



## SpanApp Service

---

This chapter describes the DCNM web services' API methods for the SpanApp service.

### Information About SpanApp Service

Switches use the switch port analyzer (SPAN) feature to copy traffic from switch source ports or VLANs to destination ports. This feature allows you to analyze traffic at destination ports by using monitoring devices such as network analyzers. The API categories are as follows:

- Query/Get APIs—Query data from the persistent database.
- Modify APIs—Modify configurations of SPAN sessions or global settings.
- Delete APIs—Delete SPAN sessions from the device.

The following sessions are supported for query APIs and delete APIs:

- Local SPAN
- RSPAN source
- RSPAN destination

You can modify only local SPAN sessions. An exception is thrown for RSPAN source or destination sessions.

The following associations are available for interfaces:

- Interface name
- Interface status

The following associations are available for VLANs:

- VLAN status

### addDestinationPortsToSpanSession

Adds destination ports/Ethernet channels to local SPAN session. Remote SPAN is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If NetworkInterface which is already used as SPAN source is used as destination

## ***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

- If the NetworkInterface is already configured as SPAN destination for any other SPAN session.
- If the SPAN permit-list is enabled and NetworkInterface is not configured as permit-list destination port.
- If SwitchedEtherChannelNetworkInterface / RoutedEtherChannelNetworkInterface is used as SPAN destination in case of Catalyst 6500 series switches
- If Etherchannel member ports are configured as SPAN destination
- If the NetworkInterface is not SPAN destination capable.
- If any of the objects corresponding to InstanceNameId in dstPortNameIds does not exist

### **Parameters**

opContext—Operational context

sessionNameId—InstanceNameId of LocalSpanSession.

dstPortNameIds—List of InstanceNameId of the interfaces to be added as SPAN destination.

### **Return Value**

void

## **addFilteredVlanRangeToSpanSession**

Adds filtered VLANs to local SPAN session. Remote SPAN is not supported. Can add a VLAN that does not exist in the device (pre-provisioning).

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If adding filtered VLANs to the local SPAN and if the given local SPAN session has source VLANs added.

### **Parameters**

opContext—Operational context

sessionNameId—InstanceNameId of LocalSpanSession.

filteredVlans—range of VLAN IDs to be configured as filtered VLANs. Eg: "2,7-9".

### **Return Value**

void

## **addFilteredVlansToSpanSession**

Adds filtered VLANs to local SPAN session. Remote SPAN is not supported. Can add a VLAN which is not existing in the device as well (preprovisioning).

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist

## ***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

- If adding filtered VLANs to the local SPAN and if the given local SPAN session has source VLANs added.

### **Parameters**

opContext—Operational context

sessionNameId—InstanceNameId of LocalSpanSession.

vlanRange—range of VLAN IDs to be configured as filtered VLANs . Eg:"2,7-9".

### **Return Value**

void

## **addPortsToSpanPermitList**

Adds new ports to SPAN destination permit list.

ValidationException is thrown if any of the following situation occurs:

- If neNameId is null or not InstanceNameId of the AbstractNetworkElement.
- If the specified AbstractNetworkElement does not exist
- If any of the element's InstanceNameId in interfaceNameIds is of type RoutedEtherChannelNetworkInterface or SwitchedEtherChannelNetworkInterface.

### **Parameters**

opContext—Operational context

neNameId—InstanceNameId of the network element.

interfaceNameIds—List of InstanceNameId of NetworkInterface

### **Return Value**

void

## **addSourcesToSpanSession**

Adds source ports/Ethernet channels to a local SPAN session. Remote SPAN is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If NetworkInterface which is already used as destination is used as SPAN source
- If SwitchedEtherChannelNetworkInterface / RoutedEtherChannelNetworkInterface is used as SPAN source in case of Catalyst 6500 series switches
- If Etherchannel member ports are configured as SPAN source
- If NetworkInterface is added as SPAN source if the session already has filtered VLANs configured in case of Catalyst 6500 series switches
- If NetworkInterface is added as SPAN source if the session already has VLANs configured as SPAN source in case of Catalyst 6500 series switches

***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

- If VLAN is added as SPAN source if the session already has interfaces configured as SPAN source in case of Catalyst 6500 series switches
- If any of the objects corresponding to InstanceNameId in srcNameIds does not exist
- If direction is null

**Parameters**

opContext—Operational context

sessionNameId—InstanceNameId of LocalSpanSession.

srcNameIds—List of InstanceNameId of NetworkInterface / VLAN to be configured as sources for the given SPAN session.

direction—Traffic direction SpanSourceDirection.

**Return Value**

void

## addSourcesToSpanSessionByRange

Adds source VLANs to a local SPAN session. Remote SPAN is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If VLAN is added as SPAN source if the session already has filtered VLANs configured in case of Catalyst 6500 series switches
- If VLAN is added as SPAN source if the session already has Interfaces configured as SPAN source in case of Catalyst 6500 series switches
- If any of the objects corresponding to VLAN ID in vlanRange does not exist
- If direction is null

**Parameters**

opContext—Operational context

sessionNameId—InstanceNameId of LocalSpanSession.

vlanRange—range of VLAN IDs to be configured as SPAN source. Eg: "2,7-9".

direction—Traffic direction SpanSourceDirection

**Return Value**

void

## createSpanSessions

Create a SPAN sessions in a network element. This method can be used to create all types of SPAN sessions.

## ***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

While calling this method, one has to set the NetworkElement reference to the session on which the corresponding session has to be created.

On the server, this method will create the new InstanceNameId for the SPAN session based on the session ID and NetworkElement.

For creating a SPAN session, the user must configure atleast one source or destination association to it.

Following associations will be updated:

- SpanSource Local SPAN Source—Source interfaces and VLANs
- VLAN Local SPAN Source—Filtered VLANs
- NetworkInterfaceLocal SPAN Destination—Destination interfaces
- SPAN status

ValidationException is thrown if any of the following situation occurs:

- RSPAN source/destination sessions are to be created
- If spanSessionList is null or empty or elements in the list are not of type LocalSpanSession.
- If the LocalSpanSession object in the spanSessionList does not have valid session ID or null AbstractNetworkElement reference.
- If the number of SPAN sessions limit in the specified AbstractNetworkElement exceeded
- If the SPAN session to be created does not have source or destination associations
- In Catalyst 6500 series switches, if the SPAN session to be created have NetworkInterface and VLAN as SPAN sources
- If the SPAN session to be created have NetworkInterface as SPAN destination interface which is already a SPAN destination some other session

PropertiesException is thrown if any of the following situation occurs:

- In the spanSessionList collection, if any one of the SpanSession attribute is not valid.

Example:

- sessionId of a SpanSession is out of range.

IntegrityException is thrown if any of the following situation occurs:

- In the spanSessionList collection, if any one of the SpanSession object is already existing in the specified AbstractNetworkElement.

### **Parameters**

opContext—Operational context

spanSessionList—List of (@link SpanSession)Objects.

### **Return Value**

List of InstanceNameId for the newly created SPAN sessions.

## **createSpanSessionsInNetworkElement**

Create a list of SPAN sessions in a network element. This method can be used to create any type of SPAN session. InstanceNameId will be created on the server based on the session ID and the NetworkElement InstanceNameId.

## ***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

Following associations will be updated:

- SpanSource Local SPAN Source—Source interfaces and VLANs
- VLAN Local SPAN Source—Filtered VLANs
- NetworkInterfaceLocal SPAN Destination—Destination interfaces
- SPAN status

ValidationException is thrown if any of the following situation occurs:

- If sessionInstanceNameIds is not a valid InstanceNameId of SpanSession object.
- If neNameId is null or it is not of InstanceNameId of AbstractNetworkElement.

ValidationException is thrown if any of the following situation occurs:

- RSPAN source/destination sessions are to be created
- If spanSessionList is null or empty or elements in the list are not of type LocalSpanSession.
- If the specified AbstractNetworkElement does not exist
- If the LocalSpanSession object in the spanSessionList does not have valid session ID or null AbstractNetworkElement reference.
- If the number of SPAN sessions limit in the specified AbstractNetworkElement exceeded
- If the SPAN session to be created does not have source or destination associations
- In Catalyst 6500 series switches, If the SPAN session to be created have NetworkInterface and VLAN as SPAN sources
- If the SPAN session to be created have NetworkInterface as SPAN destination interface which is already a SPAN destination some other session

PropertiesException is thrown if any of the following situation occurs:

- In the spanSessionList collection, if any one of the SpanSession attribute is not valid.

Example:

- sessionId of a SpanSession is out of range.

IntegrityException is thrown if any of the following situation occurs:

- In the spanSessionList collection, if any one of the SpanSession object is already existing in the specified AbstractNetworkElement.

### **Parameters**

opContext—Operational context

neNameId—InstanceNameId of the network element.

spanSessionList—List of SpanSession that need to be created.

### **Return Value**

Ordered list of newly created SPAN sessions instance name IDs

## **deleteSpanSessions**

Deletes the SPAN session corresponding to the given list of InstanceNameId. The corresponding SpanSession and its associations will be deleted.

## ***Send document comments to nexus7k-docfeedback@cisco.com***

The following sessions are allowed to be deleted:

- Local SPAN
- RSPAN Source
- RSPAN Destination

ValidationException is thrown if any of the following situation occurs:

- If sessionNameIds is null or empty or elements in the list are not InstanceNameId of SpanSession.

In the sessionNameIds collection, if any one of the SpanSession object corresponding to InstanceNameId in the sessionNameIds does not exist.

### **Parameters**

opContext—Operational context

sessionNameIds—List of SPAN session InstanceNameId that need to be deleted.

### **Return Value**

void

## **disableInterfacesMonitor**

Disables Interfaces Monitor for one or more Interfaces. Given the instance name ID of interface ,InterfacesMonitor will be disabled on those elements. This API is to disable "InterfacesMonitor" for Cisco NX-OS devices and it is not applicable for other Cisco IOS devices.

ValidationException is thrown if any of the following situation occurs:

- If the interfaceInstanceNameId is is of type NetworkInterface null.
- If the interfaceInstanceNameId contains one or more null element, or the collection is empty or it is not type InstanceNameId.
- If the given interface is running in other Cisco IOS

### **Parameters**

opContext—Operational context

interfaceInstanceNameId—List of InstanceNameId of NetworkInterface for which the information is required

### **Return Value**

void

## **enableInterfacesMonitor**

Enables Interface Monitor for the given Interfaces. Given the instance name ID of one or more interfaces, "InterfacesMonitor" will be enabled on those elements. This API is to enable InterfacesMonitor for Cisco NX-OS devices and it is not applicable for other Cisco IOS devices

ValidationException is thrown if any of the following situation occurs:

- If the interfaceInstanceNameId is null.

## ***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

- If the interfaceInstanceNameId is of type NetworkInterface contains one or more null element, or the collection is empty or it is not type InstanceNameId.
- If the given interface is running in other Cisco IOS

### **Parameters**

opContext—Operational context

interfaceInstanceNameId—List of InstanceNameId of NetworkInterface for which the information is required

### **Return Value**

void

## **getLocalSpanSessionsInNetwork**

Returns local SPAN sessions corresponding to a list of session IDs for a list of network elements.

Following associations will be available:

- Source interfaces and VLANs
- Filtered VLANs
- Destination interfaces
- SPAN status

ValidationException is thrown if any of the following situation occurs:

- If elements in the neNameIdList are not of InstanceNameId type of AbstractNetworkElement.

### **Parameters**

opContext—Operational context

sessionIds—range of session IDs (e.g. 1-2,20,50-53). If null, returns all sessions.

neNameIdList—List of (@link InstanceNameId) of network elements whose Session ID are to be queried. If null or empty, will return the sessions from all the managed network elements.

### **Return Value**

Ordered list of Local SPAN sessions

## **getSpanDestinationPorts**

Returns the destination ports for the given session NameId.

Following associations will be available for resultant Interfaces:

15.Interface Name

16.Interface Status

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the SpanSession.
- If the specified SpanSession does not exist



***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

**Parameters**

opContext—Operational context

sessionNameId—InstanceNameId of the SPAN session.

**Return Value**

List of NetworkInterface objects.

## getSpanPermitListSettingsInNetwork

Returns SPAN destination permit list for a list of network elements.

Following associations will be available:

- NetworkInterface Destination interfaces

ValidationException is thrown if any of the following situation occurs:

- If elements in the neNameIdList are not of InstanceNameId type of AbstractNetworkElement.

**Parameters**

opContext—

neNameIdList—List of InstanceNameId of network elements whose SpanPermitListSetting are to be queried. If null or empty, will return the sessions from all the managed network elements.

**Return Value**

List of SpanPermitListSetting with its associations initialized.

## getSpanSessions

Returns list of SPAN sessions with all its associations for a given list of instance name IDs of the SPAN sessions.

The return type can have the following session types:

- Local SPAN
- RSPAN Source
- RSPAN Destination

Following associations will be available:

- SpanSource Local SPAN/RSPAN Source—Source interfaces and VLANs
- VLAN Local SPAN/RSPAN Source—Filtered VLANs
- NetworkInterface Local SPAN/RSPAN Destination—Destination interfaces
- VlanRSPAN Source/RSPAN Destination VLAN—RSPAN VLAN
- SPAN status

ValidationException is thrown if any of the following situation occurs:

- If sessionInstanceNameIds is null or empty or it is not of type InstanceNameId.
- If sessionInstanceNameIds is not a valid InstanceNameId of SpanSession object.

***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

#### Parameters

opContext—Operational context

sessionInstanceNameIds—List of (@link InstanceNameId) of SPAN sessions.

#### Return Value

An ordered List of SpanSession objects. The returned objects will have all the associations initialized.

## getSpanSessionsAssociatedToVlan

Returns SPAN session objects for which given VLAN is configured as SPAN source.

Following associations will be available for each SPAN session:

25. Local SPAN to VLAN association

Following associations will be available on other end of LocalSpanSessionAppliesToVlan:

26. Reference to VLAN

ValidationException is thrown if any of the following situation occurs:

- If vlanNameId is null or not InstanceNameId of the VLAN.
- If the specified VLAN does not exist

#### Parameters

opContext—Operational context

vlanNameId—InstanceNameId of the VLAN.

#### Return Value

List of SpanSession objects for which given VLAN is configured as SPAN source.

## getSpanSessionsInNetwork

Returns SPAN sessions corresponding to a list of session IDs for a list of network elements.

The return type can have the following session types:

- Local SPAN
- RSPAN Source
- RSPAN Destination

Following associations will be available:

- SpanSource Local SPAN/RSPAN Source—Source interfaces and VLANs
- VLAN Local SPAN/RSPAN Source—Filtered VLANs
- NetworkInterface Local SPAN/RSPAN Destination—Destination interfaces
- VLAN RSPAN Source/RSPAN Destination VLAN—RSPAN VLAN
- SPAN status

ValidationException is thrown if any of the following situation occurs:

- If elements in the neNameIdList are not of InstanceNameId type of AbstractNetworkElement.

## ***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

### **Parameters**

opContext—Operational context

sessionIds—range of session IDs (e.g. 1-2,20,50-53). If null, returns all sessions.

neNameIdList—List of (@link InstanceNameId) of network elements whose Session ID are to be queried. If null or empty, will return the sessions from all the managed network elements.

### **Return Value**

An ordered List of SpanSession objects. The returned objects will have all the associations initialized.

## **getSpanSources**

Returns all source ports/etherchannels/VLANs associated to a local SPAN session or remote SPAN source session

The return type will have the list of following objects for Local SPAN session:

19.Interface associations

20.VLAN associations

Following associations will be available for LocalSpanSessionAppliesToNetworkInterface:

21.Reference to interface

Following associations will be available for LocalSpanSessionAppliesToVlan:

22.Reference to VLAN

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the SpanSession.
- If the specified SpanSession does not exist

### **Parameters**

opContext—Operational context

sessionNameId—InstanceNameId of a SPAN session

### **Return Value**

List of sources for a given SPAN session

## **modifyFilteredVlanRangeForSpanSession**

Modifies the filtered VLANs associated to local SPAN session . The existing filtered VLANs will be replaced with the given VLANs.Remote SPAN is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If any of the objects corresponding to InstanceNameId in vlanInstanceNameIds does not exist
- If adding filtered VLANs to the local SPAN and if the given local SPAN session has source VLANs added.

***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

#### Parameters

opContext—Operational context

sessionNameId—InstanceNameId of LocalSpanSession.

filteredVlans—range of VLAN IDs to be configured as filtered VLANs . Eg:"2,7-9".

#### Return Value

void

## modifyFilteredVlansByRangeForSpanSession

Modifies the filtered VLANs associated to local SPAN session . The existing filtered VLANs will be replaced with the given VLANs.Remote SPAN is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If adding filtered VLANs to the local SPAN and if the given local SPAN session has source VLANs added.

#### Parameters

opContext—Operational context

sessionNameId—InstanceNameId of LocalSpanSession.

vlanRange—range of VLAN IDs to be configured as filtered VLANs . Eg:"2,7-9".

#### Return Value

void

## modifyFilteredVlansForSpanSession

Modifies the filtered VLANs associated to local SPAN session . The existing filtered VLANs will be replaced with the given VLANs.Remote SPAN is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If any of the objects corresponding to InstanceNameId in vlanInstanceNameIds does not exist
- If adding filtered VLANs to the local SPAN and if the given local SPAN session has source VLANs added.

#### Parameters

opContext—Operational context

sessionNameId—InstanceNameId of LocalSpanSession.

vlanInstanceNameIds—List of InstanceNameId of VLAN to be replaced as filtered VLANs to the given SPAN session.

***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

**Return Value**

void

## modifySpanPermitListSettingsInNetworkElement

Modify the permit list for a given network element. This method include modification to the destination ports that are associated to SpanPermitListSetting.

Following associations will be updated:

- NetworkInterface destination interfaces associated to permit-list

ValidationException is thrown if any of the following situation occurs:

- If neNameId is null or not InstanceNameId of the AbstractNetworkElement.
- If the specified AbstractNetworkElement does not exist
- If modifiedPermitListSetting is null.
- If permitlist is Enabled without associating a port as permit-list destination port.
- If permitlist is Enabled without associating a port as permit-list destination port.
- If the NetworkInterface configured is of type RoutedEtherChannelNetworkInterface or SwitchedEtherChannelNetworkInterface.

**Parameters**

opContext—Operational context

neNameId—InstanceNameId of the AbstractNetworkElement.

modifiedPermitListSetting—Modified SpanPermitListSetting object.

**Return Value**

void

## modifySpanSessions

Modify existing SPAN sessions. This method will not modify the associations corresponding to the SPAN session.

ValidationException is thrown if any of the following situation occurs:

- If modifiedSpanSessions is null or empty or elements in the list are not of type SpanSession.

PropertiesException is thrown if any of the following situation occurs:

- 
- If any of the attributes for SpanSession object modifiedSpanSessions is not valid

Example:

- description of a SpanSession is only applicable to Cisco NX-OS platform.
- sessionId of a SpanSession is out of range.

IntegrityException is thrown if any of the following situation occurs:

- If anyone of the object in modifiedSpanSessions collection SpanSession object does not exist.

***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

#### Parameters

opContext—Operational context

modifiedSpanSessions—List of SpanSession objects to be modified.

#### Return Value

void

## modifySpanSourceTrafficDirection

Modifies the direction of the traffic flowing from a span source port/EtherChannel or VLAN. Remote SPAN is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If srcNameId is null or not
- If the specified source VLAN / NetworkInterface does not exist

#### Parameters

opContext—Operational context

sessionId—InstanceNameId of LocalSpanSession.

srcNameId—InstanceNameId of port/EtherChannel or VLAN.

modifiedDirection—modified SpanSourceDirection for the respective span source.

#### Return Value

void

## modifySpanSources

Modifies list of SpanSource associations. This will enable users to modify the traffic direction for a list of existing VLAN/Interface to SPAN associations.

ValidationException is thrown if any of the following situation occurs:

- If spanSources is null or empty or it is not of type SpanSource.
- If any of the objects corresponding to SpanSource in spanSources does not exist
- If any of the objects in spanSources are not of LocalSpanSessionAppliesToNetworkInterface or LocalSpanSessionAppliesToVlan

#### Parameters

opContext—Operational context

spanSources—List of modified SpanSource associations

#### Return Value

void

*Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)*

## removeDestinationPortsFromSpanSession

Removes destination ports/etherchannels from local SPAN session or remote SPAN destination session

ValidationException is thrown if any of the following situation occurs:

- If sessionId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If NetworkInterface is removed completely if the session doesn't have anyother associations ( filtered VLAN / source )
- If any of the objects corresponding to InstanceNameId in dstPortNameIds does not exist

### Parameters

opContext—Operational context

dstPortNameIds—List of InstanceNameId of the interfaces to be removed from the given SPAN session.

### Return Value

void

## removeFilteredVlanRangeInSpanSession

Removes filtered VLANs from local SPAN session.Remote SPAN is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If VLAN is removed completely if the session doesn't have anyother associations (source or destination)
- If any of the objects corresponding to InstanceNameId in dstPortNameIds does not exist

### Parameters

opContext—Operational context

sessionId—InstanceNameId of a LocalSpanSession.

filteredVlans—range of VLAN IDs to be removed. Eg:"2,7-9".

### Return Value

void

## removeFilteredVlansInSpanSession

Removes filtered VLANs from local SPAN session.Remote SPAN is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist

## ***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

- If VLAN is removed completely if the session doesn't have anyother associations (source or destination)
- If any of the objects corresponding to InstanceNameId in dstPortNameIds does not exist

### **Parameters**

opContext—Operational context

sessionNameId—InstanceNameId of a LocalSpanSession.

vlanRange—range of VLAN IDs to be removed. Eg:"2,7-9".

### **Return Value**

void

## **removePortsFromSpanPermitList**

Removes ports from SPAN destination permit list.

ValidationException is thrown if any of the following situation occurs:

- If neNameId is null or not InstanceNameId of the AbstractNetworkElement.
- If the specified AbstractNetworkElement does not exist
- If any of the element's InstanceNameId in interfaceNameIds is associated as SPAN destination to a SPAN session.

Validations:

14. Will not allow to remove the interface if is used as SPAN destination for any of the sessions in the corresponding device.

### **Parameters**

opContext—Operational context

neNameId—Instance name ID of the network element.

interfaceNameIds—List of network interface instance name IDs

### **Return Value**

void

## **removeSourcesFromSpanSession**

Removes source ports/etherchannels or VLANs from a local SPAN session. Remote SPAN is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If NetworkInterface or VLAN is removed completely if the session doesn't have anyother associations ( filtered VLAN / Destination port)
- If any of the objects corresponding to InstanceNameId in srcNameIds does not exist



***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***

**Parameters**

opContext—Operational context

sessionNameId—InstanceNameId of LocalSpanSession.

srcNameIds—List of InstanceNameId of VLAN / NetworkInterface.

**Return Value**

void

## removeSourcesInSpanSession

Remove source ports/etherchannels or VLANs from a local SPAN session. Remote SPAN session is not supported.

ValidationException is thrown if any of the following situation occurs:

- If sessionNameId is null or not InstanceNameId of the LocalSpanSession.
- If the specified LocalSpanSession does not exist
- If VLAN is removed completely if the session doesn't have anyother associations ( filtered VLAN / Destination port)

**Parameters**

opContext—Operational context

sessionNameId—InstanceNameId of LocalSpanSession.

vlanRange—range of VLAN IDs to be configured as SPAN source. Eg:"2,7-9".

**Return Value**

void

***Send document comments to [nexus7k-docfeedback@cisco.com](mailto:nexus7k-docfeedback@cisco.com)***