



# Nokia Service Router Linux

## Release 23.7

## NetOps Development Kit API Reference

---

3HE 19561 AAAA TQZZA  
Edition: 01  
August 2023

© 2023 Nokia.

Use subject to Terms available at: [www.nokia.com/terms](http://www.nokia.com/terms).

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

---

This document includes Nokia proprietary and confidential information, which may not be distributed or disclosed to any third parties without the prior written consent of Nokia.

This document is intended for use by Nokia's customers ("You"/"Your") in connection with a product purchased or licensed from any company within Nokia Group of Companies. Use this document as agreed. You agree to notify Nokia of any errors you may find in this document; however, should you elect to use this document for any purpose(s) for which it is not intended, You understand and warrant that any determinations You may make or actions You may take will be based upon Your independent judgment and analysis of the content of this document.

Nokia reserves the right to make changes to this document without notice. At all times, the controlling version is the one available on Nokia's site.

No part of this document may be modified.

NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF AVAILABILITY, ACCURACY, RELIABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT WILL NOKIA BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT, EVEN IN THE CASE OF ERRORS IN OR OMISSIONS FROM THIS DOCUMENT OR ITS CONTENT.

Copyright and trademark: Nokia is a registered trademark of Nokia Corporation. Other product names mentioned in this document may be trademarks of their respective owners.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

© 2023 Nokia.

# Table of contents

<b>1</b>	<b>About this guide.....</b>	<b>7</b>
1.1	Precautionary and information messages.....	7
1.2	Conventions.....	7
<b>2</b>	<b>What's new.....</b>	<b>9</b>
<b>3</b>	<b>Introduction.....</b>	<b>10</b>
3.1	Datastore.....	10
3.2	gRPC.....	10
3.3	Protocol buffers.....	10
<b>4</b>	<b>Protocol Documentation.....</b>	<b>12</b>
4.1	appid_service.proto.....	12
4.1.1	AppIdentData.....	12
4.1.2	AppIdentKey.....	12
4.1.3	AppIdentNotification.....	12
4.1.4	AppIdentSubscriptionRequest.....	13
4.2	bfd_service.proto.....	13
4.2.1	BfdmgrGeneralSessionDataPb.BfdmgrSessionSubType.....	13
4.2.2	BfdmgrSessionStatus.....	13
4.2.3	BfdmgrSessionType.....	14
4.2.4	BfdSessionNotification.....	14
4.2.5	BfdSessionSubscriptionRequest.....	14
4.2.6	BfdmgrGeneralSessionDataPb.....	15
4.2.7	BfdmgrGeneralSessionKeyPb.....	15
4.2.8	BfdmgrGeneralSessionKeyPb.MicrobfdKey.....	15
4.2.9	BfdmgrGeneralSessionKeyPb.P2pKey.....	16
4.2.10	BfdmgrGeneralSessionKeyPb.SbfdechoKey.....	16
4.3	config_service.proto.....	16
4.3.1	ConfigData.....	16
4.3.2	ConfigKey.....	17
4.3.3	ConfigNotification.....	17
4.3.4	ConfigSubscriptionRequest.....	17
4.4	interface_service.proto.....	18

---

4.4.1	InterfaceData.....	18
4.4.2	InterfaceKey.....	18
4.4.3	InterfaceNotification.....	18
4.4.4	InterfaceSubscriptionRequest.....	19
4.5	lldp_service.proto.....	19
4.5.1	LldpNeighborDataPb.PortSubType.....	19
4.5.2	LldpNeighborKeyPb.ChassisIdType.....	20
4.5.3	LldpNeighborDataPb.....	20
4.5.4	LldpNeighborKeyPb.....	21
4.5.5	LldpNeighborNotification.....	21
4.5.6	LldpNeighborSubscriptionRequest.....	21
4.6	mpls_service.proto.....	22
4.6.1	MplsRoutePb.Operation.....	22
4.6.2	MplsRouteAddRequest.....	22
4.6.3	MplsRouteAddResponse.....	22
4.6.4	MplsRouteDeleteRequest.....	22
4.6.5	MplsRouteDeleteResponse.....	23
4.6.6	MplsRouteInfo.....	23
4.6.7	MplsRouteKeyPb.....	23
4.6.8	MplsRoutePb.....	23
4.6.9	SdkMgrMplsRouteService.....	24
4.6.10	Methods with deprecated option.....	24
4.7	networkinstance_service.proto.....	24
4.7.1	NetworkInstanceData.NetInstType.....	25
4.7.2	NetworkInstanceData.....	25
4.7.3	NetworkInstanceKey.....	25
4.7.4	NetworkInstanceNotification.....	25
4.7.5	NetworkInstanceSubscriptionRequest.....	26
4.8	nexthop_group_service.proto.....	26
4.8.1	NextHop.ResolutionType.....	26
4.8.2	NextHop.ResolveToType.....	26
4.8.3	MplsNextHop.....	26
4.8.4	NextHop.....	27
4.8.5	NextHopGroup.....	27
4.8.6	NextHopGroupDeleteRequest.....	27
4.8.7	NextHopGroupDeleteResponse.....	27

4.8.8	NextHopGroupInfo.....	28
4.8.9	NextHopGroupKey.....	28
4.8.10	NextHopGroupNotification.....	28
4.8.11	NextHopGroupRequest.....	29
4.8.12	NextHopGroupResponse.....	29
4.8.13	NextHopGroupSubscriptionRequest.....	29
4.8.14	SdkMgrNextHopGroupService.....	29
4.9	route_service.proto.....	30
4.9.1	IpRouteNotification.....	30
4.9.2	IpRouteSubscriptionRequest.....	30
4.9.3	RouteAddRequest.....	30
4.9.4	RouteAddResponse.....	31
4.9.5	RouteDeleteRequest.....	31
4.9.6	RouteDeleteResponse.....	31
4.9.7	RouteInfo.....	31
4.9.8	RouteKeyPb.....	32
4.9.9	RoutePb.....	32
4.9.10	SdkMgrRouteService.....	32
4.10	sdk_common.proto.....	33
4.10.1	IfEthernetDuplexModeType.....	33
4.10.2	IfEthernetPortSpeedType.....	33
4.10.3	IfMgrIfType.....	34
4.10.4	IfOperDownReason.....	35
4.10.5	IfOperStateType.....	36
4.10.6	IfTransceiverFecType.....	37
4.10.7	IpAddressState.....	37
4.10.8	SdkMgrOperation.....	38
4.10.9	SdkMgrStatus.....	38
4.10.10	AgentReply.....	38
4.10.11	EvpnEthSegIdPb.....	38
4.10.12	GlobalIfId.....	39
4.10.13	IpAddrPrefLenPb.....	39
4.10.14	IpAddressPb.....	39
4.10.15	IpInterfaceAddrPrefixPb.....	39
4.10.16	MacAddressPb.....	40
4.10.17	MplsLabel.....	40

---

4.10.18	NetInstanceId.....	40
4.10.19	PortIdPb.....	40
4.10.20	SyncRequest.....	41
4.10.21	SyncResponse.....	41
4.11	sdk_service.proto.....	41
4.11.1	NotificationRegisterRequest.Operation.....	41
4.11.2	AgentRegistrationRequest.....	41
4.11.3	AgentRegistrationResponse.....	42
4.11.4	AppIdRequest.....	42
4.11.5	AppIdResponse.....	42
4.11.6	KeepAliveRequest.....	43
4.11.7	KeepAliveResponse.....	43
4.11.8	Notification.....	43
4.11.9	NotificationQueryRequest.....	44
4.11.10	NotificationQueryResponse.....	44
4.11.11	NotificationQuerySubscription.....	44
4.11.12	NotificationRegisterRequest.....	44
4.11.13	NotificationRegisterResponse.....	45
4.11.14	NotificationStreamRequest.....	46
4.11.15	NotificationStreamResponse.....	46
4.11.16	SdkMgrService.....	46
4.11.17	SdkNotificationService.....	47
4.12	telemetry_service.proto.....	47
4.12.1	TelemetryData.....	47
4.12.2	TelemetryDeleteRequest.....	47
4.12.3	TelemetryDeleteResponse.....	47
4.12.4	TelemetryInfo.....	48
4.12.5	TelemetryKey.....	48
4.12.6	TelemetryUpdateRequest.....	48
4.12.7	TelemetryUpdateResponse.....	48
4.12.8	SdkMgrTelemetryService.....	49
4.13	Scalar Value Types.....	49
4.13.1	Scalar.....	49

# 1 About this guide

The NetOps Development Kit (NDK) allows operators to program high-performance, integrated agents that run alongside the Nokia Service Router Linux (SR Linux). This document provides programming gRPC APIs used with the NDK.

This document is intended for users who plan to program high-performance, integrated agents for SR Linux.

**Note:**

This manual covers the current release and may also contain some content that will be released in later maintenance loads. See the *SR Linux Release Notes* for information on features supported in each load.

Configuration and command outputs shown in this guide are examples only; actual displays may differ depending on supported functionality and user configuration.

## 1.1 Precautionary and information messages

The following are information symbols used in the documentation.



**DANGER:** Danger warns that the described activity or situation may result in serious personal injury or death. An electric shock hazard could exist. Before you begin work on this equipment, be aware of hazards involving electrical circuitry, be familiar with networking environments, and implement accident prevention procedures.



**WARNING:** Warning indicates that the described activity or situation may, or will, cause equipment damage, serious performance problems, or loss of data.



**Caution:** Caution indicates that the described activity or situation may reduce your component or system performance.



**Note:** Note provides additional operational information.



**Tip:** Tip provides suggestions for use or best practices.

## 1.2 Conventions

Nokia SR Linux documentation uses the following command conventions.

- **Bold** indicates a command that the user must enter.
- Input and output examples are displayed in Courier text.
- An open right angle bracket indicates a progression of menu choices or simple command sequence (often selected from a user interface). Example: **start > connect to**

- A vertical bar (|) indicates a mutually exclusive argument.
- Square brackets ([ ]) indicate optional elements.
- Braces ({ }) indicate a required choice. When braces are contained within square brackets, they indicate a required choice within an optional element.
- *Italic* indicates a variable.

Generic IP addresses are used in examples. Replace these with the appropriate IP addresses used in the system.



## 2 What's new

Topic	Change
<a href="#">ConfigKey</a>	Updated
<a href="#">IfOperDownReason</a>	Updated

## 3 Introduction

SR Linux provides a NetOps Development Kit (NDK), with a suite of libraries to assist operators with developing agents that run alongside SR Linux applications.

Agents built with the gRPC NDK function similar to other applications provided with SR Linux. SR Linux applications share state details with each other using a publish/ subscribe (pub/sub) architecture. Agents have their own table space within the IDB and can subscribe and receive a notification to events occurring on the device, or create their own table space and publish data to it. This data can be read by other applications within SR Linux, allowing route modifications by publishing routes to the IDB for selection by the FIB manager.

### 3.1 Datastore

The gRPC NDK allows you to add your own configuration to the system in a non-persistent manner. An NDK-added configuration is considered short-term, and its state is bound to the state of the agent. If an agent fails, the configuration added by the agent using the NDK is removed.

Agents can also add configurations through normal APIs (gNMI/JSON-RPC/CLI). This configuration persists across an agent failure, and the only way to remove it is to overwrite it with a commit command. The short-term datastore is used for agent route injection, while traditional methods for configuring the device are persistent.

### 3.2 gRPC

SR Linux uses gRPC for inter-process communication. gRPC is a client application that directly calls methods on a server application on a different machine as if it was a local object. The supported external APIs (CLI, gNMI, and JSON-RPC) communicate with the SR Linux and retrieve state information using gRPC.

On the server side, the server implements the interface and runs a gRPC server to handle client calls. On the client side, the client has a stub (or client) that provides the same methods as the server.

gRPC clients and servers can run and talk to each other in a number of environments and can be written in any supported gRPC language. Clients can be created in Go, Python, Ruby, or any other language with gRPC support.

### 3.3 Protocol buffers

SR Linux's gRPC NDK uses protocol buffers. Protocol buffers are automated mechanisms for serializing structured data. You define how you want your data to be structured once, then you can use special generated source code to easily write and read your structured data to and from a variety of data streams using a variety of languages. You can also update a data structure without breaking deployed programs that are compiled against an old format.

When working with protocol buffers, structure is defined for serialized data in a proto file (a regular text file with a .proto extension). Each protocol buffer message is a small logical record of information containing a series of name-value pairs. Each message type has one or more uniquely numbered fields, and each field has a name and value type.

Messages also have optional arguments that specify if fields are optional, required, or repeated. New fields can be added to message formats without breaking backwards compatibility, and old binaries ignore any new fields when parsing the message. This allows the gRPC NDK to evolve over time without impacting current deployments.

Once the data structure is specified, a protocol buffer compiler (protoc) generates data access classes from the proto definition. These provide simple accessors for each field (like `ConfigData()` and `set_ConfigData()`) and methods to serialize and parse the complete structure to and from raw bytes.

## 4 Protocol Documentation

### 4.1 appid\_service.proto

#### 4.1.1 AppIdentData

Represents appid data.

*Table 1: AppIdentData*

Field	Type	Label	Description
name	<a href="#">string</a>		Application name
author	<a href="#">string</a>		Author name
is_connected	<a href="#">bool</a>		Connected to IDB or not
version	<a href="#">string</a>		Version string

#### 4.1.2 AppIdentKey

Represents appid key.

*Table 2: AppIdentKey*

Field	Type	Label	Description
id	<a href="#">uint32</a>		Application id

#### 4.1.3 AppIdentNotification

Represents appid notification.

*Table 3: AppIdentNotification*

Field	Type	Label	Description
op	<a href="#">SdkMgrOperation</a>		Operation such as create, delete, or update
key	<a href="#">AppIdentKey</a>		AppIdent key

Field	Type	Label	Description
data	<a href="#">AppldentData</a>		Appldent data

#### 4.1.4 AppldentSubscriptionRequest

Represents appid subscription request.

Table 4: *AppldentSubscriptionRequest*

Field	Type	Label	Description
key	<a href="#">AppldentKey</a>		Optional, to filter on name

## 4.2 bfd\_service.proto

#### 4.2.1 BfdmgrGeneralSessionDataPb.BfdmgrSessionSubType

Represents BFD session subtype.

Table 5: *BfdmgrGeneralSessionDataPb.BfdmgrSessionSubType*

Name	Number	Description
SESSION_SUB_TYPE_UNKNOWN	0	Session subtype unknown
SESSION_SUB_TYPE_SINGLE_HOP	1	Single-hop session
SESSION_SUB_TYPE_MULTI_HOP	2	Multi-hop session
SESSION_SUB_TYPE_MICROBFD	3	microbfd session
SESSION_SUB_TYPE_SBFDF_ECHO	4	microbfd session

#### 4.2.2 BfdmgrSessionStatus

Represents BFD session status.

Table 6: *BfdmgrSessionStatus*

Name	Number	Description
INVALID	0	Session invalid
ADMIN_DOWN	1	Admin down
DOWN	2	Status down
INIT	3	Status initializing
UP	4	Status up and running

### 4.2.3 BfdmgrSessionType

Represents BFD session type.

Table 7: *BfdmgrSessionType*

Name	Number	Description
SESSION_TYPE_UNKNOWN	0	Unknown session type
SESSION_TYPE_P2P	1	Peer-to-peer session type
SESSION_TYPE_MICROBFD	2	microbfd session type
SESSION_TYPE_SBFID_ECHO	3	seamless BFD session type, echo initiator

### 4.2.4 BfdSessionNotification

Represents BFD session notification.

Table 8: *BfdSessionNotification*

Field	Type	Label	Description
op	<a href="#">SdkMgrOperation</a>		Operation such as session create, delete, or update
key	<a href="#">BfdmgrGeneralSessionKeyPb</a>		Session key
data	<a href="#">BfdmgrGeneralSessionDataPb</a>		Session data

#### 4.2.5 BfdSessionSubscriptionRequest

Represents BFD session subscription request.

Table 9: BfdSessionSubscriptionRequest

Field	Type	Label	Description
key	<a href="#">BfdmgrGeneralSessionKeyPb</a>		Optional, to filter on name

#### 4.2.6 BfdmgrGeneralSessionDataPb

Represents BFD session data.

Table 10: BfdmgrGeneralSessionDataPb

Field	Type	Label	Description
status	<a href="#">BfdmgrSessionStatus</a>		Status of the session
sub_type	<a href="#">BfdmgrSessionSubType</a>		Subtype of the session
src_if_id	<a href="#">uint32</a>		src_if_id is only populated for P2P type Source interface ID

#### 4.2.7 BfdmgrGeneralSessionKeyPb

Represents BFD session key.

Table 11: BfdmgrGeneralSessionKeyPb

Field	Type	Label	Description
type	<a href="#">BfdmgrSessionType</a>		type is always present, other key field presence is determined by type Session type
p2p	<a href="#">P2pKey</a>		
microbfd	<a href="#">MicrobfdKey</a>		
sbfdecho	<a href="#">SbfdechoKey</a>		

#### 4.2.8 BfdmgrGeneralSessionKeyPb.MicrobfdKey

Table 12: *BfdmgrGeneralSessionKeyPb.MicrobfdKey*

Field	Type	Label	Description
interface_name	string		

#### 4.2.9 BfdmgrGeneralSessionKeyPb.P2pKey

Table 13: *BfdmgrGeneralSessionKeyPb.P2pKey*

Field	Type	Label	Description
src_ip_addr	IpAddressPb		Source IP address of the session
dst_ip_addr	IpAddressPb		Destination IP address of the session
instance_id	uint32		Network instance identifier
ipv6_ll_if_id	uint32		Global if id for ipv6 link local session, otherwise 0
specified_discr	bool		True if discriminators are specified

#### 4.2.10 BfdmgrGeneralSessionKeyPb.SbfdechoKey

Table 14: *BfdmgrGeneralSessionKeyPb.SbfdechoKey*

Field	Type	Label	Description
instance_id	uint32		Network instance identifier
sr_policy_segment_list_id	uint32		
sr_policy_color	uint32		
sr_policy_endpoint	IpAddressPb		

### 4.3 config\_service.proto

#### 4.3.1 ConfigData

Represents configuration data.



Table 15: ConfigData

Field	Type	Label	Description
json	string		Entire configuration fragment as JSON string
data	bytes		Entire configuration fragment as binary data

### 4.3.2 ConfigKey

Represents configuration key.

Table 16: ConfigKey

Field	Type	Label	Description
js_path	string		JSON path formatted string from YANG; for example, interface.my_field
keys	string	repeated	Value for keys
js_path_with_keys	string		JSON path formatted string from YANG; for example, interface{.name==ethernet1/1}.my_field

### 4.3.3 ConfigNotification

Represents configuration notification message to subscribe to configuration events

Table 17: ConfigNotification

Field	Type	Label	Description
op	SdkMgrOperation		Operation indicating create, delete, or update
key	ConfigKey		Configuration key
data	ConfigData		Configuration data

### 4.3.4 ConfigSubscriptionRequest

Represents configuration subscription request.

Table 18: ConfigSubscriptionRequest

Field	Type	Label	Description
key	<a href="#">ConfigKey</a>		Optional, to filter on name

## 4.4 interface\_service.proto

### 4.4.1 InterfaceData

Represents interface data.

Table 19: InterfaceData

Field	Type	Label	Description
admin_is_up	<a href="#">uint32</a>		Admin state
mtu	<a href="#">uint32</a>		Maximum transmission unit
if_type	<a href="#">IfMgrIfType</a>		Interface type; for example, loopback, physical, or LAG
port_id	<a href="#">PortIdPb</a>		Port identifier
description	<a href="#">string</a>		Interface description
mac_addr	<a href="#">MacAddressPb</a>		MAC address
aggregate_id	<a href="#">string</a>		associated aggregate id
oper_is_up	<a href="#">uint32</a>		Operational state

### 4.4.2 InterfaceKey

Represents interface key.

Table 20: InterfaceKey

Field	Type	Label	Description
if_name	<a href="#">string</a>		Interface name; for example, ethernet 1/1

### 4.4.3 InterfaceNotification

Represents interface notification.

Table 21: InterfaceNotification

Field	Type	Label	Description
op	<a href="#">SdkMgrOperation</a>		Operation such as create, delete, or update
key	<a href="#">InterfaceKey</a>		Interface key
data	<a href="#">InterfaceData</a>		Interface data

### 4.4.4 InterfaceSubscriptionRequest

Represents interface subscription request.

Table 22: InterfaceSubscriptionRequest

Field	Type	Label	Description
key	<a href="#">InterfaceKey</a>		Optional, to filter on name

## 4.5 lldp\_service.proto

### 4.5.1 LldpNeighborDataPb.PortSubType

Represents port subtype.

Table 23: LldpNeighborDataPb.PortSubType

Name	Number	Description
RESERVED	0	Reserved for future use
INTERFACE_ALIAS	1	Alias of the interface
PORT_COMPONENT	2	Port identifier based on a locally defined port component
MAC_ADDRESS	3	MAC address
NETWORK_ADDRESS	4	Network address
INTERFACE_NAME	5	Name of the interface

Name	Number	Description
AGENT_CIRCUIT_ID	6	Port identifier based on the circuit ID in the DHCP relay agent information option
LOCALLY_ASSIGNED	7	Port identifier based on a locally defined alphanumeric string

#### 4.5.2 LdpNeighborKeyPb.ChassisIdType

Represents chassis type.

Table 24: LdpNeighborKeyPb.ChassisIdType

Name	Number	Description
RESERVED	0	Reserved for future use
CHASSIS_COMPONENT	1	Chassis identifier based on a locally defined chassis component
INTERFACE_ALIAS	2	Alias of the interface
PORT_COMPONENT	3	Chassis identifier based on a locally defined port component
MAC_ADDRESS	4	MAC address
NETWORK_ADDRESS	5	Network address
INTERFACE_NAME	6	Name of the interface
LOCALLY_ASSIGNED	7	Chassis identifier based on a locally defined value

#### 4.5.3 LdpNeighborDataPb

Represents LLDP neighbor data.

Table 25: LdpNeighborDataPb

Field	Type	Label	Description
port_id	<a href="#">string</a>		Port identifier
port_type	<a href="#">PortSubType</a>		Port type
source_mac	<a href="#">MacAddressPb</a>		Port MAC address

Field	Type	Label	Description
bgp_peer_address	<a href="#">IpAddressPb</a>	repeated	LLDP BGP autodiscovered addresses
bgp_group_id	<a href="#">uint32</a>		BGP group identifier
system_name	<a href="#">string</a>		System name
system_description	<a href="#">string</a>		System description

#### 4.5.4 LldpNeighborKeyPb

Represents LLDP neighbor key.

Table 26: *LldpNeighborKeyPb*

Field	Type	Label	Description
interface_name	<a href="#">string</a>		Local interface name
chassis_id	<a href="#">string</a>		Chassis identifier
chassis_type	<a href="#">ChassisIdType</a>		Chassis type

#### 4.5.5 LldpNeighborNotification

Represents LLDP neighbor notification.

Table 27: *LldpNeighborNotification*

Field	Type	Label	Description
op	<a href="#">SdkMgrOperation</a>		Operation such as create, delete, or update
key	<a href="#">LldpNeighborKeyPb</a>		LLDP neighbor key
data	<a href="#">LldpNeighborDataPb</a>		LLDP neighbor data

#### 4.5.6 LldpNeighborSubscriptionRequest

Represents LLDP neighbor subscription request.

Table 28: *LldpNeighborSubscriptionRequest*

Field	Type	Label	Description
key	<a href="#">LldpNeighborKeyPb</a>		Optional, to filter on name

## 4.6 mpls\_service.proto

### 4.6.1 MplsRoutePb.Operation

Represents MPLS operation.

Table 29: MplsRoutePb.Operation

Name	Number	Description
INVALID_OP	0	Invalid operation
POP	1	Pop operation
SWAP	2	Swap operation

### 4.6.2 MplsRouteAddRequest

Represents MPLS route add request, which can include one or more MPLS routes.

Table 30: MplsRouteAddRequest

Field	Type	Label	Description
routes	<a href="#">MplsRouteInfo</a>	repeated	MPLS routes

### 4.6.3 MplsRouteAddResponse

Represents MPLS route add response.

Table 31: MplsRouteAddResponse

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Status of MPLS route add request
error_str	<a href="#">string</a>		Detailed error string

### 4.6.4 MplsRouteDeleteRequest

Represents MPLS route delete request, which can include one or more MPLS routes.

Table 32: *MplsRouteDeleteRequest*

Field	Type	Label	Description
routes	<a href="#">MplsRouteKeyPb</a>	repeated	MPLS routes

#### 4.6.5 MplsRouteDeleteResponse

Represents MPLS route delete response.

Table 33: *MplsRouteDeleteResponse*

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Status of MPLS route delete request
error_str	<a href="#">string</a>		Detailed error string

#### 4.6.6 MplsRouteInfo

Represents MPLS route information; contains key and data.

Table 34: *MplsRouteInfo*

Field	Type	Label	Description
key	<a href="#">MplsRouteKeyPb</a>		MPLS route key
data	<a href="#">MplsRoutePb</a>		MPLS route data

#### 4.6.7 MplsRouteKeyPb

Represents MPLS route key.

Table 35: *MplsRouteKeyPb*

Field	Type	Label	Description
top_label	<a href="#">MplsLabel</a>		Top label
net_inst_name	<a href="#">string</a>		Network instance name

#### 4.6.8 MplsRoutePb

Represents MPLS route data.

Table 36: *MplsRoutePb*

Field	Type	Label	Description
nexthop_group_name	string		Next hop group name
operation	Operation		Operation such as POP or SWAP
preference	uint32		Route preference

#### 4.6.9 SdkMgrMplsRouteService

Represents service for MPLS programming.

Table 37: *SdkMgrMplsRouteService*

Method Name	Request Type	Response Type	Description
MplsRouteAdd OrUpdate	MplsRouteAddRequest	MplsRouteAdd Response	MPLS route add or update; can add or update more than one MPLS route in one request
MplsRouteDelete	MplsRouteDeleteRequest	MplsRouteDelete Response	MPLS route delete; can delete more than one MPLS route in one request
SyncStart	SyncRequest	SyncResponse	Synchronization start to begin synchronization operation
SyncEnd	SyncRequest	SyncResponse	Synchronization end to close synchronization operation

#### 4.6.10 Methods with deprecated option

Represents methods with deprecated option.

Table 38: *Methods with deprecated option*

Method Name	Option
MplsRouteAdd OrUpdate	True
MplsRouteDelete	True



## 4.7 networkinstance\_service.proto

### 4.7.1 NetworkInstanceData.NetInstType

Represents network instance type.

Table 39: NetworkInstanceData.NetInstType

Name	Number	Description
DEFAULT	0	Default network instance type
L3VRF	1	L3VRF network instance type

### 4.7.2 NetworkInstanceData

Represents network instance data.

Table 40: NetworkInstanceData

Field	Type	Label	Description
net_inst_id	uint32		Network instance identifier
base_name	string		Base name
oper_is_up	bool		Operation status
router_id	string		Router identifier
inst_type	NetInstType		Network instance type

### 4.7.3 NetworkInstanceKey

Represents network instance key.

Table 41: NetworkInstanceKey

Field	Type	Label	Description
inst_name	string		Network instance name

### 4.7.4 NetworkInstanceNotification

Represents network instance notification.

Table 42: NetworkInstanceNotification

Field	Type	Label	Description
op	<a href="#">SdkMgrOperation</a>		Operation such as create, delete, or update
key	<a href="#">NetworkInstanceKey</a>		Network key
data	<a href="#">NetworkInstanceData</a>		Network data

#### 4.7.5 NetworkInstanceSubscriptionRequest

Represents network instance subscription request.

### 4.8 nexthop\_group\_service.proto

#### 4.8.1 NextHop.ResolutionType

Represents resolution type.

Table 43: NextHop.ResolutionType

Name	Number	Description
INVALID	0	Invalid resolution
REGULAR	1	Regular resolution
MPLS	2	MPLS resolution

#### 4.8.2 NextHop.ResolveToType

Represents resolve-to type.

Table 44: NextHop.ResolveToType

Name	Number	Description
LOCAL	0	Resolve to local routes
DIRECT	1	Resolve to direct routes
INDIRECT	2	Resolve to indirect routes

### 4.8.3 MplsNextHop

Represents MPLS next hop.

Table 45: *MplsNextHop*

Field	Type	Label	Description
ip_nexthop	<a href="#">IpAddressPb</a>		Next-hop IP address
label_stack	<a href="#">MplsLabel</a>	repeated	MPLS label stack

### 4.8.4 NextHop

Represents next-hop.

Table 46: *NextHop*

Field	Type	Label	Description
resolve_to	<a href="#">ResolveToType</a>		Resolve-to type
type	<a href="#">ResolutionType</a>		Resolution type
ip_nexthop	<a href="#">IpAddressPb</a>		IP next-hop address
mpls_nexthop	<a href="#">MplsNextHop</a>		MPLS next-hop

### 4.8.5 NextHopGroup

Represents next-hop group.

Table 47: *NextHopGroup*

Field	Type	Label	Description
next_hop	<a href="#">NextHop</a>	repeated	Next-hops

### 4.8.6 NextHopGroupDeleteRequest

Represents next-hop group delete request.

Table 48: *NextHopGroupDeleteRequest*

Field	Type	Label	Description
group_key	<a href="#">NextHopGroupKey</a>	repeated	Next-hop group key details

### 4.8.7 NextHopGroupDeleteResponse

Represents next-hop group delete response.

Table 49: NextHopGroupDeleteResponse

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Response for next-hop group request
error_str	<a href="#">string</a>		Detailed error string

### 4.8.8 NextHopGroupInfo

Represents next-hop group information.

Table 50: NextHopGroupInfo

Field	Type	Label	Description
key	<a href="#">NextHopGroupKey</a>		Next-hop group key
data	<a href="#">NextHopGroup</a>		Next-hop group data

### 4.8.9 NextHopGroupKey

Represents next-hop group key.

Table 51: NextHopGroupKey

Field	Type	Label	Description
name	<a href="#">string</a>		Next-hop group name
network_instance_name	<a href="#">string</a>		Next-hop group network instance name

### 4.8.10 NextHopGroupNotification

Represents next-hop group notification.

Table 52: NextHopGroupNotification

Field	Type	Label	Description
op	<a href="#">SdkMgrOperation</a>		Operation such as create, delete, or update

Field	Type	Label	Description
key	<a href="#">uint64</a>		Next-hop group key
data	<a href="#">NextHopGroup</a>		Next-hop group data

#### 4.8.11 NextHopGroupRequest

Represents next-hop group request.

*Table 53: NextHopGroupRequest*

Field	Type	Label	Description
group_info	<a href="#">NextHopGroupInfo</a>	repeated	Next-hop group details

#### 4.8.12 NextHopGroupResponse

Represents next-hop group response.

*Table 54: NextHopGroupResponse*

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Response for next-hop group request
error_str	<a href="#">string</a>		Detailed error string

#### 4.8.13 NextHopGroupSubscriptionRequest

Represents next-hop group subscription request.

*Table 55: NextHopGroupSubscriptionRequest*

Field	Type	Label	Description
key	<a href="#">NextHopGroupKey</a>		Optional, to filter on name

#### 4.8.14 SdkMgrNextHopGroupService

Represents service for next-hop group operations.

Table 56: SdkMgrNextHopGroupService

Method Name	Request Type	Response Type	Description
NextHopGroupAdd OrUpdate	<a href="#">NextHopGroupRequest</a>	<a href="#">NextHopGroup Response</a>	Add or update one or more next-hop groups.
NextHopGroupDelete	<a href="#">NextHopGroupDeleteRequest</a>	<a href="#">NextHopGroup DeleteResponse</a>	Delete next-hop group.
SyncStart	<a href="#">SyncRequest</a>	<a href="#">SyncResponse</a>	Synchronization start to open synchronization operation.
SyncEnd	<a href="#">SyncRequest</a>	<a href="#">SyncResponse</a>	Synchronization end to close synchronization operation.

## 4.9 route\_service.proto

### 4.9.1 IpRouteNotification

Represents IP route notification.

Table 57: IpRouteNotification

Field	Type	Label	Description
op	<a href="#">SdkMgrOperation</a>		Operation such as create, delete, or update
key	<a href="#">RouteKeyPb</a>		IP route key
data	<a href="#">RoutePb</a>		IP route data

### 4.9.2 IpRouteSubscriptionRequest

Represents IP route subscription request.

Table 58: IpRouteSubscriptionRequest

Field	Type	Label	Description
key	<a href="#">RouteKeyPb</a>		Optional, to filter on name

### 4.9.3 RouteAddRequest

Represents route add request; can contain more than one route.

Table 59: RouteAddRequest

Field	Type	Label	Description
routes	<a href="#">RouteInfo</a>	repeated	IP routes

#### 4.9.4 RouteAddResponse

Represents route add response.

Table 60: RouteAddResponse

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Status of route add operation
error_str	<a href="#">string</a>		Detailed error string

#### 4.9.5 RouteDeleteRequest

Represents route delete request; can contain more than one route.

Table 61: RouteDeleteRequest

Field	Type	Label	Description
routes	<a href="#">RouteKeyPb</a>	repeated	IP routes

#### 4.9.6 RouteDeleteResponse

Represents route delete response.

Table 62: RouteDeleteResponse

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Status of route delete operation
error_str	<a href="#">string</a>		Detailed error string

#### 4.9.7 RouteInfo

Represents route information.

Table 63: RouteInfo

Field	Type	Label	Description
key	<a href="#">RouteKeyPb</a>		Route key
data	<a href="#">RoutePb</a>		Route data

#### 4.9.8 RouteKeyPb

Represents route key.

Table 64: RouteKeyPb

Field	Type	Label	Description
net_inst_name	<a href="#">string</a>		Network instance name
ip_prefix	<a href="#">IpAddrPrefLenPb</a>		IP prefix

#### 4.9.9 RoutePb

Represents route data.

Table 65: RoutePb

Field	Type	Label	Description
nexthop_group_name	<a href="#">string</a>		Next hop group name
preference	<a href="#">uint32</a>		Preference
metric	<a href="#">uint32</a>		Metric
nexthop	<a href="#">NextHop</a>	repeated	List of next hops
owner_id	<a href="#">uint32</a>		Next hop owner identifier returned only on notification.
nhg_id	<a href="#">uint64</a>		Next-hop group identifier returned only on notification.

#### 4.9.10 SdkMgrRouteService

Represents service for IP route operations.



Table 66: SdkMgrRouteService

Method Name	Request Type	Response Type	Description
RouteAddOrUpdate	<a href="#">RouteAddRequest</a>	<a href="#">RouteAdd Response</a>	Add or update IP routes.
RouteDelete	<a href="#">RouteDeleteRequest</a>	<a href="#">RouteDelete Response</a>	Delete IP routes.
SyncStart	<a href="#">SyncRequest</a>	<a href="#">SyncResponse</a>	Synchronization start for IP routes
SyncEnd	<a href="#">SyncRequest</a>	<a href="#">SyncResponse</a>	Synchronization end for IP routes

## 4.10 sdk\_common.proto

### 4.10.1 IfEthernetDuplexModeType

Represents interface ethernet duplex mode. Corresponds to yang values

Table 67: IfEthernetDuplexModeType

Name	Number	Description
IF_ETH_DUPLEX_MODE_UNSET	0	duplex mode not supported
IF_ETH_DUPLEX_MODE_FULL	1	
IF_ETH_DUPLEX_MODE_HALF	2	

### 4.10.2 IfEthernetPortSpeedType

Represents interface ethernet port speed. Corresponds to yang values

Table 68: IfEthernetPortSpeedType

Name	Number	Description
IF_ETH_PORT_SPEED_UNSET	0	Speed unknown
IF_ETH_PORT_SPEED_10M	1	
IF_ETH_PORT_SPEED_100M	2	

Name	Number	Description
IF_ETH_PORT_SPEED_1G	3	
IF_ETH_PORT_SPEED_10G	4	
IF_ETH_PORT_SPEED_25G	5	
IF_ETH_PORT_SPEED_40G	6	
IF_ETH_PORT_SPEED_50G	7	
IF_ETH_PORT_SPEED_100G	8	
IF_ETH_PORT_SPEED_200G	9	
IF_ETH_PORT_SPEED_400G	10	
IF_ETH_PORT_SPEED_1T	11	

### 4.10.3 IfMgrIfType

Represents interface type.

Table 69: IfMgrIfType

Name	Number	Description
ETHERNET	0	Ethernet interface
LOOPBACK	1	Loopback interface
MANAGEMENT	2	Management interface
AGGREGATE	3	Aggregate(LAG) interface
IRB	4	Integrated Routing and Bridging (IRB) interface
SYSTEM	5	System interface
LIF	6	linux interface
NIC	7	linux nic interface (bus/dev/fn)
VHOST	8	vhost-net interface, vhn-<name> name for sock-path
KKLIF	9	temp name for new style of lif interface

Name	Number	Description
KKVHOST	10	temp name for new style of vhost interface
MAX	11	

#### 4.10.4 IfOperDownReason

Table 70: IfOperDownReason

Name	Number	Description
IF_OPER_DOWN_NONE	0	
IF_OPER_DOWN_PORT_ADMIN_DISABLED	1	
IF_OPER_DOWN_MDA_ADMIN_DISABLED	2	
IF_OPER_DOWN_TRANS_LASER_DISABLED	3	
IF_OPER_DOWN_MDA_NOT_PRESENT	4	
IF_OPER_DOWN_TRANS_NOT_PRESENT	5	
IF_OPER_DOWN_PHY_INIT	6	
IF_OPER_DOWN_LOOPBACK	7	
IF_OPER_DOWN_LOWER_LAYER_DOWN	8	
IF_OPER_DOWN_MTU_RESOURCES	9	
IF_OPER_DOWN_UNSUPPORTED_SPEED	10	
IF_OPER_DOWN_UNSUPPORTED_TRANS_FEC	11	
IF_OPER_DOWN_OTHER	12	
IF_OPER_DOWN_PORT_NOT_PRESENT	13	used internally by chassis mgr only - xdp never publish to IDB!

Name	Number	Description
IF_OPER_DOWN_FABRIC_AVAILABILITY	14	used internally by chassis mgr only - xdp never publish to IDB!
IF_OPER_DOWN_NO_ACTIVE_LINKS	15	lag interface only
IF_OPER_DOWN_MIN_LINK_THRESHOLD	16	lag interface only
IF_OPER_DOWN_9_12_SPEED_MISMATCH	17	Vodka port 9-12 must all be same speed as port 9
IF_OPER_DOWN_LAG_RESOURCES	18	lag interface only
IF_OPER_DOWN_LAG_MEMBER_RESOURCES	19	lag member interface only
IF_OPER_DOWN_STANDBY_SIGNALING	20	ESM multihoming
IF_OPER_DOWN_HOLD_TIME_UP_ACTIVE	21	interface hold-time up is actively holding the interface down
IF_OPER_DOWN_RELOAD_TIME_ACTIVE	22	interface reload time is actively holding the interface down
IF_OPER_DOWN_CONNECTOR_DOWN	23	parent connector oper down forces breakout port oper down
IF_OPER_DOWN_AUTO_NEG_MISMATCH	24	
IF_OPER_DOWN_EVENT_HANDLER	25	used internally by chassis mgr only - xdp never publish to IDB!
IF_OPER_DOWN_UNSUPPORTED_BREAKOUT	26	interface doesn't support breakout config

#### 4.10.5 IfOperStateType

Represents interface operational state.

Table 71: IfOperStateType

Name	Number	Description
IF_OPER_STATE_UP	0	Interface operational state up

Name	Number	Description
IF_OPER_STATE_DOWN	1	Interface operational state down
IF_OPER_STATE_TESTING	2	Interface operational state testing
IF_OPER_STATE_UNKNOWN	3	Interface operational state unknown
IF_OPER_STATE_DORMANT	4	Interface operational state dormant
IF_OPER_STATE_NOT_PRESENT	5	Interface operational state not present
IF_OPER_STATE_LOWER_LAYER_DOWN	6	Interface operational state lower layer down

#### 4.10.6 IfTransceiverFecType

Represents interface transceiver fec. Corresponds to yang values

Table 72: IfTransceiverFecType

Name	Number	Description
IF_TRANS_FEC_UNSET	0	Fec unknown
IF_TRANS_FEC_DISABLED	1	
IF_TRANS_FEC_RS528	2	
IF_TRANS_FEC_RS544	3	
IF_TRANS_FEC_BASER	4	
IF_TRANS_FEC_RS108	5	

#### 4.10.7 IpAddressState

Represents IP address state.

Table 73: IpAddressState

Name	Number	Description
IPADDR_STATE_UNKNOWN	0	IP address state unknown
IPADDR_STATE_TENTATIVE	1	IP address state tentative
IPADDR_STATE_DUPLICATED	2	IP address state duplicated

Name	Number	Description
IPADDR_STATE_INACCESSIBLE	3	IP address state inaccessible
IPADDR_STATE_DEPRECATED	4	IP address state deprecated
IPADDR_STATE_PREFERRED	5	IP address state preferred

#### 4.10.8 SdkMgrOperation

Represents enumeration value for operation in subscription.

Table 74: SdkMgrOperation

Name	Number	Description
Create	0	Create operation
Update	1	Update operation
Delete	2	Delete operation

#### 4.10.9 SdkMgrStatus

Represents status of network programming service calls.

Table 75: SdkMgrStatus

Name	Number	Description
kSdkMgrSuccess	0	Successful service call
kSdkMgrFailed	1	Failed service call

#### 4.10.10 AgentReply

Empty message from agent.

#### 4.10.11 EvpnEthSegIdPb

Table 76: *EvpnEthSegIdPb*

Field	Type	Label	Description
es_id	bytes		Type 0 for now. hard-coded id

#### 4.10.12 GlobalIfId

Represents global interface identifier.

Table 77: *GlobalIfId*

Field	Type	Label	Description
global_if_id	uint32		Global interface identifier

#### 4.10.13 IpAddrPrefLenPb

Represents IP prefix.

Table 78: *IpAddrPrefLenPb*

Field	Type	Label	Description
ip_addr	IpAddressPb		IP address
prefix_length	uint32		IP address prefix length

#### 4.10.14 IpAddressPb

Represents IP address.

Table 79: *IpAddressPb*

Field	Type	Label	Description
addr	bytes		IP address

#### 4.10.15 IpInterfaceAddrPrefixPb

Represents IP prefix state.

Table 80: *IpInterfaceAddrPrefixPb*

Field	Type	Label	Description
prefix	<a href="#">IpAddrPrefLenPb</a>		IP prefix
state	<a href="#">IpAddressState</a>		IP prefix state

#### 4.10.16 MacAddressPb

Represents MAC address.

Table 81: *MacAddressPb*

Field	Type	Label	Description
mac_address	<a href="#">bytes</a>		MAC address

#### 4.10.17 MplsLabel

Represents MPLS label.

Table 82: *MplsLabel*

Field	Type	Label	Description
mpls_label	<a href="#">uint32</a>		MPLS label

#### 4.10.18 NetInstanceld

Represents network instance identifier.

Table 83: *NetInstanceld*

Field	Type	Label	Description
instance_id	<a href="#">uint32</a>		Network instance identifier

#### 4.10.19 PortIdPb

Represents port identifier.



Table 84: PortIdPb

Field	Type	Label	Description
port_id	uint64		Port identifier

#### 4.10.20 SyncRequest

Empty message for synchronization request.

#### 4.10.21 SyncResponse

Empty message for synchronization end.

Table 85: SyncResponse

Field	Type	Label	Description
status	SdkMgrStatus		Error code
error_str	string		Detailed error string

### 4.11 sdk\_service.proto

#### 4.11.1 NotificationRegisterRequest.Operation

Represents notification stream subscription request operation.

Table 86: NotificationRegisterRequest.Operation

Name	Number	Description
Create	0	Create a subscription
Delete	1	Delete all subscriptions
AddSubscription	2	Add subscription to existing subscriptions
DeleteSubscription	3	Delete one subscription from existing subscriptions

#### 4.11.2 AgentRegistrationRequest

Represents registration request message used in agent register and unregister.

Table 87: AgentRegistrationRequest

Field	Type	Label	Description
js_path	string	repeated	Optional, JSON path formatted strings, which are used in telemetry. Format of js_path follows hierarchical YANG. for example: .interface{.name=*}.my_app. .my_app.tunnel{.name==*} "*" needs to be replaced with a specific key.
agent_liveliness	uint32		Kill this agent unless a keepalive is received within this many seconds. Value of 0 means do not monitor this agent for liveliness.

### 4.11.3 AgentRegistrationResponse

Represents registration response in reply to registration request.

Table 88: AgentRegistrationResponse

Field	Type	Label	Description
status	SdkMgrStatus		Status of the register; for example: kOk, kFailed
error_str	string		Detailed error text
app_id	uint32		Application ID assigned by SDK manager.

### 4.11.4 AppIdRequest

Represents application identifier request from agent. All applications are assigned an identifier by IDB.

Table 89: AppIdRequest

Field	Type	Label	Description
name	string		Application name

### 4.11.5 AppIdResponse

Represents application identifier response to agent.

Table 90: *AppldResponse*

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Status of the call; for example, kOk, kFailed
id	<a href="#">uint32</a>		Identifier for the given application name

#### 4.11.6 KeepAliveRequest

Represents keep alive request from agent to refresh liveliness of the agent.

#### 4.11.7 KeepAliveResponse

Represents keepalive response.

Table 91: *KeepAliveResponse*

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Status of keepalive; for example, kOk or kFailed

#### 4.11.8 Notification

Represents notification stream response.

Table 92: *Notification*

Field	Type	Label	Description
sub_id	<a href="#">uint64</a>		Subscription identifier
intf	<a href="#">InterfaceNotification</a>		Interface details
nw_inst	<a href="#">NetworkInstanceNotification</a>		Network instance details
lldp_neighbor	<a href="#">LldpNeighborNotification</a>		LLDP neighbor details
config	<a href="#">ConfigNotification</a>		Configuration notification
bfd_session	<a href="#">BfdSessionNotification</a>		BFD session details
route	<a href="#">IpRouteNotification</a>		IP route details
appid	<a href="#">AppIdentNotification</a>		App identification details

Field	Type	Label	Description
nhg	<a href="#">NextHopGroupNotification</a>		Next-hop group details

#### 4.11.9 NotificationQueryRequest

Represents notification query to return specific subscription details.

Table 93: NotificationQueryRequest

Field	Type	Label	Description
stream_id	<a href="#">uint64</a>		Stream identifier, in Notification RegisterResponse

#### 4.11.10 NotificationQueryResponse

Represents notification query response.

Table 94: NotificationQueryResponse

Field	Type	Label	Description
subscriptions	<a href="#">NotificationQuerySubscription</a>	repeated	List of subscription details
status	<a href="#">SdkMgrStatus</a>		Status of the query

#### 4.11.11 NotificationQuerySubscription

Represents notification subscription.

Table 95: NotificationQuerySubscription

Field	Type	Label	Description
sub_id	<a href="#">uint64</a>		Subscription identifier
description	<a href="#">string</a>		Subscription description

#### 4.11.12 NotificationRegisterRequest

Represents notification request from agent. Agent uses this message to subscribe to router events such as interface create, delete, or update, as well as LLDP neighbor create, delete, or update, and so on.

Table 96: NotificationRegisterRequest

Field	Type	Label	Description
stream_id	uint64		Unset on create, set otherwise
op	Operation		Specific operation in the notification register request
sub_id	uint64		Set for delete subscription, unset otherwise
intf	InterfaceSubscriptionRequest		Interface subscription request
nw_inst	NetworkInstanceSubscriptionRequest		Network instance subscription request
lldp_neighbor	LldpNeighborSubscriptionRequest		LLDP neighbor subscription request
config	ConfigSubscriptionRequest		Configuration subscription request
bfd_session	BfdSessionSubscriptionRequest		BFD session subscription request
route	IpRouteSubscriptionRequest		IP route subscription request
appid	AppIdentSubscriptionRequest		App identification subscription request
nhg	NextHopGroupSubscriptionRequest		Nexthop Group subscription request

#### 4.11.13 NotificationRegisterResponse

Represents notification response.

Table 97: NotificationRegisterResponse

Field	Type	Label	Description
stream_id	uint64		Stream identifier. This needs to be passed to the SDK manager for further notification subscription changes specific to the current subscription
sub_id	uint64		Subscription identifier. Each subscription gets an identifier, which can be used to delete a subscription

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Status of subscription

#### 4.11.14 NotificationStreamRequest

Represents notification stream request.

Table 98: NotificationStreamRequest

Field	Type	Label	Description
stream_id	<a href="#">uint64</a>		Stream identifier

#### 4.11.15 NotificationStreamResponse

Represents notification stream response that contains one or more notification.

Table 99: NotificationStreamResponse

Field	Type	Label	Description
notification	<a href="#">Notification</a>	repeated	Notification details

#### 4.11.16 SdkMgrService

Represents base service that defines agent registration, unregistration, notification subscriptions, and keepalive messages.

Table 100: SdkMgrService

Method Name	Request Type	Response Type	Description
AgentRegister	<a href="#">AgentRegistrationRequest</a>	<a href="#">AgentRegistrationResponse</a>	Register agent
AgentUnRegister	<a href="#">AgentRegistrationRequest</a>	<a href="#">AgentRegistrationResponse</a>	Unregister agent
NotificationRegister	<a href="#">NotificationRegisterRequest</a>	<a href="#">NotificationRegisterResponse</a>	Register for event notifications
NotificationQuery	<a href="#">NotificationQueryRequest</a>	<a href="#">NotificationQueryResponse</a>	Returns current or specific notification subscription details
KeepAlive	<a href="#">KeepAliveRequest</a>	<a href="#">KeepAliveResponse</a>	Send periodic keepalive message

Method Name	Request Type	Response Type	Description
GetAppld	<a href="#">AppldRequest</a>	<a href="#">AppldResponse</a>	Get application name from application identifier

#### 4.11.17 SdkNotificationService

Represents service for handling notifications.

*Table 101: SdkNotificationService*

Method Name	Request Type	Response Type	Description
NotificationStream	<a href="#">NotificationStreamRequest</a>	<a href="#">NotificationStreamResponse</a>	Send stream of event notifications based on the agent subscriptions

## 4.12 telemetry\_service.proto

### 4.12.1 TelemetryData

Represents telemetry data.

*Table 102: TelemetryData*

Field	Type	Label	Description
json_content	<a href="#">string</a>		Structured JSON telemetry data

### 4.12.2 TelemetryDeleteRequest

Represents telemetry delete request.

*Table 103: TelemetryDeleteRequest*

Field	Type	Label	Description
key	<a href="#">TelemetryKey</a>	repeated	Telemetry key

### 4.12.3 TelemetryDeleteResponse

Represents telemetry delete response.

Table 104: TelemetryDeleteResponse

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Status of delete request
error_str	<a href="#">string</a>		Detailed error message

#### 4.12.4 TelemetryInfo

Represents telemetry information.

Table 105: TelemetryInfo

Field	Type	Label	Description
key	<a href="#">TelemetryKey</a>		Telemetry key
data	<a href="#">TelemetryData</a>		Telemetry data

#### 4.12.5 TelemetryKey

Represents telemetry key.

Table 106: TelemetryKey

Field	Type	Label	Description
js_path	<a href="#">string</a>		JSON path referencing the key for telemetry data

#### 4.12.6 TelemetryUpdateRequest

Represents telemetry update request.

Table 107: TelemetryUpdateRequest

Field	Type	Label	Description
state	<a href="#">TelemetryInfo</a>	repeated	State of application

#### 4.12.7 TelemetryUpdateResponse

Represents telemetry update response.



Table 108: TelemetryUpdateResponse

Field	Type	Label	Description
status	<a href="#">SdkMgrStatus</a>		Status of telemetry update request
error_str	<a href="#">string</a>		Detailed error message

## 4.12.8 SdkMgrTelemetryService

Represents service for telemetry service to store state data.

Table 109: SdkMgrTelemetryService

Method Name	Request Type	Response Type	Description
TelemetryAdd OrUpdate	<a href="#">TelemetryUpdateRequest</a>	<a href="#">TelemetryUpdate Response</a>	Add or update telemetry data
TelemetryDelete	<a href="#">TelemetryDeleteRequest</a>	<a href="#">TelemetryDelete Response</a>	Delete telemetry data

## 4.13 Scalar Value Types

### 4.13.1 Scalar

Scalar description

Table 110: Scalar

.proto Type	Notes	C++	C#	Go	Java	PHP	Python	Ruby
double		double	double	float64	double	float	float	Float
float		float	float	float32	float	float	float	Float
int32	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint32 instead.	int32	int	int32	int	integer	int	Bignum or Fixnum (as required)

.proto Type	Notes	C++	C#	Go	Java	PHP	Python	Ruby
int64	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint64 instead.	int64	long	int64	long	integer/string	int/long	Bignum
uint32	Uses variable-length encoding.	uint32	uint	uint32	int	integer	int/long	Bignum or Fixnum (as required)
uint64	Uses variable-length encoding.	uint64	ulong	uint64	long	integer/string	int/long	Bignum or Fixnum (as required)
sint32	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int32s.	int32	int	int32	int	integer	int	Bignum or Fixnum (as required)
sint64	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int64s.	int64	long	int64	long	integer/string	int/long	Bignum
fixed32	Always four bytes. More efficient than uint32 if values are often greater than 2 <sup>28</sup> .	uint32	uint	uint32	int	integer	int	Bignum or Fixnum (as required)
fixed64	Always eight bytes. More efficient than uint64 if values are often greater than 2 <sup>56</sup> .	uint64	ulong	uint64	long	integer/string	int/long	Bignum
sfixed32	Always four bytes.	int32	int	int32	int	integer	int	Bignum or Fixnum (as required)

.proto Type	Notes	C++	C#	Go	Java	PHP	Python	Ruby
sfixed64	Always eight bytes.	int64	long	int64	long	integer/string	int/long	Bignum
bool		bool	bool	bool	boolean	boolean	boolean	TrueClass/False Class
string	A string must always contain UTF-8 encoded or 7-bit ASCII text.	string	string	string	String	string	str/unicode	String (UTF-8)
bytes	May contain any arbitrary sequence of bytes.	string	Byte String	[]byte	Byte String	string	str	String (ASCII-8BIT)

# Customer document and product support



## **Customer documentation**

[Customer documentation welcome page](#)



## **Technical support**

[Product support portal](#)



## **Documentation feedback**

[Customer documentation feedback](#)