Network Management Layer3 Interface Profile

Information for Work-in-Progress version:

IMPORTANT: This document is not a standard. It does not necessarily reflect the views of the DMTF or all of its members. Because this document is a Work in Progress, it may still change, perhaps profoundly. This document is available for public review and comment until the stated expiration date.

It expires on: 2013-12-22

Provide any comments through the DMTF Feedback Portal:
http://www.dmtf.org/standards/feedback

Document Type: Specification
Document Status: Work In Progress
Document Language: en-US
Copyright Notice

Copyright © 2012-2013 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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

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

34 Foreword .......................................................................................................................... 5
35 Introduction ..................................................................................................................... 6
36 1 Scope .............................................................................................................................. 7
37 2 Normative references ................................................................................................... 7
38 3 Terms and definitions ................................................................................................... 7
39 4 Symbols and abbreviated terms .................................................................................... 8
40 5 Synopsis ........................................................................................................................ 9
41 6 Description .................................................................................................................... 9
42 6.1 Class diagram ............................................................................................................. 10
43 6.2 CIM_IPProtocolEndpoint .......................................................................................... 10
44 6.3 CIM_SwitchVirtualInterface ..................................................................................... 11
45 6.4 CIM_Subinterface ...................................................................................................... 11
46 6.5 CIM_Loopback ........................................................................................................... 11
47 6.6 CIM_HostedIPInterface ............................................................................................ 11
48 6.7 CIM_IPEncapsulationInterface ................................................................................. 11
49 6.8 CIM_IPConfigurationService .................................................................................... 11
50 7 Implementation .............................................................................................................. 11
51 7.1 Representing the layer3 interface management capabilities ..................................... 11
52 7.1.1 CIM_IPConfigurationService ............................................................................. 11
53 7.2 Representing Layer 3 Interfaces ................................................................................ 12
54 7.2.1 CIM_IPProtocolEndpoint .................................................................................... 12
55 8 Methods ........................................................................................................................ 13
56 8.1 Extrinsic Methods ...................................................................................................... 13
57 8.1.2 CIM_IPConfigurationService. AddIPProtocolEndpoint() .................................. 14
58 8.1.3 CIM_IPConfigurationService. RemoveIPProtocolEndpoint() .......................... 15
59 8.2 Profile conventions for operations ........................................................................... 16
60 8.3 CIM_BindsToLANEndpoint ...................................................................................... 16
61 8.4 CIM_HostedService ................................................................................................... 16
62 8.5 CIM_HostedIPInterface ............................................................................................ 17
63 8.6 CIM_L3InterfaceConfigurationService ................................................................... 17
64 8.7 CIM_IPEncapsulationInterface ................................................................................. 17
65 8.8 CIM_Subinterface ...................................................................................................... 17
66 8.9 CIM_LoopbackInterface ........................................................................................... 17
67 8.10 CIM_SwitchVirtualInterface .................................................................................. 17
68 8.11 CIM_IPProtocolInterface ....................................................................................... 17
69 8.12 CIM_Subinterface .................................................................................................... 18
70 9 Use cases ........................................................................................................................ 19
71 9.1 Profile Registration .................................................................................................... 19
72 9.2 Subinterface ............................................................................................................... 20
73 9.3 Switch Virtual Interface ........................................................................................... 21
74 9.4 Tunnel Interface and Loopback Interface .................................................................. 22
75 9.5 Add an IPProtocolEndpoint to an Ethernet Port ..................................................... 23
76 10 CIM Elements ............................................................................................................ 24
77 10.1 CIM_BindsToLANEndpoint .................................................................................... 25
78 10.2 CIM_HostedService ............................................................................................... 25
79 10.3 CIM_IPConfigurationService ............................................................................... 26
80 10.4 CIM_IPProtocolEndpoint ...................................................................................... 26
81 10.5 CIM_IPEncapsulationInterface .............................................................................. 27
82 10.6 CIM_SubInterface .................................................................................................. 27
83 10.7 CIM_SwitchVirtualInterface .................................................................................. 28
84 10.8 CIM_RegisteredProfile ........................................................................................ 28
85 ANNEX A (informative) Change log ................................................................................ 29
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

Contributors:
- Hemal Shah – Broadcom Corporation
- John Crandall – Brocade Communications System
- Alex Zhdankin – Cisco Systems
- Steve Neely – Cisco Systems
- Shishir Pardikar – Citrix
- John Parchem – Microsoft Corporation
- Lawrence Lamers – VMware
- Dr. Bhumip Khasnabish - ZTE 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 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 DSP1033, Profile Registration Profile 1.0, http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

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: 2.39e
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>
<thead>
<tr>
<th>Profile Name</th>
<th>Organization</th>
<th>Version</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Registration</td>
<td>DMTF</td>
<td>1.0</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>Virtual Ethernet Switch</td>
<td>DMTF</td>
<td>1.1</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>IP Configuration Profile</td>
<td>DMTF</td>
<td>1.1</td>
<td>Optional</td>
<td>None</td>
</tr>
<tr>
<td>IP Interface Profile</td>
<td>DMTF</td>
<td>1.1.1</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>Host LAN Network</td>
<td>DMTF</td>
<td>1.0.2</td>
<td>Mandatory</td>
<td>None</td>
</tr>
<tr>
<td>Network Management</td>
<td>DMTF</td>
<td>1.0</td>
<td>Optional</td>
<td>None</td>
</tr>
<tr>
<td>Network Management Routing</td>
<td>DMTF</td>
<td>1.0</td>
<td>Optional</td>
<td>None</td>
</tr>
</tbody>
</table>

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 *CIM_IPProtocolEndpoint* and the subclasses of *IPProtocolEndpoint*, *CIM_IPEncapsulationInterface*, *CIM_Subinterface*, *CIM_Loopback* and *CIM_SwitchVirtualInterface* all represent the management aspects of 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_Subinterface
A Subinterface subdivides a single switch port into multiple IP subnets. This is typically done using Dot1Q encapsulation using VLANIds to distinguish the subnets. Even though a Subinterface 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_Loopback
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_IPEncapsulationInterface
An IP tunnel interface is used to connect two disjoined IP networks.

6.8 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_ServiceAvailableToElement.

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() method shall comply with the requirements of [DSP1036 IP Interface Profile 1.1.1](#). Where CIM_IPProtocolEndpoint is the central class of DSP1036. The additional requirements listed in this cause and its sub clauses are in addition 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_IPEncapsulationInterface, CIM_LoopbackInterface and CIM_SwitchVirtualInterface

Instances of CIM_IPProtocolEndpoint created as a result of the CIM_IPConfigurationService.CreateIPProtocolEndpoint() method where the TargetInterface is an instance of CIM_System shall be associated to the TargetInterface through an instance of CIM_HostedIPInterface.

7.2.1.3 CIM_LoopbackInterface

Represents a single IP endpoint communication channel. CIM_LoopbackInterface shall conform to 7.2.1.2. The instance of CIM_System described in 7.2.1.2 shall be the instance of the class scoping class instance of CIM_ComputerSystem.

7.2.1.4 CIM_Subinterface

Represents the subdivision of a single port into multiple IP subnets. CIM_Subinterface 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.

7.2.1.5 CIM_SwitchVirtualInterface

Represents the IP settings for a VLAN to allow layer 3 routing between VLANs. CIM_SwitchVirtualInterface shall conform to 7.2.1.2. The instance of CIM_System described in 7.2.1.2 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.

### 7.2.1.6 CIM_IPEncapsulationInterface

Represents an interface to tunnel between disjoint IP networks, generally through encapsulation. CIM_IPEncapsulationInterface shall conform to 7.2.1.2. The instance of CIM_System described in 7.2.1.2 shall be the instance of the scoping class instance of CIM_ComputerSystem. The properties inherited from the class CIM_IPProtocolEndpoint shall represent the IP configuration for the represented IP tunnel Interface and shall conform to DSP1036 IP Configuration Profile.

The value of the property TunnelMode shall be implemented and contain a value that represents the type of tunnel this interface is implementing. If the value matches 1 (Other) the OtherTunnelMode property shall be implemented and contain the tunnel mode represented as a free form string.

If the VRFContext is implemented it may contain an identifier of the virtual routing and forwarding table (VRF) used to determine the tunnel IP destination address.

If the TunnelDestinationIPAddress is implemented it may contain the destination IP address for this tunnel interface. The value of the property shall be specified in dotted decimal notation as defined in IETF RFC 1208 if an IPv4 address or if an IPv6 address the value of the property shall be specified in the notation specified in IETF RFC 4291, section 2.2.

If implemented the TunnelSourceInterface may contain the identifier of the source interface, an instance of CIM_EthernetPort.

### 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.

<table>
<thead>
<tr>
<th>Qualifiers</th>
<th>Name</th>
<th>Type</th>
<th>Description/Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>TargetInterface</td>
<td>CIM_ManagedElement REF</td>
<td>See 8.1.2.1.1</td>
</tr>
<tr>
<td>IN</td>
<td>IPProtocolEndpoint</td>
<td>String[]</td>
<td>See 8.1.2.1.2</td>
</tr>
<tr>
<td>IN</td>
<td>EndpointSettings</td>
<td>String[]</td>
<td>See 8.1.2.1.3</td>
</tr>
<tr>
<td>OUT</td>
<td>ResultingEndpoint</td>
<td>CIM_IPProtocolEndpoint REF</td>
<td>See 8.1.2.1.4</td>
</tr>
<tr>
<td>OUT</td>
<td>Job</td>
<td>CIM_ConcreteJob REF</td>
<td>See 8.1.1.1</td>
</tr>
</tbody>
</table>
8.1.2.1.1 TargetInterface

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

8.1.2.1.2 IPProtocolEndpoint[]

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

8.1.2.1.3 EndpointSettings[]

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

8.1.2.1.4 ResultingEndpoint[]

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

8.1.2.1.5 Job

See 8.1.1.1

8.1.3 CIM_IPConfigurationService. RemoveIPProtocolEndpoint()

The implementation of the RemoveIPProtocolEndpoint() 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 RemoveIPProtocolEndpoint() method shall remove the instances referenced in the methods Endpoint parameter and should remove any associated CIM_SettingData instances.

Table 2 contains requirements for parameters of this method.

<table>
<thead>
<tr>
<th>Qualifiers</th>
<th>Name</th>
<th>Type</th>
<th>Description/Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td>Endpoint</td>
<td>CIM_IPProtocolEndpoint REF</td>
<td>See 8.1.3.1.1</td>
</tr>
<tr>
<td>OUT</td>
<td>Job</td>
<td>CIM_ConcreteJob REF</td>
<td>See 8.1.1.1</td>
</tr>
</tbody>
</table>

8.1.3.1.1 Endpoint

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

8.1.3.1.2 Job

See 8.1.1.1
8.2 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

8.3 CIM_BindsToLANEndpoint

Table 4 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 4, 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>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associators</td>
<td>Unspecified</td>
<td>None</td>
</tr>
<tr>
<td>AssociatorNames</td>
<td>Unspecified</td>
<td>None</td>
</tr>
<tr>
<td>References</td>
<td>Unspecified</td>
<td>None</td>
</tr>
<tr>
<td>ReferenceNames</td>
<td>Unspecified</td>
<td>None</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associators</td>
<td>Unspecified</td>
<td>None</td>
</tr>
<tr>
<td>AssociatorNames</td>
<td>Unspecified</td>
<td>None</td>
</tr>
<tr>
<td>References</td>
<td>Unspecified</td>
<td>None</td>
</tr>
<tr>
<td>ReferenceNames</td>
<td>Unspecified</td>
<td>None</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associators</td>
<td>Unspecified</td>
<td>None</td>
</tr>
<tr>
<td>AssociatorNames</td>
<td>Unspecified</td>
<td>None</td>
</tr>
<tr>
<td>References</td>
<td>Unspecified</td>
<td>None</td>
</tr>
<tr>
<td>ReferenceNames</td>
<td>Unspecified</td>
<td>None</td>
</tr>
</tbody>
</table>

8.6 CIM_L3InterfaceConfigurationService

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

8.7 CIM_IPEncapsulationInterface

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_Subinterface

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_LoopbackInterface

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_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.

8.11 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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Requirement</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModifyInstance</td>
<td>Conditional. See DSP1036_1.1.1</td>
<td>None</td>
</tr>
</tbody>
</table>

8.12 CIM_Subinterface

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 Subinterface

The object diagram shown in Figure 3 contains the basic elements used to model configuration of the Subinterfaces of an Ethernet switch port. The diagram shows that Ethernet port E0/4 has three associated instances of CIM_Subinterface, SI10, SI20, and SI30 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 Subinterfaces 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_Subinterface {
    ElementName = E0/4.10
    EncapsulationType = 2
    EncapsulationValue = 10
    IPv4Address=10.10.10.
    ProtocolIFType=4060
  }
  - Embedded Instance of CIM_Subinterface {
    ElementName = E0/4.20
    EncapsulationType = 2
    EncapsulationValue = 20
    IPv4Address=20.20.20.1
    ProtocolIFType=4060
  }
  - Embedded Instance of CIM_Subinterface {
    ElementName = E0/4.30
    EncapsulationType = 2
    EncapsulationValue = 30
    IPv4Address=30.30.30.1
    ProtocolIFType=4060
  }
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 instances 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, 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[] –

Figure 3 – Subinterface
9.4 Tunnel Interface and Loopback Interface

The object diagram shown in Figure 5 contains the basic elements used to model configuration of an IP tunnel interface and a loopback interface. The diagram shows that CIM_ComputerSystem has two associated instances through the association CIM_HostedIPInterface. These are a loopback interface CIM_Loopback:LB0 and an IP tunnel interface CIM_IPEncapsulationInterface:GRETunnel0. This is a very simple diagram, not shown are many of the required properties of the relative profiles for the objects shown.

The two interfaces 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_IPEndPoint 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
  }
  - Embedded Instance of CIM_IPEncapsulationInterface {
    - TunnelMode=2 (GRE)
    - TunnelSourceInterface = WBEM URI reference to CIM_EthernetPort: E0/4
    - VRFContext = WBEM URI reference to CIM_VirtualRoutingandForwarding: Management
    - IPv4Address=127.0.0.1
    - ProtocolIFType=4060
  }

Figure 4 - Switch Virtual Interface

```
Embedded Instance of CIM_SwitchVirtualInterface {
  IPv4Address=10.10.10.1
  ProtocolIFType=4060
}
```
9.5 Add an IPProtocolEndpoint to an Ethernet Port.

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

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

- TargetInterface – WBEM URI reference to E0/1:CIM_EthernetPort
- IPProtocolEndpoint[] –
  - Embedded Instance of CIM_IPProtocolEndpoint {
    - IPv4Address=9.44.125.168
    - SubnetMask: 255.255.240.0
    - ProtocolIFType=4096
  }
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

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIM_BindsToLANEndpoint</td>
<td>Optional</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>CIM_HostedService</td>
<td>Conditional</td>
<td>See 7.1.1</td>
</tr>
</tbody>
</table>
### Element Name

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIM_HostedIPInterface</td>
<td>Conditional</td>
<td>See 7.2.1.1 and 7.2.1.2</td>
</tr>
<tr>
<td>CIM_IPProtocolEndpoint</td>
<td>Conditional</td>
<td>See 7.2.1</td>
</tr>
<tr>
<td>CIM_IPConfigurationService</td>
<td>Mandatory</td>
<td>See 7.1.1</td>
</tr>
<tr>
<td>CIM_LoopbackInterface</td>
<td>Conditional</td>
<td>See 7.2.1.3</td>
</tr>
<tr>
<td>CIM_RegisteredProfile</td>
<td>Optional</td>
<td>See</td>
</tr>
<tr>
<td>CIM_ServiceAffectsElement</td>
<td>Conditional</td>
<td>See 7.1.1</td>
</tr>
<tr>
<td>CIM_ServiceAvailableToElement</td>
<td>Conditional</td>
<td>See Error! Reference source not found.</td>
</tr>
<tr>
<td>CIM_Subinterface</td>
<td>Optional</td>
<td>See 7.2.1.4</td>
</tr>
<tr>
<td>CIM_SwitchVirtualInterface</td>
<td>Optional</td>
<td>See 7.2.1.5</td>
</tr>
<tr>
<td>CIM_IPEncapsulationInterface</td>
<td>Optional</td>
<td>See 7.2.1.6</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>Mandatory</td>
<td><strong>Key:</strong> This shall be a reference to an instance of CIM_LANEndpoint. Cardinality 0..1</td>
</tr>
<tr>
<td>Dependent</td>
<td>Mandatory</td>
<td><strong>Key:</strong> This shall be a reference to the Central Instance. Cardinality 1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedent</td>
<td>Mandatory</td>
<td><strong>Key:</strong> This shall be a reference to the Central Instance. Cardinality 1</td>
</tr>
<tr>
<td>Dependent</td>
<td>Mandatory</td>
<td><strong>Key:</strong> This shall be a reference to an instance of CIM_IPConfigurationService. Cardinality *</td>
</tr>
</tbody>
</table>
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 IP Interface Profile 1.1.1.

Table 11 – Class: CIM_IPConfigurationService

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemCreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>SystemName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>Name</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>ElementName</td>
<td>Mandatory</td>
<td>Pattern &quot;.**&quot;</td>
</tr>
<tr>
<td>AddIPProtocolEndpoint( )</td>
<td>Mandatory</td>
<td>See 8.1.2.</td>
</tr>
<tr>
<td>RemoveIPProtocolEndpoint( )</td>
<td>Mandatory</td>
<td>See 8.1.3.</td>
</tr>
</tbody>
</table>

10.4 CIM_IPProtocolEndpoint

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

Table 12 – Class: CIM_IPProtocolEndpoint

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemCreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>SystemName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>Name</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>NameFormat</td>
<td>Mandatory</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>ProtocolIFType</td>
<td>Mandatory</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>RequestedState</td>
<td>Mandatory</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>EnabledState</td>
<td>Mandatory</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>ElementName</td>
<td>Mandatory</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>RequestStateChange( )</td>
<td>Conditional</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>IPv4Address</td>
<td>Conditional</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>SubnetMask</td>
<td>Conditional</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>AddressOrigin</td>
<td>Mandatory</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>IPv6Address</td>
<td>Conditional</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
<tr>
<td>IPv6AddressType</td>
<td>Conditional</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
</tbody>
</table>
### 10.5 CIM_IPEncapsulationInterface

CIM_IPEncapsulationInterface represents the IP tunnel interface used to route to connect two disjoined IP networks. Table 13 provides information about the additional properties of CIM_IPEncapsulationInterface that are in addition to those in CIM_IPProtocolEndpoint 10.3 Table 12.

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6SubnetPrefixLength</td>
<td>Conditional</td>
<td>See DSP1036 IP Interface Profile 1.1.1</td>
</tr>
</tbody>
</table>

### Table 13 – Class: CIM_IPEncapsulationInterface

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemCreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>SystemName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>Name</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>ElementName</td>
<td>Mandatory</td>
<td>Pattern &quot;.*&quot;</td>
</tr>
<tr>
<td>TunnelMode</td>
<td>Mandatory</td>
<td>See 7.2.1.6</td>
</tr>
<tr>
<td>VRFContext</td>
<td>Optional</td>
<td>See 7.2.1.6</td>
</tr>
<tr>
<td>TunnelDestinationIPAddress</td>
<td>Optional</td>
<td>See 7.2.1.6</td>
</tr>
<tr>
<td>TunnelSourceInterface</td>
<td>Optional</td>
<td>See 7.2.1.6</td>
</tr>
</tbody>
</table>

### 10.6 CIM_SubInterface

CIM_SubInterface represents a sub division of an Ethernet interface interface. Table 14 provides information about the additional properties of CIM_SubInterface that are in addition to those in CIM_IPProtocolEndpoint 10.3 Table 12.

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemCreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>SystemName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>Name</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>ParentInterface</td>
<td>Mandatory</td>
<td>See 7.2.1.4</td>
</tr>
<tr>
<td>ElementName</td>
<td>Mandatory</td>
<td>Pattern &quot;.*&quot;</td>
</tr>
<tr>
<td>EncapsulationType</td>
<td>Mandatory</td>
<td>See 7.2.1.4</td>
</tr>
</tbody>
</table>
## Elements Requirement Description

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OtherEncapsulationType</td>
<td>Conditional</td>
<td>See 7.2.1.4</td>
</tr>
<tr>
<td>EncapsulationValue</td>
<td>Conditional</td>
<td>See 7.2.1.4</td>
</tr>
</tbody>
</table>

### 10.7 CIM_SwitchVirtualInterface

CIM_SwitchVirtualInterface represents the IP protocol endpoint used to route a VLAN within a switch. Table 15 provides information about the additional properties of CIM_SwitchVirtualInterface that are in addition to those in CIM_IPProtocolEndpoint 10.3 Table 12.

### Table 15 – Class: CIM_SwitchVirtualInterface

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemCreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>SystemName</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>Name</td>
<td>Mandatory</td>
<td>Key</td>
</tr>
<tr>
<td>ElementName</td>
<td>Mandatory</td>
<td>Pattern &quot;.*&quot;</td>
</tr>
<tr>
<td>VLANId</td>
<td>Mandatory</td>
<td>See 7.2.1.5</td>
</tr>
</tbody>
</table>

### 10.8 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 16, the behavior of the CIM_RegisteredProfile instance is in accordance with the Profile Registration Profile.

### Table 16 – Class: CIM_RegisteredProfile

<table>
<thead>
<tr>
<th>Elements</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegisteredName</td>
<td>Mandatory</td>
<td>This property shall have a value of &quot;Network Management L3 Interface Profile&quot;.</td>
</tr>
<tr>
<td>RegisteredVersion</td>
<td>Mandatory</td>
<td>This property shall have a value of &quot;1.0.0&quot;.</td>
</tr>
<tr>
<td>RegisteredOrganization</td>
<td>Mandatory</td>
<td>This property shall have a value of &quot;DMTF&quot;.</td>
</tr>
</tbody>
</table>
ANNEX A
(informative)

Change log

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0a</td>
<td>2013-09-22</td>
<td>DMTF Work in Progress</td>
</tr>
</tbody>
</table>