedComponent.

   b) Find all instances of CIM_StaticIPAssignmentSetttingData that are associated through an instance of CIM_OrderedComponent.

3) Determine whether there is an instance of CIM_DHCPSettingData such that the value of the AssignedSequence property of the CIM_OrderedComponent that associates the instance of CIM_DHCPSettingData with the instance of CIM_IPAssignmentSettingData is less than the value of the AssignedSequence property of an instance of CIM_OrderedComponent that associates the CIM_StaticIPAssignmentSettingData with the instance of CIM_IPAssignmentSettingData. If so, DHCP then static is supported.

9.6 View default configuration

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

1) Find all instances of CIM_ElementSettingData that associate an instance of CIM_IPAssignmentSettingData (the parent class and not subclasses) with the CIM_IPProtocolEndpoint instance.

2) For each instance of CIM_ElementSettingData, see if the value of the IsDefault property is 1 (Is Default).

9.7 Configure the interface to use DHCP

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

1) Find all instances of CIM_IPAssignmentSettingData (the parent class and not subclasses) that are associated with the CIM_IPProtocolEndpoint instance.

2) For each instance of CIM_IPAssignmentSettingData:

   a) Find an instance of CIM_DHCPSettingData that is associated through an instance of CIM_OrderedComponent.

   b) Verify that no instances of CIM_StaticIPAssignmentSettingData are associated with the instance of CIM_IPAssignmentSettingData.

   This instance of CIM_IPAssignmentSettingData represents a DHCP configuration.

3) Find an instance of CIM_IPConfigurationService that is associated with the CIM_IPProtocolEndpoint instance through an instance of CIM_ServiceAffectsElement.

4) Invoke the ApplySettingToIPProtocolEndpoint( ) method of the CIM_IPConfigurationService instance, specifying the instances of CIM_IPProtocolEndpoint and CIM_IPAssignmentSettingData.

1130   ### 9.8 Establish a static IP configuration for an interface

1131   A client can manually assign an IP configuration to an interface as follows:

1132   1) Find all instances of CIM_IPAssignmentSettingData (the parent class and not subclasses) that
1133   are associated with the CIM_IPProtocolEndpoint instance.

1134   2) For each instance of CIM_IPAssignmentSettingData:

1135   a) Find an instance of CIM_StaticIPAssignmentSettingData that is associated through an
1136   instance of CIM_OrderedComponent.

1137   b) Verify that no other instances of CIM_StaticIPAssignmentSettingData or instances of
1138   CIM_DHCPSettingData are associated with the instance of CIM_IPAssignmentSettingData
1139   through an instance of CIM_OrderedComponent.

1140   c) For the instance of CIM_ElementSettingData that associates the
1141   CIM_IPAssignmentSettingData instance with the instance of CIM_IPProtocolEndpoint,
1142   verify that the value of the IsDefault property is 2 (Is Not Default).

1143   This instance of CIM_IPAssignmentSettingData represents a modifiable, static configuration for
1144   the IP interface.

1145   3) Modify the properties of the CIM_StaticIPAssignmentSettingData instance to contain the
1146   appropriate configuration for the IP interface.

1147   4) Apply the pending configuration using the steps in 9.9 or 9.10.

1148   ### 9.9 Apply a pending configuration — synchronously

1149   Some implementations may support modifying the configuration of an IP interface without requiring a
1150   restart of the underlying network interface. If this behavior is supported by the implementation, then given
1151   an instance of CIM_IPProtocolEndpoint for which the configuration should be modified and an instance of
1152   CIM_IPAssignmentSettingData that represents the new configuration, a client can:

1153   1) Find an instance of CIM_IPConfigurationService that is associated with the
1154   CIM_IPProtocolEndpoint instance through an instance of CIM_ServiceAffectsElement.

1155   2) Invoke the ApplySettingToIPProtocolEndpoint( ) method of the CIM_IPConfigurationService,
1156   specifying the instances of CIM_IPProtocolEndpoint and CIM_IPAssignmentSettingData.

1157   ### 9.10 Apply a pending configuration — upon restart

1158   Some implementations may require that the IP interface be restarted in order for a new configuration that
1159   is bound to the interface to take effect. If an implementation requires that the IP interface be restarted,
1160   then given an instance of CIM_IPProtocolEndpoint for which the configuration should be modified and an
1161   instance of CIM_IPAssignmentSettingData that represents the new configuration, a client can:

1162   1) Find an instance of CIM_ElementSettingData that associates the CIM_IPAssignmentSettingData
1163   instance with the CIM_IPProtocolEndpoint instance.

1164   2) Set the IsNext property of the CIM_ElementSettingData instance to a value of 1 (Is Next).

1165   3) Invoke the RequestStateChange( ) method of the CIM_IPProtocolEndpoint instance, with a
1166   RequestedState of 11 (Reset).

1167 ## 9.11 Determine whether DNS configuration was DHCP assigned

1168 Starting at the CIM_DNSProtocolEndpoint instance, a client can determine if any elements of the DNS
1169 configuration were assigned through DHCP as follows:

1170   1)   Find the instance of CIM_IPProtocolEndpoint that is associated through an instance of
1171   CIM_SAPSAPDependency.

1172   2)   Find the instance of CIM_DHCPProtocolEndpoint that is associated with the
1173   CIM_IPProtocolEndpoint instance through an instance of CIM_SAPSAPDependency.

1174   3)   Query the EnabledState property of the CIM_DHCPProtocolEndpoint instance for the value 2
1175   (Enabled) to ensure that the DHCP client was used.

1176   4)   Query the OptionsReceived property of the CIM_DHCPProtocolEndpoint instance to determine
1177   if one of the DNS-related options (8, 14, or 17) was received.

1178 ## 9.12 Determine whether ElementName can be modified

1179 A client can determine whether it can modify the ElementName property of an instance of
1180 CIM_IPProtocolEndpoint as follows:

1181   1)   Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
1182   CIM_IPProtocolEndpoint instance.

1183   2)   Query the value of the ElementNameEditSupported property of the
1184   CIM_EnabledLogicalElementCapabilities instance. If the value is TRUE, the client can modify
1185   the ElementName property of the target instance.

1186 ## 9.13 Determine whether state management is supported

1187 A client can determine whether state management is supported for an instance of
1188 CIM_IPProtocolEndpoint as follows:

1189   1)   Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
1190   CIM_IPProtocolEndpoint instance.

1191   2)   Query the value of the RequestedStatesSupported property. If at least one value is specified,
1192   state management is supported.

1193 # 10 CIM Elements

1194 Table 15 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
1195 implemented as described in Table 15. Clauses 7 ("Implementation") and 8 ("Methods") may impose
1196 additional requirements on these elements.

1197 **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 (IPAssignmentSettingData)	Conditional	See 7.4.2, and 10.4.
CIM_ElementSettingData (StaticIPAssignmentData)	Conditional	See 7.4.4, and 10.4.

Element Name	Requirement	Description
CIM_HostedAccessPoint (IPProtocolEndpoint)	Mandatory	See 10.7.
CIM_HostedAccessPoint (RemoteServiceAccessPoint)	Conditional	See 10.6.
CIM_HostedService	Conditional	See 7.4.1 and 10.8.
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		

10.1 CIM_BindsToLANEndpoint

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

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

10.2 CIM_ElementCapabilities

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

Table 17 – Class: CIM_ElementCapabilities

Elements	Requirement	Description
ManagedElement	Mandatory	**Key:** This shall be a reference to the Central Instance. Cardinality 1..*

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

10.3 CIM_ElementSettingData — CIM_IPAssignmentSettingData Reference

1208

1209 CIM_ElementSettingData associates instances of CIM_IPAssignmentSettingData with the
1210 CIM_IPProtocolEndpoint instance. Table 18 provides information about the properties of
1211 CIM_ElementSettingData.

1212 **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)

10.4 CIM_ElementSettingData — CIM_StaticIPAssignmentSettingData Reference

1213

1214 CIM_ElementSettingData associates instances of CIM_StaticIPAssignmentSettingData with the
1215 CIM_IPProtocolEndpoint instance. Table 19 provides information about the properties of
1216 CIM_ElementSettingData.

1217 **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)

1218 ## 10.5 CIM_EnabledLogicalElementCapabilities

1219 CIM_EnabledLogicalElementCapabilities indicates support for managing the IP interface. Table 20
1220 provides information about the properties of CIM_EnabledLogicalElementCapabilities.

1221 **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.

1222 ## 10.6 CIM_HostedAccessPoint — CIM_RemoteServiceAccessPoint Reference

1223 An instance of CIM_HostedAccessPoint Association between an instance of CIM_ProtocolEndpoint and
1224 CIM_RemoteServiceAccessPoint shall only be instantiated if CIM_RemoteServiceAccessPoint is
1225 supported.

1226 CIM_HostedAccessPoint relates the CIM_RemoteServiceAccessPoint instance that represents the
1227 default gateway with its scoping CIM_ComputerSystem instance. Table 21 provides information about the
1228 properties of CIM_HostedAccessPoint.

1229 **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 *

1230 ## 10.7 CIM_HostedAccessPoint — CIM_IPProtocolEndpoint Reference

1231 CIM_HostedAccessPoint relates the Central Instance with its Scoping Instance. Table 22 provides
1232 information about the properties of CIM_HostedAccessPoint.

1233 **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..*

1234   ## 10.8 CIM_HostedService

1235   CIM_HostedService relates the CIM_IPConfigurationService instance to its scoping
1236   CIM_ComputerSystem instance. Table 23 provides information about the properties of
1237   CIM_HostedService.

1238   **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 *

1239   ## 10.9 CIM_IPAssignmentSettingData

1240   CIM_IPAssignmentSettingData is the aggregation point for the SettingData instances that define a
1241   configuration that can be applied to an IP interface. Table 24 provides information about the properties of
1242   CIM_IPAssignmentSettingData.

1243   **Table 24 – Class: CIM_IPAssignmentSettingData**

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

1244   ## 10.10 CIM_IPConfigurationService

1245   CIM_IPConfigurationService represents the ability to configure an IP interface. Table 25 provides
1246   information about the properties of CIM_IPConfigurationService.

1247   **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.

1248   **10.11 CIM_IPProtocolEndpoint**

1249   CIM_IPProtocolEndpoint represents an IP interface that is associated with an Ethernet interface. Table 26
1250   provides information about the properties of CIM_IPProtocolEndpoint.

1251   **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.
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.
IPv6AddressType	Conditional	See 7.1.1.2.
IPv6SubnetPrefixLength	Conditional	See 7.1.1.2.

1252   **10.12 CIM_OrderedComponent**

1253   CIM_OrderedComponent associates an instance of CIM_IPAssignmentSettingData to the instances of
1254   CIM_StaticIPAssignmentSettingData, CIM_DHCPSettingData, CIM_DNSSettingData, and
1255   CIM_DNSGeneralSettingData that compose a configuration. Table 27 provides information about the
1256   properties of CIM_OrderedComponent.

1257   **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.

10.13 CIM_RegisteredProfile

CIM_RegisteredProfile identifies the *IP Interface Profile* in order for a client to determine whether an instance of CIM_IPProtocolEndpoint is conformant with this profile. The CIM_RegisteredProfile class is defined by the *Profile Registration Profile*. With the exception of the mandatory values specified for the properties in Table 28, the behavior of the CIM_RegisteredProfile instance is in accordance with the *Profile Registration Profile*.

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 "1.1.2".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

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

10.14 CIM_RemoteAccessAvailableToElement

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

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.

10.15 CIM_RemoteServiceAccessPoint

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

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 ".*"

1278 **10.16 CIM_ServiceAffectsElement**

1279 CIM_ServiceAffectsElement associates an instance of CIM_IPConfigurationService with an instance of
1280 CIM_IPProtocolEndpoint that the service is able to configure. Table 31 provides information about the
1281 properties of CIM_ServiceAffectsElement.

1282   **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)

1283 **10.17 CIM_StaticIPAssignmentSettingData**

1284 CIM_StaticIPAssignmentSettingData represents a static configuration that can be applied to an instance
1285 of CIM_IPProtocolEndpoint. Table 32 provides information about the properties of
1286 CIM_StaticIPAssignmentSettingData.

1287   **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	
IPv6AddressType	Optional	
IPv6SubnetPrefixLength	Optional	
GatewayIPv6Address	Optional	

1288

1289
1290

ANNEX A
(informative)

1291
1292

Change log

Version	Date	Description
1.0.0	2008-07-27	Final Standard & addition of IPv6 support as Experimental
1.0.1	2009-09-24	Errata 1.0.1
1.1.0	2010-10-21	Experimental sections were removed, and the document was formatted for DMTF Standard release.
1.1.1	2012-01-09	Errata 1.1.1 Section 8 - Removed CIM_SystemDevice. Section 9 - Correction in association for CIM_RemoteServiceAccessPoint. Section 10 - Spelling for CIM_ServiceAffectsElement.ElementEffects
1.1.2	2018-12-18	This errata addresses these issues: • Updated RegisteredVersion to reflect errata version number in clause 10.2 • Updated RegisteredOrganization description to reflect correct value for DMTF in clause 10.2. • Updated CIM_Elements table in Section 10 with additional instances and relevant clarification in Section 7 Updated use cases to reflect the above fixes.

1293
1294