Document Identifier: DSP1036

Date: 2018-12-18

Version: 1.0.3

# IP Interface Profile

Supersedes: 1.0.2

Document Class: Normative

Document Status: Published

Document Language: en-US

This document's normative language is English. Translation into other languages is permitted.

# CONTENTS

# Figures

# Tables

# Foreword

The *IP Interface Profile* (DSP1036) was prepared by the Server Management Working Group and the Physical Platform Profiles Working Group of the DMTF.

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

## Acknowledgments

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

- RadhaKrishna Dasari – Dell Inc.
- Jon Hass – Dell Inc.
- Jeff Hilland – Hewlett Packard Enterprise
- John Leung – Intel Corporation
- Aaron Merkin – IBM
- Khachatur Papanyan – Dell Inc.
- Sivakumar Sathappan – Advanced Micro Devices
- Hemal Shah – Broadcom Inc.
- Christina Shaw – Hewlett Packard Enterprise
- Enoch Suen – Dell Inc.
- Perry Vincent – Intel Corporation

# Introduction

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

## Document conventions

### Experimental material

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

The following typographical convention indicates experimental material:

**EXPERIMENTAL**

Experimental material appears here.

**EXPERIMENTAL**

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

# IP Interface Profile

## 1 Scope

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

## 2 Normative references

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

### 2.1 Approved references

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

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

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

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

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

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

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

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

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

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

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

# 3 Terms and definitions

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

**3.1**

**can**

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

**3.2**

**cannot**

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

**3.3**

**conditional**

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

**3.4**

**mandatory**

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

**3.5**

**may**

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

**3.6**

**need not**

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

**3.7**

**optional**

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

**3.8**

**pending configuration**

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

**3.9**

**referencing profile**

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

**3.10**

**shall**

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

**3.11**

**shall not**

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

**3.12**

**should**

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

**3.13**

**should not**

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

**3.14**

**unspecified**

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

# 4 Symbols and abbreviated terms

The following abbreviations are used in this document.

**4.1**

**DHCP**

Dynamic Host Configuration Protocol

**4.2**

**DNS**

Domain Name System

**4.3**

**IP**

Internet Protocol

# 5 Synopsis

**Profile Name:** IP Interface

**Version:** 1.0.1

**Organization:** DMTF

**CIM Schema Version:** 2.19

**Central Class:** CIM_IPProtocolEndpoint

**Scoping Class:** CIM_ComputerSystem

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

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

**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. |

| Profile Name | Organization | Version | Requirement | Description |
|---|---|---|---|---|
| DHCP Client | DMTF | 1.0 | Optional | See 7.2. |
| Host LAN Network Port | DMTF | 1.0 | Optional | See 7.6. |

# 6 Description

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

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

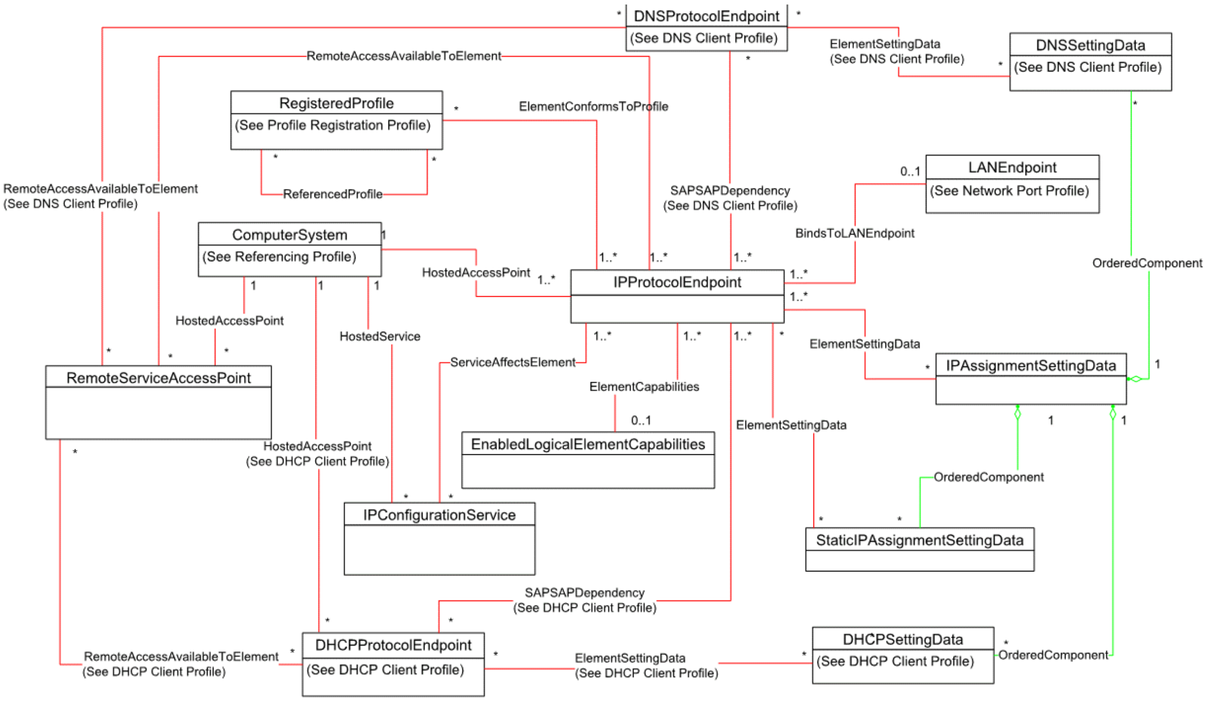


**Figure 1 – IP Interface Profile: Class diagram**

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

- IPv4 interface (optionally associated with a network interface)
- optional relationship with a DNS client

- optional relationship with a DHCP client
- current and pending configurations

Functionality explicitly excluded from the scope of this profile includes:

- modeling of the network gateway
- modeling of TCP/UDP ports

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

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

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

## 6.1 Pending and alternate configuration management

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

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

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

# 7 Implementation

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

## 7.1 Basic IP configuration

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

### 7.1.1 CIM_IPProtocolEndpoint

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

#### 7.1.1.1 CIM_IPProtocolEndpoint.AddressOrigin

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

##### 7.1.1.1.1 AddressOrigin — Static

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

##### 7.1.1.1.2 AddressOrigin — DHCP

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

#### 7.1.1.2 CIM_IPProtocolEndpoint.ProtocolIFType

The ProtocolIFType property shall indicate the current IP address type.

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

The value of CIM_IPProtocolEndpoint.ProtocolIFType shall be 4096,

**EXPERIMENTAL**

4097, or 4098.

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

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

**EXPERIMENTAL**

#### 7.1.1.3 CIM_IPProtocolEndpoint.IPv4Address

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

#### 7.1.1.4 CIM_IPProtocolEndpoint.SubnetMask

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

**EXPERIMENTAL**

#### 7.1.1.5 CIM_IPProtocolEndpoint.IPv6Address

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

**EXPERIMENTAL**

### 7.1.2 IP interface state management is supported — conditional

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

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

#### 7.1.2.1 CIM_EnabledLogicalElementCapabilities

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

#### 7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

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

#### 7.1.2.2 CIM_IPProtocolEndpoint.RequestedState

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

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

#### 7.1.2.3 CIM_IPProtocolEndpoint.EnabledState

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

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

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

### 7.1.3 IP interface state management is not supported

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

#### 7.1.3.1 CIM_EnabledLogicalElementCapabilities

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

##### 7.1.3.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall not contain any values.

#### 7.1.3.2 CIM_IPProtocolEndpoint.RequestedState

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

#### 7.1.3.3 CIM_IPProtocolEndpoint.EnabledState

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

### 7.1.4 Modifying ElementName is supported — conditional

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

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

#### 7.1.4.1 CIM_EnabledLogicalElementCapabilities

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

##### 7.1.4.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

The ElementNameEditSupported property shall have a value of TRUE.

##### 7.1.4.1.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

The MaxElementNameLen property shall be implemented.

### 7.1.5 Modifying ElementName is not supported

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

#### 7.1.5.1 CIM_EnabledLogicalElementCapabilities

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

##### 7.1.5.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

The ElementNameEditSupported property shall have a value of FALSE.

##### 7.1.5.1.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

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

### 7.1.6 Default gateway

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

**EXPERIMENTAL**

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

**EXPERIMENTAL**

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

#### 7.1.6.1 CIM_RemoteServiceAccessPoint.AccessInfo

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

**EXPERIMENTAL**

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

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

**EXPERIMENTAL**

#### 7.1.6.2 CIM_RemoteAccessAvailableToElement.Antecedent

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

#### 7.1.6.3 CIM_RemoteAccessAvailableToElement.Dependent

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

#### 7.1.6.4 CIM_RemoteAccessAvailableToElement.OrderOfAccess

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

## 7.2 DHCP client is supported

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

## 7.3 DNS client is supported

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

## 7.4 Managing alternate configurations — optional

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

### 7.4.1 Configuration management is supported

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

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

### 7.4.2 Representing an alternate configuration using CIM_IPAssignmentSettingData

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

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

#### 7.4.2.1 Associating an alternate configuration with an IP interface

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

##### 7.4.2.1.1 CIM_ElementSettingData.IsCurrent

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

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

### 7.4.3 Associating settings using CIM_OrderedComponent

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

#### 7.4.3.1 CIM_OrderedComponent.GroupComponent

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

#### 7.4.3.2 CIM_OrderedComponent.PartComponent

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

#### 7.4.3.3 Interpretation of CIM_OrderedComponent.AssignedSequence

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

##### 7.4.3.3.1 Use of 0 (zero)

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

##### 7.4.3.3.2 Discreteness

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

### 7.4.4 Alternate static configuration

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

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

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

### 7.4.5 Alternate DHCP configuration

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

### 7.4.6 DNS client alternate configuration

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

### 7.4.7 Relationship between DHCP and DNS configuration

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

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

#### 7.4.7.1 Relationship between DHCP options and the DNS configuration

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

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

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

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

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

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

- When the OptionsReceived property of the DHCPPE instance and the DHCPOptionsToUse property of the DNSPE instance both contain the value 8 (Domain Name Server), the DNS Servers instrumented in accordance with the "DNS Server Representation" clause of the *DNS Client Profile* shall identify the DNS server addresses specified by the DHCP server as the data for the Domain Name Server DHCP option.
- When the OptionsReceived property of the DHCPPE instance and the DHCPOptionsToUse property of the DNSPE instance both contain the value 14 (Host Name), the value of the Hostname property of the DNSPE instance shall be the hostname specified by the DHCP server as the data for the Host Name DHCP option.
- When the OptionsReceived property of the DHCPPE instance and the DHCPOptionsToUse property of the DNSPE instance both contain the value 17 (Domain Name), the value of the DomainName property of the DNSPE instance shall be the domain name specified by the DHCP server as the data for the Domain Name DHCP option.

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

### 7.4.8 Representing a pending configuration

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

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

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

## 7.5 Applying an alternate configuration

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

#### 7.5.3.1 CIM_StaticIPAssignmentSettingData.GatewayIPv4Address

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

**EXPERIMENTAL**

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

#### 7.5.3.2 CIM_StaticIPAssignmentSettingData.GatewayIPv6Address

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

**EXPERIMENTAL**

#### 7.5.3.3 Successful application of settings

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

### 7.5.4 Applying DHCP settings

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

### 7.5.5 Applying DNS settings

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

### 7.5.6 Resolving overlapped settings

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

## 7.6 Relationship with a network interface

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

# 8 Methods

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

## 8.1 CIM_IPProtocolEndpoint.RequestStateChange( )

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

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

No standard messages are defined.

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

**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 |

**Table 3 – CIM_IPProtocolEndpoint.RequestStateChange( ) method: Parameters**

| Qualifiers | Name | Type | Description/Values |
|---|---|---|---|
| IN, REQ | RequestedState | uint16 | Valid state values:<br>2 (Enabled)<br>3 (Disabled)<br>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:<br>0 or NULL – No time requirements<br>&lt;interval&gt; – Maximum time allowed |

#### 8.1.1.1 CIM_IPProtocolEndpoint.RequestStateChange( ) — Conditional support

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

## 8.2 CIM_IPConfigurationService.ApplySettingToIPProtocolEndpoint( )

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

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

No standard messages are defined.

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

**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 |

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

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

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

## 8.3 Profile conventions for operations

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

The default list of operations is as follows:

- GetInstance
- EnumerateInstances
- EnumerateInstanceNames
- Associators
- AssociatorNames
- References
- ReferenceNames

## 8.4 CIM_BindsToLANEndpoint

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

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

**Table 6 – Operations: CIM_BindsToLANEndpoint**

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

## 8.5 CIM_ElementSettingData

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

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

**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 |

### 8.5.1 CIM_ElementSettingData — ModifyInstance

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

#### 8.5.1.1 CIM_ElementSettingData Referencing CIM_IPAssignmentSettingData

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

- The ModifyInstance operation shall not allow the IsDefault property to be modified.
- The ModifyInstance operation shall not allow the IsCurrent property to be modified.
- When the ModifyInstance operation is used to set the IsNext property to a value of 1 (Is Next), the ModifyInstance operation shall implement the following behavior:
  1) The ModifyInstance operation shall find all other instances of CIM_ElementSettingData that associate an instance of CIM_IPAssignmentSettingData with the instance of CIM_IPProtocolEndpoint referenced by the target instance of CIM_ElementSettingData where the IsNext property has a value of 1 (Is Next).
  2) For each instance of CIM_ElementSettingData found, the ModifyInstance operation shall modify the value of its IsNext property to have a value of 2 (Is Not Next).
- When the IsNext property has a value of 1 (Is Next), the ModifyInstance operation shall not be supported.
- When the ModifyInstance operation is used to set the IsNext property to a value of 3 (Is Next for Single Use), the ModifyInstance operation shall implement the following behavior:
  1) The ModifyInstance operation shall find all other instances of CIM_ElementSettingData that associate an instance of CIM_IPAssignmentSettingData with the instance of CIM_IPProtocolEndpoint referenced by the target instance of CIM_ElementSettingData where the IsNext property has a value of 3 (Is Next For Single Use).
  2) For each instance of CIM_ElementSettingData found, the ModifyInstance operation shall modify the value of its IsNext property to have a value of 2 (Is Not Next).

#### 8.5.1.2 CIM_ElementSettingData Referencing CIM_StaticIPAssignmentSettingData

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

## 8.6 CIM_HostedAccessPoint

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

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

**Table 8 – Operations: CIM_HostedAccessPoint**

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

## 8.7 CIM_HostedService

Table 9 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 9, all operations in the default list in 8.3 shall be implemented as defined in DSP0200.

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

**Table 9 – Operations: CIM_HostedService**

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

## 8.8 CIM_IPAssignmentSettingData

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

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

## 8.9 CIM_IPConfigurationService

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

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

## 8.10 CIM_IPProtocolEndpoint

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

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

**Table 10 – Operations: CIM_IPProtocolEndpoint**

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

### 8.10.1 CIM_IPProtocolEndpoint — ModifyInstance operation

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

#### 8.10.1.1 CIM_IPProtocolEndpoint.ElementName property

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

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

## 8.11 CIM_OrderedComponent

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

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

**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 |

### 8.11.1 CIM_OrderedComponent — ModifyInstance

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

## 8.12 CIM_RemoteAccessAvailableToElement

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

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

**Table 12 – Operations: CIM_RemoteAccessAvailableToElement**

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

## 8.13 CIM_RemoteServiceAccessPoint

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

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

## 8.14 CIM_ServiceAffectsElement

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

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

**Table 13 – Operations: CIM_ServiceAffectsElement**

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

## 8.15 CIM_StaticIPAssignmentSettingData

Table 14 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 14, all operations in the default list in 8.3 shall be implemented as defined in DSP0200.

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

**Table 14 – Operations: CIM_StaticIPAssignmentSettingData**

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

## 8.16 CIM_SystemDevice

Table 15 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 15, all operations in the default list in 8.3 shall be implemented as defined in DSP0200.

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

**Table 15 – Operations: CIM_SystemDevice**

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

# 9 Use cases

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

## 9.1 Miscellaneous object diagrams

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

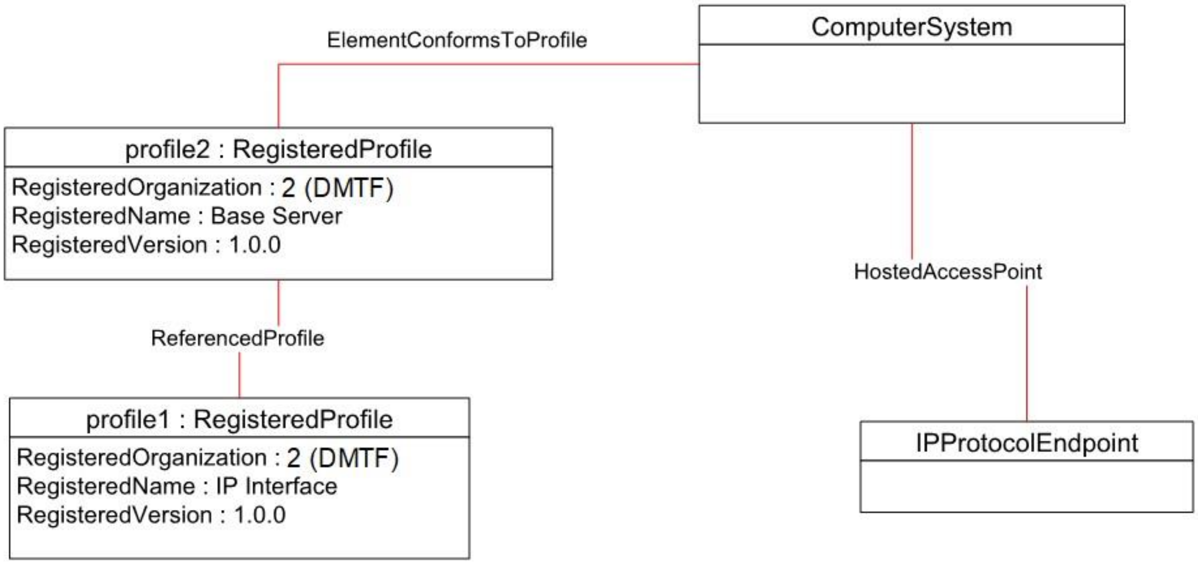
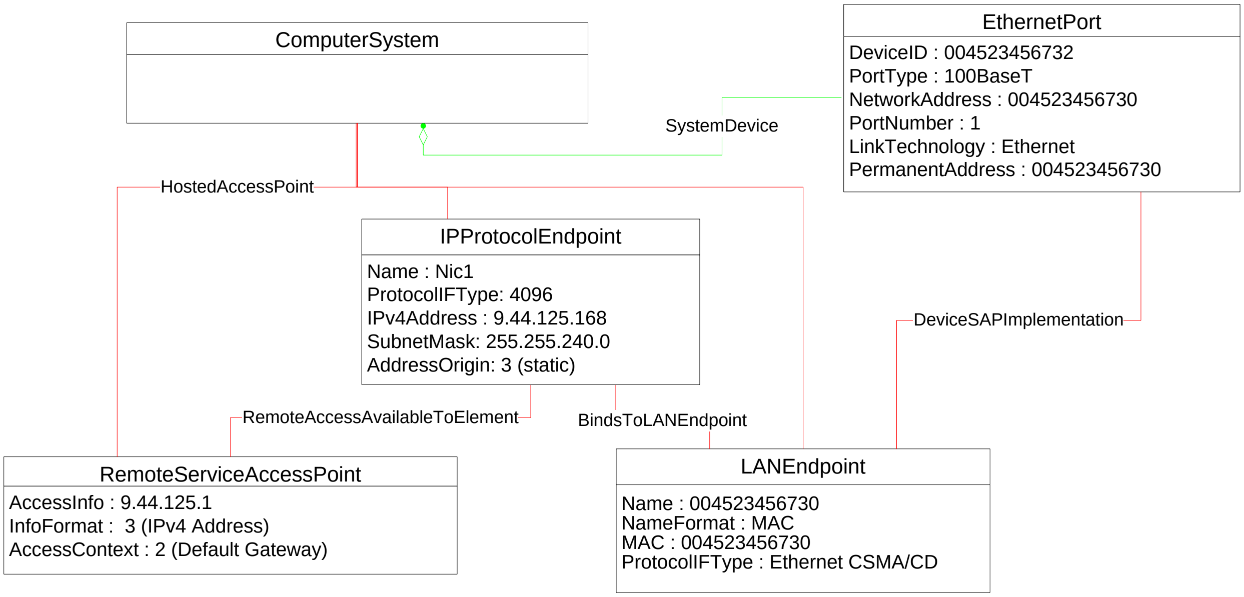

ElementConformsToProfile
ComputerSystem
profile2 : RegisteredProfile
RegisteredOrganization : 2 (DMTF)
RegisteredName : Base Server
RegisteredVersion : 1.0.0
HostedAccessPoint
ReferencedProfile
profile1 : RegisteredProfile
RegisteredOrganization : 2 (DMTF)
RegisteredName : IP Interface
RegisteredVersion : 1.0.0
IPProtocolEndpoint

**Figure 2 – Registered Profile**

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

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



954

955 **Figure 3 – Basic configuration — IPv4**

956 **EXPERIMENTAL**

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

961

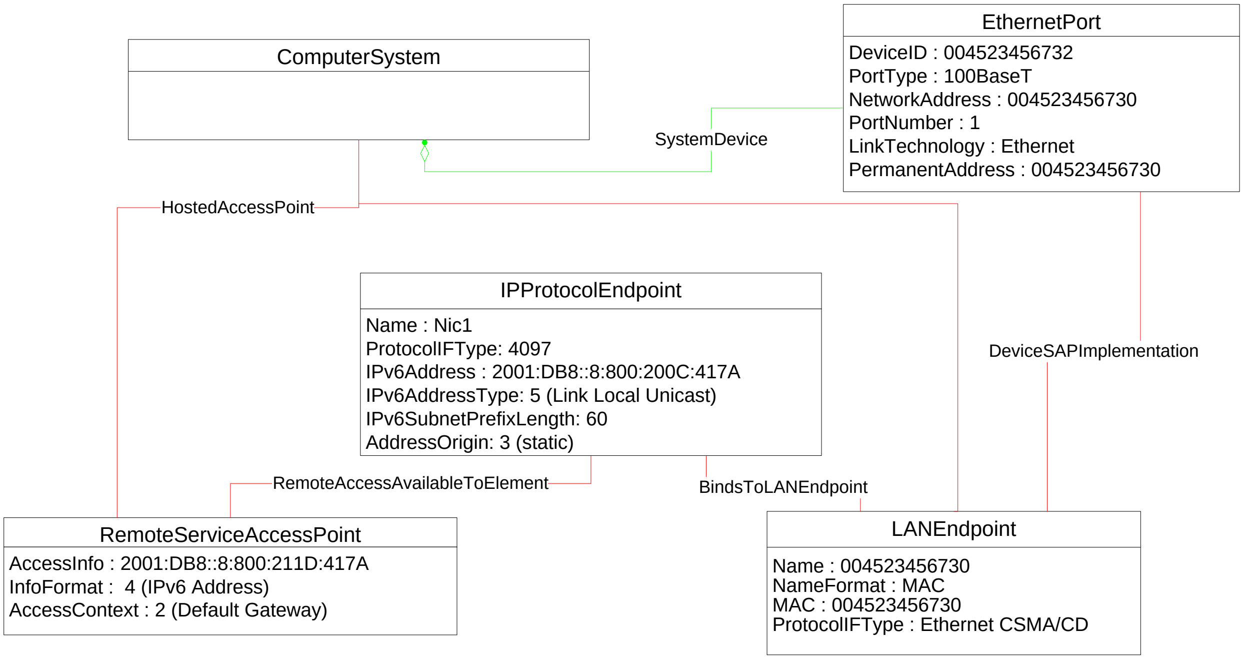

ComputerSystem
EthernetPort
DeviceID : 004523456732
PortType : 100BaseT
NetworkAddress : 004523456730
PortNumber : 1
LinkTechnology : Ethernet
PermanentAddress : 004523456730
SystemDevice
HostedAccessPoint
IPProtocolEndpoint
Name : Nic1
ProtocolIFType: 4097
IPv6Address : 2001:DB8::8:800:200C:417A
IPv6AddressType: 5 (Link Local Unicast)
IPv6SubnetPrefixLength: 60
AddressOrigin: 3 (static)
DeviceSAPImplementation
RemoteAccessAvailableToElement
BindsToLANEndpoint
RemoteServiceAccessPoint
AccessInfo : 2001:DB8::8:800:211D:417A
InfoFormat : 4 (IPv6 Address)
AccessContext : 2 (Default Gateway)
LANEndpoint
Name : 004523456730
NameFormat : MAC
MAC : 004523456730
ProtocolIFType : Ethernet CSMA/CD

**Figure 4 – Basic configuration — IPv6**

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

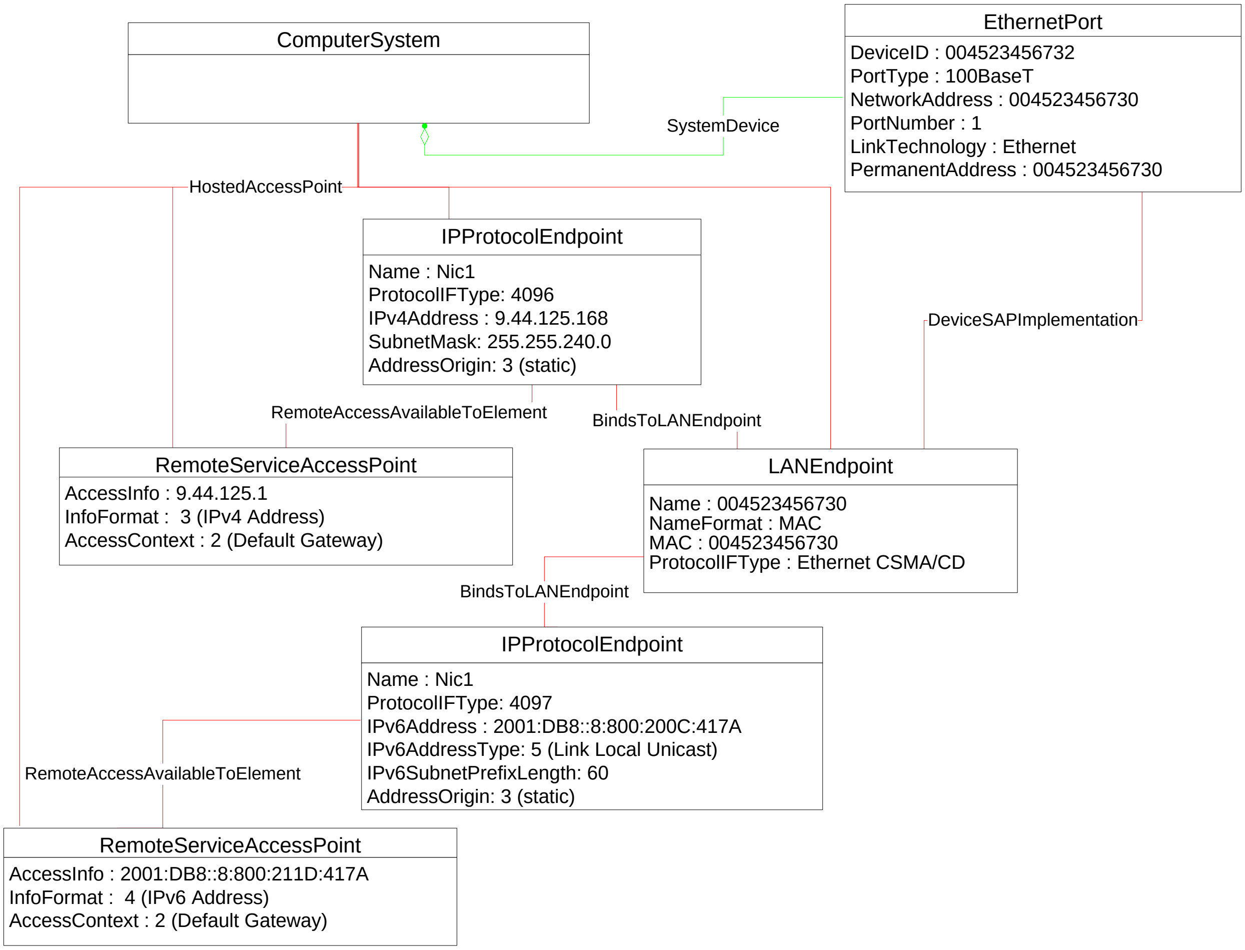

ComputerSystem
EthernetPort
DeviceID : 004523456732
PortType : 100BaseT
NetworkAddress : 004523456730
PortNumber : 1
LinkTechnology : Ethernet
PermanentAddress : 004523456730
SystemDevice
HostedAccessPoint
IPProtocolEndpoint
Name : Nic1
ProtocolIFType: 4096
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
AddressOrigin: 3 (static)
DeviceSAPImplementation
RemoteAccessAvailableToElement
BindsToLANEndpoint
RemoteServiceAccessPoint
AccessInfo : 9.44.125.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 2 (Default Gateway)
LANEndpoint
Name : 004523456730
NameFormat : MAC
MAC : 004523456730
ProtocolIFType : Ethernet CSMA/CD
BindsToLANEndpoint
IPProtocolEndpoint
Name : Nic1
ProtocolIFType: 4097
IPv6Address : 2001:DB8::8:800:200C:417A
IPv6AddressType: 5 (Link Local Unicast)
IPv6SubnetPrefixLength: 60
AddressOrigin: 3 (static)
RemoteAccessAvailableToElement
RemoteServiceAccessPoint
AccessInfo : 2001:DB8::8:800:211D:417A
InfoFormat : 4 (IPv6 Address)
AccessContext : 2 (Default Gateway)

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

## EXPERIMENTAL

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

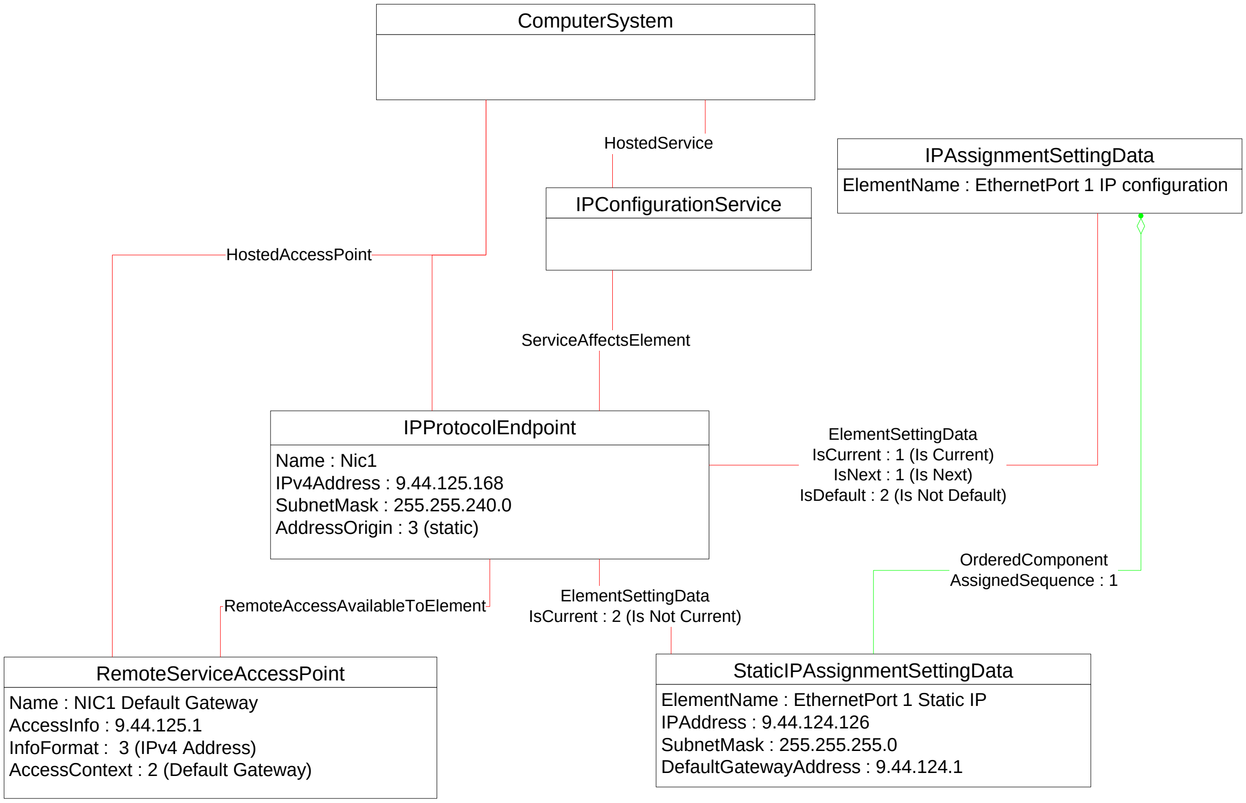

ComputerSystem
HostedService
IPAssignmentSettingData
ElementName : EthernetPort 1 IP configuration
IPConfigurationService
HostedAccessPoint
ServiceAffectsElement
IPProtocolEndpoint
Name : Nic1
IPv4Address : 9.44.125.168
SubnetMask : 255.255.240.0
AddressOrigin : 3 (static)
ElementSettingData
IsCurrent : 1 (Is Current)
IsNext : 1 (Is Next)
IsDefault : 2 (Is Not Default)
OrderedComponent
AssignedSequence : 1
RemoteAccessAvailableToElement
ElementSettingData
IsCurrent : 2 (Is Not Current)
RemoteServiceAccessPoint
Name : NIC1 Default Gateway
AccessInfo : 9.44.125.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 2 (Default Gateway)
StaticIPAssignmentSettingData
ElementName : EthernetPort 1 Static IP
IPAddress : 9.44.124.126
SubnetMask : 255.255.255.0
DefaultGatewayAddress : 9.44.124.1

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

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

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

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

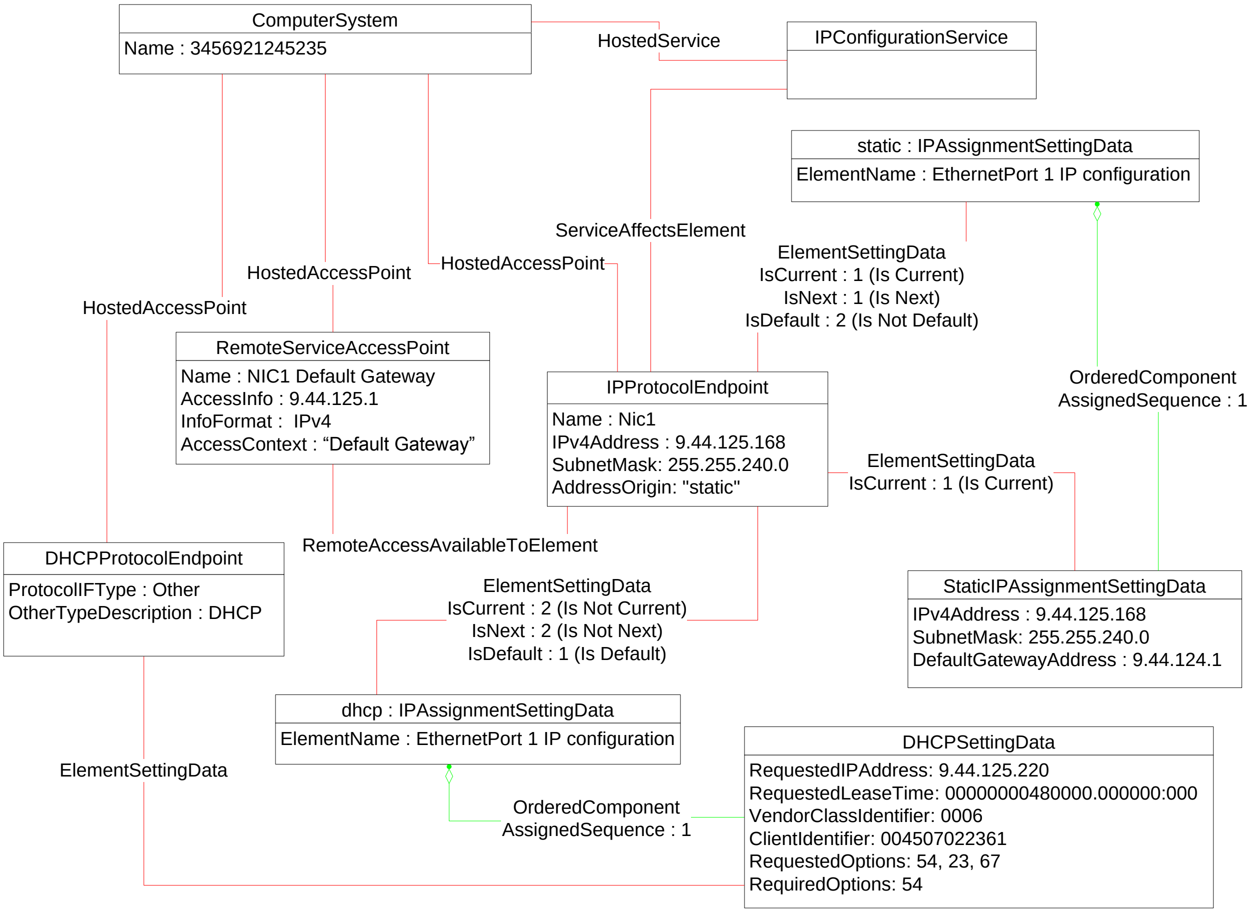

ComputerSystem
Name : 3456921245235
HostedService
IPConfigurationService
static : IPAssignmentSettingData
ElementName : EthernetPort 1 IP configuration
ServiceAffectsElement
ElementSettingData
IsCurrent : 1 (Is Current)
IsNext : 1 (Is Next)
IsDefault : 2 (Is Not Default)
HostedAccessPoint
HostedAccessPoint
HostedAccessPoint
RemoteServiceAccessPoint
Name : NIC1 Default Gateway
AccessInfo : 9.44.125.1
InfoFormat : IPv4
AccessContext : “Default Gateway”
IPProtocolEndpoint
Name : Nic1
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
AddressOrigin: "static"
OrderedComponent
AssignedSequence : 1
ElementSettingData
IsCurrent : 1 (Is Current)
RemoteAccessAvailableToElement
DHCPProtocolEndpoint
ProtocolIFType : Other
OtherTypeDescription : DHCP
StaticIPAssignmentSettingData
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
DefaultGatewayAddress : 9.44.124.1
ElementSettingData
IsCurrent : 2 (Is Not Current)
IsNext : 2 (Is Not Next)
IsDefault : 1 (Is Default)
dhcp : IPAssignmentSettingData
ElementName : EthernetPort 1 IP configuration
DHCPSettingData
RequestedIPAddress: 9.44.125.220
RequestedLeaseTime: 00000000480000.000000:000
VendorClassIdentifier: 0006
ClientIdentifier: 004507022361
RequestedOptions: 54, 23, 67
RequiredOptions: 54
ElementSettingData
OrderedComponent
AssignedSequence : 1

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

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

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

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

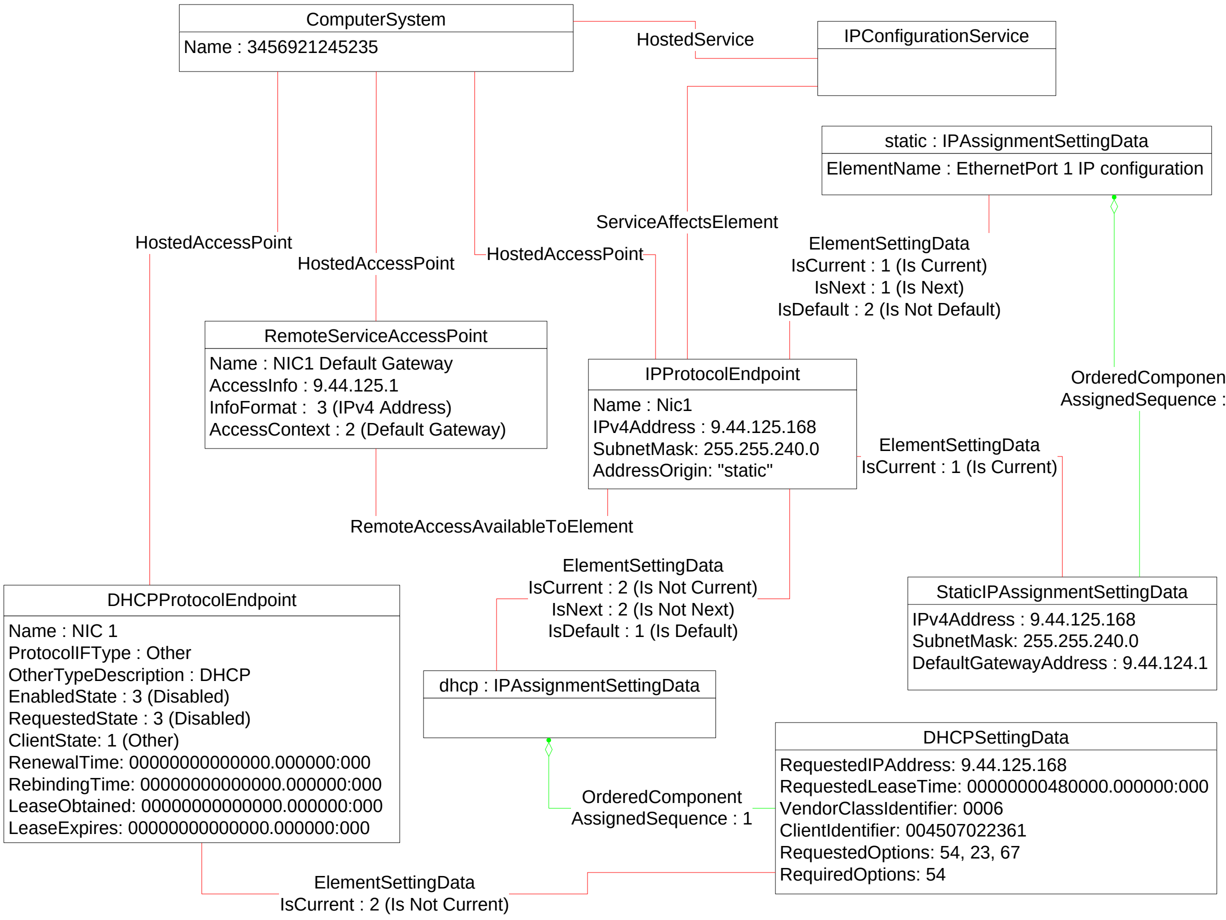

ComputerSystem
Name : 3456921245235
HostedService
IPConfigurationService
static : IPAssignmentSettingData
ElementName : EthernetPort 1 IP configuration
ServiceAffectsElement
HostedAccessPoint
HostedAccessPoint
HostedAccessPoint
ElementSettingData
IsCurrent : 1 (Is Current)
IsNext : 1 (Is Next)
IsDefault : 2 (Is Not Default)
RemoteServiceAccessPoint
Name : NIC1 Default Gateway
AccessInfo : 9.44.125.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 2 (Default Gateway)
IPProtocolEndpoint
Name : Nic1
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
AddressOrigin: "static"
OrderedComponent
AssignedSequence : 1
ElementSettingData
IsCurrent : 1 (Is Current)
RemoteAccessAvailableToElement
ElementSettingData
IsCurrent : 2 (Is Not Current)
IsNext : 2 (Is Not Next)
IsDefault : 1 (Is Default)
DHCPProtocolEndpoint
Name : NIC 1
ProtocolIFType : Other
OtherTypeDescription : DHCP
EnabledState : 3 (Disabled)
RequestedState : 3 (Disabled)
ClientState: 1 (Other)
RenewalTime: 00000000000000.000000:000
RebindingTime: 00000000000000.000000:000
LeaseObtained: 00000000000000.000000:000
LeaseExpires: 00000000000000.000000:000
StaticIPAssignmentSettingData
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
DefaultGatewayAddress : 9.44.124.1
dhcp : IPAssignmentSettingData
OrderedComponent
AssignedSequence : 1
DHCPSettingData
RequestedIPAddress: 9.44.125.168
RequestedLeaseTime: 00000000480000.000000:000
VendorClassIdentifier: 0006
ClientIdentifier: 004507022361
RequestedOptions: 54, 23, 67
RequiredOptions: 54
ElementSettingData
IsCurrent : 2 (Is Not Current)

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

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

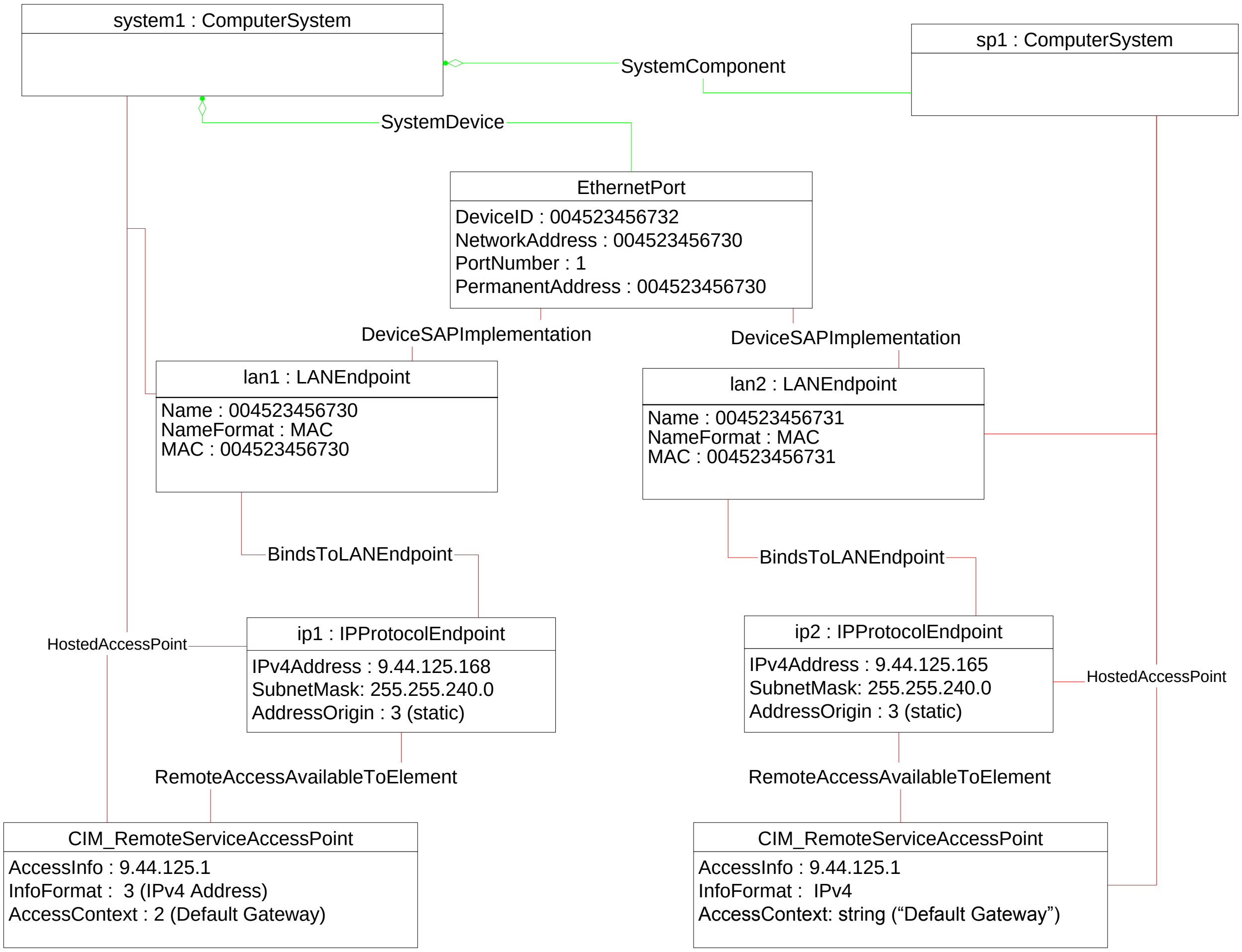


**Figure 9 – Service processor and server share an NIC**

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

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

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

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

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

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

dhcp_only represents a configuration that uses DHCP to obtain the IP configuration. This behavior is indicated by the aggregated instance dhcp1. The DNS configuration can be assigned through DHCP or statically assigned. This behavior is indicated by the aggregated instances dns1 and dnsgen1. In the event the DHCP client is unable to obtain a configuration, the system is implemented to default to a hard-coded, well-known default static IP configuration. The existence of a default configuration is indicated by the aggregated instance static3. Note that no advertisement mechanism is specified in the profile to indicate that static3 represents hard-coded values that cannot be modified by the client. If the system were implemented such that the DHCP client would be continually in use without a timeout to a static configuration, the aggregated instance static3 would not exist.

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

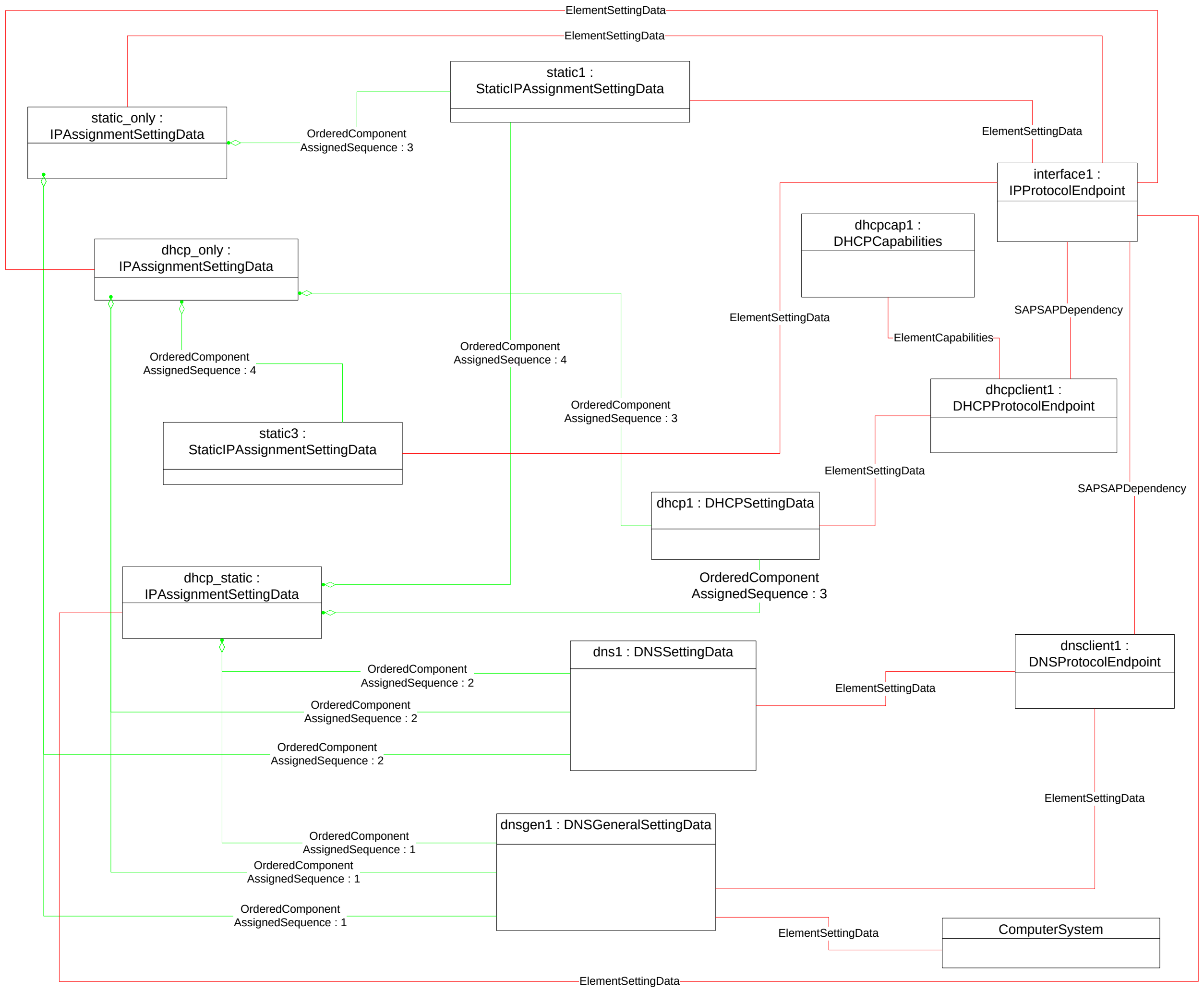

ElementSettingData
ElementSettingData
static1 : StaticIPAssignmentSettingData
static_only : IPAssignmentSettingData
OrderedComponent AssignedSequence : 3
ElementSettingData
interface1 : IPProtocolEndpoint
dhcpcap1 : DHCPCapabilities
dhcp_only : IPAssignmentSettingData
ElementSettingData
SAPSAPDependency
ElementCapabilities
OrderedComponent AssignedSequence : 4
OrderedComponent AssignedSequence : 4
OrderedComponent AssignedSequence : 3
dhcpclient1 : DHCPProtocolEndpoint
static3 : StaticIPAssignmentSettingData
ElementSettingData
SAPSAPDependency
dhcp1 : DHCPSettingData
dhcp_static : IPAssignmentSettingData
OrderedComponent AssignedSequence : 3
dns1 : DNSSettingData
dnsclient1 : DNSProtocolEndpoint
OrderedComponent AssignedSequence : 2
ElementSettingData
OrderedComponent AssignedSequence : 2
OrderedComponent AssignedSequence : 2
ElementSettingData
dnsgen1 : DNSGeneralSettingData
OrderedComponent AssignedSequence : 1
OrderedComponent AssignedSequence : 1
OrderedComponent AssignedSequence : 1
ElementSettingData
ComputerSystem
ElementSettingData

**Figure 10 – Configuration choices**

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

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

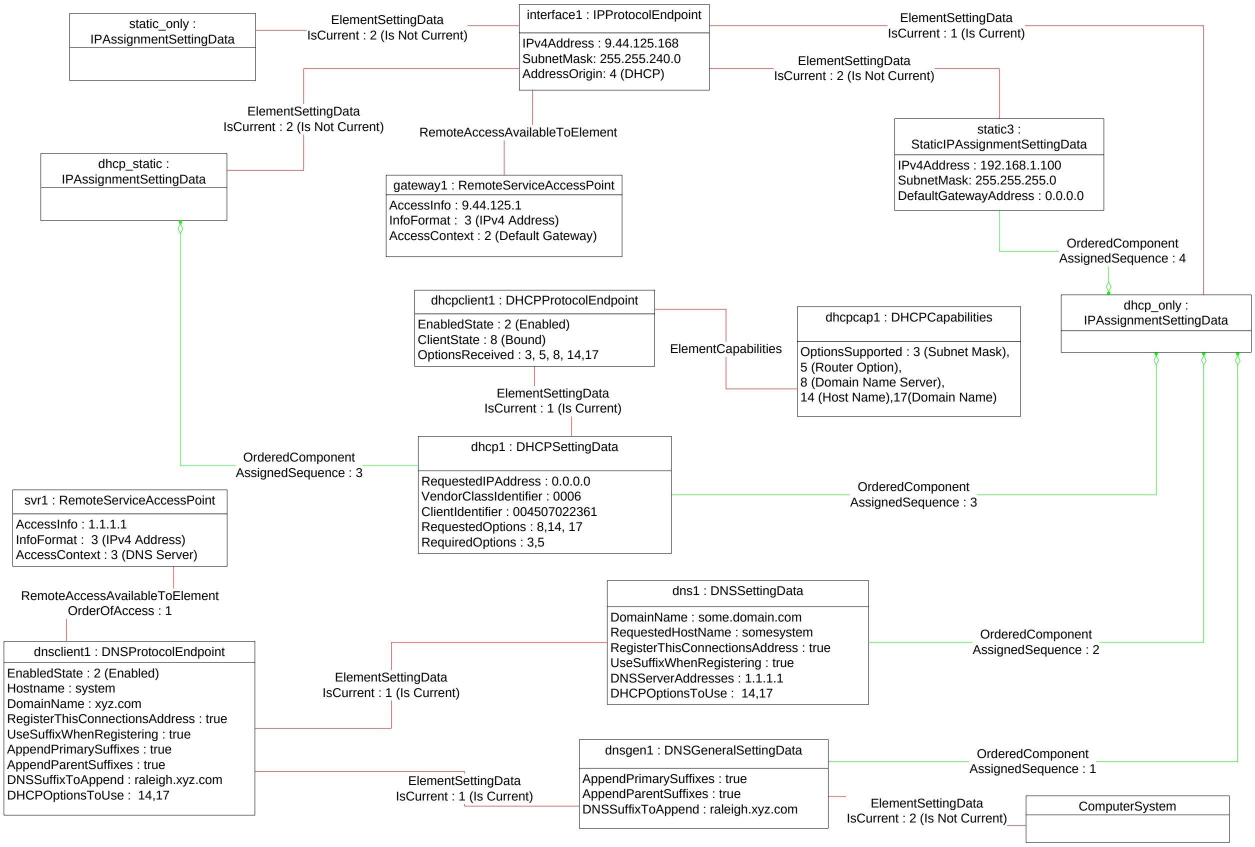

static_only : IPAssignmentSettingData
ElementSettingData IsCurrent : 2 (Is Not Current)
interface1 : IPProtocolEndpoint
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
AddressOrigin: 4 (DHCP)
ElementSettingData IsCurrent : 1 (Is Current)
ElementSettingData IsCurrent : 2 (Is Not Current)
ElementSettingData IsCurrent : 2 (Is Not Current)
RemoteAccessAvailableToElement
dhcp_static : IPAssignmentSettingData
gateway1 : RemoteServiceAccessPoint
AccessInfo : 9.44.125.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 2 (Default Gateway)
static3 : StaticIPAssignmentSettingData
IPv4Address : 192.168.1.100
SubnetMask: 255.255.255.0
DefaultGatewayAddress : 0.0.0.0
OrderedComponent AssignedSequence : 4
dhcpclient1 : DHCPProtocolEndpoint
EnabledState : 2 (Enabled)
ClientState : 8 (Bound)
OptionsReceived : 3, 5, 8, 14,17
ElementCapabilities
dhcpcap1 : DHCPCapabilities
OptionsSupported : 3 (Subnet Mask),
5 (Router Option),
8 (Domain Name Server),
14 (Host Name),17(Domain Name)
dhcp_only : IPAssignmentSettingData
ElementSettingData IsCurrent : 1 (Is Current)
OrderedComponent AssignedSequence : 3
dhcp1 : DHCPSettingData
RequestedIPAddress : 0.0.0.0
VendorClassIdentifier : 0006
ClientIdentifier : 004507022361
RequestedOptions : 8,14, 17
RequiredOptions : 3,5
OrderedComponent AssignedSequence : 3
svr1 : RemoteServiceAccessPoint
AccessInfo : 1.1.1.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 3 (DNS Server)
RemoteAccessAvailableToElement OrderOfAccess : 1
dns1 : DNSSettingData
DomainName : some.domain.com
RequestedHostName : somesystem
RegisterThisConnectionsAddress : true
UseSuffixWhenRegistering : true
DNSServerAddresses : 1.1.1.1
DHCPOptionsToUse : 14,17
OrderedComponent AssignedSequence : 2
dnsclient1 : DNSProtocolEndpoint
EnabledState : 2 (Enabled)
Hostname : system
DomainName : xyz.com
RegisterThisConnectionsAddress : true
UseSuffixWhenRegistering : true
AppendPrimarySuffixes : true
AppendParentSuffixes : true
DNSSuffixToAppend : raleigh.xyz.com
DHCPOptionsToUse : 14,17
ElementSettingData IsCurrent : 1 (Is Current)
dnsgen1 : DNSGeneralSettingData
AppendPrimarySuffixes : true
AppendParentSuffixes : true
DNSSuffixToAppend : raleigh.xyz.com
OrderedComponent AssignedSequence : 1
ElementSettingData IsCurrent : 1 (Is Current)
ElementSettingData IsCurrent : 2 (Is Not Current)
ComputerSystem

**Figure 11 – DHCP assigned partial DNS**

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

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

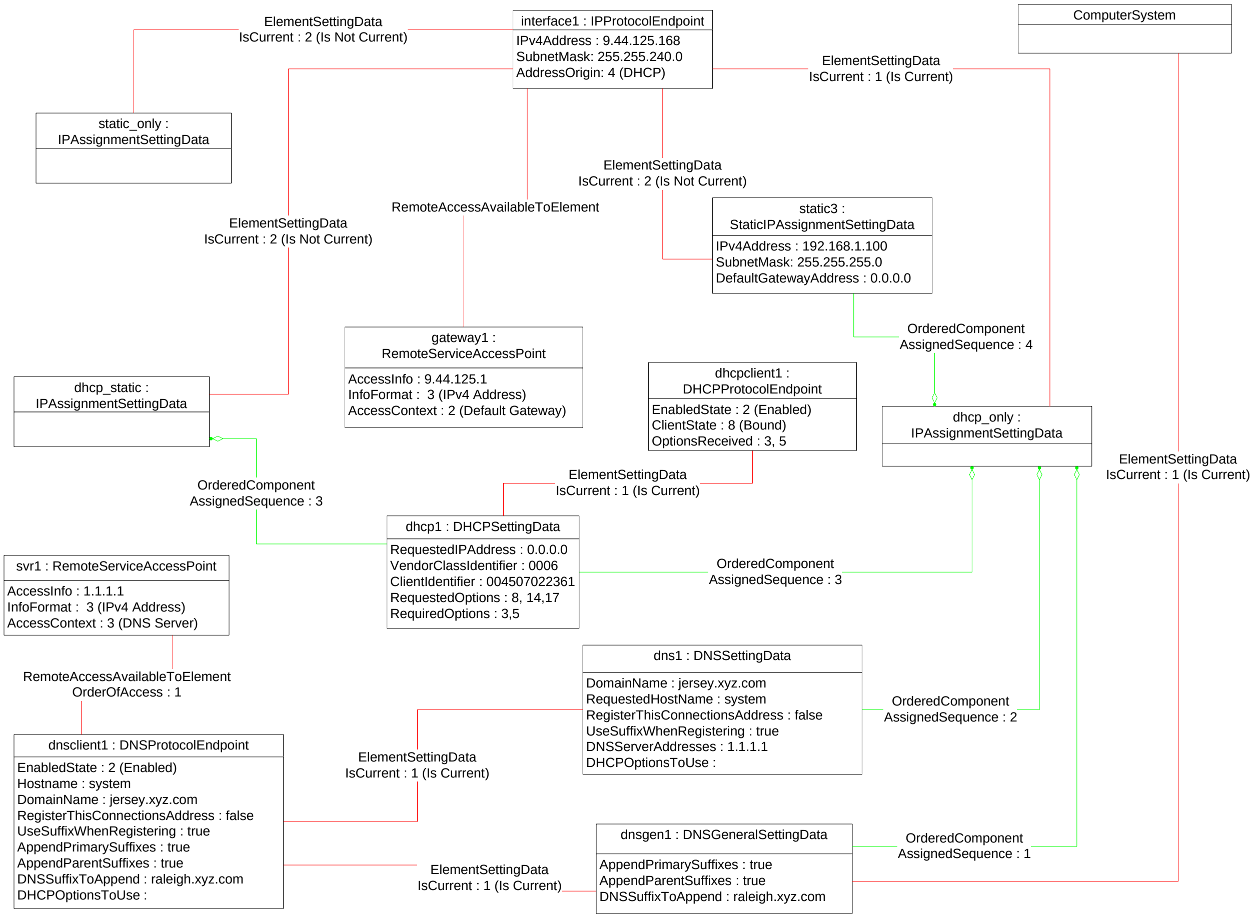

interface1 : IPProtocolEndpoint
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
AddressOrigin: 4 (DHCP)
ComputerSystem
ElementSettingData
IsCurrent : 2 (Is Not Current)
ElementSettingData
IsCurrent : 1 (Is Current)
static_only :
IPAssignmentSettingData
ElementSettingData
IsCurrent : 2 (Is Not Current)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

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

### 9.1.1 Sequence for disabled DNS client

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

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

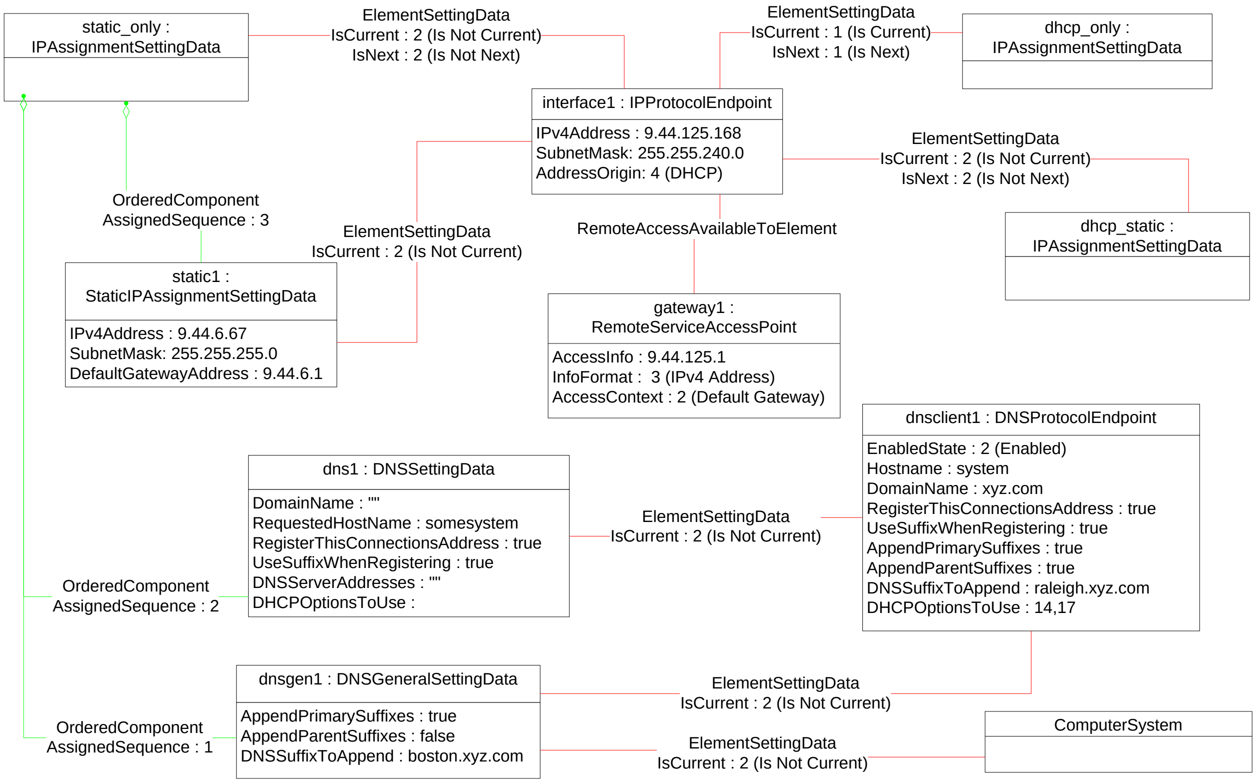

static_only :
IPAssignmentSettingData
ElementSettingData
IsCurrent : 2 (Is Not Current)
IsNext : 2 (Is Not Next)
ElementSettingData
IsCurrent : 1 (Is Current)
IsNext : 1 (Is Next)
dhcp_only :
IPAssignmentSettingData
interface1 : IPProtocolEndpoint
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
AddressOrigin: 4 (DHCP)
ElementSettingData
IsCurrent : 2 (Is Not Current)
IsNext : 2 (Is Not Next)
dhcp_static :
IPAssignmentSettingData
OrderedComponent
AssignedSequence : 3
ElementSettingData
IsCurrent : 2 (Is Not Current)
RemoteAccessAvailableToElement
static1 :
StaticIPAssignmentSettingData
IPv4Address : 9.44.6.67
SubnetMask: 255.255.255.0
DefaultGatewayAddress : 9.44.6.1
gateway1 :
RemoteServiceAccessPoint
AccessInfo : 9.44.125.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 2 (Default Gateway)
dnsclient1 : DNSProtocolEndpoint
EnabledState : 2 (Enabled)
Hostname : system
DomainName : xyz.com
RegisterThisConnectionsAddress : true
UseSuffixWhenRegistering : true
AppendPrimarySuffixes : true
AppendParentSuffixes : true
DNSSuffixToAppend : raleigh.xyz.com
DHCPOptionsToUse : 14,17
dns1 : DNSSettingData
DomainName : ""
RequestedHostName : somesystem
RegisterThisConnectionsAddress : true
UseSuffixWhenRegistering : true
DNSServerAddresses : ""
DHCPOptionsToUse :
ElementSettingData
IsCurrent : 2 (Is Not Current)
OrderedComponent
AssignedSequence : 2
dnsgen1 : DNSGeneralSettingData
AppendPrimarySuffixes : true
AppendParentSuffixes : false
DNSSuffixToAppend : boston.xyz.com
ElementSettingData
IsCurrent : 2 (Is Not Current)
OrderedComponent
AssignedSequence : 1
ElementSettingData
IsCurrent : 2 (Is Not Current)
ComputerSystem

**Figure 13 – Static without DNS configuration — One**

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

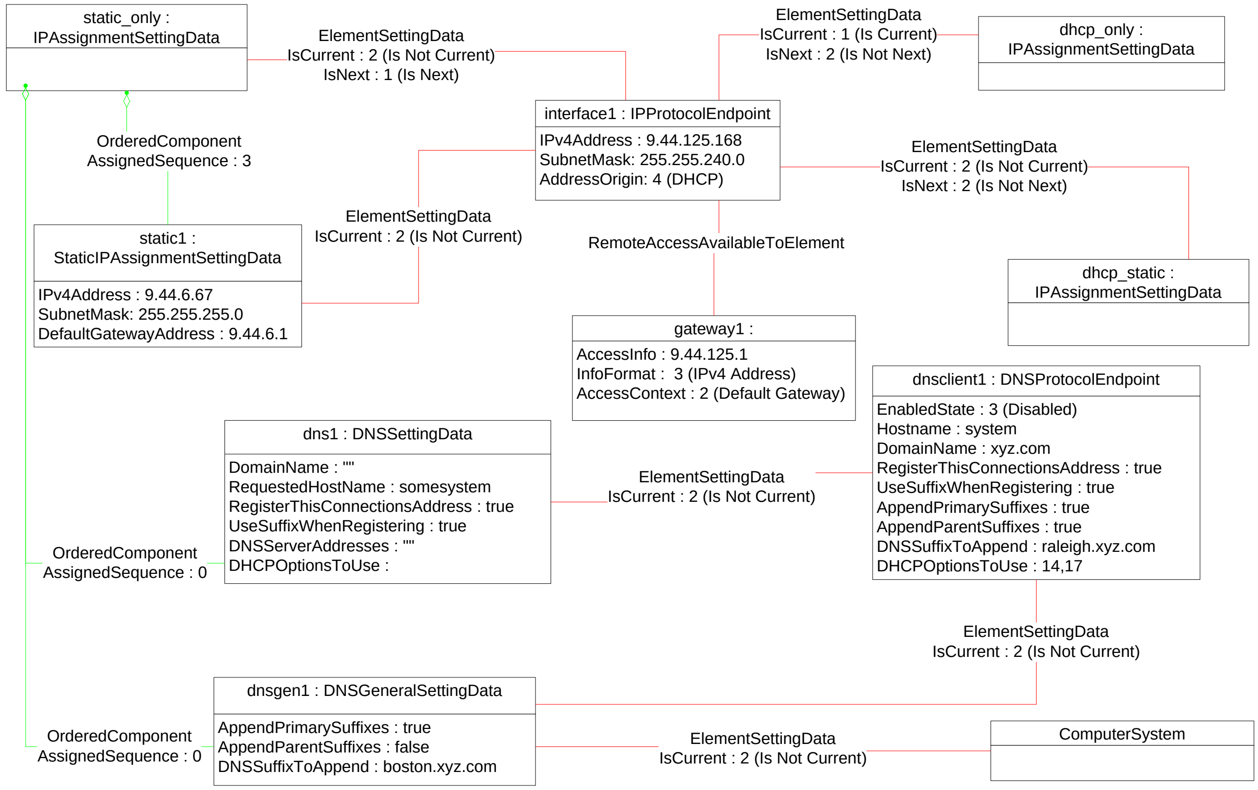

static_only : IPAssignmentSettingData
ElementSettingData
IsCurrent : 2 (Is Not Current)
IsNext : 1 (Is Next)
ElementSettingData
IsCurrent : 1 (Is Current)
IsNext : 2 (Is Not Next)
dhcp_only : IPAssignmentSettingData
interface1 : IPProtocolEndpoint
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
AddressOrigin: 4 (DHCP)
OrderedComponent
AssignedSequence : 3
ElementSettingData
IsCurrent : 2 (Is Not Current)
IsNext : 2 (Is Not Next)
ElementSettingData
IsCurrent : 2 (Is Not Current)
static1 : StaticIPAssignmentSettingData
IPv4Address : 9.44.6.67
SubnetMask: 255.255.255.0
DefaultGatewayAddress : 9.44.6.1
RemoteAccessAvailableToElement
dhcp_static : IPAssignmentSettingData
gateway1 :
AccessInfo : 9.44.125.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 2 (Default Gateway)
dnsclient1 : DNSProtocolEndpoint
EnabledState : 3 (Disabled)
Hostname : system
DomainName : xyz.com
RegisterThisConnectionsAddress : true
UseSuffixWhenRegistering : true
AppendPrimarySuffixes : true
AppendParentSuffixes : true
DNSSuffixToAppend : raleigh.xyz.com
DHCPOptionsToUse : 14,17
dns1 : DNSSettingData
DomainName : ""
RequestedHostName : somesystem
RegisterThisConnectionsAddress : true
UseSuffixWhenRegistering : true
DNSServerAddresses : ""
DHCPOptionsToUse :
ElementSettingData
IsCurrent : 2 (Is Not Current)
OrderedComponent
AssignedSequence : 0
ElementSettingData
IsCurrent : 2 (Is Not Current)
dnsgen1 : DNSGeneralSettingData
AppendPrimarySuffixes : true
AppendParentSuffixes : false
DNSSuffixToAppend : boston.xyz.com
OrderedComponent
AssignedSequence : 0
ElementSettingData
IsCurrent : 2 (Is Not Current)
ComputerSystem

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

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

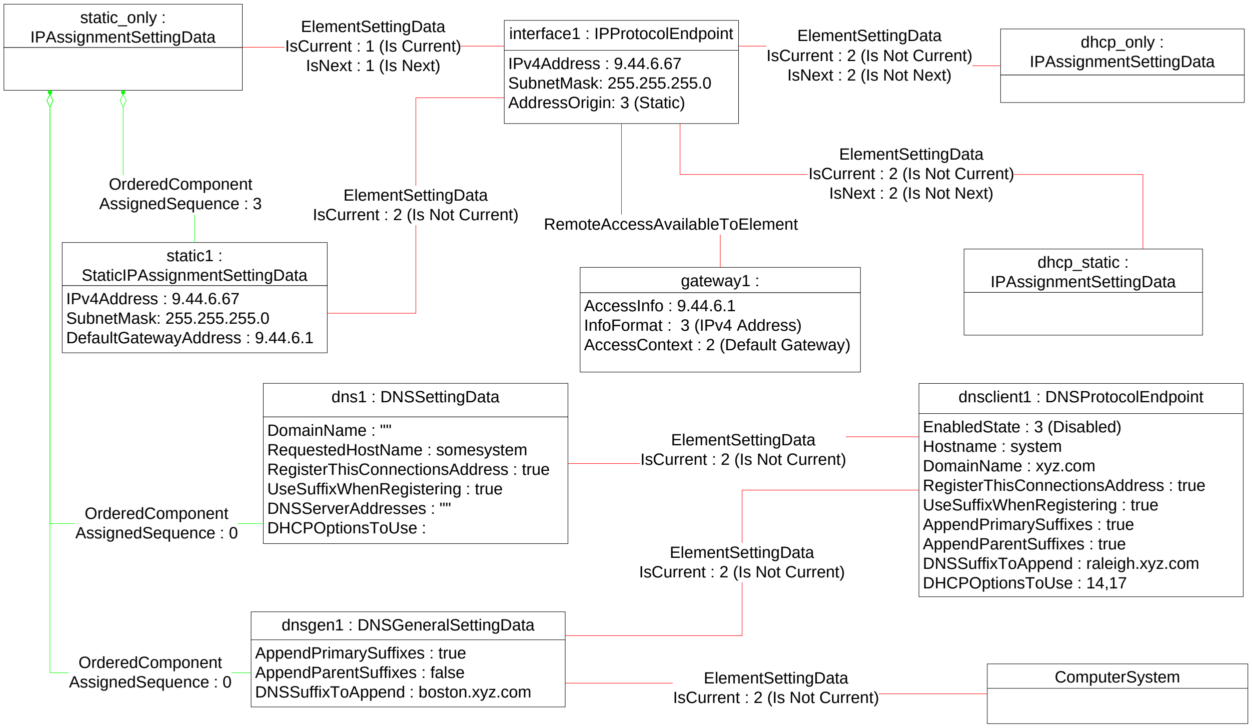

static_only : IPAssignmentSettingData
ElementSettingData
IsCurrent : 1 (Is Current)
IsNext : 1 (Is Next)
interface1 : IPProtocolEndpoint
IPv4Address : 9.44.6.67
SubnetMask: 255.255.255.0
AddressOrigin: 3 (Static)
ElementSettingData
IsCurrent : 2 (Is Not Current)
IsNext : 2 (Is Not Next)
dhcp_only : IPAssignmentSettingData
OrderedComponent
AssignedSequence : 3
ElementSettingData
IsCurrent : 2 (Is Not Current)
RemoteAccessAvailableToElement
ElementSettingData
IsCurrent : 2 (Is Not Current)
IsNext : 2 (Is Not Next)
static1 : StaticIPAssignmentSettingData
IPv4Address : 9.44.6.67
SubnetMask: 255.255.255.0
DefaultGatewayAddress : 9.44.6.1
gateway1 :
AccessInfo : 9.44.6.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 2 (Default Gateway)
dhcp_static : IPAssignmentSettingData
dns1 : DNSSettingData
DomainName : ""
RequestedHostName : somesystem
RegisterThisConnectionsAddress : true
UseSuffixWhenRegistering : true
DNSServerAddresses : ""
DHCPOptionsToUse :
ElementSettingData
IsCurrent : 2 (Is Not Current)
dnsclient1 : DNSProtocolEndpoint
EnabledState : 3 (Disabled)
Hostname : system
DomainName : xyz.com
RegisterThisConnectionsAddress : true
UseSuffixWhenRegistering : true
AppendPrimarySuffixes : true
AppendParentSuffixes : true
DNSSuffixToAppend : raleigh.xyz.com
DHCPOptionsToUse : 14,17
OrderedComponent
AssignedSequence : 0
ElementSettingData
IsCurrent : 2 (Is Not Current)
dnsgen1 : DNSGeneralSettingData
AppendPrimarySuffixes : true
AppendParentSuffixes : false
DNSSuffixToAppend : boston.xyz.com
OrderedComponent
AssignedSequence : 0
ElementSettingData
IsCurrent : 2 (Is Not Current)
ComputerSystem

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

## 9.2 Determine supported configuration methods

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

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

2) For each instance of CIM_IPAssignmentSettingData:

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

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

## 9.3 Determine gateway address

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

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

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

## 9.4 Determine method used for current configuration

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

## 9.8 Establish a static IP configuration for an interface

A client can manually assign an IP configuration to an interface 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_StaticIPAssignmentSettingData that is associated through an instance of CIM_OrderedComponent.

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

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

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

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

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

## 9.9 Apply a pending configuration — synchronously

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

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

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

## 9.10 Apply a pending configuration — upon restart

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

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

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

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

## 9.11 Determine whether DNS configuration was DHCP assigned

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

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

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

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

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

## 9.12 Determine whether ElementName can be modified

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

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

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

## 9.13 Determine whether state management is supported

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

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

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

# 10 CIM Elements

Table 16 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 16. Clauses 7 (“Implementation”) and 8 (“Methods”) may impose additional requirements on these elements.

**Table 16 – 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 10.43. |
| 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 17 provides information about the properties of CIM_BindsToLANEndpoint.

**Table 17 – Class: CIM_BindsToLANEndpoint**

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

## 10.2 CIM_ElementCapabilities

CIM_ElementCapabilities associates an instance of CIM_EnabledLogicalElementCapabilities with the CIM_IPProtocolEndpoint instance. Table 18 provides information about the properties of CIM_ElementCapabilities.

**Table 18 – Class: CIM_ElementCapabilities**

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

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

## 10.3 CIM_ElementSettingData — CIM_IPAssignmentSettingData reference

CIM_ElementSettingData associates instances of CIM_IPAssignmentSettingData with the CIM_IPProtocolEndpoint instance. Table 19 provides information about the properties of CIM_ElementSettingData.

**Table 19 – Class: CIM_ElementSettingData — CIM_IPAssignmentSettingData**

| Elements | Requirement | Description |
|---|---|---|
| ManagedElement | Mandatory | **Key:** This shall be a reference to the Central Instance.<br>Cardinality 1..* |
| SettingData | Mandatory | **Key:** This shall be a reference to an instance of CIM_IPAssignmentSettingData.<br>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

CIM_ElementSettingData associates instances of CIM_StaticIPAssignmentSettingData with the CIM_IPProtocolEndpoint instance. Table 20 provides information about the properties of CIM_ElementSettingData.

**Table 20 – Class: CIM_ElementSettingData — CIM_StaticIPAssignmentSettingData**

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

## 10.5 CIM_EnabledLogicalElementCapabilities

CIM_EnabledLogicalElementCapabilities indicates support for managing the IP interface. Table 21 provides information about the properties of CIM_EnabledLogicalElementCapabilities.

**Table 21 – 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. |

## 10.6 CIM_HostedAccessPoint — CIM_RemoteServiceAccessPoint reference

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

CIM_HostedAccessPoint relates the CIM_RemoteServiceAccessPoint instance that represents the default gateway with its scoping CIM_ComputerSystem instance. Table 22 provides information about the properties of CIM_HostedAccessPoint.

**Table 22 – Class: CIM_HostedAccessPoint — CIM_RemoteServiceAccessPoint**

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

## 10.7 CIM_HostedAccessPoint — CIM_IPProtocolEndpoint reference

CIM_HostedAccessPoint relates the Central Instance with its Scoping Instance. Table 23 provides information about the properties of CIM_HostedAccessPoint.

**Table 23 – Class: CIM_HostedAccessPoint — CIM_IPProtocolEndpoint**

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

## 10.8 CIM_HostedService

CIM_HostedService relates the CIM_IPConfigurationService instance to its scoping CIM_ComputerSystem instance. Table 24 provides information about the properties of CIM_HostedService.

**Table 24 – Class: CIM_HostedService**

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

## 10.9 CIM_IPAssignmentSettingData

CIM_IPAssignmentSettingData is the aggregation point for the SettingData instances that define a configuration that can be applied to an IP interface. Table 25 provides information about the properties of CIM_IPAssignmentSettingData.

**Table 25 – Class: CIM_IPAssignmentSettingData**

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

## 10.10 CIM_IPConfigurationService

CIM_IPConfigurationService represents the ability to configure an IP interface. Table 26 provides information about the properties of CIM_IPConfigurationService.

**Table 26 – 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. |

## 10.11 CIM_IPProtocolEndpoint

CIM_IPProtocolEndpoint represents an IP interface that is associated with an Ethernet interface. Table 27 provides information about the properties of CIM_IPProtocolEndpoint.

**Table 27 – 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 – EXPERIMENTAL |
| IPv6AddressType | Conditional | See 7.1.1.2 – EXPERIMENTAL |
| IPv6SubnetPrefixLength | Conditional | See 7.1.1.2 – EXPERIMENTAL |

## 10.12 CIM_OrderedComponent

CIM_OrderedComponent associates an instance of CIM_IPAssignmentSettingData to the instances of CIM_StaticIPAssignmentSettingData, CIM_DHCPSettingData, CIM_DNSSettingData, and CIM_DNSGeneralSettingData that compose a configuration. Table 28 provides information about the properties of CIM_OrderedComponent.

**Table 28 – 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 29, the behavior of the CIM_RegisteredProfile instance is in accordance with the *Profile Registration Profile*.

**Table 29 – Class: CIM_RegisteredProfile**

| Elements | Requirement | Description |
|---|---|---|
| RegisteredName | Mandatory | This property shall have a value of "IP Interface". |
| RegisteredVersion | Mandatory | This property shall have a value of "1.0.3". |
| 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 30 provides information about the properties of CIM_RemoteAccessAvailableToElement.

**Table 30 – 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 31 provides information about the properties of CIM_RemoteServiceAccessPoint.

**Table 31 – 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 ".*" |

## 10.16 CIM_ServiceAffectsElement

CIM_ServiceAffectsElement associates an instance of CIM_IPConfigurationService with an instance of CIM_IPProtocolEndpoint that the service is able to configure. Table 32 provides information about the properties of CIM_ServiceAffectsElement.

**Table 32 – Class: CIM_ServiceAffectsElement**

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

## 10.17 CIM_StaticIPAssignmentSettingData

CIM_StaticIPAssignmentSettingData represents a static configuration that can be applied to an instance of CIM_IPProtocolEndpoint. Table 33 provides information about the properties of CIM_StaticIPAssignmentSettingData.

**Table 33 – Class: CIM_StaticIPAssignmentSettingData**

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

# ANNEX A
# (informative)

# 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 |
| 1.0.2 | 2010-08-02 | Version 1.0.2 of the Final Standard formatted for DMTF Standard release |
| 1.0.3 | 2018-12-18 | This errata addresses these issues:<br>• Updated RegisteredVersion to reflect errata version number in clause 10.2<br>• Updated RegisteredOrganization description to reflect correct value for DMTF in clause 10.2.<br>• Updated CIM_Elements table in clause 10 with additional instances and relevant clarification in clause 7<br>Updated use cases to reflect the above fixes. |