HP Network Node Manager iSPI for MPLS

For the Windows ® and Linux operating systems

Software Version: 10.00

Developer's Toolkit

Document Release Date: July 2014

Software Release Date: July 2014

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: **http://h20230.www2.hp.com/selfsolve/manuals**

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: **http://h20229.www2.hp.com/passport-registration.html**

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at: **http://www.hp.com/go/hpsoftwaresupport**

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

HP Software Solutions Now accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is **http://h20230.www2.hp.com/sc/solutions/index.jsp**

Contents

Chapter 1: Introduction

HP Network Node Manager i Software Smart Plug-in for MPLS (Multi Protocol Label Switching) (NNMi iSPI for MPLS) helps you to extend the capability of HP Network Node Manager iSoftware (NNMi) to monitor the overall health of the network.

The iSPI for MPLS provides real-time data that enables you to monitor the health of L3 Virtual Private Network (L3 VPN), Layer 2 VPNs (L2 VPNs), Multicast VPNs (MVPNs), PseudoWire VCs, and Traffic Engineering (TE) tunnels.

The iSPI for MPLS Developer's Toolkit guides you to use the MPLS web services. The MPLS web services offer read and write capabilities on the MPLS objects. You can customize and enhance the functionality of the iSPI for MPLS and NNMi by using NNMi and MPLS web services.

This MPLS API documentation enables you to access and use MPLS web services.

Related Documentation

For more information about iSPI for MPLS, see the following:

- iSPI for MPLS Online Help
- iSPI for MPLS Release Notes
- iSPI for MPLS Support Matrix
- iSPI for MPLS Deployment Guide

Also see, *NNMi 10.00 SDK Guide*

Chapter 2: API - MPLS Web Services

The iSPI for MPLS in conjunction to NNMi offers the Web Services (WS) on the MPLS objects. By using the MPLS WS, you can add, remove, and update the MPLS objects. You can also add and remove the custom attributes of the MPLS objects or update the value for any custom attribute.

Security

The web services available with iSPI for MPLS require the basic authentication and credentials of NNMi user roles. The NNMi user roles include Web Services client and admin. Use the NNMi console configuration to create a user account with one of specified roles.

Web Service Interoperability (WS-I) APIs

The iSPI for MPLS provides you the web services on the MPLS object. Each of the MPLS services provides access to fetch MPLS objects. You can use the filter option to fetch these MPLS objects. The custom attributes setting is enabled for the following MPLS objects:

- "L3 VPN" on page 10
- "L3 VRF" on page 11
- "MVPN" on page 12
- "MVRF" on page 14
- "PE Inteface" on page 15
- "Pseudowire" on page 15
- "TE Tunnel" on page 16
- "VPLS VPN" on page 17
- "VPWS VPN" on page 18
- "Attachment Circuit" on page 19
- "VC LSP" on page 20
- "VFI" on page 20

Accessing the MPLS WS-I Services

To access the MPLS WS-I services, follow these steps:

1. Use the following URL to log on for Web Services registry:

   http:// <host>:<port>/NmsSdkService/WsRegistryBean?wsdl

   > **Note:** Default port for NNM is 80. You can customize the port when you are installing iSPI for MPLS.

2. Use the following getEndpoint list to find out the actual Web Services Description Language (WSDL) URL for MPLS WS: .

   - L3VpnBeanService
   - L3VrfBeanService
   - MVpnBeanService
   - MVrfBeanService
   - PeInterfaceBeanService
   - PseudoWireBeanService
   - TunnelBeanService
   - VPLSBeanService
   - VPWSBeanService
   - AttachmentCircuitBeanService
   - VcLspBeanService
   - VfiBeanService

     The relevant WSDL URL appears.

     For example, for L3VpnBeanService, the following wsdl URL appears: http:// <host>:<port>/MplsSDK/L3VpnBeanService?wsdl.

3. Use the wsdl URL to use the following MPLS WS:

   - L3 VPN
   - L3 VRF
   - MVPN
   - MVRF
   - PE Interface

- Pseudowire
- TE Tunnel
- VPLS VPN
- VPWS VPN
- AttachmentCircuit
- VCLSP
- VFI

Chapter 3: WS-I Filter Definition

The WS-I services enable you to filter the MPLS objects from the data store.

Filter is defined as one of the following:

- Expression

  BooleanOperator (AND, OR)

  Filter[]

- Condition

  String: name

  Operator ('=', '!=', '<', '<=', '>', '>', 'like')

  String: value

- Constraint

  String: name

  String: value

Valid Constraints

Name	Default Value
offset	0
maxObjects	1000
includeCustomAttributes	false

Alternatively:

<filter> ::= <booleanOperator> <list>

<list> ::= <subFilter> | <subFilter> <list>

<subFilter> ::= <constraint> | <condition> | <filter>

<constraint> ::= <constraintName> <constraintValue>

<condition> ::= <propertyName> <operator> <propertyValue>

<booleanOperator> ::= <AND> | <OR>

<operator> ::= <=> | <!=> | <<> | <<=> | <>> | <>=> | <like>

<constraintName> ::= <offset> | <maxObjects>

<constraintValue> ::= <string representation of a constraint value>

<propertyName> ::= <name of an exposed NNMi attribute>

<propertyValue> ::= <string representation of value>

Example to Use the Filters

The folowing code explains how to apply a constraint and expression filter to get a list of objects that have a full-mesh topotype and name is not equal to Galaxy.

```
Filter c1 = new Condition("topoType", Operator.EQ, "FULL_MESH");

Filter c2 = new Condition("name", Operator.NE, "Galaxy");

Filter[] nf = {c1,c2};

Filter exp1 = new Expression(BooleanOperator.AND, nf );
```

You can use different combinations of the three filters and create multiple or subfilters based on your requirement.

Filters for Custom Attributes

If the includeCustomAttributes is set to TRUE as a constraint then the *getMPLSObjectname* method will return the MPLS Objects and include the custom attributes for them.

For example, if the includeCustomAttributes is set to TRUE for getVpns(), then the method will return all the VPNs and also the Custom Attributes for them.

Chapter 4: MPLS WS- I

The MPLS web services help you to read and set MPLS objects and the custom attributes. The MPLS web services also help you to update values for the custom attributes.

L3 VPN

In an MPLS-enabled network, the provider edge (PE) routers reside on the perimeter of the service provider’s network and communicate with each other using the label-switched paths. Each PE router maintains a Virtual Routing and Forwarding (VRF) table. A Virtual Private Network (VPN) is formed by the set of VRFs tables on a PE router. Each Layer 3 Virtual Private Network (L3 VPNs) contains the backbone routers (P routers), the provider edge (PE) routers, and the customer edge (CE) routers.

The iSPI for MPLS monitors the L3 VPNs configured on the provider edge nodes of the network.

The L3 VPN WS enables you to read and set the L3 VPN name. The list of L3 VPN function calls is as follows:

- L3Vpn[] getVpns(Filter filter, String sgUser)
- L3Vpn getVpnByUuid(String uuid, String sgUser)
- L3Vpn getVpnById(Long id, String sgUser)
- L3Vpn getVpnByName(String name, String sgUser)
- setVpnName(Long id,String name)

  Note: setVpnName is a unique function available exclusively for L3 VPN.

- updateCustomAttributes(Long id,CustomAttribute[] customAttributes)
- addCustomAttributes(Long id,CustomAttribute[] customAttributes)
- removeCustomAttributes(Long id,String[] customAttributes)

The L3 VPN object provides the following attributes:

- String uuid
- String name
- String description
- String status // *status of the selected L3 VPN*
- Long id

- Date createTime
- Date updateTime
- String importRTList // *Imports a list of Route Targets from VRFs participating to form the L3 VPN.*
- String exportRTList // *Exports a list of Route Targets to VRFs participating to form the L3 VPN.*
- String topoType
- String vpnId
- String managementMode

Strings used for the topoType object:

- FULL_MESH // *Full Mesh VPN is formed if all the VRFs have the same route target. All the Provider Edge (PE) routers communicate with each other.*
- NON_FULLMESH //*All the VRFs are not communicating with the other VRFs belonging to a VPN.*
- SINGLETON // *A VRF participating to form a single L3 VPN.*
- HUB_AND_SPOKE // *Hub and spoke is a star shaped topology where the Hub VRF is in the center and all the spoke-VRFs connect to the Hub.*

Date statusLastModifiedTime

CustomAttribute[] customAttributes

CustomAttribute object:

- String name
- String value

L3 VRF

Each PE router maintains a Virtual Routing and Forwarding (VRF) table. The purpose of the VRF is to transfer traffic towards the correct customer edge (CE) router. An L3 VPN is formed by a set of VRFs. A VRF participates only in a single VPN and is grouped on the basis of the Route Targets (RTs).

The L3 VRF WS enables you to read the L3 VRFs participating to form an L3 VPN. The list of L3 VRF function calls is as follows:

- L3Vrf[] getVrfs(Filter filter, String sgUser)
- L3Vrf getVrfByUuid(String uuid, String sgUser)

- L3Vrf getVrfById(Long Id, String sgUser)
- L3Vrf getVrfByName(String name, String sgUser)
- updateCustomAttributes(Long id,CustomAttribute[] customAttributes)
- addCustomAttributes(Long id,CustomAttribute[] customAttributes)
- removeCustomAttributes(Long id,String[] customAttributes)

The L3 VRF object provides the following attributes that are used as filters:

- String uuid
- String name
- String status
- Long id
- Date createTime
- Date statusLastModifiedTime
- String description
- String routeDistinguisher // *Route Distinguisher is a unique numerical value of the VRF that provides the accurate resolution of the overlapping IP address domains.*
- String importRTList
- String exportRTList
- String hostedOnNodeUuid // *Name of the PE node on which the VRF is hosted.*
- String vpnUuid // *Name of the VPN to which the VRF belongs.*
- String managementMode
- CustomAttribute[] customAttributes

  CustomAttribute object:

  - String name
  - String value

MVPN

In the MVPN topology, the provider edge (PE) routers sit on the perimeter of the service provider’s network and communicate with Provider routers inside the MPLS cloud and customer edge (CE) routers that are located and managed at the customer sites. The PE routers are configured with multicast-enabled VRF (MVRF) and use the multicast services to transmit the customer multicast traffic. A Layer 3 VPN (L3 VPN) can contain more than one Multicast Domain (MD) participating to form multiple MVPNs. Each MVPN consists of one default Multicast Distribution Tree (MDT). The iSPI for MPLS discovers and monitors the MVPNs participating in the network.

The MVPN WS enables you to read the MVPNs in the network. The list of MVPN function calls is as follows:

- MVpn[] getMVpns(Filter filter)
- MVpn getMVpnByUuid(String uuid)
- MVpn getMVpnById(Long Id)
- MVpn getMVpnByName(String name)
- updateCustomAttributes(Long id,CustomAttribute[] customAttributes)
- addCustomAttributes(Long id,CustomAttribute[] customAttributes)
- removeCustomAttributes(Long id,String[] customAttributes)

The MVPN object provides the following attributes:

- String uuid
- String name
- String status
- Long id
- Date statusLastModifiedTime
- String defautMDTAddr // *The MDT group addresses used for forwarding the multicast packets in the MVPN network.*
- String vpnUuid
- CustomAttribute[] customAttributes

  CustomAttribute object:

  - String name
  - String value

MVRF

The PE routers participating in the MPLS cloud are configured with multicast-enabled VRF (MVRF) and use the multicast services to transmit the customer multicast traffic. The MVRF is the multicast-enabled VRF that participates to form an MVPN. The iSPI for MPLS discovers and monitors the MVRFs participating to form an MVPN.

The MVRF WS enables you to read the MVRFs. The list of MVRF function calls is as follows:

- MVrf[] getMVrfs(Filter filter)
- MVrf getMVrfByUuid(String uuid)
- MVrf getMVrfById(Long Id)
- MVrf getMVrfByName(String name)

The MVRF object provides the following attributes:

- String uuid
- String status
- Long id
- Date statusLastModifiedTime
- String defaultMDTAddr // *The MDT group addresses used for forwarding the multicast packets in the MVPN network.*
- String dataMDTAddrRange // *The Range of the group addresses for a specific MVPN group.*
- String dataMDTWildCardBits
- String defaultMDTEncapType
- Long dataMDTThreshold // *The maximum bandwidth value configured on the selected MVRF for the MVPN traffic using the data MDT.*
- String tunnelIfUuid // *Name of the TE Tunnel associated with the MVRF.*
- String vrfUuid // *Name of the VRF associated with the MVRF.*

CustomAttribute object:

- String name
- String value

> **Note:** MVRF is Multicast enabled VRF and CustomAttribute object mentioned for MVRF and L3 VRF are same.

PE Inteface

The PE interface WS enables you to read the existing PE interfaces.

The list of PE Interface function calls is as follows:

- PeInterface[] getPeInterfaces(Filter filter, String sgUser)
- PeInterface getPeInterfaceByUuid(String uuid, String sgUser)
- PeInterface getPeInterfaceById(Long id, String sgUser)

The PE Interface object provides the following attributes:

- String uuid
- Long id
- String ifUUID
- String hostedOnNodeUuid
- String routingProtocol // *Protocol used for data transmission between PE and CE routers.*
- String vrfUuid
- String[] ceifUuidList // *List of CE interfaces associated with the PE.*

Pseudowire

A PseudoWire VC is a point-to-point link for data transmission between the two nodes using the L2 technology. There are two types of L2 VPNs - Virtual Private Wire Service (VPWS) and Virtual Private LAN Service (VPLS). The iSPI for MPLS discovers and monitors the PW participating in the network.

The Pseudowire WS enables you to read the Pseudowire VC. The list of Pseudowire function calls is as follows:

- PseudoWire[] getPseudoWires(Filter filter, String sgUser)
- PseudoWire getPseudoWireByUuid(String uuid, String sgUser)
- PseudoWire getPseudoWireById(Long id, String sgUser)

- updateCustomAttributes(Long id,CustomAttribute[] customAttributes)
- addCustomAttributes(Long id,CustomAttribute[] customAttributes)
- removeCustomAttributes(Long id,String[] customAttributes)

The Pseudowire object provides the following attributes:

- String uuid
- Long id
- Long vcId // *Unique index identifier for each virtual circuit.*
- String status
- Date createTime
- Date statusLastModifiedTime
- String vcType // *Encapsulation type used for data transmision.*
- CustomAttribute[] customAttributes
- String vpwsUuid // *Name of the Pseudowire VC participating to form a VPWS VPN.*
- String managementMode

TE Tunnel

The TE Tunnel WS enables you to read the TE Tunnels. The list of TE Tunnel function calls is as follows:

- Tunnel[] getTunnels(Filter filter, String sgUser)
- Tunnel getTunnelByUuid(String uuid, String sgUser)
- Tunnel getTunnelById(Long id, String sgUser)
- updateCustomAttributes(Long id,CustomAttribute[] customAttributes)
- addCustomAttributes(Long id,CustomAttribute[] customAttributes)
- removeCustomAttributes(Long id,String[] customAttributes)

The TE tunnel object provides the following attributes:

- String uuid;
- String name;

- String status;
- Long id;
- Date createTime;
- Date statusLastModifiedTime;
- String description;
- String attributes;
- Long bandWidth; // *The bandwidth configured for the selected TE Tunnel.*
- int setupPriority; // *The priority used to determine if the selected TE Tunnel is eligible to be preempted.*
- int holdPriority; // *The holding priority value specifies the priority used when protecting the tunnel from preemption by other tunnels*.
- String destNodeUuid; // *Name of the node at which the TE Tunnel terminates.*
- String sourceNodeUuid; // *Name of the node at which the TE Tunnel starts.*
- CustomAttribute[] customAttributes

  CustomAttribute object:

  - String name
  - String value

VPLS VPN

A VPLS VPN is formed by Layer 2 VPNs where multiple sites communicate using Ethernet-based multipoint to multipoint communication over a Packet Switched Network (PSN). The iSPI for MPLS monitors the VPLS VPNs participating in the network.

The VPLS VPN service allows you to read the VPLS VPN. This service allows you to determine the VPLS VPNs in your topology.

The list of VPLS VPN function calls is as follows:

- VPLS[] getVPLSs(Filter filter, String sgUser)
- VPLS getVPLSByUuid(String uuid, String sgUser)
- VPLS getVPLSById(Long Id, String sgUser)
- VPLS getVPLSByName(String name, String sgUser)

- updateCustomAttributes(Long id,CustomAttribute[] customAttributes)
- addCustomAttributes(Long id,CustomAttribute[] customAttributes)
- removeCustomAttributes(Long id,String[] customAttributes)

The VPLS VPN object provides the following attributes:

- String uuid
- Long id
- String name
- String status
- Date createTime
- Date statusLastModifiedTime
- Long vplsId
- String managementMode
- CustomAttribute[] customAttributes

  CustomAttribute object:

  - String name
  - String value

VPWS VPN

A VPWS VPN is formed by Layer 2 VPNs where point-to-point link connects the CE devices through a Packet Switched Network (PSN) using PseudoWires VCs. The iSPI for MPLS discovers and monitors the VPWS VPNs participating in the network.

The VPWS VPN WS enables you to read the VPWS VPNs in the network. The list of VPWS VPN function calls is as follows:

- VPWS[] getVPWSs(Filter filter, String sgUser)
- VPWS getVPWSByUuid(String uuid, String sgUser)
- VPWS getVPWSById(Long Id, String sgUser)
- VPWS getVPWSByName(String name, String sgUser)
- updateCustomAttributes(Long id,CustomAttribute[] customAttributes)

- addCustomAttributes(Long id,CustomAttribute[] customAttributes)
- removeCustomAttributes(Long id,String[] customAttributes)

The VPWS VPN object provides the following attributes:

- String uuid
- Long id
- String name
- String status
- Date createTime
- Date statusLastModifiedTime
- String managementMode
- CustomAttribute[] customAttributes

  CustomAttribute object:

  - String name
  - String value

Attachment Circuit

The Attachment Circuit WS enables you to read the Attachment Circuit. The function call for Attachment Circuit is given below:

- AttachmentCircuit[] getAttachmentCircuits(Filter filter, String sgUser);

The AttachmentCircuit object provides the following attributes:

- String uuid;
- Long id;
- String nnmUuid;
- String hostedOnNodeUuid;
- String acType; // *Port or Interface*.
- String vfiUuid;
- String vclspUuid;

VC LSP

The VC LSP WS enables you to read the VC LSP. The function call for VC LSP is given below:

- VcLsp[] getVcLsps(Filter filter, String sgUser);

The VC LSP object provides the following attributes:

- String uuid;
- String name;
- Long id;
- String description;
- String hostedNodeUUID;
- String endPointAddress;
- String acType;
- Long vcId;
- String status;
- Date statusLastModifiedTime;
- String managementMode;
- String vfiUuid;
- String pseudoWireUuid;

VFI

The VFI WS enables you to read the VFI. The function call for VFI is given below:

- Vfi[] getVfis(Filter filter, String sgUser);
- updateCustomAttributes(Long id,CustomAttribute[] customAttributes);
- addCustomAttributes(Long id,CustomAttribute[] customAttributes);
- removeCustomAttributes(Long id,String[] customAttributes);

The VFI object provides the following attributes:

- String uuid;
- String name;
- String status;
- Long id;
- Date statusLastModifiedTime;
- String description;
- Long vpnId;
- String hostedOnNodeUuid;
- String vplsUuid;
- String managementMode;
- CustomAttribute[] customAttributes

  CustomAttribute object:

  - String name
  - String value

Status for the MPLS Objects

The status for the MPLS objects is as follows:

- NOSTATUS
- NORMAL
- DISABLED
- UNKNOWN
- WARNING
- MINOR
- MAJOR
- CRITICAL

Chapter 5: WS Registry

NNMi maintains a registry bean. Every client accessing the MPLS web services has to contact the NNMi registry to obtain actual URLs for given endpoints. For more information see, *"Accessing the MPLS WS-I Services" on page 5*

We appreciate your feedback!

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Developer's Toolkit (Network Node Manager iSPI for MPLS 10.00)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to docfeedback@hp.com.