Document Identifier: DSP1063

Date: 2018-06-11

Version: 1.0.0

Network Management Layer3 Interface Profile

Supersedes: None

Document Class: Normative

Document Status: Published

Document Language: en-US

CONTENTS

Figures

Tables

Foreword

The *Network Management Layer3* Interface Profile (DSP1063) was prepared by the Network Services Management 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:

Editors:

- John Parchem – Microsoft Corporation – DMTF Fellow

Contributors:

- John Crandall – Brocade Communications System
- Dr. Bhumip Khasnabish - ZTE Corporation
- Lawrence Lamers – VMware
- John Leung – Intel
- Steve Neely – Cisco Systems
- Shishir Pardikar – Citrix
- Hemal Shah – Broadcom Corporation
- Alex Zhdankin – Cisco Systems

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 Network Services and the 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

Typographical conventions

The following typographical conventions are used in this document:

- Document titles are marked in *italics*.
- ABNF rules are in `monospaced font`.

Network Management Layer3 Interface Profile

1 Scope

The *Network Management Layer3 Interface Profile* is a profile that specifies the CIM schema and use cases associated with the general and common aspects of typical layer 3 interfaces found in an Ethernet Switch. This profile includes a specification of the Layer 3 interface configuration service, Sub-Interface, IP Tunnel Interface, switch virtual interface and loopback interface.

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.

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 DSP0223, *Generic Operations 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0223_1.0.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 DSP1097, *Virtual Ethernet Switch Profile 1.1*,
http://dmtf.org/sites/default/files/standards/documents/DSP1097_1.1.0.pdf

DMTF DSP1036 *IP Interface Profile 1.1.1,*
http://www.dmtf.org/sites/default/files/standards/documents/DSP1036_1.1.1.pdf

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

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

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

The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 5.

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

The terms defined in DSP0004, DSP0223, and DSP1001 apply to this document. The following additional terms are used in this document.

3.1
conditional

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

3.2
mandatory

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

3.3
optional

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

3.4
pending configuration

indicates the configuration that will be applied to an IP network connection the next time the IP network connection accepts a configuration

3.5
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

3.6
unspecified

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

4 Symbols and abbreviated terms

The abbreviations defined in DSP0004, DSP0223, and DSP1001 apply to this document. The following additional abbreviations are used in this document.

4.1
IP

Internet Protocol

4.2
VLAN

Virtual Local Area Network

4.3
VSI
Virtual Switch Interface

5 Synopsis

Profile name: Network Management Layer3 Interface Profile

Version: 1.0.0

Organization: DMTF

CIM Schema version: 2.51

Central class: CIM_IPConfigurationService

Scoping class: CIM_System

The *Network Management Layer3 Interface Profile* is a profile that specifies the CIM schema and use cases associated with managing the IP layer 3 interfaces in an Ethernet switch. This profile includes a specification for configuration and life cycle management of the IP configuration of an Ethernet switch port, Subinterfaces, Switch Virtual Interfaces, Loopback and IP tunnel interfaces.

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
Virtual Ethernet Switch	DMTF	1.1	Mandatory	None
IP Configuration Profile	DMTF	1.1	Optional	None
IP Interface Profile	DMTF	1.1.1	Mandatory	None
Host LAN Network	DMTF	1.0.2	Mandatory	None
Network Management	DMTF	1.0	Optional	None
Network Management Routing	DMTF	1.0	Optional	None

6 Description

The *Network Management Layer3 Interface Profile* is a profile that will specify the CIM schema and use cases associated with the general and common aspects of creating and configuring layer 3 interfaces in a typical Ethernet switch. These interfaces include IP configuration of an Ethernet switch port, Subinterfaces, Switch Virtual Interfaces, Loopback and IP tunnel interfaces.

6.1 Class diagram

Figure 1 represents the class schema for the *Network Management Layer3 Interface Profile*. For simplicity, the CIM_ prefix has been removed from the names of the classes.

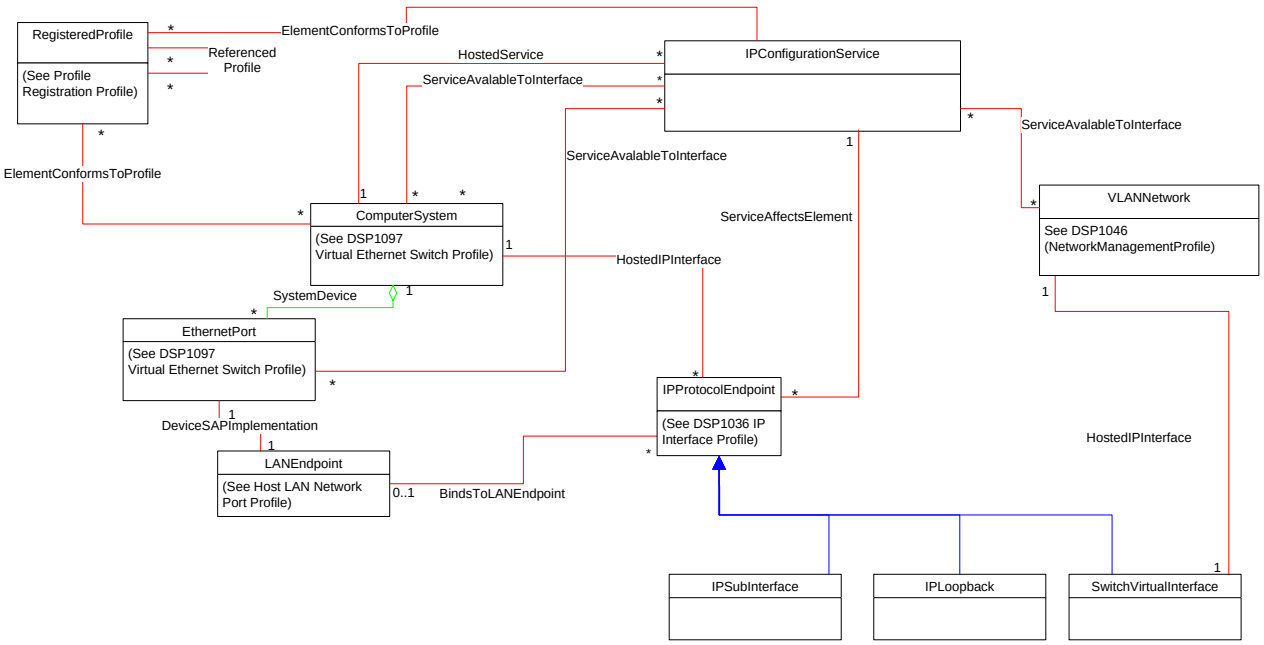


Figure 1 – Network Management Layer3 Interface Profile: Class diagram

Figure 1 is a class diagram for the network layer 3 interfaces within the context of an Ethernet switch represented as a DSP1097 Virtual Ethernet Switch Profile compliant switch. The class IPProtocolEndpoint and the subclasses of IPProtocolEndpoint, CIM_Subinterface, CIM_Loopback and CIM_SwitchVirtualInterface all represent the management aspects the typical layer 3 interfaces found in an Ethernet switch to facilitate IP routing capabilities.

6.2 CIM_IPProtocolEndpoint

In this profile the CIM_IPProtocolEndpoint is used to provide IP configuration for the Ethernet ports in the switch. It is also the super class for all of the Layer 3 interfaces in the switch. The CIM_IPProtocolEndpoints that are representing switch interfaces are associated through an instance of CIM_HostedIPInterface either to the CIM_ComputerSystem instance representing the scoping class or to a Network class or subclass such as CIM_VLANNetwork for the case of a switch virtual interface.

6.3 CIM_SwitchVirtualInterface

A switch virtual interface allows IP routing across VLANS. A CIM_VLANNetwork instance can only have one CIM_SwitchVirtualInterface instance associated to it through an instance of CIM_HostedIPInterface.

6.4 CIM_IPSubinterface

An IPSubinterface subdivides a single switch port into multiple IP subnets. This is typically done using Dot1Q encapsulation using VLANIds to distinguish the subnets. Even though an IPSubinterface may have a VLANId within the scoped router this is a layer 3 interface and this interface is not a part of an internal VLANNetwork with the same VLANId.

6.5 CIM_IPLoopback

A loopback interface is a virtual Layer 3 interface typically found in an Ethernet Switch or router. It is has a single endpoint that is always up. Packets that are transmitted over a loopback interface are immediately received by this interface.

6.6 CIM_HostedIPInterface

An association allowing for the discovery of all IP interfaces that are hosted by a switch (CIM_System) or a network (CIM_Network).

6.7 CIM_IPConfigurationService

The CIM_IPConfigurationService is the central class of this profile. The service has a set of extrinsic methods to control the creation and removal layer 3 IP interfaces. The service can be available to physical interfaces represented with instances of CIM_EthernetPort, a switch represented by CIM_ComputerSystem and VLAN networks represented with instances of CIM_VLANNetwork.

7 Implementation

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

7.1 Representing the layer3 interface management capabilities

7.1.1 CIM_IPConfigurationService

One or more instances of CIM_IPConfigurationService shall be instantiated.

These instances of CIM_IPConfigurationService shall be associated with an instance of the scoping CIM_ComputerSystem class through an instance of CIM_HostedService.

The instances of the CIM_IPConfigurationService class shall also be associated to each CIM_ManagedElement subclass instance that may be used as the TargetInterface parameter of its AddIPProtocolEndpoint () method through an instance of CIM_ServiceAvalableToElement.

IPProtocolEndpoint instances created through the use of an instance of CIM_IPConfigurationService shall be associated to the CIM_IPConfigurationService instance through an instance of CIM_ServiceAffectsElement.

7.2 Representing Layer 3 Interfaces

7.2.1 CIM_IPProtocolEndpoint

Instances of CIM_IPProtocolEndpoint created as a result of the CIM_IPConfigurationService.AddIPProtocolEndpoint () shall comply with the requirements of DSP1036 IP Interface Profile 1.1 where CIM_IPProtocolEndpoint is the central class of DSP1036. The additional requirements listed in this cause and its sub clauses are in additional to requirements in DSP1036.

7.2.1.1 CIM_IPProtocolEndpoint (CIM_EthernetPort)

Instances of CIM_IPProtocolEndpoint created as a result of the CIM_IPConfigurationService.AddIPProtocolEndpoint () method targeting an instance of CIM_EthernetPort shall be associated with the instance of CIM_LANEndpoint associated to the CIM_EthernetPort instance, that was specified as the TargetInterface of the method call, through an instance of CIM_BindsToLANEndpoint. This instance of CIM_IPProtocolEndpoint shall also be associated through an instance of CIM_HostedIPInterface to the scoping instance of CIM_ComputerSystem.

7.2.1.2 CIM_IPLoopbackInterface

Represents a single IP endpoint communication channel. CIM_IPLoopbackInterface shall conform to 7.2.1.1. The instance of CIM_System described in 7.2.1.1 shall be the instance of the class scoping class instance of CIM_ComputerSystem.

7.2.1.3 CIM_IPSubinterface

Represents the subdivision of a single port into multiple IP subnets. CIM_IPSubinterface shall conform to 7.2.1.1.

The value of EncapsulationType shall be 1 or 2. If the value matches 1 (Other) the OtherEncapsulationType property shall be implemented and contain the encapsulation type represented as a free form string. If the value matches 2 (Dot1Q) the EncapsulationValue property shall be implemented and contain the 12 bit VLANId value represented as a string.

The ParentInterface property shall be implemented and contain a reference to the port interface, the instance of CIM_EthernetPort that is being subdivided. This value shall be formatted as a URI per RFC3986 and should be a WBEM URI (DSP0207). If this interface was created using the CIM_IPConfigurationService.AddIPProtocolEndpoint (), this value shall be the reference passed in the TargetInterface parameter of the method call.

7.2.1.4 CIM_SwitchVirtualInterface

Represents the IP settings for a VLAN to allow layer 3 routing between VLANs. CIM_SwitchVirtualInterface shall conform to 7.2.1.1. The instance of CIM_System described in 7.2.1.1 shall be an instance of the class CIM_VLANNetwork.

The VLANId property shall be implemented and contain the 12 bit VLANId that this interface is depended on. If this interface was created using the CIM_IPConfigurationService.AddIPProtocolEndpoint (), this value shall be the VLANId of the CIM_VLANNetwork Instance passed in the TargetInterface parameter of the method call.

8 Methods

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

8.1 Extrinsic methods

If synchronous execution of a method succeeds, the implementation shall set a return value of 0 (Completed with No Error).

If synchronous execution of a method fails, the implementation shall set a return value of 2 (Failed) or a more specific return code as specified with the respective method.

If a method is executed as an asynchronous task, the implementation shall perform all of the following actions:

- Set a return value of 4096 (Job Started).
- Set the value of the Job output parameter to refer to an instance of the CIM_ConcreteJob class that represents the asynchronous task.
- Set the values of the JobState and TimeOfLastStateChange properties in that instance to represent the state and last state change time of the asynchronous task.

In addition, the implementation may present state change indications as task state changes occur.

If the method execution as an asynchronous task succeeds, the implementation shall perform all of the following actions:

- Set the value of the JobState property to 7 (Completed).
- Provide an instance of the CIM_AffectedJobEntity association with property values set as follows:
  - The value of the AffectedElement property shall refer to the object that represents the top-level entity that was created or modified by the asynchronous task. For example, for the CIM_IPConfigurationService. AddIPProtocolEndpoint() method, this is an instance of the CIM_IPProtocolEndpoint class
  - The value of the AffectingElement property shall refer to the instance of the CIM_ConcreteJob class that represents the completed asynchronous task.
  - The value of the first element in the ElementEffects[ ] array property (ElementEffects[0]) shall be set to 5 (Create) for the CIM_IPConfigurationService. AddIPProtocolEndpoint() method. Otherwise, this value shall be 0 (Unknown).

If the method execution as an asynchronous task fails, the implementation shall set the value of the JobState property to 9 (Killed) or 10 (Exception).

8.1.1.1 Job parameter

The implementation shall set the value of the Job parameter as a result of an asynchronous execution of a method of the CIM_IPConfigurationService as follows:

- If the method execution is performed synchronously, the implementation shall set the value to NULL.
- If the method execution is performed asynchronously, the implementation shall set the value to refer to the instance of the CIM_ConcreteJob class that represents the asynchronous task.

8.1.2 CIM_IPConfigurationService. AddIPProtocolEndpoint()

The implementation of the AddIPProtocolEndpoint( ) method is required, the provisions in this sub clause apply in addition behavior applicable to all extrinsic methods as specified in 8.1.

The successful execution of the AddIPProtocolEndpoint( ) method shall create an index array of instance of the CIM_IPProtocolEndpoint class or a subclass of IPProtocolEndpoint and any required associations as described in the sub clauses of 7.2. In addition if the optional method parameter EndpointSettings is populated corresponding instances of the embedded CIM_SettingData classes should be associated with the newly instantiated CIM_IPProtocolEndpoint through an instance of CIM_ElementSettingData.

Table 2 contains requirements for parameters of this method.

377 **Table 2 – AddIPProtocolEndpoint ( ) Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	TargetInterface	CIM_ManagedElement REF	See 8.1.2.1.1
IN	IPProtocolEndpoint	String[]	See 8.1.2.1.2
IN	EndpointSettings	String[]	See 8.1.2.1.3
OUT	ResultingEndpoint	CIM_IPProtocolEndpoint REF[]	See 8.1.2.1.4
OUT	Job	CIM_ConcreteJob REF	See 8.1.1.1

378 #### 8.1.2.1.1 TargetInterface

379 A required reference to an associated target interface, system or network. The supported target interfaces
380 for a CIM_IPProtocolEndpoint class or subclass supported should be as described in the sub clauses of
381 7.2.

382 #### 8.1.2.1.2 IPProtocolEndpoint[]

383 A required array of string an containing one or more embedded instances of the class-subclass of
384 CIM_IPProtocolEndpoint that describes the configuration of the resultant CIM_IPProtocolEndpoints. The
385 populated properties of the embedded CIM_IPProtocolEndpoints should not contain key properties, and
386 any key property values may be ignored.

387 #### 8.1.2.1.3 EndpointSettings[]

388 An optional array of strings containing embedded instances of the class-subclass of CIM_SettingData
389 that describes the additional configuration properties for the resultant CIM_IPProtocolEndpoints. The
390 array shall be indexed to the IPProtocolEndpoint array property. The populated properties of the
391 embedded CIM_SettingData instances should not contain key properties, and any key property values
392 may be ignored. The resulting CIM_SettingData instance should be associated with the corresponding
393 resultant instance of CIM_IPProtocolEndpoint through an instance of CIM_ElementSettingData.

394 #### 8.1.2.1.4 ResultingEndpoint[]

395 If the assignment of a protocol endpoint is successfully, an array of references to the resultant instances
396 of class CIM_IPProtocolEndpoint that represents the newly defined endpoints shall be returned.

397 #### 8.1.2.1.5 Job

398 See 8.1.1.1

399 ### 8.1.3 CIM_IPConfigurationService. RemoveIPProtocolEndpoint()

400 The implementation of the RemoveIPProtocolEndpoint( ) method is required, the provisions in this sub
401 clause apply in addition behavior applicable to all extrinsic methods as specified in 8.1.

402 The successful execution of the RemoveIPProtocolEndpoint ( ) method shall remove the instances
403 referenced in the methods Endpoint parameter and should remove any associated CIM_SettingData
404 instances.

405 Table 3 contains requirements for parameters of this method.

406 **Table 3 – RemoveIPProtocolEndpoint ( ) Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	Endpoint	CIM_IPProtocolEndpoint REF	See 8.1.3.1.1
OUT	Job	CIM_ConcreteJob REF	See 8.1.1.1

407 #### 8.1.3.1.1 Endpoint

408 An array of references to instances of the class CIM_IPProtocolEndpoint that shall be removed.

409 #### 8.1.3.1.2 Job

410 See 8.1.1.1.

411 ## 8.2 Profile conventions for operations

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

414 The default list of operations is as follows:

415 • GetInstance
416 • EnumerateInstances
417 • EnumerateInstanceNames
418 • Associators
419 • AssociatorNames
420 • References
421 • ReferenceNames

422 ## 8.3 CIM_BindsToLANEndpoint

423 Table 4 lists implementation requirements for operations. If implemented, these operations shall be
424 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 4, all operations in
425 the default list in 8.2 shall be implemented as defined in DSP0200.

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

427 **Table 4 – Operations: CIM_BindsToLANEndpoint**

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

8.4 CIM_HostedService

Table 5 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 5, all operations in the default list in 8.2 shall be implemented as defined in DSP0200.

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

Table 5 – Operations: CIM_HostedService

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

8.5 CIM_HostedIPInterface

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 8.2 shall be implemented as defined in DSP0200.

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

Table 6 – Operations: CIM_HostedIPInterface

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

8.6 CIM_L3InterfaceConfigurationService

All operations in the default list in 8.2 shall be implemented as defined in DSP0200.

8.7 CIM_IPSubinterface

All operations in the default list in 8.2 shall be implemented as defined in DSP0200.

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

8.8 CIM_IPLoopbackInterface

All operations in the default list in 8.2 shall be implemented as defined in DSP0200.

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

8.9 CIM_SwitchVirtualInterface

All operations in the default list in 8.2 shall be implemented as defined in DSP0200.

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

8.10 CIM_IPProtocolInterface

All operations in the default list in 8.2 shall be implemented as defined in DSP0200.

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

Table 7 – Operations: CIM_IPProtocolEndpoint

Operation	Requirement	Messages
ModifyInstance	Conditional. See DSP1036_1.1	None

8.11 CIM_IPSubinterface

All operations in the default list in 8.2 shall be implemented as defined in DSP0200.

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

9 Use cases

This clause contains object diagrams and use cases for the *Network Management Layer3 Interface Profile*.

9.1 Profile registration

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 Network Management Layer3 Interface Profile with which an instance of CIM_IPConfigurationService is conformant. An instance of CIM_RegisteredProfile exists for each profile that is instrumented in the system. One instance of CIM_RegisteredProfile identifies the “VirtualEthernetSwitch1.1.0”. The other instance identifies the “Network Management Layer3 Interface Profile”. The CIM_IPConfigurationService instance is scoped to an instance of CIM_ComputerSystem. This instance of CIM_ComputerSystem is conformant with the DMTF *Virtual Ethernet Switch Profile* version 1.1.0 as indicated by the CIM_ElementConformsToProfile association to the CIM_RegisteredProfile instance.

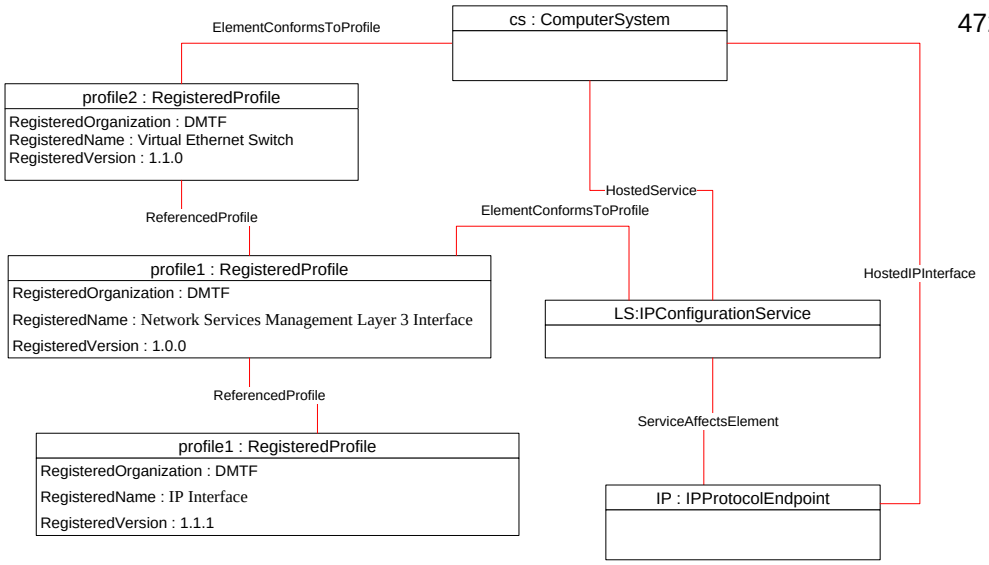


Figure 2 – Registered profile

9.2 IPSubinterface

The object diagram shown in Figure 3 contains the basic elements used to model configuration of the IPSubinterfaces of an Ethernet switch port. The diagram shows that Ethernet port E0/4 has three associated instances of CIM_IPSubinterface, SI10, SI20, and SI 30 each using Dot1Q encapsulation to separate the three IP subnets (10.10.10.10.1, 10.20.20.20.1 and 10.30.30.30.10. The Dot1Q encapsulation respectively uses VLANId 10, 20 and 30 to provide the isolation in the layer 2 switch. This is a very simple diagram, not shown are many of the required properties of the relative profiles for the objects shown.

The IPSubinterfaces were created with a CIM_IPConfigurationService.AddProtocolEndpoint() method with the following parameters. Note this is for illustration purposes and other properties from the super class CIM_IPProtocolEndpoint and other base classes may be specified as required.

- TargetInterface – WBEM URI reference to E0/4
- IPProtocolEndpoint[] –
    - Embedded Instance of CIM_IPSubinterface {
      ElementName = E0/4.10
      EncapsulationType = 2
      EncapsulationValue = 10
      IPv4Address=10.10.10.
      ProtocolIFType=4060}

    - Embedded Instance of CIM_IPSubinterface {
      ElementName = E0/4.20
      EncapsulationType = 2
      EncapsulationValue = 20
      IPv4Address=20.20.20.1
      ProtocolIFType=4060}

    - Embedded Instance of CIM_IPSubinterface {
      ElementName = E0/4.30
      EncapsulationType = 2
      EncapsulationValue = 30
      IPv4Address=30.30.30.1
      ProtocolIFType=4060}

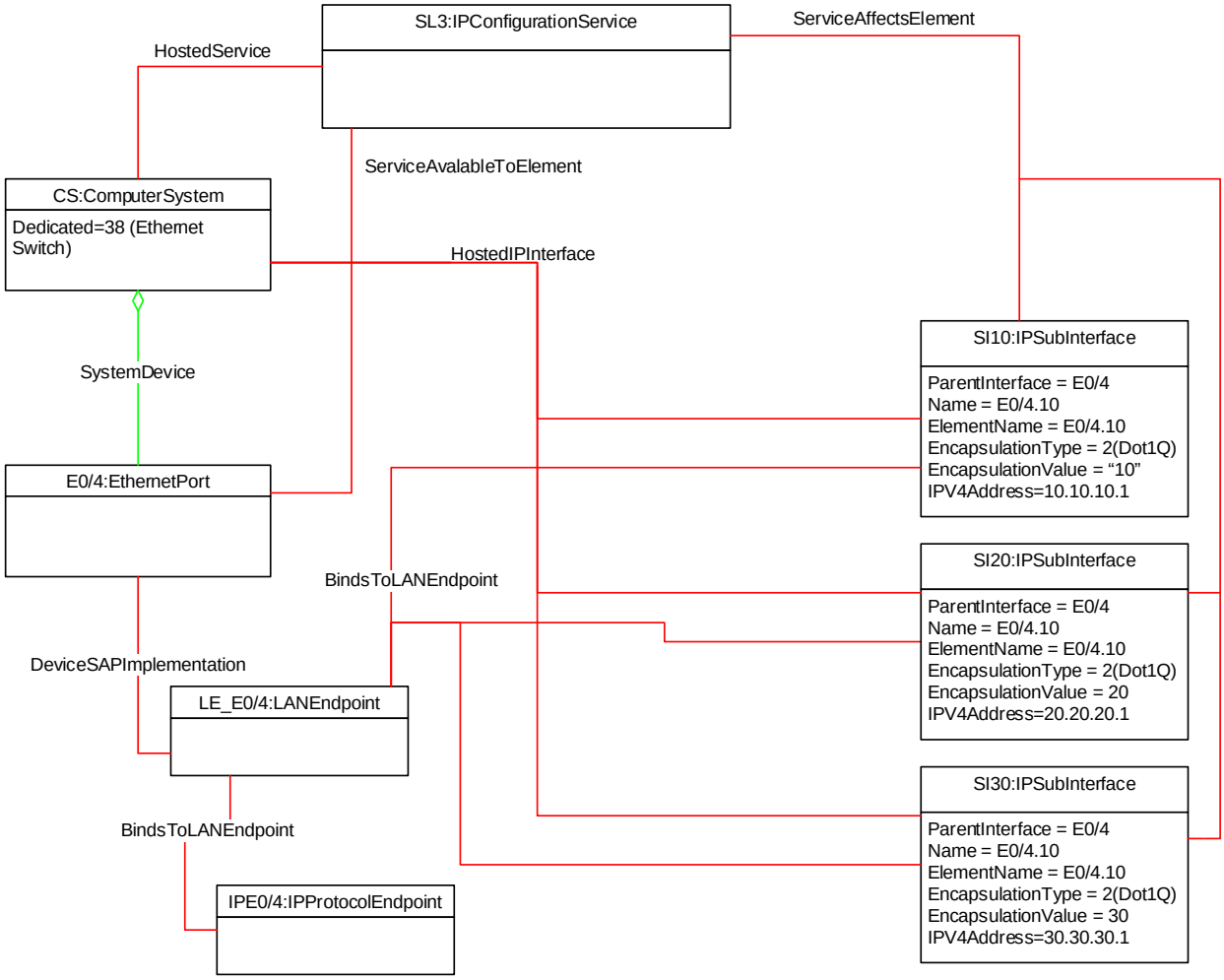


Figure 3 – IPSubinterface

9.3 Switch Virtual Interface

The object diagram shown in Figure 4 contains the basic elements used to model configuration of a Switch Virtual Interface (SVI) of a VLAN. The diagram shows that a CIM_VLANNetwork, VN30 has an associated instance of CIM_SwitchVirtualInterface, SVI30. This interface provides the VLAN an IP address allowing a routing component in the switch to bridge VLANs. Note that in the method description below the caller did not populate the VLANId property in the embedded instance. In this example the provider populated the property in the resultant instance with the value of the VLANId property from the TargetInterface. This is a very simple diagram, and not shown are many of the required properties of the relative profiles for the objects shown.

The SVI was created with a IPConfigurationService.AddProtocolEndpoint() method with the following parameters. Note this is for illustration purposes and other properties from the super class CIM_IPProtocolEndpoint and other base classes may be specified as required.

- TargetInterface – WBEM URI reference to VN30
- IPProtocolEndpoint[] –
  - Embedded Instance of CIM_SwitchVirtualInterface {
    IPv4Address=10.10.10.1
    ProtocolIFType=4060}

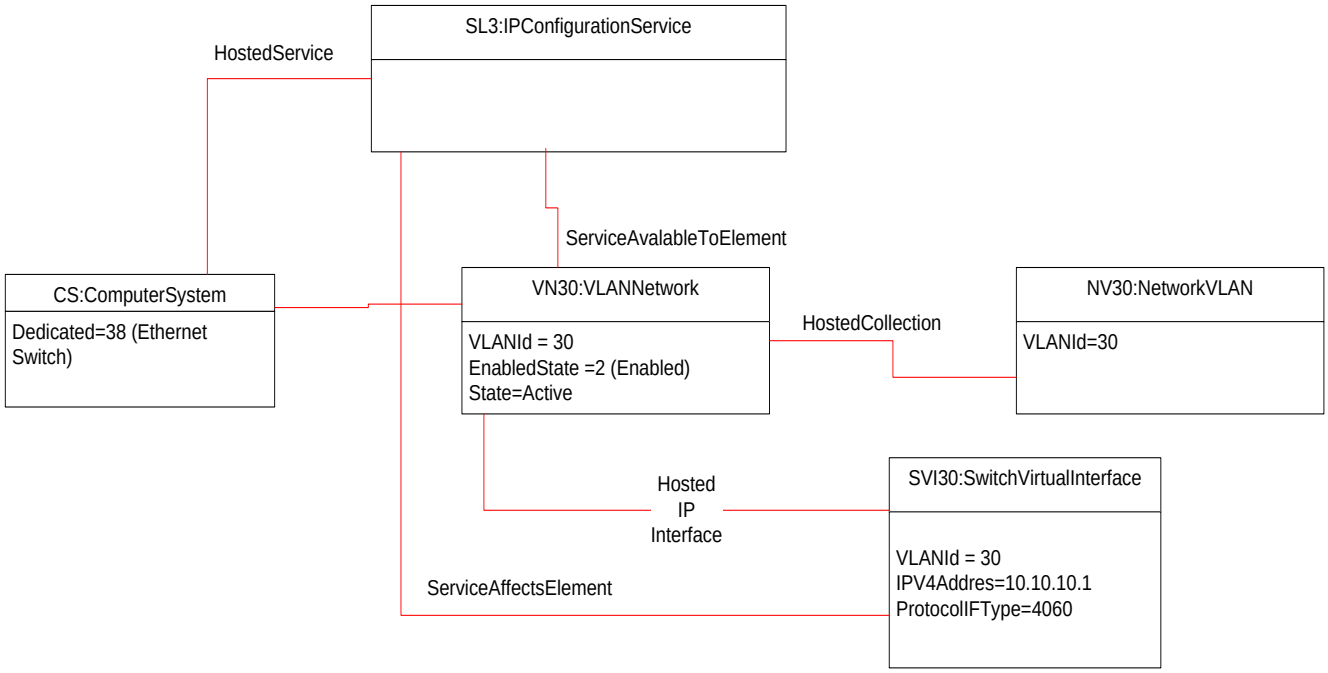


Figure 4 - Switch Virtual Interface

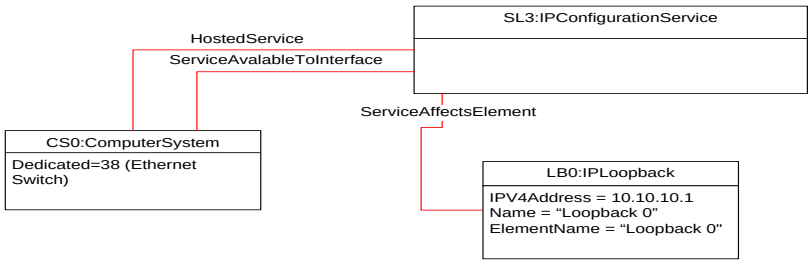
9.4 Loopback interface

The object diagram shown in Figure 5 contains the basic elements used to model configuration of a loopback interface. The diagram shows that CIM_ComputerSystem has an associated instance through the association CIM_HostedIPInterface. This is a loopback interface CIM_Loopback:LB0. This is a very simple diagram, not shown are many of the required properties of the relative profiles for the objects shown.

The interface was created with a CIM_IPConfigurationService.AddProtocolEndpoint() method with the following parameters. Note this is for illustration purposes and other properties from the super class CIM_IPProtocolEndpoint and other base classes may be specified as required.

- TargetInterface – WBEM URI reference to CIM_ComputerSystem:CS0
- IPProtocolEndpoint[] –
  - Embedded Instance of CIM_Loopback {
    IPv4Address=10.10.10.1
    ProtocolIFType=4060}

552



553

554 **Figure 5 - Loopback interface**

555

556 9.5 Add an IPProtocolEndpoint to an Ethernet Port.

557 The object diagram shown in Figure 6 contains the basic elements used to add an IP address to an
558 Ethernet Port. The diagram shows an instance of CIM_IPProtocolEndpoint associated to
559 CIM_LANEndpoint instance LE_E0/1, the CIM_LANEndpoint instance for the CIM_EthernetPort instance
560 E0\1. The diagram also shows that CIM_IPProtocolEndpoint instance is associated with the scoping
561 CIM_ComputerSystem instance through CIM_HostedIPInterface. This is a very simple diagram, and not
562 shown are many of the required properties of the relative profiles for the objects shown.

563 The CIM_IPProtocolEndpoint interface was created using the CIM_IPConfigurationService instance, SL3,
564 associated with the target CIM_EthernetPort through CIM_ServiceAvalableToElement. The
565 IPProtocolEndpoint instance was added through the CIM_IPConfigurationService.AddProtocolEndpoint()
566 method with the following parameters. Note this is for illustration purposes and other properties from the
567 class CIM_IPProtocolEndpoint and other base classes may be specified as required.

568 • TargetInterface – WBEM URI reference to E0\1:CIM_EthernetPort
569 • IPProtocolEndpoint[] –
570 o Embedded Instance of CIM_IPProtocolEndpoint {
571 IPv4Address=9.44.125.168
572 SubnetMask: 255.255.240.0
573 ProtocolIFType=4096}
574

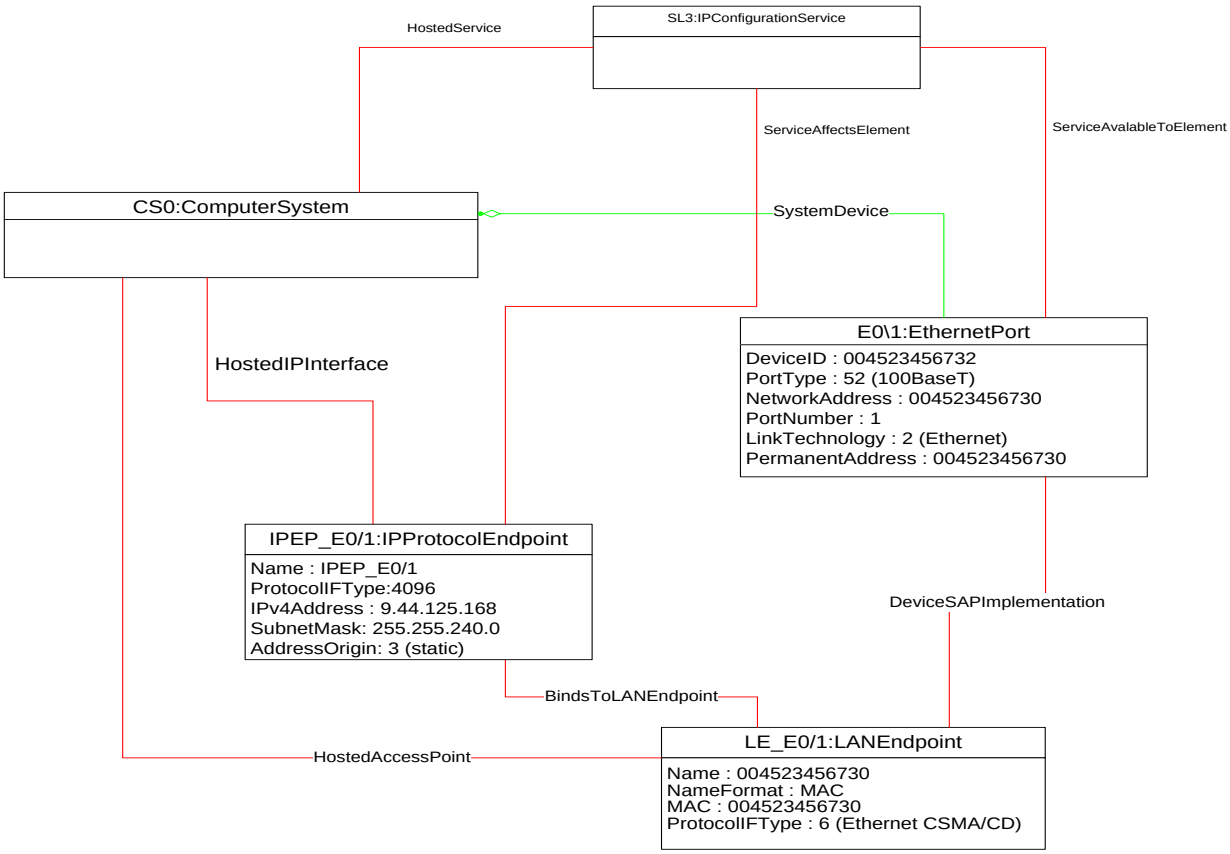


Figure 6 - IPProtocolEndpoint

10 CIM Elements

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

Table 8 – CIM Elements: Network Management Layer 3 Interface Profile

Element Name	Requirement	Description
Classes		
CIM_BindsToLANEndpoint	Optional	See DSP1036 IP Interface Profile 1.1.1
CIM_HostedService	Conditional	See 7.1.1
CIM_HostedIPInterface	Conditional	See 7.2.1.1
CIM_IPProtocolEndpoint	Conditional	See 7.2.1
CIM_IPConfigurationService	Mandatory	See 7.1.1
CIM_IPLoopbackInterface	Conditional	See 7.2.1.2
CIM_RegisteredProfile	Optional	

Element Name	Requirement	Description
CIM_ServiceAffectsElement	Conditional	See 7.1.1
CIM_ServiceAvalableToElement	Conditional	See 7.1.1
CIM_IPSubinterface	Optional	See 7.2.1.3
CIM_SwitchVirtualInterface	Optional	See 7.2.1.4
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 9 provides information about the properties of CIM_BindsToLANEndpoint.

Table 9 – Class: CIM_BindsToLANEndpoint

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

10.2 CIM_HostedService

CIM_HostedService relates the CIM_IPConfigurationService instance to its scoping CIM_ComputerSystem instance. Table 10 provides information about the properties of CIM_HostedService.

Table 10 – Class: CIM_HostedService

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

10.3 CIM_IPConfigurationService

CIM_IPConfigurationService provides the methods to create and delete a Layer 3 interface. Table 11 provides information about the properties of CIM_IPConfigurationService that are in addition to those specified in DSP1036.

598 **Table 11 – Class: CIM_IPConfigurationService**

Elements	Requirement	Description
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
ElementName	Mandatory	Pattern ".*"
AddIPProtocolEndpoint( )	Mandatory	See 8.1.2.
RemoveIPProtocolEndpoint( )	Mandatory	See 8.1.3.

599 ## 10.4 CIM_IPProtocolEndpoint

600 CIM_IPProtocolEndpoint represents an IP interface that is associated with an Ethernet interface. Table 12
601 provides information about the properties of CIM_IPProtocolEndpoint.

602 **Table 12 – Class: CIM_IPProtocolEndpoint**

Elements	Requirement	Description
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
NameFormat	Mandatory	See DSP1036 IP Interface Profile 1.1.1
ProtocolIFType	Mandatory	See DSP1036 IP Interface Profile 1.1.1
RequestedState	Mandatory	See DSP1036 IP Interface Profile 1.1.1
EnabledState	Mandatory	See DSP1036 IP Interface Profile 1.1.1
ElementName	Mandatory	See DSP1036 IP Interface Profile 1.1.1
RequestStateChange( )	Conditional	See DSP1036 IP Interface Profile 1.1.1
IPv4Address	Conditional	See DSP1036 IP Interface Profile 1.1.1
SubnetMask	Conditional	See DSP1036 IP Interface Profile 1.1.1
AddressOrigin	Mandatory	See DSP1036 IP Interface Profile 1.1.1
IPv6Address	Conditional	See DSP1036 IP Interface Profile 1.1.1
IPv6AddressType	Conditional	See DSP1036 IP Interface Profile 1.1.1
IPv6SubnetPrefixLength	Conditional	See DSP1036 IP Interface Profile 1.1.1

603

10.5 CIM_IPSubinterface

CIM_IPSubinterface represents a subdivision of an Ethernet interface. Table 13 provides information about the additional properties of CIM_IPSubinterface that are in addition to those in CIM_IPProtocolEndpoint 10.3 Table 12.

Table 13 – Class: CIM_IPSubinterface

Elements	Requirement	Description
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
ParentInterface	Mandatory	**See 7.2.1.3**
ElementName	Mandatory	Pattern ".*"
EncapsulationType	Mandatory	**See 7.2.1.3**
OtherEncapsulationType	Conditional	**See 7.2.1.3**
EncapsulationValue	Conditional	**See 7.2.1.3**

10.6 CIM_SwitchVirtualInterface

CIM_SwitchVirtualInterface represents the IP protocol endpoint used to route a VLAN within a switch. Table 14 provides information about the additional properties of CIM_SwitchVirtualInterface that are in addition to those in CIM_IPProtocolEndpoint 10.3, Table 12.

Table 14 – Class: CIM_SwitchVirtualInterface

Elements	Requirement	Description
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
ElementName	Mandatory	Pattern ".*"
VLANId	Mandatory	**See 7.2.1.4**

10.7 CIM_RegisteredProfile

CIM_RegisteredProfile identifies the *Network Management Layer3 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 15, the behavior of the CIM_RegisteredProfile instance is in accordance with the *Profile Registration Profile*.

Table 15 – Class: CIM_RegisteredProfile

Elements	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of "Network Management L3 Interface Profile".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of "DMTF".

ANNEX A
(informative)

Change log

Version	Date	Description
1.0.0	2018-06-11	