NEC

PF6800 Ver. 6.0
WebAPI User's Guide

PFC00EN0600-01

Copyrights

Trademarks

Introduction

Thank you for purchasing PF6800 (referred to as PFC). The PF6800 is a path control device used for centralized management of networks, and conforms to OpenFlow 1.0/1.3.

Unlike in conventional switch products, packet transfer and path control functions are separated, thus enabling greater flexibility in the network configuration.

To take full advantage of the functions of this product, please read this manual carefully and become fully familiar with the handling of this device.

Symbols

In this manual, the following three types of symbols are used. These symbols and their meanings are important for proper handling of the PFC.

Important

Indicates items for which special care should be taken to follow regarding handling of equipment and software operation.

Remember

Points that should be checked when operating devices or software.

Tip

Helpful, good-to-know information

Structure of this Manual

This manual has four chapters.

Disclaimer

Unless explicitly set forth in a license agreement, NEC Corporation makes no explicit or implicit guarantees regarding this product and the related documentation, including its commercial use or fitness for a particular purpose, and disclaims all liability pertaining to its handling, use, or attendant trade practices.

Acknowledgment

We would like to express our thanks to Mr. Linus Torvalds and all the people involved in Linux development.

Contents

Chapter 1. WebAPI

Following describes an overview of WebAPI and common information.

1.1 Overview of WebAPI

Following describes an overview of the WebAPI.

The WebAPI provides an interface to access the network information managed by the PFC via HTTPS or HTTP. The WebAPI enables you to perform operations equivalent to those available with the PFC CLI.

Remember

Before using the WebAPI, refer to the PFC CLI-related manuals such as the *Command Reference* and *Configuration Guide*.

1.1.1 Operations That Can Be Performed Using the WebAPI

There are the following WebAPI operations.

Table 1-1 Operations That Can Be Performed Using the WebAPI

Operation Category	Operation
base	Operate startup-configuration
	Operate PFC process alarm
	Operate API version
	Operate physical network flow
	Operate physical network topology
	Operate OFS
	Operate VTN station information
	Operate trunk port
L2	Operate VTN
	Operate VTN flow
	Operate OFS map information (VTN mode)
	Operate vBridge
	Operate vBridge IP address
	Operate interface (vBridge mode)
	Operate MAC entry
	Operate VLAN map
	Operate OFS map information (vExternal mode)
	Operate vLink
	Operate L2 domain configuration information
L2/L3	Operate VTN topology

Operation Category	Operation
	Operate vExternal
	Operate interface (vExternal mode)
L3	Operate vRouter
	Operate DHCP relay
	Operate Interface(vRouter mode)
	Operate IP Routing information
Policy management	Operate flowlist
	Operate flowlist entry
	Operate flow filter (vBridge interface)
	Operate flow filter entry (vBridge interface)
	Operate flow filter (vExternal interface)
	Operate flow filter entry (vExternal interface)
	Operate Flow filter (vRouter interface)
	Operate flow filter entry (vRouter interface)
	Operate network monitor group
	Operate monitoring host
	Operate global path map
	Operate path policy
	Operate linkweight
	Operate VTN path map
	Operate IP routing information using the net monitoring function
	Operate flow filter (VTN)
	Operate flow filter entry (VTN)
	Operate flow filter (vBridge)
	Operate flow filter entry (vBridge)
	Operate safe flow filter (vExternal)
	Operate safe flow filter entry (vExternal)
QoS	Operate policing profile
	Operate Policing profile entry
	Operate policing setting information for flooding packets
	Operate policing setting information (vtn mode)
	Operate policing setting information (vbridge mode)
	Operate policing setting information (vexternal mode)
	Operate OFS QoS flow entry information

1.1.2 Connection Mode

The WebAPI is connected as shown in the figure below.

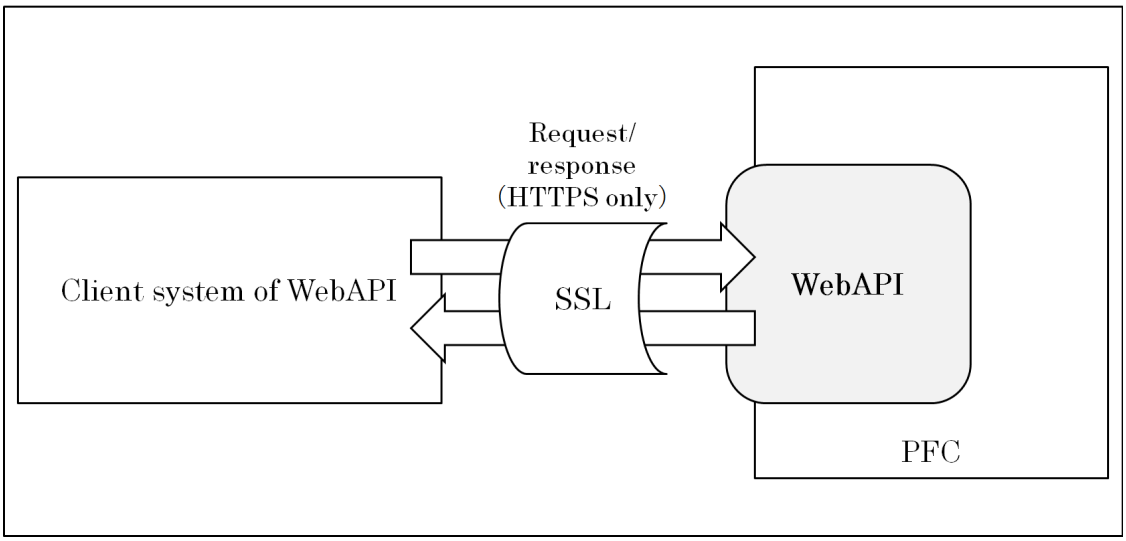


Figure 1-1 Connection to the WebAPI

Each function in the above figure is described below.

Table 1-2 Description of Each Function

Function name	Description
Client system of the WebAPI	This indicates a system that accesses the WebAPI via HTTPS, references the setup information and sets the configurations for the PFC.
WebAPI	A suite of APIs that process accepted requests from the client system of the WebAPI for the PFC and return results as responses.

The function to access WebAPI ("Client system of the WebAPI" in the above figure) issues a request to the target URI (refer to "Chapter 3. WebAPI (base-related) Reference (page 31)") and sends the POST, GET, PUT, DELETE or other method and the parameters necessary for each of them. The WebAPI retrieves the result and details for the request from the network information managed by the PFC and then returns the information to the system outside the PFC as a response.

The protocols and data formats that can be used in the WebAPI are listed in the table below.

Table 1-3 Available Protocols and Data Format

Protocol	Data format of request/response
HTTPS	XML format or JSON format

Tip

The data format that can be used in WebAPI is determined based on specifications made when the request is issued.

1.1.3 WebAPI Version

The WebAPI user can acquire WebAPI version information with the following steps.

Issue a request to the following URI with the GET method. The request body is not necessary.

`https://`<IP address of WebAPI>`:8080/pfc_webapi_version.xml` (XML format)

`https://` <IP address of WebAPI>`:8080/pfc_webapi_version.json`(JSON format)

The response body of the response returns the WebAPI version information in the XML or JSON format. The format of data to be returned is determined by the URI when the request is issued.

Version correspondence between PF6800 and WebAPI are as listed in the table below.

PF6800 version	WebAPI version
V3.0	V1.0
V4.0	V1.1
V5.0	V1.2
V5.1	V1.3

1.1.4 Notes and Restrictions

1. Manually match the settings on the active and standby nodes because the WebAPI settings are not synchronized between the cluster nodes. For how to specify the settings, refer to "2.1.2 Specifying the WebAPI Startup Parameters (page 15)" and "2.1.3 Specifying the SSL Settings (page 15)".

   * Specify the Firewall (iptables settings on the active and standby nodes. For how to specify the settings, refer to "2.1.1 Setting up the Use of Port for Access to WebAPI (page 11)".

2. Some query character strings and parameter values specified in request body can contain underscores (such as VTN name, vBridge name, and vRouter name). However, these values must not start with an underbar.

3. In each API, there are the following notes in the expression of the request body and the response body in WebAPI.

   - Both the XML and JSON formats are indented using line feeds and spaces to represent data structure. Actual data does not include line feeds and spaces for indentation.
   - In both the XML and JSON formats, "[" and "]" are used to indicate that multiple data elements of the same type appear. Actual XML-format data does not contain these brackets. That is, data elements of the same type just consecutively appear. In contrast, JSON-format data does contain these brackets, between which data elements of the same type consecutively appear.
   - In both the XML and JSON formats, be sure to specify the data elements of the request body in the order that is defined in each API reference.

4. A percent-encoded character "question mark (?)" cannot be specified for URI of each API.

5. When making a change to the configuration of the PFC by using the WebAPI, the operation will fail if another user sets configuration mode on the PFC shell or through another session. The following message will be returned in the response body.

   ```
   "Error:Acquire config mode failed."
   ```

   Also, when the value as config_mode_retry_count in sys.properties is greater than 0 and config-mode acquisition is retry-out, the following message will be returned in the response body.

   ```
   "Error:Acquire config mode timeout."
   ```

6. When using both Accept and Content-Type in HTTP header, specify same value.

7. When using the WebGUI, use the default WebAPI user name and password.

   For settings, refer to "1.2.3 User Authentication (page 7)"

8. Check the version of the WebAPI following the steps described in "1.1.3 WebAPI Version (page 3)" before accessing the WebAPI.

9. Data returned from some URIs may increase in future. To ensure compatibility, it is recommended to implement client applications that use the WebAPI in a way that they ignore unknown elements if such elements are returned.
10. This version does not support the API for the vRouter IPv6-related functions.
11. There is a problem with HTTP Keep-Alive function in the reception of the JSON form by the influence by which chunked-transfer-coding function of Web application server became unavailable. Therefore it's different in the way to judge reception complete of response body of the JSON form as follows.

    (V3.0)

    "Transfer-Encoding: chunked" is given to the HTTP header which is at the time of data reception, and chunked transmission is performed. If the chunk data size detects description of "0" in response body, it can be judged as reception complete.

    (V4.0 or later)

    A HTTP session (connection) is closed after the whole data reception. It can be judged as reception complete with detection of the closing.

    Further, I don't change one in case of the XML form with the past and am included in the HTTP header which is at the time of data reception, if I refer to "Content-Length", and data of the number of part-time work concerned is received, it can be judged as reception complete.
12. All URI need suffix (.xml or .json).
13. The API to "show ofs-maps on VTN" returns the element "logical_if_name," which is not included in the WebAPI User's Guide. Do not refer to this element because it is for components of the PFC.
14. At the time of an error reply of WebAPI, the log which is not described in manual may be output to webapi.log. As the factor, there is "the value outside the scope of specification is specified as a request parameter" or "combination of request parameters is incorrect (such as combination that cannot be specified at the same time)". If such a log was output, confirm whether it is not applicable to these factors.
15. It isn't possible to change the configuration of virtual network on the PFC server which is managed by UNC (Unified Network Coordinator). When trying to change the configuration of virtual network, the following error message (status cord :500) is returned in response body.

    ```
    "Error: Due to control of another coordinator, cannot be executed."
    ```

1.2 WebAPI Common Information

1.2.1 Versions of SSL and TLS During HTTPS Communication

The following versions of SSL and TLS can be used for HTTPS communication to the WebAPI.

- SSL 3.0
- TLS 1.0

1.2.2 HTTP Header at Request

Several HTTP headers are required to send an HTTP request to the WebAPI. Necessary HTTP headers differ depending on the method as shown below.

Table 1-4 HTTP Header at Request

HTTP headers	Method				Description
	POST	PUT	GET	DELETE	
Authorization	Y	Y	Y	Y	Specify this when setting user authentication information in the HTTP header. The maximum assignable size is 256 bytes, and one-byte alphanumeric characters and `~!@#$%^&*()-=_+[]\{}|;"'",./<>? can be used. For details on the setup method, see "1.2.3.2 Specifying the user authentication information in the HTTP header (Authorization) (page 7)".
Accept	*2	*2	*2	*2	Specify the data format for a request or response. JSON format: application/json XML format: application/xml
Content-Length	Y	Y	N	N	Specify the send data size using half-width digits. Up to 1024 can be specified in half-width digits.
Content-Type	Y	Y	N	N	Specify the data format for a request or response. JSON format: application/json XML format: application/xml
Host	Y	Y	Y	Y	Specify the host name of the PFC. The maximum assignable size is 256 bytes, and one-byte alphanumeric characters and ~!@#$&*()-_+=[];': can be used. Example: RESTClient: 192.168.231.231:8080 CURL: 192.168.231.231:8080
User-Agent	Y	Y	Y	Y	Specify the applications, such as browsers, you are using, as well as their versions. The maximum assignable size is 256 bytes, and one-byte alphanumeric characters and `~!@#$%^&*()-=_+[]\{}|;':",./<>? can be used. Example: RESTClient: Apache-HttpClient/4.0.1 (java 1.5) CURL: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0 zlib/1.2.3 libidn/1.18 libssh2/1.4.2

*1 "Y" indicates that the relevant header is required. "N" indicates that the header is not required.

*2 This header is not required but can be specified in the format accepted by the client.

Remember

When specifying both Content-Type and Accept, it must be set to same value.

Remember

When specifying multiple HTTP headers, "\r\n (a set of 0x0d and 0x0a in ascii code)" is required as each header's separator.

1.2.3 User Authentication

1.2.3.1 Selection method of user authentication method

The WebAPI performs use authentication using the user name and password for each requests. The authentication method can be selected with one of the following three methods.

[Method-1]

Use the default WebAPI user name and password.

[Method-2]

Use the user name and password on OS (/etc/passwd).

[Method-3]

Use the user name and password registered in the WebAPI configuration file.

Important

When using only the Web GUI, select method-1.

When using only the Web client application except Web GUI, select method-2 or method-3.

When using both the Web GUI and the other Web client application, select method-3.

For settings to selected authentication method, Refer to "2.2.2.6 pwd.properties (page 26)" .

1.2.3.2 Specifying the user authentication information in the HTTP header (Authorization)

The Web client must specify authentication information specified according to "1.2.3.1 Selection method of user authentication method (page 7)" in the HTTP header (Authorization) to send it in order to authenticate each request for the WebAPI. To specify authentication information in the HTTP header (Authorization), perform the following procedure.

1. Generate a character string by concatenating the user name and password specified in"1.2.3.1 Selection method of user authentication method (page 7)" with a one-byte colon (:).
2. Encode the character string at (1) with BASE64.
3. Input the data encoded at (2) in "<BASE64 encode data>" in the following syntax.

```
Authorization:_Basic_<BASE64 encoded character string>
```

Remember

A one-byte space (0x20 in ascii code) is required in the underlined portion above.

1.2.4 Common Message

The following messages are common to all commands and may be displayed when a command is executed. The displayed messages are as follows.

Table 1-5 Common Message

Message	Meaning
*** Error: Acquire config mode failed.	Config mode is occupied.
*** Error: Host is down.	Failed to execute invoke after the pfcapi_config_commit_check (Host: down). Failed to execute invoke after the pfcapi_config_commit_execute (Host: down).
*** Error: Internal error.	Internal processing failed.
*** Error: No Entry.	There is insufficient setup information to execute the command.
*** Error: PFC daemon is not started.	PFC has not started.
*** Error: Resource busy.	Failed to execute invoke after the pfcapi_config_commit_check (Busy state). Failed to execute invoke after the pfcapi_config_commit_execute (Busy state).
*** Error: Invalid argument.	The argument is invalid.
*** Error: Out of memory.	Failed to allocate required memory.
*** Error: Response timed out.	Timed out waiting for a response to a command.
*** Error:Operation Canceled.	Cancel the IPC service.
*** Error: Acquire config mode timeout.	config mode is busy.
*** Error: Due to control of another coordinator, cannot be executed.	It's connected with UNC, you can't save and carry out setting API of Virtual Network.

1.2.5 Error Codes

The error codes common to the WebAPI are listed in the table below. For each WebAPI-specific error code, refer to the description of WebAPI.

Table 1-6 Error Codes Common to the WebAPI

Error Code	Meaning	Generation condition
400 Bad Request	The syntax of request may be incorrect.	This error code is generated if the request does not exactly follow the syntax described in this reference guide.
401 Unauthorized	User authentication failed.	This error code is generated if there is no user name and password in the HTTP header of the request for WebAPI, or the user name or password is wrong.
404 Not Found	Invalid URI. The function for the requested URI does not exist.	This error code is generated when a nonexistent WebAPI URI is accessed. Example: https://<IP Address>:8080/abcd
405 Method Not Allowed	Although the URI is correct, the corresponding HTTP method (PUT/POST/GET/DELETE) contains an error.	This error code is generated if the HTTP method used to access the WebAPI contains an error. Example: When accessing https://<IP Address>:8080/vtns with the DELETE method
415 Unsupported Media Type	The Content-Type which is specified for requests is incorrect.	This error code is generated when Content-type is not "application/xml" for XML format data, or not "application/json" for JSON format data.

Error Code	Meaning	Generation condition
422 Unprocessable Entity	A parameter for the request is not appropriate.	This error code is generated if the value specified in the request already exists as resource.
500 Internal Server Error	A Web server error or an internal error during the WebAPI processing has occurred.	-

Remember

When nonexistent information is deleted, an error may not be reported and the system may determine that the process was normally terminated.

1.2.6 Operations Common to the WebAPI

- If a parameter and/or tag not defined in WebAPI are specified, they are ignored.
- XML or JSON is the valid format for request body and response body in the WebAPI. Which of them is valid is determined by applying the following rules one by one.
  1. If the accessed URI has an extension (.xml or .json), the valid format is determined by that extension.
  2. If a value (application/xml or application/json) is specified for Accept in the HTTP header, the valid format is determined by that value. In addition, when adding an extension to URI, Content-Type is not necessary.
  3. If a value (application/xml or application/json) is specified for Content-Type in the HTTP header, the valid format is determined by that value. In addition, when adding an extension to URI, Content-Type is not necessary.
  4. If no URI extension, Accept, and Content-Type in the HTTP header are specified, the XML format is valid.
  5. There is no difference in performance regardless of whether json or xml is specified.
- The latest settings are applied to running-configuration every time the API that registers or updates settings is called.
- If a request body element is omitted and the API automatically creates vBridge, vExternal, interface, or other items, these items are named based on sequential numbers. If the maximum number is reached, automatic creation fails and an HTTP error (error code 500) is returned.

1.2.7 Applying the Settings by WebAPI to startup-configuration

Settings which have been registered or updated by WebAPI are not automatically applied to startup-configuration by default.

To automatically apply the information set up by API to startup-configuration, execute the configuration commit auto-save command in the PFC shell cli mode before using WebAPI. Then, the latest settings are applied not only to running-configuration but also to startup-configuration every time the API that registers or updates settings is called.

For details about the function of configuration commit auto-save, refer to the *Command Reference*.

1.2.8 WebAPI log

The WebAPI log is output to /var/opt/nec/pfc/Agent/log/webapi. The following log files are output.

Table 1-7 WebAPI log

File Name	Description
webapi_daemon.log	Logs the daemon process that starts the WebAPI.
webapi_access.log{.N} *N is a number equal to or lager than 1.	Records the APIs (URI paths) that the Web client accesses. When reaching a certain file size, this file is saved as a past file under the file name which a number is added to. The file size, the number of saved files, and the log level output to the file can be changed. For details, refer to "2.2.2.10 logaccess.properties (page 29)".
webap.log{.N} *N is a number equal to or larger than 1.	Records WebAPI operations. When reaching a certain file size, this file is saved as a past file under the file name which a number is added to. The file size, the number of saved files, and the log level output to the file can be changed. For details, refer to "2.2.2.9 log.properties (page 28)".

Chapter 2. Implementing and Setting up the WebAPI

Following describes the work and various settings necessary for implementing WebAPI.

2.1 Setting up When Implementing the WebAPI

Following describes a variety of settings necessary for implementing WebAPI.

The following settings are required before using WebAPI.

1. Setting up the use of port for access to WebAPI.
2. Specify the WebAPI startup parameters.
3. Specify the SSL settings.
4. Activate WebAPI
5. Check the connection from the client

Important

Specify this setting on both of the active and standby nodes in the cluster, when you use redundant cluster system. When you use single-node cluster system, set up only the node.

Important

Be sure to perform the settings described in the following section as a root account.

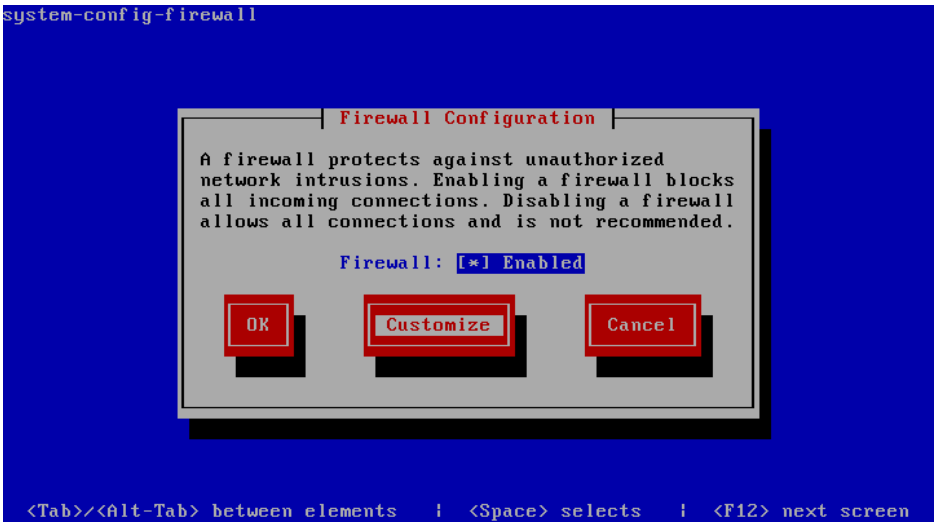
2.1.1 Setting up the Use of Port for Access to WebAPI

Need to set up the use of port for access to WebAPI. The procedures are followings.

1. Execute the command following.

```
# system-config-firewall-tui
```

2. The Firewall Configuration screen is displayed.

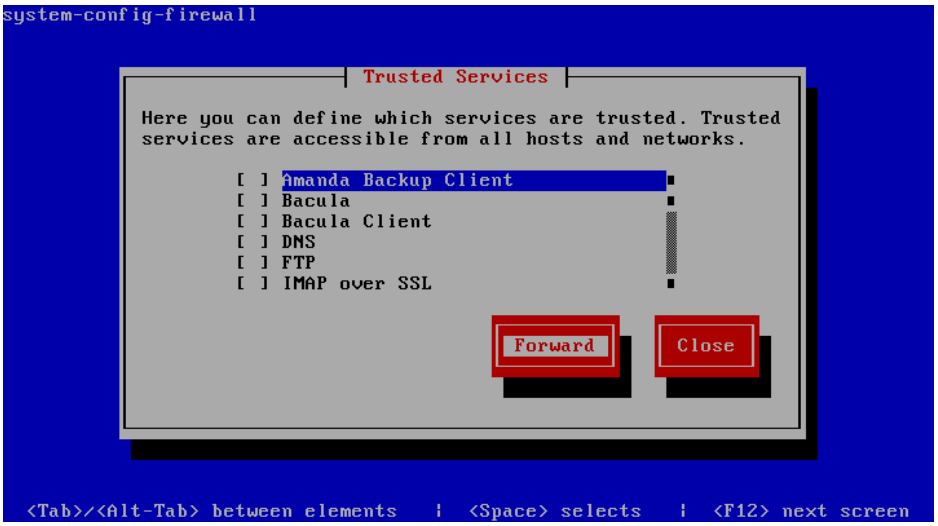


Set the focus on **Customize**, and then press the **space** key.

Tip

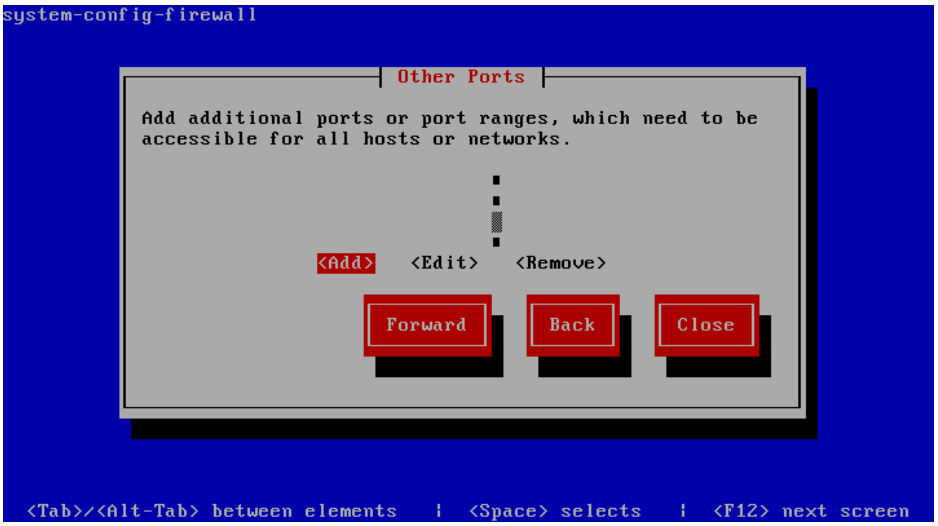
You can move the focus by pressing the **Tab** key.

3. The Trusted Services screen is displayed.



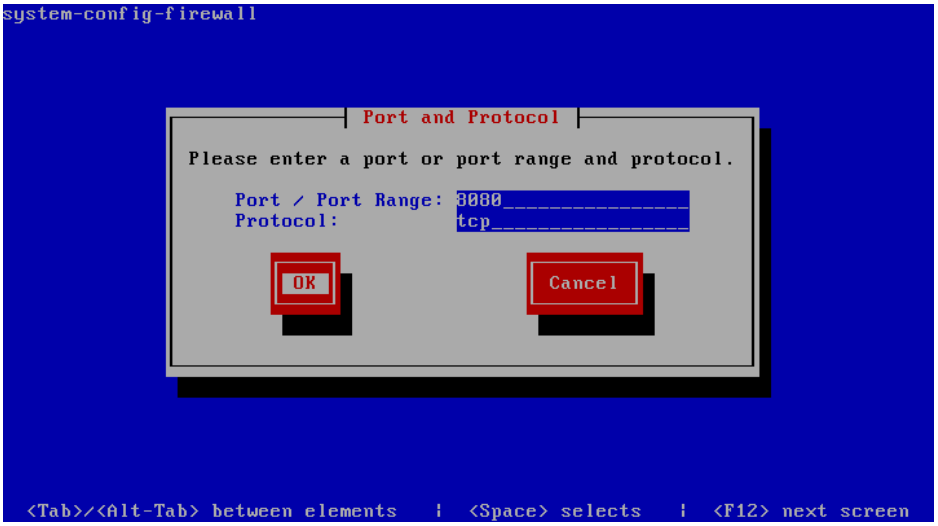
Set the focus on **Forward**, and then press the **space** key.

4. The Other Ports screen is displayed.



Set the focus on **<Add>**, and then press the **space** key.

5. The Port and Protocol screen is displayed.

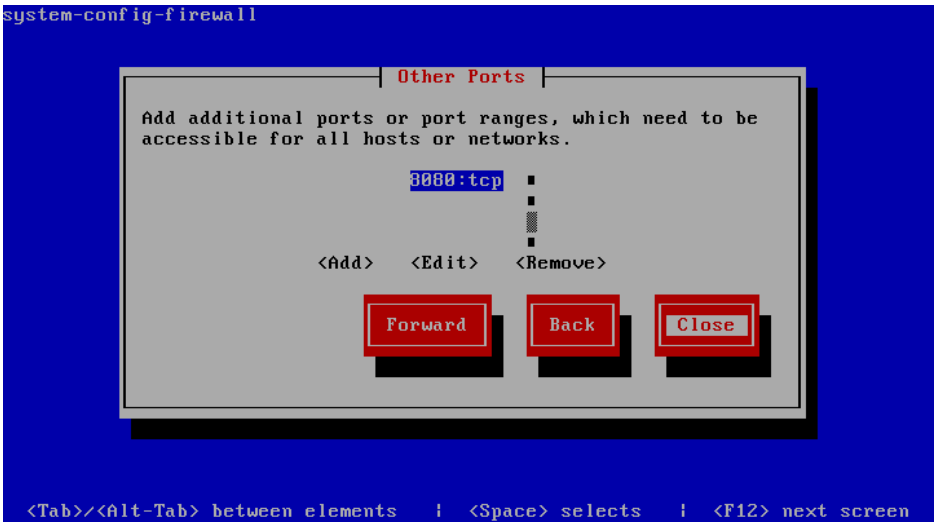


Input the following values.

Name	Value
Port / Port Range	8080 (Default for WebAPI. You can change the value.)
Protocol	tcp

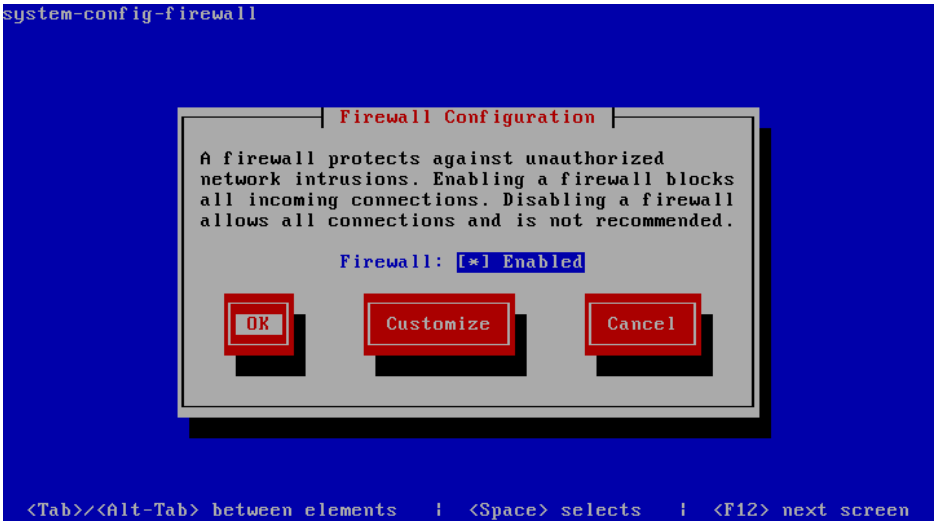
After input the values, set the focus on **OK** and then press the **space** key.

6. The Other Ports screen is displayed again.



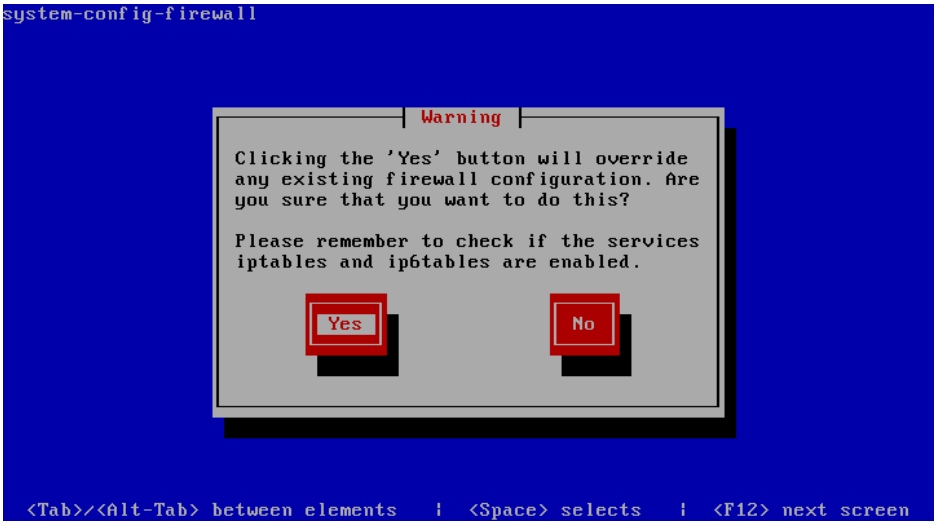
After checking that the specified port number and protocol are displayed, set the focus on **Close** and then press the **space** key.

7. The Firewall Configuration screen is displayed again.



Set the focus on **OK** and then press the **space** key.

8. The Warning screen is displayed.



Set the focus on **Yes** and then press the **space** key.

2.1.2 Specifying the WebAPI Startup Parameters

Because the WebAPI function is disabled in the PFC server initial settings, change the WebAPI startup parameters following the procedure below.

1. Editing webapi-daemon.ini

   Edit the WebAPI configuration file /opt/nec/pfc/Agent/sg/webapi/webapi-daemon.ini to change the WebAPI startup parameter auto_start to ON by using an editor.

```
[webapi-daemon-init-config]
livecheck_interval = 10
auto_start = ON
mdc_start_flag = OFF
```

   Important

   To use the WebGUI, specify the mdc_start_flag parameter to ON.

2. Saving webapi-daemon.ini

   Save webapi-daemon.ini.

2.1.3 Specifying the SSL Settings

Specify the SSL settings on the PFC server where the WebAPI operates. The setting flow is as shown below. When using the pre-installed default configuration file, steps 1 and 2 can be skipped.

Tip

In the initial state that PFC is installed, the SSL configuration files (keystore and truststore) of WebAPI are pre-installed. You can check the content of each file by using the following commands.

- Keystore (File name: *keystore_server*)

  `# keytool -list -v -keystore keystore_server -storepass` **keystore_password**

- Truststore (File name: *truststore_client*)

  `# keytool -list -v -keystore truststore_client -storepass` **truststore_password**

The passwords to be specified for **keystore_password** and **truststore_password** are as follows:

keystore_password	`keystorepfcpfc`
truststore_password	`truststorepfcpfc`

You can operate by using the above configuration files. If you want to create new configuration files, execute the procedure below in the order from step 1.

1. Creating the parameter file for the setup tool

   Although default values can be used for most of parameters, it is necessary to specify the values to some parameters according to your environment. Specify the setting values for these parameters in the file before executing the setup tool.

2. Creating a certificate

   Create the server certificate or other necessary items for your node and store them in the keystore and truststore. Use the setup tool for this work.

3. Acquiring the certificate from the counterpart node and adding it to the truststore

Store the server certificates for your node and the counterpart node in a single truststore. Use the setup tool for this work. This work is not required in single-node cluster system.

Use the following setup tool provided by WebAPI to set up the SSL used by WebAPI.

```
/opt/nec/pfc/Agent/bin/webapi/webapi_ssl_setup
```

Tip

Following describes how to create the key and import the certificate to use HTTPS in the WebAPI. In the description, the file to store the local certificate (key) is called "keystore" and the one to store the remote (communication counterpart's) certificate is called "truststore."

Tip

There is no problem if two truststores can be confirmed either the active or standby node after executing "keytool-list," a standard JDK command.

1. Creating the parameter file for the setup tool
   a. Open the following parameter file with an arbitrary text editor.

      ```
      /opt/nec/pfc/Agent/sg/webapi/webapi_ssl_setup.param
      ```

   b. Specify necessary parameter information referring to the table below. The values in the "Default value" column are used for the parameters whose values are not specified.

      Parameters related to the server certificate

Parameter name	Description	Default value
server_key_alias	Name to uniquely identify the created key (certificate) in the keystore. Any name can be specified. This parameter is case-insensitive.	serverKey
server_validity	Validity period of certificate. Specify a value equal to or larger than 1.	365
server_dname	Information required to issue the certificate (X. 500 identification name). This information includes the issuer and his organization. If you execute a command omitting the -dname option, you are prompted to enter necessary information for this argument.	"CN=WebAPI, OU=PFC, O=NEC, L=City, ST=State, C=JP"
server_cert_alias	Name to uniquely identify the imported certificate in the truststore. Any name can be specified. This parameter is case-insensitive.	serverCert
server_cert_alias2	Name to uniquely identify the server certificate for the counterpart node in the truststore on your node in the cluster configuration. Specify a value different from **server_cert_alias**. This parameter is case-insensitive.	serverCert2

      Parameters related to acquiring the server certificate from the counterpart node

Parameter name	Description	Default value
opposite_node	IP address of counterpart node in the cluster configuration. In the redundant configuration, specify the IP address of eth0 of the counterpart node. This parameter is required when executing webapi_ssl_setup with the -r option.	None (This parameter cannot be omitted.)

c. Overwrite and save `webapi_ssl_setup.param`.

2. Creating a certificate

   Execution Example:

```
# /opt/nec/pfc/Agent/bin/webapi/webapi_ssl_setup   (1)
WARNING: "/opt/nec/pfc/Agent/sg/webapi/keys" directory has some files. Do you continue?
[y/n] y   (2)
Input keystore password : keystore_password   (3)
Input truststore password : truststore_password   (4)
# ls -l /opt/nec/pfc/Agent/sg/webapi/keys   (5)
total 24
-rw-r--r-- 1 root root 1224 Aug 16 13:25 keystore_server
-rw-r--r-- 1 root root 1072 Aug 16 13:25 server.cert
-rw-r--r-- 1 root root  818 Aug 16 13:25 truststore_client
```

   a. Execute the following command. (**(1)** in the above execution example)

```
/opt/nec/pfc/Agent/bin/webapi/webapi_ssl_setup
```

   b. If there is some data in the /opt/nec/pfc/Agent/sg/webapi/keys directory, to which the certificate or other data is output, the following message is displayed to ask you whether to continue the processing. To continue the processing, enter 'y' and press the **Enter** key (The processing is aborted if any other character than 'y' is entered). (**(2)** in the above execution example)

   c. Specify the keystore password. (Execution Example(3))

   d. Specify the truststore password. (Execution Example(4))

   Tip

   You can specify different passwords for each node in a redundant-node cluster system. This applies to both the keystore and truststore.

   e. The certificate has been successfully created when no message beginning with `ERROR:` is output. Check that there are three files in the /opt/nec/pfc/Agent/sg/webapi/keys directory. (Execution Example(5))

   f. This completes the specification of the WebAPI SSL settings in a single-node cluster system. In a redundant-node cluster system, create another certificate on the counterpart node and follow the steps described in "3. Acquiring the certificate from the counterpart node and adding it to the truststore".

3. Acquiring the certificate from the counterpart node and adding it to the truststore

   Important

   If you use single-node cluster system, you do not need the steps in this section.

Important

Perform the steps described in this section on either the active or standby node. This manual shows the steps on the active node.

Execution Example:

```
# /opt/nec/pfc/Agent/bin/webapi/webapi_ssl_setup -r  (1)
Input truststore password : truststore_password   (2)
root@<IP address of counterpart node>'s password:   (3)
server.cert                                   100% 1072     1.1KB/s   00:00
Enter keystore password:
Keystore type: JKS
Keystore provider: SUN

Your keystore contains 2 entries

Alias name: servercert   (4)
Creation date: Aug 16, 2012
Entry type: trustedCertEntry

Owner: CN=WebAPI, OU=PFC, O=NEC, L=City, ST=State, C=JP
Issuer: CN=WebAPI, OU=PFC, O=NEC, L=City, ST=State, C=JP
Serial number: 502c763c
Valid from: Thu Aug 16 13:25:32 JST 2012
      until: Fri Aug 16 13:25:32 JST 2013
Certificate fingerprints:
         MD5:  08:22:E8:4C:53:03:F4:5B:C9:A8:EE:D9:BE:DE:5D:62
         SHA1: 6B:C3:0F:36:64:90:23:5B:6D:EA:C8:53:71:21:9D:2A:8C:5A:C7:02
         Signature algorithm name: SHA1withDSA
         Version: 3


*******************************************
*******************************************


Alias name: servercert2   (5)
Creation date: Aug 16, 2012
Entry type: trustedCertEntry

Owner: CN=WebAPI, OU=PFC, O=NEC, L=City, ST=State, C=JP
Issuer: CN=WebAPI, OU=PFC, O=NEC, L=City, ST=State, C=JP
Serial number: 502bb06b
Valid from: Wed Aug 15 23:21:31 JST 2012
      until: Thu Aug 15 23:21:31 JST 2013
Certificate fingerprints:
         MD5:  44:A6:0C:B8:44:B8:7B:E2:D8:6C:5D:D0:E1:86:94:5A
         SHA1: 51:95:2E:35:8A:99:82:62:B8:80:9A:EF:54:47:94:5F:1A:D5:84:F1
         Signature algorithm name: SHA1withDSA
         Version: 3


*******************************************
*******************************************
```

a. Execute the following command. (**(1)** in the above execution example)

```
# /opt/nec/pfc/Agent/bin/webapi/webapi_ssl_setup -r
```

b. Specify the truststore password. (Execution Example(2))

Specify the same password as **truststore_password** specified in "2. Creating a certificate".

- Passwords described in "Tips" at the beginning of "2.1.3 Specifying the SSL Settings (page 15)" (when using the default configuration files)

- Passwords specified in "**(2)** Creating a certificate" (when creating new SSL configuration files)

Important

The passwords must be managed with great care.

c. Enter the password for the root account on the counterpart node. (**(3)** in the above execution example)

d. The contents in /opt/nec/pfc/Agent/sg/webapi/keys/truststore_client (truststore file located on the client) are displayed. Check that the server certificate for your node (**(4)** in the above execution example) and that for the counterpart node (**(5)** in the above execution example) are displayed.

Important

The truststore file stores the certificate of the server in which the WebAPI operates. RestClient, which is the software used to access the WebAPI, uses the truststore file as a certificate. The file can be specified with "Trust store file" in the RestClient tool. For details, see "2.1.6 Checking the Connection from Web Client (page 20)". It is also possible to acquire a certificate from the truststore file by using the keytool, a JDK standard, by setting it for the existing truststore and using it.

Tip

Although the display order of server certificates differs depending on the node, this does not affect the operation. In addition, there is no problem if the merged truststore file is used on the node of either the active or standby node which executed the merge operation.

e. This completes specifying the WebAPI SSL settings in the redundant-node cluster system.

2.1.4 Registering authentication information

The authentication method to link with the Web GUI is enabled by default in the WebAPI. However, the password for the method is not opened. Therefore, register other username and password for authentication to check the connection to WebAPI

Before checking to connection to WebAPI, refer to "2.2.2.6 pwd.properties (page 26)" and then register "type" (2 or 3) and username and password.

2.1.5 Activating WebAPI

Execute the following command to activate the WebAPI.

```
# /opt/nec/pfc/Agent/bin/webapi/pfcwebapi start webapi
```

How to check that WebAPI has been activated

You can check that WebAPI has been activated by using the following procedure.

1. Check the presence of the process.

   Execute the following command to check that a java process and a pfcshell process exist.

```
# ps -ef | grep -i webapi | grep -v "grep"
root   7969     1  0 16:05 pts/0   00:00:00   java -jar
   -Djava.library.path=/opt/nec/pfc/Agent/bin/webapi webapi.jar --start
```

```
root   7984  7969  0 16:05 pts/0   00:00:00
   /opt/nec/pfc/Agent/bin/pfcshell --api WebAPI
```

* Although the command may be displayed on multiple lines depending on the layout, enter this command on two lines.

2. Check the presence of the service port

   Execute the following command to check that the WebAPI service port (TCP port 8080) exists in the LISTEN state.

```
# netstat -ano | grep 8080
tcp     0     0 :::8080         :::*          LISTEN      off (0.00/0/0)
```

2.1.6 Checking the Connection from Web Client

Perform the following operation on the client machine from which you access WebAPI.

Important

Following describes the procedure to check the WebAPI operation by using a Web browser or RESTClient tool. However, Web client applications that operate as a WebAPI client is not described in this manual.

Access WebAPI by using a Web browser

Execute the following to check that you can access WebAPI.

1. Enter URI of WebAPI in the toolbar of a Web browser.

   Example: https://<server IP address>:8080/pfc_webapi_version

2. You are prompted to enter the user name and password to access WebAPI. Enter the username and password registered in "2.1.4 Registering authentication information (page 19)".

3. Confirm that there is a response from WebAPI.

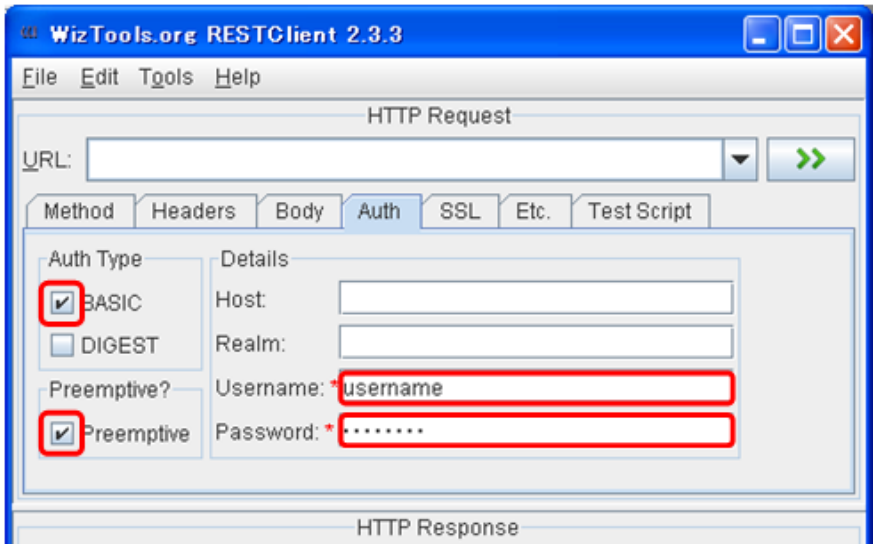
Access WebAPI by using the RESTClient tool

Execute the following to check that you can access WebAPI. In the following description, RESTClient 2.3.3 is used as an example.

1. Start the RESTClient tool.

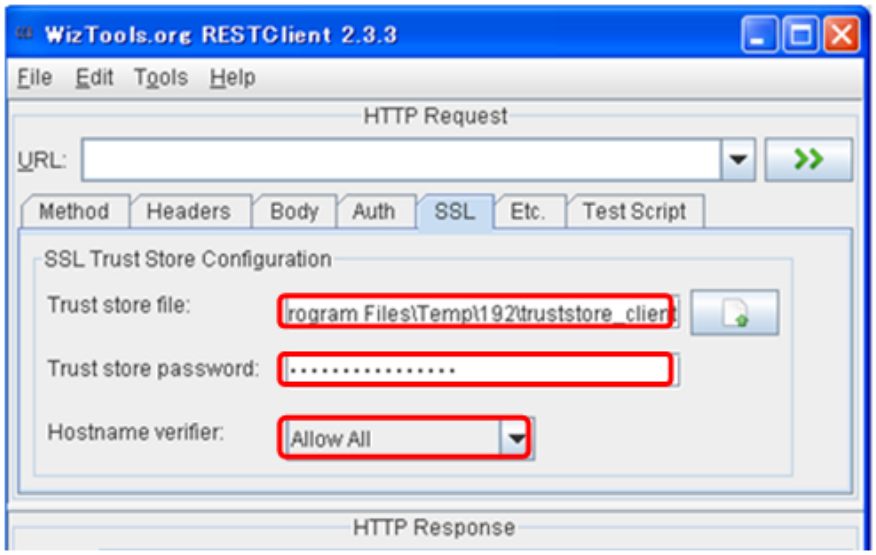
2. Specify the following items on the [Auth] tab.

Item Name	Setting
BASIC	Select this check box.
Preemptive	Select this check box.
Username	Enter the username registered in "2.1.4 Registering authentication information (page 19)".
Password	Enter the password registered in "2.1.4 Registering authentication information (page 19)".

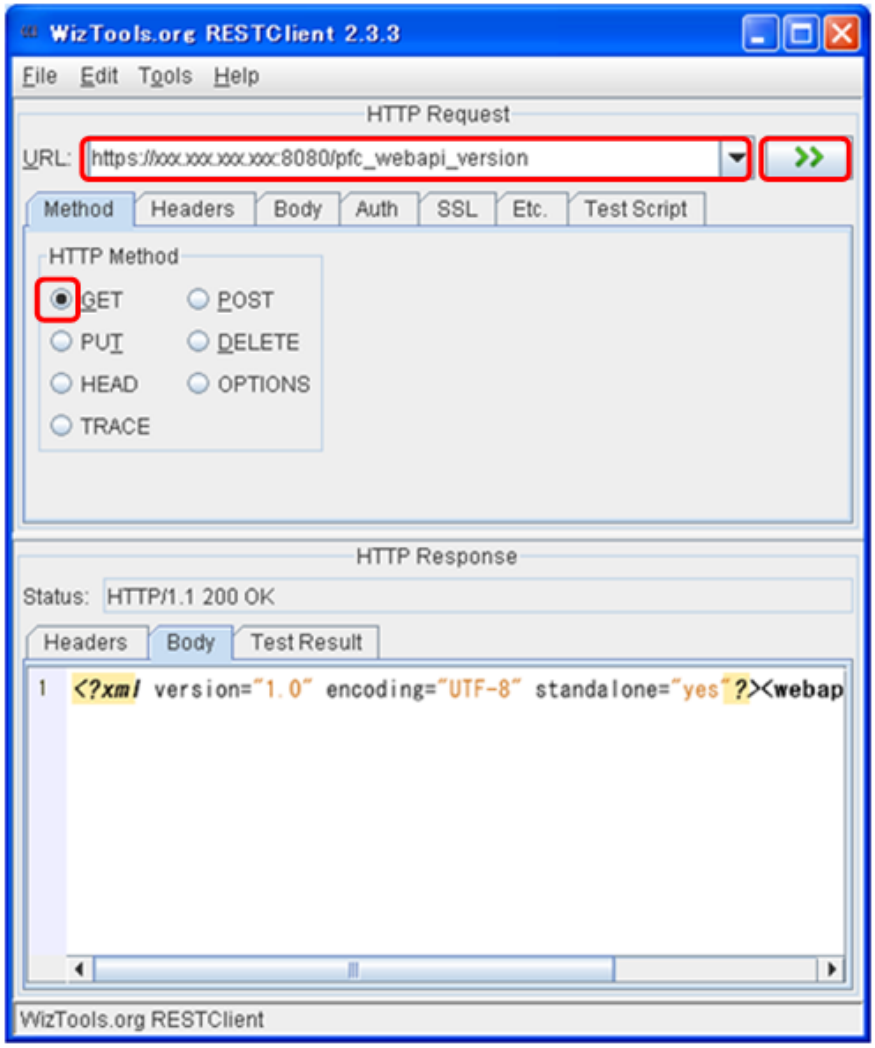


3. Specify the following items on the [SSL] tab.

Item Name	Setting
Trust store file	Copy the **truststore_client** file created in (3) of "3. Acquiring the certificate from the counterpart node and adding it to the truststore" in "2.1.3 Specifying the SSL Settings (page 15)", which contains the server certificates for both the active and standby nodes, in a desired folder of the client machine, and then enter the path to the file.
Trust store password	Enter the **truststore_pass** parameter that is specified in (2) of "1. Creating the parameter file for the setup tool" in "2.1.3 Specifying the SSL Settings (page 15)".
Hostname verifier	Select **Allow All**.



4. Select **GET** as a method on the **Method** tab. Enter the URI and click the button at the right of the field to check that there is a response from WebAPI.



Tip

The following requirements must be satisfied by the HTTP Client which accesses the PFC server.

1. Be compatible with HTTP/1.1;
2. Have HTTPS functions (for example, SSL3.0, TLS 1.0);
3. Be capable of sending UID/PWD information in the Authorization header; and
4. Be capable of continuing processing by skipping an unknown data element (for example, data item, tag) without handling it as an error upon the detection of an unknown data element.

2.2 WebAPI Configuration File

Following describes the configuration files used in the WebAPI.

2.2.1 Configuration Files

There are the following configuration files for WebAPI.

Table 2-1 Configuration Files

Directory	File Name	Description
/opt/nec/pfc/Agent/sg/webapi	config.properties	Specifies the temporary file path

Directory	File Name	Description
		Specifies the interfaces that accept HTTP requests
	sys.properties	Specifies internal operations (such as caching)
	access.properties	Specifies the access permission to WebAPI
	webapi-daemon.ini	Specifies the service activation
	ssl.properties	Specifies the HTTPS (SSL)
	pwd.properties	Specifies the HTTP authentication
	loglist-webapi-daemon	Specifies the log output environment
	logfile-webapi-daemon	Specifies the service process log
	log.properties	Specifies the operation log
	logaccess.properties	Specifies the access log
	listenport.properties	Specifies the port for access to WebAPI

Important

The WebAPI settings are not synchronized between cluster nodes. Manually match the ACT and SBY settings. For how to specify the settings, refer to "2.1.2 Specifying the WebAPI Startup Parameters (page 15)" and "2.1.3 Specifying the SSL Settings (page 15)" .

Tip

The above configuration files are backed up and restored. For details about the backup/restore functions, refer to the *Configuration Guide*.

2.2.2 Details of Configuration Files

Following describes each of the configuration files used in WebAPI.

2.2.2.1 config.properties

```
# Define the path for temporary files
#
tmp_file_path=/var/opt/nec/pfc/Agent/tmp/webapi/

# If choose to listen on all interfaces, comment the following option (default).
# If choose to listen on defined IPv4 address, define the IPv4 address
# (ex. listen_ip=10.0.0.1).
# If define IP address as 0.0.0.0, it is the same as default that will be listen
# on all interfaces.
#
# listen_ip=10.0.0.1
```

- tmp_file_path

  Specify the storage path for the WebAPI temporary file.When a partition of the path mentioned by default has a small capacity, please designate the path in the large capacity partition.

- listen_ip

  Specify the interfaces (IP addresses) that accept HTTP requests.

  If this is not specified, all the interfaces on the server accept HTTP requests. The same applies when 0.0.0.0 is specified.

2.2.2.2 sys.properties

```
# Define on/off of cache.
#   0: Enable cache
#   1: Disable cache
# (Default: 1)
cache_disable=1

# Specifies the maximum size, in Mbytes, of the memory allocation pool.
# The default value is chosen at runtime based on system configuration.
#
#heap_max_size=1024

# Chooses the default media type from XML or JSON (Default: XML).
#
#default_media_type=XML

# Specify the retry number of times on conflict of cofig-mode acquisition (Default: 8).
#
#config_mode_retry_count=8
```

- cache_disable

  0: Enables the cache.

  1: Disables the cache (default value).

  This parameter is defined for functional enhancement in the future. Please don't do setting change and make it a condition of a fixed value.

- heap_max_size

  Specify the maximum size of the heap memory used by WebAPI, in MB. Set a value between 64 MB and 1536 MB.

  If this setting is omitted, a value is automatically assigned according to the amount of physical memory.

  A physical memory of a server is used much for WebAPI so that the value is big. Please establish it according to the physical memory size.

- default_media_type

  Choose the default media type that is used when it was specified in neither the URI or the request header.

  You can choose from XML or JSON (Default: XML).

- config_mode_retry_count

  Specify the retry number of times, which WebAPI retry config-mode acquisition in case of confliction of config-mode acquisition (retry interval is always 1 sec).

  Specify the value more than 0. Default is 8.

  When it's set as the big value, detection of a retry expiration error of configure mode acquisition becomes slow.

2.2.2.3 access.properties

```
# If remote access is through a proxy, define 1. If not, define 0.
# (Default: 0)
#
PROXY_OPT=0

# Define http method for GET/POST/DELETE/PUT separately.
#
```

```
# If * is defined, which means it is allowed for all IP address.
# It can be defined by IP address like 10.0.0.1, 10.0.0.2, 10.0.0.3.
#
# The value for "ALL" is prior to value for other method. (So, if IP address or
# * is defined for "ALL", defined IP address or * for GET/POST/DELETE/PUT will
# be ignored.)
#
ALL=*
GET=*
POST=*
DELETE=*
PUT=*
```

- PROXY_OPT

  0: Specify for the web client application no using Proxy and accesses WebAPI (default value).

  1: Specify for the web client application using Proxy and accesses WebAPI.

- ALL

  Specify the IP address of the client that is allowed to use all the methods.

- GET

  Specify the IP address of the client that is allowed to use the GET method.

- POST

  Specify the IP address of the client that is allowed to use the POST method.

- DELETE

  Specify the IP address of the client that is allowed to use the DELETE method.

- PUT

  Specify the IP address of the client that is allowed to use the PUT method.

Remember

If an asterisk (*) is specified, the IP addresses are permitted.

Remember

Multiple IP addresses can be specified by separating them with a comma (,).

2.2.2.4 webapi-daemon.ini

```
[webapi-daemon-init-config]
livecheck_interval=10
auto_start=OFF
mdc_start_flag=OFF
```

- livecheck_interval

  Specify the keep alive monitoring interval for the WebAPI function in seconds.Please designate it at the reach 5 to 360.

- auto_start

  This is the specification of the automatic activation of the WebAPI function.

  ON: The WebAPI function also starts automatically when the PFC server software starts.

  OFF: The WebAPI function does not start automatically when the PFC server software starts. (Default)

- mdc_start_flag

This is the specification of the automatic activation of the Web GUI.

ON: The Web GUI function also starts automatically when the PFC server software starts.

OFF: The Web GUI function does not start automatically when the PFC server software starts. (Default)

2.2.2.5 ssl.properties

```
# HTTPS's keystorefile
# (Default: ./keys/keystore_server)
#
keystorefile=/opt/nec/pfc/Agent/sg/webapi/keys/keystore_server

# HTTPS's keystore password (AES)
# (Default: MEn51Xe8VzT/nd/udDnztQ==)
#
keystorepass=MEn51Xe8VzT/nd/udDnztQ==
```

- keystorefile

  Specify the file path for the keystore. A full path of the key store file made by "2.1.3 Specifying the SSL Settings (page 15)" is mentioned.

- keystorepass

  Specify the password of the keystore. Please indicate the one coded in AES on a password.

2.2.2.6 pwd.properties

```
# Type of Username/Password for HTTP authentication
#
# Define the type of Username/Password.
#   1: Default Username/Password
#   2: Username/Password defined in system password file
#   3: Username/Password defined in this file
# (Default: 1)
#
type=1

# If 3 is defined for "type", use the following Username/Password.
# At most, 1000 Username/Password can be defined.
# The format is following.
#
#   username:MD5(password)
#
# (ex.)
#   user1:178d0ff61c7ab8684b9aecae846b1b44
#   userABC:d41d8cd98f00b204e9800998ecf8427e
#
```

- type

  Specify the authentication type.

  1: Use the default WebAPI user name and password.

  2: Use the user name and password for the OS (/etc/passwd).

  3: Use the user name and password specified in this file.

  When you set 1 to type, you do not need to register the user name and password clearly.

  When you set 2 or 3 to type, you need to register the user name and password by below.

Important

Be sure to register the settings as a root account.

Tip

Apply user name and password which is appropriate for your settings.

Registering the user name and password when you set 2 to type

Register the user name and password to /etc/passwd file. The procedure is below.

1. Execute the following command. The user name "**username**" is registered.

```
# useradd username
```

2. Execute the following command. The password for "**username**" is set.

```
# passwd username
Changing password for user username.
New password:password (No echo back)
Retype new password: password (Input same password above)
passwd: all authentication tokens updated successfully.
```

3. Stop or start the WebAPI.

```
# /opt/nec/pfc/Agent/bin/webapi/pfcwebapi stop webapi
stopping PFC WebAPI                                        [  OK  ]
# /opt/nec/pfc/Agent/bin/webapi/pfcwebapi start webapi
starting PFC WebAPI                                        [  OK  ]
#
```

Registering the user name and password when you set 3 to type

Add the string which consists of the user name (plain string) and ":" and the password (MD5 hashed) to pwd.properties file. The procedure is below.

1. Execute below command. The user name "**username**" is hashed by MD5, and then the string of 32 characters by hexadecimal number is displayed.

```
# echo -n password | /usr/bin/md5sum
00112233445566778899aabbccddeeff  -
```

2. Copy the displayed string by hexadecimal number.
3. Join "**username**" and ":" and the copied string in 2). And then, add the joined string into pwd.properties file.

Important

When you use both WebGUI and the other Web client application, copy and paste the following description, which is the authentication information for WebGUI, to pwd.properties file.

```
pfcpfc:03693c71b9375f574f98fa63a337dbdd
```

Tip

Refer to the example in the end of pwd.properties file. Do not insert blank in front of the joined string.

4. Stop or start the WebAPI.

```
# /opt/nec/pfc/Agent/bin/webapi/pfcwebapi stop webapi
stopping PFC WebAPI                                        [  OK  ]
# /opt/nec/pfc/Agent/bin/webapi/pfcwebapi start webapi
starting PFC WebAPI                                        [  OK  ]
#
```

2.2.2.7 loglist-webapi-daemon

```
[webapi_daemon_sys_localfile]
category=sys
log=file
config=/opt/nec/pfc/Agent/sg/webapi/logfile-webapi-daemon
section=webapi_daemon_log
```

This is the definition file for the log output environment. This file does not require correction.

2.2.2.8 logfile-webapi-daemon

```
[webapi_daemon_log]
path=/var/opt/nec/pfc/Agent/log/webapi/webapi_daemon.log
size=10000
rotates=3
level=debug6
```

- path

  Specify the file storage path.When a partition of the path mentioned by default has a small capacity, please designate the path in the large capacity partition.

- size

  Specify the maximum file size.

  unit : KBytes

  min : 1

  max : 50000

  default : 1000

- rotates

  Specify the generation to be rotated.

  min : 1

  max : 10

  default : 3

- level

  Specify the log level.It's possible to designate below.

  emerg,alert,crit,err,warning,notice,info,debug1,debug2,debug3,debug4,debug5,debug6,debug7,debug8,debug9.

2.2.2.9 log.properties

```
# The log file path (must specified, no default value)
#
log_file=/var/opt/nec/pfc/Agent/log/webapi/webapi.log
```

```
# Max size (MB) of a single log file
#
# The legal value is a positive number or 0 which means unlimited.
# (Default: 1)
#
log_limit=1

# Log level
#
# The following values are available.
#   CRITICAL or CRIT or 1
#   ERROR    or ERRO or 2
#   WARNING  or WARN or 3
#   NOTE     or NOTE or 4
#   INFO     or INFO or 5
#   DEBUG    or DEBG or 6
#   TRACE    or TRAC or 7
# (Default: NOTE)
#
log_level=NOTE

# Max log file count
#
# If log_limit is 0, log_count will be ignored. Otherwise, the legal value is
# between 1 and 256.
# (Default: 5)
#
log_count=5
```

- log_file

  Specify the storage path for WebAPI log file.When a partition of the path mentioned by default has a small capacity, please designate the path in the large capacity partition.

- log_level

  Specify the WebAPI log level.

- log_limit

  Specify the maximum WebAPI log size in MB. There is no limit if 0 is specified.

- log_count

  Specify the maximum number of saved WebAPI log files that reach the maximum size specified with log_limit. 1 to 256 can be specified.

Remember

The maximum log storage amount is 5 MB by default.

2.2.2.10 logaccess.properties

```
# The log file path (must specified, no default value)
#
log_file=/var/opt/nec/pfc/Agent/log/webapi/webapi_access.log

# Max size (MB) of a single log file
#
# The legal value is a positive number or 0 which means unlimited.
# (Default: 1)
#
log_limit=1

# Max log file count
#
# If log_limit is 0, log_count will be ignored. Otherwise, the legal value is
# between 1 and 256.
# (Default: 5)
```

```
#
log_count=5
```

- log_file

  Specify the storage path for WebAPI log file.When a partition of the path mentioned by default has a small capacity, please designate the path in the large capacity partition.

- log_limit

  Specify the maximum WebAPI log size in MB. There is no limit if 0 is specified.

- log_count

  Specify the maximum number of saved WebAPI log files that reach the maximum size specified with log_limit. 1 to 256 can be specified.

Remember

The maximum log storage amount is 5 MB by default.

2.2.2.11 listenport.properties

```
# Define listen port between 0 and 65535.
#   0: Does not start the server.
#   1-65535: Start the server and listen on the defined port.
# If not defined any, listen on the default port.

# Basic Port
# (Default: 8080)
#
#webapi_port=8080
```

- webapi_port

  Specify the port number for WebAPI. It is not necessary to deregister the default comment when using the default value, 8080. When using except for 8080, please release a comment and designate used TCP portnumber. The value by which designation is possible is 0 or 1-65535, and service doesn't start in case of 0.

Chapter 3. WebAPI (base-related) Reference

Following describes the URI, parameters, request/response data and other details about the base-related API.

Tip

WebAPI uses expressed data as input/output of API structurally at XML and JSON. The data element which becomes the same hierarchy on the structure is being gathered in one table and an element name is being transcribed by bold face like **"data"** about each input/output data of API at this chapter. When there is more data of the lower hierarchy, the data name is described by like "data" (not bold-faced) and the table numbers of the reference destination are indicated in the "Return Value" column.

3.1 Operate startup-configuration

3.1.1 Update startup-configuration

Following describes how to upgrade PFC startup-configuration.

Processing request

Method

PUT

request URI

- XML format

  `/pfc/startup_configuration.xml`

  `(or /startup_configuration.xml)`
- JSON format

  `/pfc/startup_configuration.json`

  `(or /startup_configuration.json)`

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Update startup-configuration succeeded."`

- On failure

  `"Update startup-configuration failed.(<Error information from pfcshell>)"`

Remark

None

3.2 Operate PFC Process Alarm

3.2.1 Show alarms occurring in PFC process

Following describes how to acquire the PFC process alarm information.

Processing request

Method

GET

request URI

- XML format

  `/pfc/alarms.xml`

- JSON format

  `/pfc/alarms.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<alarms>
  [
  <alarm date="date"
    severity="severity"
    alarmnumber="alarmnumber"
```

```
    type="type"
    vtn_name="vtn_name"
    alarminfo="alarminfo"/>
  ]
</alarms>
```

- JSON format

```
{
  "alarms": [
    {
      "date": "date",
      "severity": "severity",
      "alarmnumber": "alarmnumber",
      "type": "type",
      "vtn_name": "vtn_name",
      "alarminfo": "alarminfo"
    }
  ]
}
```

Table 3-1 Description of Elements in alarms

Element	Description	Return Value
date	Alarm generation date	Date information (Format of YYYY-MM-DD hh:mm:ss)
severity	Type of severity	{emergency|alert|critical|error|warning|notice|information|debug}
alarmnumber	Alarm number	Up to 64 one-byte numeric characters
type	Alarm generation/recovery	{occurred|recovered}
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
alarminfo	Details of alarm	Up to 254 characters including NULL characters. The line feed (LF) is treated as two characters.

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.3 Operate API Version

3.3.1 Show version of API

Following describes how to acquire the API version.

Processing request

Method

GET

request URI

- XML format

  `/pfc_webapi_version.xml`

- JSON format

  `/pfc_webapi_version.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<webapi_version version="version"/>
```

- JSON format

```
{
  "webapi_version": {
    "version": "version"
  }
}
```

Table 3-2 Description of Elements in version

Element	Description	Return Value
version	WebAPI version information	V*.* format (*: decimal integer value)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.4 Operate Physical Network Flow

3.4.1 Show data-flow for real network

Following describes how to acquire the physical network flow information.

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/dataflows.xml`

- JSON format

  `/realnetwork/dataflows.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `df_id=`**`df_id`**

  `dp_id=`**`dp_id`**

  `port_name=`**`port_name`**

Table 3-3 Description of Parameters in URI

Element	Description	Valid Value
df_id	Dataflow ID	Decimal number (1 to 4294967295)
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number). It is not permitted to specify F's for all of them.
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

Remember

If port_name contains the following symbols, they must be percent-encoded (converted to "%xx") on the Web client.

/ < > % , { }

Remember

For parameters in a query character string, "df_id only," "dp_id only," or a combination of "dp_id and port_ name" can be specified.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataflows>
  [
  <dataflow flowid="flowid"
    status="status"
    flowtype="flowtype"
    pathpolicyindex="pathpolicyindex"
    vtnid="vtnid"
    ingressofsdpid="ingressofsdpid"
    inport="inport"
    instationid="instationid"
    egressofsdpid="egressofsdpid"
    outport="outport"
    outstationid="outstationid"/>
  ]
  <totalcount value="value"/>
</dataflows>
```

- JSON format

```
{
  "dataflows": [
    {
      "flowid": "flowid",
      "status": "status",
      "flowtype": "flowtype",
      "pathpolicyindex": "pathpolicyindex",
      "vtnid": "vtnid",
      "ingressofsdpid": "ingressofsdpid",
      "inport": "inport",
      "instationid": "instationid",
      "egressofsdpid": "egressofsdpid",
      "outport": "outport",
      "outstationid": "outstationid"
    }
  ],
  "totalcount": {
    "value": "value"
  }
}
```

Table 3-4 Description of Elements in dataflows

Element	Description	Return Value
flowid	Flow ID	Decimal number (1 to 4294967295)
flowtype	Flow type	{dynamic|static|emergency}
status	Status	{init|activating|active|switching|down}
pathpolicyindex	Index of pathpolicy	One character

Element	Description	Return Value
vtnid	VTN ID	Decimal number (1 to 65536)
ingressofsdpid	OFS datapath ID for Ingress	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
inport	Input port	Decimal number (1 to 65280)
instationid	Input station ID	Decimal number (1 to 524287)
egressofsdpid	OFS datapath ID for Egress	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
outport	Output port	Decimal number (1 to 65280)
outstationid	Output station ID	Up to six characters (1 to 524287)

Table 3-5 Description of Elements in total count

Element	Description	Return Value
value	Total count	Decimal number (1 to 4294967295)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On failure

  "Get dataflow of real-network failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

3.4.2 Show data-flow details for real network

Following describes how to acquire the detailed information on the physical network flow including statistical information.

Processing request

Method

GET

request URI

- XML format

  /realnetwork/dataflows/detail.xml

- JSON format

  `/realnetwork/dataflows/detail.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `df_id=`**`df_id`**

  `dp_id=`**`dp_id`**

  `port_name=`**`port_name`**

Table 3-6 Description of Parameters in URI

Element	Description	Valid Value
df_id	Dataflow ID	Decimal number (1 to 4294967295)
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number). It is not permitted to specify F's for all of them.
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

Remember

If port_name contains the following symbols, they must be percent-encoded (converted to "%xx") on the Web client.

/ < > % , { }

Remember

For parameters in a query character string, "df_id only," "dp_id only," or a combination of "dp_id and port_ name" can be specified.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataflows>
  [
  <dataflow  flowid="flowid"
    status="status"
    flowtype="flowtype"
    pathpolicyindex="pathpolicyindex"
    vtnid="vtnid"
    ingressofsdpid="ingressofsdpid"
    inport="inport"
    instationid="instationid"
    egressofsdpid="egressofsdpid"
    outport="outport"
    outstationid="outstationid">
    <flowentryinformation>
      <match dstmac="dstmac"
        inport="inport"
```

```
        srcmac="srcmac"
        vlan_id="vlan_id"
        vlanpriority="vlanpriority"
        ethernettype="ethernettype"
        srcip="srcip"
        dstip="dstip"
        iptos="iptos"
        ipprotocol="ipprotocol"
        l4srcport_icmptype="l4srcport_icmptype"
        l4dstport_icmpcode="l4dstport_icmpcode"/>
      <action enqueueport="enqueueport"
        outport="outport"
        setvlan_id="setvlan_id"
        setvlan_priority="setvlan_priority"
        stripvlanheader="stripvlanheader"
        setsrcmac="setsrcmac"
        setdstmac="setdstmac"
        setsrcip="setsrcip"
        setdstip="setdstip"
        setiptos="setiptos"
        setl4srcport_icmptype="setl4srcport_cmptype"
        setl4dstport_icmpcode="setl4dstport_icmpcode"
        multicast_group_id="multicast_group_id"/>
    </flowentryinformation>
    <statistics packets="packets"
      octets="octets"
      duration="duration"/>
    <pathinformations>
      [
      <pathinformation hop="hop"
        inport="inport"
        ofs="ofs"
        outport="outport"/>
      ]
    </pathinformations>
  </dataflow>
  ]
  <totalcount value="value"/>
</dataflows>
```

- JSON format

```
{
  "dataflows": [
    {
      "flowid": "flowid",
      "status": "status",
      "flowtype": "flowtype",
      "pathpolicyindex": "pathpolicyindex",
      "vtnid": "vtnid",
      "ingressofsdpid": "ingressofsdpid",
      "inport": "inport",
      "instationid": "instationid",
      "egressofsdpid": "egressofsdpid",
      "outport": "outport",
      "outstationid": "outstationid",
      "flowentryinformation": {
        "match": {
          "dstmac": "dstmac",
          "inport": "inport",
          "srcmac": "srcmac",
          "vlan_id": "vlan_id",
          "vlanpriority": "vlanpriority",
          "ethernettype": "ethernettype",
          "srcip": "srcip",
          "dstip": "dstip",
          "iptos": "iptos",
          "ipprotocol": "ipprotocol",
          "l4srcport_icmptype": "l4srcport_icmptype",
          "l4dstport_icmpcode": "l4dstport_icmpcode"
        },
        "action": {
```

```
          "enqueueport": "enqueueport",
          "outport": "outport",
          "setvlan_id": "setvlan_id",
          "setvlan_priority": "setvlan_priority",
          "stripvlanheader": "stripvlanheader",
          "setsrcmac": "setsrcmac",
          "setdstmac": "setdstmac",
          "setsrcip": "setsrcip",
          "setdstip": "setdstip",
          "setiptos": "setiptos",
          "setl4srcport_icmptype": "setl4srcport_icmptype",
          "setl4dstport_icmpcode": "setl4dstport_icmpcode"
        }
      },
      "statistics": {
        "packets": "packets",
        "octets": "octets",
        "duration": "duration"
      },
      "pathinformations": [
        {
          "hop": "hop",
          "inport": "inport",
          "ofs": "ofs",
          "outport": "outport"
        }
      ]
    }
  ],
  "totalcount": {
    "value": "value"
  }
}
```

Table 3-7 Description of Elements in dataflows

Element	Description	Return Value
flowid	Flow ID	Decimal number (1 to 4294967295)
status	Status	{init|activating|active|switching|down}
flowtype	Flow type	{dynamic|static|emergency}
pathpolicyindex	Index of pathpolicy	One character
vtnid	VTN ID	Decimal number (1 to 65536)
ingressofsdpid	Ingress OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
inport	Input port	Up to 15 characters
instationid	Input station ID	Up to six characters (1 to 524287)
egressofsdpid	Egress OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
outport	Output port	Up to 15 characters
outstationid	Output station ID	Up to six characters (1 to 524287)
flowentryinformation	flowEntry information	flowEntryInformationType (For details, refer to "Table 3-9 Description of Elements in flowEntryInformationType (page 41)" below.)
statistics	Statistical information list	statisticsType (For details, refer to "Table 3-12 Description of Elements in statisticsType (page 42)"".)
pathinformation	Path information	pathInformationType (For details, refer to "Table 3-13 Description of Elements in pathInformation (page 43)")

Table 3-8 Description of Elements in totalcountType

Element	Description	Return Value
value	Total count	Decimal number (1 to 4294967295)

Table 3-9 Description of Elements in flowEntryInformationType

Element	Description	Return Value
match	Match	matchType (For details, refer to "Table 3-10 Description of Elements in matchType of flowEntryInformation (page 41)")
action	Action	actionType (For details, refer to "Table 3-11 Description of Elements in actionType of flowEntryInformation (page 42)")

Table 3-10 Description of Elements in matchType of flowEntryInformation

Element	Description	Return Value
dstmac	Destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
inport	Input port number	Decimal number (1 to 65280)
srcmac	Source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
vlan_id	VLAN ID	Decimal number (1 to 4095)
vlanpriority	VLAN priority	0 to 7
ethernettype	Ether type	Hexadecimal format including "0x"
srcip	Source IP address	Hexadecimal IPv4 dot-separated format (Example: 192.168.1.1)
dstip	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
iptos	ToS value	Hexadecimal format including "0x"
ipprotocol	Protocol field of the IPv4 header or Next Header field of the IPv6 header	1 to 255
l4srcport_icmptype	source port number ICMP Type	1. Displayed if Ethernet Type/IP protocol is neither IP/ICMP nor IPv6/ICMPv6. 2. Displayed if Ethernet Type/IP protocol is IP/ICMP or IPv6/ICMPv6. 3. Displayed in the decimal format (hexadecimal format including "0x") (Example: 1023(0x3ff). Up to 29 characters
l4dstport_icmpcode	destination port number ICMP Code	1. Displayed if Ethernet Type/IP protocol is neither IP/ICMP nor IPv6/ICMPv6. 2. Displayed if Ethernet Type/IP protocol is IP/ICMP or IPv6/ICMPv6. 3. Displayed in the decimal format (hexadecimal format including "0x") (Example: 1023(0x3ff).

Element	Description	Return Value
		Up to 29 characters

Table 3-11 Description of Elements in actionType of flowEntryInformation

Element	Description	Return Value
enqueueport	Port number	Although the port number is normally displayed with a (decimal) number, the special port is displayed with a string. String displayed with the special port number **Port number** / **Displayed string** 0xfff8 / INPORT 0xfff9 / TABLE 0xfffa / NORMAL 0xfffb / FLOOD 0xfffc / ALL 0xfffd / CONTROLLER 0xfffe / LOCAL
outport	Output port	Decimal number (1 to 65280) * Either enqueueport or outport is displayed. Both of them are not displayed at the same time.
setvlan_id	VLAN ID	Decimal number (1 to 4095)
setvlan_priority	LAN priority	Value of the PCP filed of the VLAN tag
stripvlanheader	Strip VLAN header	Delete the VLAN tag from the packet.
setsrcmac	Source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
setdstmac	Destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
setsrcip	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
setdstip	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
setiptos	ToS value	Hexadecimal format including "0x"
setl4srcport_icmptype	L4Source port number or ICMP Type	Decimal format (hexadecimal format including "0x"). Example: 1023(0x3ff)
setl4dstport_icmpcode	L4Destination port number or ICMP Code	Decimal format (hexadecimal format including "0x"). Example: 1023(0x3ff)
multicast_group_id	Multicast group ID	ID to be used to identify a multicast group (1 to 1000) and OFS queue ID

Table 3-12 Description of Elements in statisticsType

Element	Description	Return Value
packets (if statistics is on)	Number of packets that match the flow	1. "N/A" if Packets or Octets is 0xFFFFFFFFFFFFFFFF. 2. "N/A" if statistical information cannot be acquired.

Element	Description	Return Value
octets (if statistics is on)	Number of octets that match the flow	1. "N/A" if Packets or Octets is 0xFFFFFFFFFFFFFFFF. 2. "N/A" if statistical information cannot be acquired.
duration (if statistics is on)	Flow retention period	1. "N/A" if statistical information cannot be acquired.

Table 3-13 Description of Elements in pathInformation

Element	Description	Return Value
hop	Hop	Up to three characters
inport	Input port	Decimal number (1 to 65280)
ofs	OFS ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
outport	Output port	Decimal number (1 to 65280)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On failure

  `"Get dataflow detail of real-network failed. [Invalid argument(parameter subdivision)](<Error information from pfcshell>)"`

Remark

None

3.4.3 Show data-flow info for real network

Following describes how to acquire the physical network flow information (excluding detailed/statistical information).

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/dataflows/info.xml`

- JSON format

  `/realnetwork/dataflows/info.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `df_id=`**`df_id`**

  `dp_id=`**`dp_id`**

  `port_name=`**`port_name`**

Table 3-14 Description of Parameters in URI

Element	Description	Valid Value
df_id	Dataflow ID	Decimal number (1 to 4294967295)
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number). It is not permitted to specify F's for all of them.
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

Remember

If port_name contains the following symbols, they must be percent-encoded (converted to "%xx") on the Web client.

/ < > % , { }

Remember

For parameters in a query character string, "df_id only," "dp_id only," or a combination of "dp_id and port_ name" can be specified.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataflows>
  [
  <dataflow flowid="flowid"
    status="status"
    flowtype="flowtype"
    pathpolicyindex="pathpolicyindex"
    vtnid="vtnid"
    ingressofsdpid="ingressofsdpid"
    inport="inport"
    instationid="instationid"
    egressofsdpid="egressofsdpid"
    outport="outport">
    outstationid="outstationid"
    <flowentryinformation>
      <match dstmac="dstmac"
        inport="inport"
```

```
        srcmac="srcmac"
        vlan_id="vlan_id"
        vlanpriority="vlanpriority"
        ethernettype="ethernettype"
        srcip="srcip"
        dstip="dstip"
        iptos="iptos"
        ipprotocol="ipprotocol"
        l4srcport_icmptype="l4srcport_icmptype"
        l4dstport_icmpcode="l4dstport_icmpcode"/>
      <action enqueueport="enqueueport"
        outport="outport"
        setvlan_id="setvlan_id"
        setvlan_priority="setvlan_priority"
        stripvlanheader="stripvlanheader"
        setsrcmac="setsrcmac"
        setdstmac="setdstmac"
        setsrcip="setsrcip"
        setdstip="setdstip"
        setiptos="setiptos"
        setl4srcport_icmptype="setl4srcport_icmptype"
        setl4dstport_icmpcode="setl4dstport_icmpcode"
        multicast_group_id="multicast_group_id"/>
    </flowentryinformation>
    <pathinformations>
      [
      <pathinformation hop="hop"
        inport="inport" ofs="ofs" outport="outport"/>]
    </pathinformations>
  </dataflow>
  ]
  <totalcount value="value"/>
</dataflows>
```

- JSON format

```
{
  "dataflows": [
    {
      "flowid": "flowid",
      "status": "status",
      "flowtype": "flowtype",
      "pathpolicyindex": "pathpolicyindex",
      "vtnid": "vtnid",
      "ingressofsdpid": "ingressofsdpid",
      "inport": "inport",
      "instationid": "instationid",
      "egressofsdpid": "egressofsdpid",
      "outport ": "outport",
      "outstationid": "outstationid",
      "flowentryinformation": {
        "match": {
          "dstmac": "dstmac",
          "inport": "inport",
          "srcmac": "srcmac",
          "vlan_id": "vlan_id",
          "vlanpriority": "vlanpriority",
          "ethernettype": "ethernettype",
          "srcip": "srcip",
          "dstip": "dstip",
          "iptos": "iptos",
          "ipprotocol": "ipprotocol",
          "l4srcport_icmptype": "l4srcport_icmptype",
          "l4dstport_icmpcode": "l4dstport_icmpcode"
        },
        "action": {
          "enqueueport": "enqueueport",
          "outport": "outport",
          "setvlan_id": "setvlan_id",
          "setvlan_priority": "setvlan_priority",
          "stripvlanheader": "stripvlanheader",
          "setsrcmac": "setsrcmac",
```

```
          "setdstmac": "setdstmac",
          "setsrcip": "setsrcip",
          "setdstip": "setdstip",
          "setiptos": "setiptos",
          "setl4srcport_icmptype": "setl4srcport_icmptype",
          "setl4dstport_icmpcode": "setl4dstport_icmpcode"
          "multicast_group_id" : "multicast_group_id"
        },
        "pathinformations": [
          {
            "hop": "hop",
            "inport": "inport",
            "ofs": "ofs",
            "outport": "outport"
          }
        ]
      }
    }
  ],
  "totalcount": {
    "value": "value"
  }
}
```

Table 3-15 Description of Elements in dataflows

Element	Description	Return Value
flowid	Flow ID	Decimal number (1 to 4294967295)
flowtype	Flow type	{dynamic|static|emergency}
status	Status	{init|activating|active|switching|down}
pathpolicyindex	Index of pathpolicy	Up to one character
vtnid	VTN ID	Decimal number (1 to 65536)
ingressofsdpid	OFS datapath ID of Ingress	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
inport	Input port	Decimal number (1 to 65280)
instationid	Input station ID	Decimal number (1 to 524287)
egressofsdpid	OFS datapath ID of Egress	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
outport	Output port	Decimal number (1 to 65280)
outstationid	Output station ID	Decimal number (1 to 524287)
flowentryinformation	flowEntry information	flowEntryInformation type (For details, refer to "Table 3-17 Description of Elements in flowEntryInformationType (page 47)")
pathinformation	Path information	pathInformation type (For details, refer to "Table 3-20 Description of Elements in pathinformation (page 48)")

Table 3-16 Description of Elements in total count

Element	Description	Return Value
value	Total count	Decimal number (1 to 4294967295)

Table 3-17 Description of Elements in flowEntryInformationType

Element	Description	Return Value
match	Match	match type (For details, refer to "Table 3-18 Description of Elements in match of flowentryinformation (page 47)")
action	Action	action type (For details, refer to "Table 3-19 Description of Elements in action of flowentryinformation (page 47)")

Table 3-18 Description of Elements in match of flowentryinformation

Element	Description	Return Value
dstmac	Destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
inport	Input port number	Decimal number (1 to 65280)
srcmac	Source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
vlan_id	VLAN ID	Decimal number (1 to 4095)
vlanpriority	VLAN priority	Decimal number (0 to 7)
ethernettype	Ether type	Hexadecimal format including "0x"
srcip	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
dstip	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
iptos	ToS value	Hexadecimal format including "0x"
ipprotocol	Protocol field of IPv4 header or Next Header field of IPv6 header	Decimal number (1 to 255)
l4srcport_icmptype	source port number ICMP Type	1. Displayed if Ethernet Type/IP protocol is neither IP/ICMP nor IPv6/ICMPv6. 2. Displayed if Ethernet Type/IP protocol is IP/ICMP or IPv6/ICMPv6. 3. Displayed in the decimal format (hexadecimal format including "0x") (Example: 1023(0x3ff).
l4dstport_icmpcode	destination port number ICMP Code	1. Displayed if Ethernet Type/IP protocol is neither IP/ICMP nor IPv6/ICMPv6. 2. Displayed if Ethernet Type/IP protocol is IP/ICMP or IPv6/ICMPv6. 3. Displayed in the decimal format (hexadecimal format including "0x") (Example: 1023(0x3ff).

Table 3-19 Description of Elements in action of flowentryinformation

Element	Description	Return Value
enqueueport	Port number	Although the port number is normally displayed with a (decimal) number, the special port is displayed with a string. String displayed with the special port number

Element	Description	Return Value
		Port number / **Displayed string** 0xfff8 / INPORT 0xfff9 / TABLE 0xfffa / NORMAL 0xfffb / FLOOD 0xfffc / ALL 0xfffd / CONTROLLER 0xfffe / LOCAL
outport	Output port	Decimal number (1 to 65280) * Either enqueueport or outport is displayed. Both of them are not displayed at the same time.
setvlan_id	VLAN ID	Decimal number (1 to 4095)
setvlan_priority	VLAN priority	
stripvlanheader	Strip VLAN header	
setsrcmac	Source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
setdstmac	Destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
setsrcip	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
setdstip	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
setiptos	ToS value	Hexadecimal format including "0x"
setl4srcport_icmptype	L4Source port number or ICMP Type	Displayed in the decimal format (hexadecimal format including "0x") (Example: 1023(0x3ff).
setl4dstport_icmpcode	L4Destination port number or ICMP Code	Displayed in the decimal format (hexadecimal format including "0x") (Example: 1023(0x3ff).
multicast_group_id	Multicast group ID	ID to be used to identify a multicast group (1 to 1000) and OFS queue ID

Table 3-20 Description of Elements in pathinformation

Element	Description	Return Value
hop	Hop	Up to three characters
inport	Input port	Up to 15 characters
ofs	OFS ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
outport	Output port	Up to 15 characters

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On failure

  `"Get dataflow info of real-network failed. [Invalid argument(`**`parameter subdivision`**`)](<Error information from pfcshell>)"`

Remark

None

3.4.4 Show data-flow count for real network

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/dataflows/count.xml`

- JSON format

  `/realnetwork/dataflows/count.json`

- Parameter

  The following parameters are passed in the `URI` query string (`?param1=***&param2=*** format`).

  `df_id=`**`df_id`**

  `dp_id=`**`dp_id`**

  `port_name=`**`port_name`**

Table 3-21 Description of Parameters in URI

Element	Description	Valid Value
df_id	Dataflow ID	Decimal number (1 to 4294967295)
dp_id	OFS datapath ID	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number. It is not permitted to specify F's for all of them.)
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

Remember

If port_name contains the following symbols, they must be percent-encoded (converted to "%xx") on the Web client.

/ < > % , { }

Remember

For parameters in a query character string, "df_id only," "dp_id only," or a combination of "dp_id" and "port_ name" can be specified.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataflow count="count"/>
```

- JSON format

```
{
  "dataflow" : {
    "count" : "count"
  }
}
```

- On error
  - Insufficient information is returned from pfcshell (Total count)

    `"*** Error:Internal error. (no total count)"`

Table 3-22 Description of Elements in dataflow

Element	Description	Return Value
count	Number of dataflows	Decimal number (Up to20 characters)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.5 Operate Real Network Topology

3.5.1 Show topology for real network

Following describes how to acquire the physical network topology information.

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/topologies.xml`

- JSON format

  `/realnetwork/topologies.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<topologies>
  [
  <topology dp_id="dp_id">
    <ports>
      [
      <port port_name="port_name">
        <neighbor dp_id="dp_id"
          port_name="port_name"/>
      </port>
      ]
    </ports>
  </topology>
  ]
</topologies>
```

- JSON format

```
{
  "topologies": [
    {
      "dp_id": "dp_id",
      "ports": [
        {
          "port_name": "port_name",
```

```
            "neighbor": {
              "dp_id": "dp_id",
              "port_name": "port_name"
            }
          }
        ]
      }
    ]
}
```

Table 3-23 Description of Elements in topologies

Element	Description	Return Value
dp_id	datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
ports	Port information	ports type (For details, refer to "Table 3-24 Description of Elements in portsType (page 52)")

Table 3-24 Description of Elements in portsType

Element	Description	Return Value
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
neighbor	Neighbor OFS	neighborType (For details, refer to "Table 3-25 Description of Elements in neighborType (page 52)")

Table 3-25 Description of Elements in neighborType

Element	Description	Return Value
dp_id	datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.5.2 Show topology details for real network

Following describes how to acquire the detailed physical network topology information.

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/topologies/detail.xml`

- JSON format

  `/realnetwork/topologies/detail.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<topologies>
  [
  <topology dp_id="dp_id"
    ipaddr="ipaddr"
    status="status"
    avoidstatus="avoidstatus">
    <avoidstatuspolicy policyindex0="policyindex0"
     policyindex1="policyindex1"
     policyindex2="policyindex2"
     policyindex3="policyindex3"/>
    <ports>
      [
      <port port_name="port_name"
        id="id"
        adminstatus="adminstatus"
        portstatus="portstatus"
        avoidstatus="avoidstatus"
        macaddr="macaddr"
        duplex="duplex"
        speed="speed">
        <avoidstatuspolicy policyindex0="policyindex0"
          policyindex1="policyindex1"
          policyindex2="policyindex2"
          policyindex3="policyindex3"/>
        <neighbor dp_id="dp_id"
          port_name="port_name"
          linkstatus="linkstatus"
          bcmc_spt_use="bcmc_spt_use"
          weight="weight"/>
        <neighborpolicy policyindex0="policyindex0"
          policyindex1="policyindex1"
          policyindex2="policyindex2"
          policyindex3="policyindex3"/>
        </port>
      ]
    </ports>
  </topology>
  ]
</topologies>
```

- JSON format

```
{
  "topologies": [
    {
      "dp_id": "dp_id",
      "ipaddr": "ipaddr",
      "status": "status",
      "avoidstatus": "avoidstatus",
      "avoidstatuspolicy" : {
       "policyindex0": "policyindex0",
       "policyindex1": "policyindex1",
       "policyindex2": "policyindex2",
       "policyindex3": "policyindex3"
      },
      "ports": [
        {
          "port_name": "port_name",
          "id": "id",
          "adminstatus": "adminstatus",
          "portstatus": "portstatus",
          "avoidstatus": "avoidstatus",
          "macaddr": "macaddr",
          "duplex": "duplex",
          "speed": "speed",
          "avoidstatuspolicy" : {
           "policyindex0": "policyindex0",
           "policyindex1": "policyindex1",
           "policyindex2": "policyindex2",
           "policyindex3": "policyindex3"
         },
          "neighbor": {
            "dp_id": "dp_id",
            "port_name": "port_name",
            "linkstatus" : "linkstatus",
            "bcmc_spt_use": "bcmc_spt_use",
            "weight": "weight"
          },
           "neighborpolicy" : {
            "policyindex0": "policyindex0",
            "policyindex1": "policyindex1",
            "policyindex2": "policyindex2",
            "policyindex3": "policyindex3"
          }
        }
      ]
    }
  ]
}
```

Table 3-26 Description of Elements in topologies

Element	Description	Return Value
dp_id	datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)
status	Connection status	{connected|disconnected}
avoidstatus	avoid status	{on|off|active}
avoidstatuspolicy	Status of OFS exclusion setting for each path policy	avoidstatuspolicy type (For details, refer to "Table 3-27 Description of Elements in avoidstatuspolicyType (page 55)")
ports	Port information	porttype (For details, refer to "Table 3-28 Description of Elements in portType (page 55)")

Table 3-27 Description of Elements in avoidstatuspolicyType

Element	Description	Return Value
policyindex0	Status of OFS exclusion setting for policy 0	{on|off}
policyindex1	Status of OFS exclusion setting for policy 1	{on|off}
policyindex2	Status of OFS exclusion setting for policy 2	{on|off}
policyindex3	Status of OFS exclusion setting for policy 3	{on|off}

Table 3-28 Description of Elements in portType

Element	Description	Return Value
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
id	Port ID	Decimal number (1 to 4294967295)
avoidstatus	avoid status	{on|off|active}
adminstatus	admin status	{up|down}
portstatus	port status	{up|down}
macaddr	MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
duplex	Line mode	{full|half}
speed	Line speed	Number + unit Example: 10Mbps, 1Gbps
avoidstatuspolicy	Status of OFS exclusion setting for each path policy	avoidstatuspolicyType (For details, refer to "Table 3-27 Description of Elements in avoidstatuspolicyType (page 55)")
neighbor	Neighbor OFS	neighborType (For details, refer to "Table 3-30 Description of Elements in neighborType (page 56)")
neighborpolicy	Link weight value for each path policy	neighborpolicyType (For details, refer to "Table 3-31 Description of Elements in neighborpolicyType (page 56)")

Table 3-29 Description of Elements in avoidstatuspolicyType

Element	Description	Return Value
policyindex0	Status of port exclusion setting for policy 0	{on|off}
policyindex1	Status of port exclusion setting for policy 1	{on|off}
policyindex2	Status of port exclusion setting for policy 2	{on|off}

Element	Description	Return Value
policyindex3	Status of port exclusion setting for policy 3	{on|off}

Table 3-30 Description of Elements in neighborType

Element	Description	Return Value
dp_id	datapath ID	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
linkstatus	Status of connection to neighbor OFS	{up|down}
bcmc_spt_use	Displays whether the link is used as the BCMC transfer spanning tree.	{on|off} on: Used as the BCMC transfer spanning tree. off: Not used as the BCMC transfer spanning tree.
weight	Weight	Decimal number (1 to 4294967295)

Table 3-31 Description of Elements in neighborpolicyType

Element	Description	Return Value
policyindex0	Link weight value for policy 0	Decimal number (1 to 4294967295)
policyindex1	Link weight value for policy 1	Decimal number (1 to 4294967295)
policyindex2	Link weight value for policy 2	Decimal number (1 to 4294967295)
policyindex3	Link weight value for policy 3	Decimal number (1 to 4294967295)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.6 Operate OFS

3.6.1 List OFSs

Following describes how to acquire the list of OFSs managed by the PFC.

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/ofses.xml`

- JSON format

  `/realnetwork/ofses.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ofses>
  [
  <ofs dp_id="dp_id"/>
  ]
</ofses>
```

- JSON format

```
{
  "ofses": [
    {
      "dp_id": "dp_id"
    }
  ]
}
```

Table 3-32 Description of Elements in ofses

Element	Description	Return Value
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)

HTTP status code for response

- On success

  200 (OK)

- On failure

400, 500

Miscellaneous

Log

None

Remark

None

3.6.2 List OFSs with descriptions

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/ofses/description.xml`

- JSON format

  `/realnetwork/ofses/description.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ofses>
  [
  <ofs dp_id="dp_id">
    <description
      manufacture="manufacture"
      hardware="hardware"
      software="software"/>
  </ofs>
  ]
</ofses>
```

- JSON format

```
{
  "ofses" : [
    {
      "dp_id" : "dp_id",
      "description" : {
        "manufacture" : "manufacture",
        "hardware" : "hardware",
```

```
        "software" : "software"
      }
    }
  ]
}
```

Table 3-33 Description of Elements in ofses

Element	Description	Valid Value
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number).
description	Detailed OFS information	description type (For details, refer to "Table 3-34 Description of Elements in descriptionType (page 59)" below.)

Table 3-34 Description of Elements in descriptionType

Element	Description	Valid Value
manufacture	OFS manufacturer information	Character string consisting of up to 255 characters
hardware	Detailed OFS hardware information	Character string consisting of up to 255 characters
software	Detailed OFS software information	Character string consisting of up to 255 characters

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.6.3 Show OFS

Following describes how to acquire the OFS information (excluding statistical information).

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/ofses/dp_id.xml`
- JSON format

  `/realnetwork/ofses/dp_id.json`

Table 3-35 Description of Parameters in URI

Element	Description	Valid Value
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number). It is not permitted to specify F's for all of them.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ofs dp_id="dp_id" ipaddr="ipaddr">
  <ports>
    [
    <port port_name="port_name"/>
    ]
  </ports>
</ofs>
```

- JSON format

```
{
  "ofs": {
    "dp_id": "dp_id",
    "ipaddr": "ipaddr",
    "ports": [
      {
        "port_name": "port_name"
      }
    ]
  }
}
```

Table 3-36 Description of Elements in ofs

Element	Description	Return Value
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
ipaddr	OFS IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ports	Port list	ports type (For details, refer to "Table 3-37 Description of Elements in portsType (page 60)")

Table 3-37 Description of Elements in portsType

Element	Description	Return Value
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

HTTP status code for response

- On success

  200 (OK)

- On failure

400, 500

Miscellaneous

Log

- On failure

  "Get ofs(dpid **dp_id**) info failed. [invalid argument(dp_id)](<Error information from pfcshell>)"

Remark

None

3.6.4 Show OFS details

Following describes how to acquire detailed information including the statistics on the OFS physical ports.

Processing request

Method

GET

request URI

- XML format

  /realnetwork/ofses/**dp_id**/detail.xml

- JSON format

  /realnetwork/ofses/**dp_id**/detail.json

Table 3-38 Description of Parameters in URI

Element	Description	Valid Value
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ofs dp_id="dp_id"
  ipaddr="ipaddr"
  status="status"
  avoidstatus="avoidstatus"
  connectedtime="connectedtime"
  connectedfor="connectedfor">
  <ports>
    [
```

```
    <port port_name="port_name"
      id="id"
      adminstatus="adminstatus"
      operstatus="operstatus"
      avoidstatus="avoidstatus"
      macaddr="macaddr"
      lagchannelgrp="lagchannelgrp">
      <statistics rx_packets="rx_packets"
        rx_bytes="rx_bytes"
        rx_dropped="rx_dropped"
        rx_errors="rx_errors"
        rx_frameerr="rx_frameerr"
        rx_crcerr="rx_crcerr"
        tx_packets="tx_packets"
        tx_bytes="tx_bytes"
        tx_dropped="tx_dropped"
        tx_errors="tx_errors"
        rx_overerr="rx_overerr"
        collisions="collisions"/>
    </port>]
  </ports>
</ofs>
```

- JSON format

```
{
  "ofs": {
    "dp_id": "dp_id",
    "ipaddr": "ipaddr",
    "status": "status",
    "avoidstatus": "avoidstatus",
    "connectedtime": "connectedtime",
    "connectedfor": "connectedfor",
    "ports": [
      {
        "port_name": "port_name",
        "id": "id",
        "adminstatus": "adminstatus",
        "operstatus": "operstatus",
        "avoidstatus": "avoidstatus",
        "macaddr": "macaddr",
        "lagchannelgrp": "lagchannelgrp",
        "statistics": {
          "rx_packets": "rx_packets",
          "rx_bytes": "rx_bytes",
          "rx_dropped": "rx_dropped",
          "rx_errors": "rx_errors",
          "rx_frameerr": "rx_frameerr",
          "rx_crcerr": "rx_crcerr",
          "tx_packets": "tx_packets",
          "tx_bytes": "tx_bytes",
          "tx_dropped": "tx_dropped",
          "tx_errors": "tx_errors",
          "rx_overerr": "rx_overerr",
          "collisions": "collisions"
        }
      }
    ]
  }
}
```

Table 3-39 Description of Elements in ofs

Element	Description	Return Value
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
ipaddr	OFS IP address	IPv4 dot-separated format (Example: 192.168.1.1)

Element	Description	Return Value
status	Connection status of secure channel	{connected|disconnected}
avoidstatus	Status of OFS exclusion setting	{on|off}
connectedtime	When the secure channel was established	YYYY-MM-DD hh:mm:ss format
connectedfor	Elapsed time since establishment of secure channel	Format: DD hh mm ss DD: day hh: hour mm: minute ss: seconds
ports	Port list	ports type (For details, refer to "Table 3-40 Description of Elements in portsType (page 63)")

Table 3-40 Description of Elements in portsType

Element	Description	Return Value
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
id	port ID	Decimal number (1 to 4294967295)
adminstatus	Admin status of port	{up|down} up: Operating down: Operation stopped according to the setting
operstatus	Oper status of port	{up|down} up: Normally operating down: Line failed
avoidstatus	Status of port exclusion setting	{on|off} on: The port is excluded in the real-network mode. off: The port is not excluded in the real-network mode.
macaddr	MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
lagchannelgrp	Channel group number for link aggregation	"none" is displayed if the relevant port is not registered in the channel group.
statistics	Statistical information list	statistics type (For details, refer to "Table 3-41 Description of Elements in statistics Type (page 63)")

Table 3-41 Description of Elements in statistics Type

Element	Description	Return Value
rx_packets	Number of received packets	Counter
rx_bytes	Number of received bytes	Counter
rx_dropped	Number of packets dropped at reception	Counter
rx_errors	Number of receiving errors	Counter
rx_frameerr	Number of receiving frame errors	Counter
rx_crcerr	Number of receiving CRC errors	Counter

Element	Description	Return Value
tx_packets	Number of transferred packets	Counter
tx_bytes	Number of transferred bytes	Counter
tx_dropped	Number of dropped packets	Counter
tx_errors	Number of sending errors	Counter
rx_overerr	Number of reception over errors	Counter
collisions	Number of collisions	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On failure

  `"Get ofs(dpid dp_id) detail info failed. [invalid argument(dp_id)](<Error information from pfcshell>)"`

Remark

None

3.6.5 Show OFS descriptions

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/ofses/dp_id/description.xml`
- JSON format

  `/realnetwork/ofses/dp_id/description.json`

Table 3-42 Description of Parameters in URI

Element	Description	Valid Value
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ofs dp_id="dp_id">
  <description
    manufacture="manufacture"
    hardware="hardware"
    software="software"/>
</ofs>
```

- JSON format

```
{
  "ofs" : {
    "dp_id" : "dp_id",
    "description" : {
      "manufacture" : "manufacture",
      "hardware" : "hardware",
      "software" : "software"
    }
  }
}
```

Table 3-43 Description of Elements in ofs

Element	Description	Valid Value
dp_id	OFSdatapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number).
description	Detailed OFS information	description type (For details, refer to "Table 3-44 Description of Elements in descriptionType (page 65)" below.)

Table 3-44 Description of Elements in descriptionType

Element	Description	Valid Value
manufacture	OFS manufacturer information	Character string consisting of up to 255 characters
hardware	Detailed OFS hardware information	Character string consisting of up to 255 characters
software	Detailed OFS software information	Character string consisting of up to 255 characters

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.7 Operate OFS domain

3.7.1 Show OFS domain

Following describes how to acquire the OFS domain configuration information.

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/ofsdomains.xml`

- JSON format

  `/realnetwork/ofsdomains.json`

- Parameters

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `ofs_domain_name=`**`ofs_domain_name`**

  `ofs_subdomain_name=`**`ofs_subdomain_name`**

Table 3-45 Description of Parameters in URI

Element	Description	Valid Value
ofs_domain_name	OFS domain name	Up to 31 characters including one-byte alphanumeric characters and underscores
ofs_subdomain_name	OFS subdomain name	Up to 31 characters including one-byte alphanumeric characters and underscores

Remember

The ofs_domain_name and ofs_subdomain_name are not required.

If, however, you specify ofs_subdomain_name, you must also specify ofs_domain_name.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ofsdomaininfo>
  <ofsdomains>
    [<ofsdomain
        domain_name="domain_name"
        conn_ofs_count="conn_ofs_count"
        default_ofs_domain="default_ofs_domain">
      <ofscoredomain
          conn_ofs_count="conn_ofs_count">
        <ofses>
          [<ofs
              dp_id="dp_id"
              status="status"/>]
        </ofses>
      </ofscoredomain>
      <ofssubdomains>
        [<ofssubdomain
            subdomain_name="subdomain_name"
            conn_ofs_count="conn_ofs_count"
            path_ctrl_method="path_ctrl_method">
            <gateway
              value="value"
              active_gateway_port_count="active_gateway_port_count">
            <connectedofses>
              [<ofs
                  dp_id="dp_id"
                  port_cnt="port_cnt"
                  active_cnt="active_cnt"/>]
            </connectedofses>
            <disconnectedofses>
              [<ofs
                  dp_id="dp_id"
                  port_cnt="port_cnt"
                  active_cnt="active_cnt"/>]
            </disconnectedofses>
            <configofses>
                [<ofs
                    dp_id="dp_id"
                    port_cnt="port_cnt"
                    active_cnt="active_cnt"/>]
            </configofses>
            <memberofses>
              [<ofs
                  dp_id="dp_id"
                  port_cnt="port_cnt"
                  status="status"/>]
            </memberofses>
            <split>
              <memberofses>
                [<ofs
                    dp_id="dp_id"
                    port_cnt="port_cnt"
                    status="status"/>]
              </memberofses>
            </split>
          </gateway>
          <memberofses>
            [<ofs
                dp_id="dp_id"
                port_cnt="port_cnt"
                status="status">
                <ports>
                  [<port
                      port_name="port_name"
                      status="status"/>]
                </ports>
            </ofs>]
          </memberofses>
        </ofssubdomain>]
      </ofssubdomains>
    </ofsdomain>]
  </ofsdomains>
  <ofsdomainstatistics>
```

```
    <ofsdomaincount
        config_count="config_count"
        connected_count="connected_count"
        disconnected_count="disconnected_count"/>
    <ofssubdomaincount
        config_count="config_count"
        connected_count="connected_count"
        disconnected_count="disconnected_count/>
    <gatewaycount
        config_count="config_count"
        connected_count="connected_count"
        disconnected_count="disconnected_count"/>
  </ofsdomainstatistics>
</ofsdomaininfo>
```

- JSON format

```
{
  "ofsdomaininfo": {
    "ofsdomains": [{
      "domain_name": "domain_name",
      "conn_ofs_count": "conn_ofs_count",
      "default_ofs_domain": "default_ofs_domain",
      "ofscoredomain": {
        "conn_ofs_count": "conn_ofs_count",
        "ofses": [{
          "dp_id": "dp_id",
          "status": "status"
        }]
      },
      "ofssubdomains": [{
        "subdomain_name": "subdomain_name",
        "conn_ofs_count": "conn_ofs_count",
        "path_ctrl_method": "path_ctrl_method",
        "gateway": {
          "value": "value",
          "active_gateway_port_count": "active_gateway_port_count",
          "connectedofses": [{
            "dp_id": "dp_id",
            "port_cnt": "port_cnt",
            "active_cnt": "active_cnt"
          }],
          "disconnectedofses": [{
            "dp_id": "dp_id",
            "port_cnt": "port_cnt",
            "active_cnt": "active_cnt"
          }],
          "configofses": [{
              "dp_id": "dp_id",
              "port_cnt": "port_cnt",
              "active_cnt": "active_cnt"
            }],
          "memberofses": [{
            "dp_id": "dp_id",
            "port_cnt": "port_cnt",
            "status": "status"
          }],
          "split": {
            "memberofses": [{
              "dp_id": "dp_id",
              "port_cnt": "port_cnt",
              "status": "status"
            }]
          }
        },
        "memberofses": [{
          "dp_id": "dp_id",
          "port_cnt": "port_cnt",
          "status": "status",
          "ports": [{
            "port_name": "port_name",
            "status": "status"
```

```
            }]
          }]
        }]
      }],
      "ofsdomainstatistics": {
        "ofsdomaincount": {
          "config_count": "config_count",
          "connected_count": "connected_count",
          "disconnected_count": "disconnected_count"
        },
        "ofssubdomaincount": {
          "config_count": "config_count",
          "connected_count": "connected_count",
          "disconnected_count": "disconnected_count"
        },
        "gatewaycount": {
          "config_count": "config_count",
          "connected_count": "connected_count",
          "disconnected_count": "disconnected_count"
        }
      }
    }
}
```

Table 3-46 Description of Elements in ofsdomaininfo

Element	Description	Return Value
ofsdomains	OFS domain	ofsdomains type (For details, refer to "Table 3-47 Description of Elements in ofsdomains Type (page 69)" below.)
ofsdomainstatistics	OFS domain statistical information	ofsdomainstatistics type (For details, refer to "Table 3-57 Description of Elements in ofsdomainstatistics Type (page 72)" below.)

Table 3-47 Description of Elements in ofsdomains Type

Element	Description	Return Value
domain_name	OFS domain name	Up to 31 characters including one-byte alphanumeric characters and underscores
conn_ofs_count	Number of OFSs which belong to the OFS domain or OFS subdomain and for which "status" is "connected"	Decimal number
default_ofs_domain	Indicates that the domain is the default OFS domain.	{true} (Lowercase)
ofscoredomain	Information of the OFS core domain	ofscoredomain type (For details, refer to "Table 3-48 Description of Elements in ofscoredomain Type (page 69)" below.)
ofssubdomains	Information of OFS subdomain	ofssubdomains type (For details, refer to "Table 3-50 Description of Elements in ofssubdomains Type (page 70)" below.)

Table 3-48 Description of Elements in ofscoredomain Type

Element	Description	Return Value
conn_ofs_count	Number of OFSs which belong to the OFS core domain and for which "status" is "connected"	Decimal number

Element	Description	Return Value
ofses	Information of OFS belonging to the OFS core domain	ofses type (For details, refer to "Table 3-49 Description of Elements in ofses Type (page 70)" below.)

Table 3-49 Description of Elements in ofses Type

Element	Description	Return Value
dp_id	OFS datapath ID	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
status	Secure channel status	{connected|disconnected|config} (Lowercase)

Table 3-50 Description of Elements in ofssubdomains Type

Element	Description	Return Value
subdomain_name	OFS subdomain name	Up to 31 characters including one-byte alphanumeric characters and underscores
conn_ofs_count	Number of OFSs which belong to the OFS domain or OFS subdomain and for which "status" is "connected"	Decimal number
path_ctrl_method	Path control method in the OFS subdomain	{pf-standard|pf-mac-forwarding|pf-extension} (Lowercase)
gateway	OFS subdomain gateway information Not outputted if the path control method is pf-extension.	gateway type (For details, refer to "Table 3-51 Description of Elements in gateway Type (page 70)" below.)
memberofses	OFS belonging to the OFS subdomain Outputted if the path control method is pf-extension.	memberofses type (For details, refer to "Table 3-55 Description of Elements in memberofses Type (page 71)" below.)

Table 3-51 Description of Elements in gateway Type

Element	Description	Return Value
value	existence of gateway	{yes|no} (Lowercase)
active_gateway_port_count	Number of selected active ports of the OFS subdomain gateway. Total value of all gateways in the OFS subdomain.	Decimal number
connectedofses	List of the OFSs which are registered as an OFS subdomain gateway and connected from the secure channel	connectedofses type (For details, refer to "Table 3-52 Parameter Elements in connectedofses, disconnectedofses, configofses (page 71)" below.)
disconnectedofses	List of the OFSs which are registered as an OFS subdomain gateway and disconnected from the secure channel	disconnectedofses type (For details, refer to "Table 3-52 Parameter Elements in connectedofses, disconnectedofses, configofses (page 71)" below.)
configofses	List of the OFSs which are registered as an OFS subdomain gateway and have never been connected to the secure channel	configofses type (For details, refer to "Table 3-52 Parameter Elements in connectedofses, disconnectedofses, configofses (page 71)" below.)
memberofses	OFS which is registered as an OFS subdomain gateway and connected to the OFS subdomain gateway port	memberofses type (For details, refer to "Table 3-54 Description of Elements in memberofses Type (page 71)" below.)

Element	Description	Return Value
split	List of the OFSs which are registered as an OFS subdomain gateway but disconnected from the OFS subdomain gateway ports	split type (For details, refer to "Table 3-53 Description of Elements in split Type (page 71)" below.)

Table 3-52 Parameter Elements in connectedofses, disconnectedofses, configofses

Element	Description	Return Value
dp_id	datapath ID of the OFS registered as a subdomain gateway	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
port_cnt	Number of ports of the OFSs registered as an OFS subdomain gateway	Decimal number or all all: Indicates that all OFS ports belong to the OFS subdomain gateway.
active_cnt	Number of selected active ports of OFS subdomain gateway	Decimal number

Table 3-53 Description of Elements in split Type

Element	Description	Return Value
memberofs es	List of the OFSs which are registered as an OFS subdomain gateway and connected to the OFS subdomain gateway ports	memberofses type (For details, refer to "Table 3-54 Description of Elements in memberofses Type (page 71)" below.)

Table 3-54 Description of Elements in memberofses Type

Element	Description	Return Value
dp_id	datapath ID of the OFS registered in the OFS subdomain	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
port_cnt	Number of ports of the OFSs registered in the OFS subdomain	Decimal number or all all: Indicates that all OFS ports belong to the OFS subdomain gateway.
status	Port connection status of the OFSs registered in the OFS subdomain	{connected|disconnected|config} (Lowercase)

Remember

The "status" is always "connected" because memberofses directly under the gateway type is the OFS connected to the port of the OFS subdomain gateway.

Table 3-55 Description of Elements in memberofses Type

Element	Description	Return Value
dp_id	datapath ID of the OFS registered in the OFS subdomain	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
port_cnt	Number of ports of the OFSs registered in the OFS subdomain	Decimal number or all all: Indicates that all OFS ports belong to the OFS subdomain gateway.
status	Port connection status of the OFSs registered in the OFS subdomain	{connected|disconnected|config} (Lowercase)
ports	List of ports of the OFSs registered in the OFS subdomain	ports type (For details, refer to "Table 3-56 Description of Elements in ports Type (page 72)" below.

Table 3-56 Description of Elements in ports Type

Element	Description	Return Value
port_name	Name of a port of the OFS registered in the OFS subdomain	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
status	Connection status of a port of the OFS registered in the OFS subdomain	{connected|config} (Lowercase)

Table 3-57 Description of Elements in ofsdomainstatistics Type

Element	Description	Return Value
ofsdomaincount	OFS domain statistical information	ofsdomaincount type (For details, refer to "Table 3-58 Description of Elements in ofsdomaincount, ofssubdomaincount, gatewaycount Type (page 72)" below.)
ofssubdomaincount	OFS subdomain statistical information	ofssubdomaincount type (For details, refer to "Table 3-58 Description of Elements in ofsdomaincount, ofssubdomaincount, gatewaycount Type (page 72)" below.)
gatewaycount	OFS subdomain gateway statistical information	gatewaycount type (For details, refer to "Table 3-58 Description of Elements in ofsdomaincount, ofssubdomaincount, gatewaycount Type (page 72)" below.)

Table 3-58 Description of Elements in ofsdomaincount, ofssubdomaincount, gatewaycount Type

Element	Description	Return Value
config_count	Number of OFSs to which the secure channel has never been connected	Decimal number
connected_count	Number of OFSs to which the secure channel is connected Decimal	Decimal number
disconnected_count	Number of OFSs from which the secure channel is disconnected	Decimal number

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On failure

  "Get ofsdomain **ofs_domain_name** of ofssubdomain **ofs_subdomain_name** of real-network failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

3.7.2 Show OFS domain details

Following describes how to acquire the detailed OFS domain.

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/ofsdomains/detail.xml`

- JSON format

  `/realnetwork/ofsdomains/detail.json`

- Parameters

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `ofs_domain_name=`**`ofs_domain_name`**

  `ofs_subdomain_name=`**`ofs_subdomain_name`**

Table 3-59 Description of Parameters in URI

Element	Description	Valid Value
ofs_domain_name	OFS domain name	Up to 31 characters including one-byte alphanumeric characters and underscores
ofs_subdomain_name	OFS subdomain name	Up to 31 characters including one-byte alphanumeric characters and underscores

Remember

The ofs_domain_name and ofs_subdomain_name are not required.

If, however, you specify ofs_subdomain_name, you must also specify ofs_domain_name.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ofsdomaininfo>
  <ofsdomains>
    [<ofsdomain
```

```
        domain_name="domain_name"
        conn_ofs_count="conn_ofs_count"
        default_ofs_domain="default_ofs_domain"
        domain_creation_time="domain_creation_time"
        last_member_config_time="last_member_config_time"
        last_member_event_time="last_member_event_time">
      <ofscoredomain
          conn_ofs_count="conn_ofs_count">
        <ofses>
          [<ofs
              dp_id="dp_id"
              status="status">
            <ports>
              [<port
                  port_name="port_name"
                  status="status">
                <ofssubdomaingateways>
                  [<ofssubdomaingateway
                      subdomain_name="subdomain_name"
                      active="active">
                    <neighbors>
                      [<neighbor
                          dp_id="dp_id"
                          port_name="port_name"/>]
                    </neighbors>
                  </ofssubdomaingateway>]
                </ofssubdomaingateways>
              </port>]
            </ports>
          </ofs>]
        </ofses>
      </ofscoredomain>
      <ofssubdomains>
        [<ofssubdomain
            subdomain_name="subdomain_name"
            conn_ofs_count="conn_ofs_count"
            subdomain_creation_time="subdomain_creation_time"
            last_member_config_time="last_member_config_time"
            last_member_event_time="last_member_event_time"
            path_ctrl_method="path_ctrl_method">
          <gateway
              value="value"
              gateway_creation_time="gateway_creation_time"
              last_gateway_config_time="last_gateway_config_time"
              last_gateway_event_time="last_gateway_event_time"
              active_gateway_port_count="active_gateway_port_count">
            <connectedofses>
              [<ofs
                  dp_id="dp_id"
                  port_cnt="port_cnt"
                  active_cnt="active_cnt"
                  priority="priority">
                <ports>
                  [<port
                      port_name="port_name"
                      status="status"
                      active="active"
                      priority="priority">
                    <neighbors>
                      [<neighbor
                          dp_id="dp_id"
                          port_name="port_name"/>]
                    </neighbors>
                    <memberofses>
                      [<ofs
                          dp_id="dp_id"
                          port_cnt="port_cnt"
                          status="status">
                        <ports>
                          [<port
                              port_name="port_name"
                              active="active">
                            <neighbors>
```

```
                        [<neighbor
                            dp_id="dp_id"
                            port_name="port_name"/>]
                    </neighbors>
                  </port>]
              </ports>
            </ofs>]
          </memberofses>
        </port>]
      </ports>
    </ofs>]
  </connectedofses>
  <disconnectedofses>
    [<ofs
        dp_id="dp_id"
        port_cnt="port_cnt"
        active_cnt="active_cnt"
        priority="priority">
      <ports>
        [<port
            port_name="port_name"
            status="status"
            active="active"
            priority="priority">
          <neighbors>
            [<neighbor
                dp_id="dp_id"
                port_name="port_name"/>]
          </neighbors>
          <memberofses>
            [<ofs
                dp_id="dp_id"
                port_cnt="port_cnt"
                status="status">
              <ports>
                [<port
                    port_name="port_name"
                    active="active">
                  <neighbors>
                    [<neighbor
                        dp_id="dp_id"
                        port_name="port_name"/>]
                  </neighbors>
                </port>]
              </ports>
            </ofs>]
          </memberofses>
        </port>]
      </ports>
    </ofs>]
  </disconnectedofses>
  <configofses>
      [<ofs
          dp_id="dp_id"
          port_cnt="port_cnt"
          active_cnt="active_cnt">
        <ports>
          [<port
              port_name="port_name"
              status="status"
              active="active"
              priority="priority"/>]
        </ports>
      </ofs>]
  </configofses>
  <split>
    <memberofses>
      [<ofs
          dp_id="dp_id"
          port_cnt="port_cnt"
          status="status">
        <ports>
          [<port
```

```
                        port_name="port_name"
                        status="status"/>]
                    </ports>
                  </ofs>]
                </memberofses>
              </split>
            </gateway>
            <memberofses>
              [<ofs
                  dp_id="dp_id"
                  port_cnt="port_cnt"
                  status="status">
                <ports>
                  [<port
                      port_name="port_name"
                      status="status"/>]
                </ports>
              </ofs>]
            </memberofses>
            <vtnsubdomainmapping
                mapping="mapping">
              <vtns>
                [<vtn
                    vtn_name="vtn_name">
                  <mappings>
                    [<mapping
                        vbr_name="vbr_name"
                        vlan_id="vlan_id"/>]
                  </mappings>
                </vtn>]
              </vtns>
            </vtnsubdomainmapping>
          </ofssubdomain>]
        </ofssubdomains>
      </ofsdomain>]
    </ofsdomains>
    <ofsdomainstatistics>
      <ofsdomaincount
          config_count="config_count"
          connected_count="connected_count"
          disconnected_count="disconnected_count"/>
      <ofssubdomaincount
          config_count="config_count"
          connected_count="connected_count"
          disconnected_count="disconnected_count"/>
      <gatewaycount
          config_count="config_count"
          connected_count="connected_count"
          disconnected_count="disconnected_count"/>
    </ofsdomainstatistics>
  </ofsdomaininfo>
```

- JSON format

```
{
  "ofsdomaininfo": {
    "ofsdomains": [{
      "domain_name": "domain_name",
      "conn_ofs_count": "conn_ofs_count",
      "default_ofs_domain": "default_ofs_domain",
      "domain_creation_time": "domain_creation_time",
      "last_member_config_time": "last_member_config_time",
      "last_member_event_time": "last_member_event_time",
      "ofscoredomain": {
        "conn_ofs_count": "conn_ofs_count",
        "ofses": [{
          "dp_id": "dp_id",
          "status": "status",
          "ports": [{
            "port_name": "port_name",
            "status": "status",
            "ofssubdomaingateways": [{
```

```
            "subdomain_name": "subdomain_name",
            "active": "active",
            "neighbors": [{
              "dp_id": "dp_id",
              "port_name": "port_name"
            }]
          }]
        }]
      }]
    },
    "ofssubdomains": [{
      "subdomain_name": "subdomain_name",
      "conn_ofs_count": "conn_ofs_count",
      "subdomain_creation_time": "subdomain_creation_time",
      "last_member_config_time": "last_member_config_time",
      "last_member_event_time": "last_member_event_time",
      "path_ctrl_method": "path_ctrl_method",
      "gateway": {
        "value": "value",
        "gateway_creation_time": "gateway_creation_time",
        "last_gateway_config_time": "last_gateway_config_time",
        "last_gateway_event_time": "last_gateway_event_time",
        "active_gateway_port_count": "active_gateway_port_count",
        "connectedofses": [{
          "dp_id": "dp_id",
          "port_cnt": "port_cnt",
          "active_cnt": "active_cnt",
          "priority": "priority",
          "ports": [{
            "port_name": "port_name",
            "status": "status",
            "active": "active",
            "priority": "priority",
            "neighbors": [{
              "dp_id": "dp_id",
              "port_name": "port_name"
            }]
            "memberofses": [{
              "dp_id": "dp_id",
              "port_cnt": "port_cnt",
              "status": "status",
              "ports": [{
                "port_name": "port_name",
                "active": "active",
                "neighbors": [{
                  "dp_id": "dp_id",
                   "port_name": "port_name"
                }]
              }]
            }]
          }]
        }],
        "disconnectedofses": [{
          "dp_id": "dp_id",
          "port_cnt": "port_cnt",
          "active_cnt": "active_cnt",
          "priority": "priority",
          "ports": [{
            "port_name": "port_name",
            "status": "status",
            "active": "active",
            "priority": "priority",
            "neighbors": [{
              "dp_id": "dp_id",
              "port_name": "port_name"
            }]
            "memberofses": [{
              "dp_id": "dp_id",
              "port_cnt": "port_cnt",
              "status": "status",
              "ports": [{
                "port_name": "port_name",
                "active": "active",
```

```
                  "neighbors": [{
                    "dp_id": "dp_id",
                    "port_name": "port_name"
                  }]
                }]
              }]
            }]
          }],
          "configofses": [{
              "dp_id": "dp_id",
              "port_cnt": "port_cnt",
              "active_cnt": "active_cnt",
              "priority": "priority",
              "ports": [{
                "port_name": "port_name",
                "status": "status",
                "active": "active",
                "priority": "priority"
              }]
            }],
          "split": {
            "memberofses": [{
              "dp_id": "dp_id",
              "port_cnt": "port_cnt",
              "status": "status",
              "ports": [{
                "port_name": "port_name",
                "status": "status"
              }]
            }]
          }
        },
        "memberofses": [{
          "dp_id": "dp_id",
          "port_cnt": "port_cnt",
          "status": "status",
          "ports": [{
            "port_name": "port_name",
            "status": "status"
          }]
        }],
        "vtnsubdomainmapping": {
          "mapping": "mapping",
          "vtns": [{
            "vtn_name": "vtn_name",
            "mappings": [{
              "vbr_name": "vbr_name",
              "vlan_id": "vlan_id"
            }]
          }]
        }
      }]
    }],
    "ofsdomainstatistics": {
      "ofsdomaincount": {
        "config_count": "config_count",
        "connected_count": "connected_count",
        "disconnected_count": "disconnected_count"
      },
      "ofssubdomaincount": {
        "config_count": "config_count",
        "connected_count": "connected_count",
        "disconnected_count": "disconnected_count"
      },
      "gatewaycount": {
        "config_count": "config_count",
        "connected_count": "connected_count",
        "disconnected_count": "disconnected_count"
      }
    }
  }
}
```

Table 3-60 Description of Elements in ofsdomaininfo

Element	Description	Return Value
ofsdomains	OFS domain	ofsdomains type (For details, refer to "Table 3-61 Description of Elements in ofsdomains Type (page 79)" below.)
ofsdomainstatistics	OFS domain statistical information	ofsdomainstatistics type (For details, refer to "Table 3-81 Description of Elements in ofsdomainstatistics Type (page 85)" below.)

Table 3-61 Description of Elements in ofsdomains Type

Element	Description	Return Value
domain_name	OFS domain name	Up to 31 characters including one-byte alphanumeric characters and underscores
conn_ofs_count	Number of OFSs which belong to the OFS domain or OFS subdomain and for which "status" is "connected"	Decimal number
default_ofs_domain	Indicates that the domain is the default OFS domain.	{true} (Lowercase)
domain_creation_time	Date when the OFS domain was created	Format of YYYY-MM-DD hh:mm:ss
last_member_config_time	Last date when a member was registered/deleted to/from the OFS domain by using the member command	Format of YYYY-MM-DD hh:mm:ss
last_member_event_time	Last date when the status of the OFS registered in the OFS domain was changed	Format of YYYY-MM-DD hh:mm:ss
ofscoredomain	Information of the OFS core domain	ofscoredomain type (For details, refer to "Table 3-62 Description of Elements in ofscoredomain Type (page 79)" below.)
ofssubdomains	Information of OFS subdomain	ofssubdomains type (For details, refer to "Table 3-67 Description of Elements in ofssubdomains Type (page 80)" below.)

Table 3-62 Description of Elements in ofscoredomain Type

Element	Description	Return Value
conn_ofs_count	Number of OFSs which belong to the OFS core domain and for which "status" is "connected"	Decimal number
ofses	Information of OFS belonging to the OFS core domain	ofses type (For details, refer to "Table 3-63 Description of Elements in ofses Type (page 79)" below.)

Table 3-63 Description of Elements in ofses Type

Element	Description	Return Value
dp_id	OFS datapath ID	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
status	Secure channel status	{connected|disconnected|config} (Lowercase)
ports	OFS port	ports type (For details, refer "Table 3-64 Description of Elements in ports Type (page 80)" below.)

Table 3-64 Description of Elements in ports Type

Element	Description	Return Value
port_name	OFS port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
status	Connection status of OFS port	{connected|config} (Lowercase)
ofssubdomaingateways	OFS subdomain gateway information	ofssubdomaingateways type (For details, refer to "Table 3-65 Description of Elements in ofssubdomaingateways Type (page 80)" below.)

Table 3-65 Description of Elements in ofssubdomaingateways Type

Element	Description	Return Value
subdomain _name	Name of the connected subdomain when the OFS port is connected as an OFS subdomain gateway	Up to 31 characters including one-byte alphanumeric characters and underscores
active	Indicates that the OFS port is selected as an active port.	{true} (Lowercase)
neighbors	Information of the neighbor OFS connected to the OFS subdomain gateway port	neighbors type (For details, refer to "Table 3-66 Description of Elements in neighbors Type (page 80)" below.)

Table 3-66 Description of Elements in neighbors Type

Element	Description	Return Value
dp_id	datapath ID of the neighbor OFS connected to the OFS subdomain gateway port	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
port_name	Port name of the neighbor OFS connected to the OFS subdomain gateway port	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

Table 3-67 Description of Elements in ofssubdomains Type

Element	Description	Return Value
subdomain_name	OFS subdomain name	Up to 31 characters including one-byte alphanumeric characters and underscores
conn_ofs_count	Number of OFSs which belong to the OFS domain or OFS subdomain and for which "status" is "connected"	Decimal number
subdomain_creation_time	Date when the OFS subdomain was created	Format of YYYY-MM-DD hh:mm:ss
last_member_config_time	Last date when the status of the OFS registered in the OFS subdomain was changed	Format of YYYY-MM-DD hh:mm:ss
last_member_event_time	Last date when the status of the OFS registered in the OFS subdomain was changed	Format of YYYY-MM-DD hh:mm:ss

Element	Description	Return Value
path_control_method	Path control method in the OFS subdomain	{pf-standard|pf-mac-forwarding|pf-extension} (Lowercase)
gateway	OFS subdomain gateway information	gateway type (For details, refer to "Table 3-68 Description of Elements in gateway Type (page 81)" below.)
memberofses	OFS belonging to the OFS subdomain Not displayed if the path control method is "PF extension".	memberofses type (For details, refer to "Table 3-76 Description of Elements in memberofses Type (page 84)" below.)
vtnsubdomainmapping	VTN mapping information of the OFS subdomain	vtnsubdomainmapping type (For details, refer to "Table 3-78 Description of Elements in vtnsubdomainmapping Type (page 84)" below.)

Table 3-68 Description of Elements in gateway Type

Element	Description	Return Value
value	existence of gateway	{yes|no} (Lowercase)
gateway_creation_time	Date when the first OFS subdomain gateway was added	Format of YYYY-MM-DD hh:mm:ss
last_gateway_config_time	Last date when the OFS subdomain gateway was added by the gateway command	Format of YYYY-MM-DD hh:mm:ss
last_gateway_event_time	Last date when the status of the registered OFS or OFS port was changed	Format of YYYY-MM-DD hh:mm:ss
active_gateway_port_count	Number of selected active ports of the OFS subdomain gateway. Total value of all gateways in the OFS subdomain.	Decimal number
connectedofses	List of the OFSs which are registered as an OFS subdomain gateway and connected to the secure channel	connectedofses type (For details, refer to "Table 3-69 Description of Elements in connectedofses, disconnectedofses Type (page 81)" below.)
disconnectedofses	List of the OFSs which are registered as an OFS subdomain gateway but disconnected from the secure channel	disconnectedofses type (For details, refer to "Table 3-69 Description of Elements in connectedofses, disconnectedofses Type (page 81)" below.)
configofses	List of the OFSs which are registered as an OFS subdomain gateway and have never been connected to the secure channel	configofses type (For details, refer to "Table 3-73 Description of Elements in configofses Type (page 83)" below.)
split	List of the OFSs which are registered as an OFS subdomain gateway but disconnected from the OFS subdomain gateway ports	split type (For details, refer to "Table 3-75 Description of Elements in split Type (page 83)" below.)

Table 3-69 Description of Elements in connectedofses, disconnectedofses Type

Element	Description	Return Value
dp_id	datapath ID of the OFS registered as a OFS subdomain gateway	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
port_cnt	Number of ports of the OFSs registered as an OFS subdomain gateway	Decimal number or all all: Indicates that all OFS ports belong to the OFS subdomain gateway.

Element	Description	Return Value
active_cnt	Number of selected active ports of OFS subdomain gateway	Decimal number
priority	Priority of the OFS port registered as an OFS subdomain gateway	Decimal number (0 to 255)
ports	Information of the OFS port registered as an OFS subdomain gateway	ports type (For details, refer to "Table 3-70 Description of Elements in ports Type (page 82)" below.)

Table 3-70 Description of Elements in ports Type

Element	Description	Return Value
port_name	Name of a port of the OFSs registered as an OFS subdomain gateway	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
status	Connection status of the OFS port registered as an OFS subdomain gateway	{connected|config} (Lowercase)
active	Indicates that the OFS port is selected as an active port.	{true} (Lowercase)
priority	Priority of the OFS port registered as an OFS subdomain gateway	Decimal number (0 to 255) If the priority is not set when specifying the OFS port setting, 128 is assumed.
neighbors	Information of the neighbor OFS connected to the OFS subdomain gateway port	neighbors type (For details, refer to "Table 3-66 Description of Elements in neighbors Type (page 80)" below.)
memberofses	OFS registered as an OFS subdomain gateway	memberofses type (For details, refer to "Table 3-71 Description of Elements in memberofses Type (page 82)" below.)

Table 3-71 Description of Elements in memberofses Type

Element	Description	Return Value
dp_id	datapath ID of the OFS registered as an OFS belonging to the OFS subdomain gateway	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
port_cnt	Number of OFS ports registered as an OFS port belonging to the OFS subdomain gateway	Decimal number or all all: Indicates that all OFS ports belong to the OFS subdomain gateway.
status	Connection status of the OFS port registered as an OFS port belonging to the OFS subdomain gateway	{connected|disconnected|config} (Lowercase)
ports	Information of the OFS port registered as an OFS subdomain gateway	ports type (For details, refer to "Table 3-72 Description of Elements in ports Type (page 83)" below.)

Table 3-72 Description of Elements in ports Type

Element	Description	Return Value
port_name	Name of the OFS port registered as an OFS belonging to the OFS subdomain gateway	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
active	Indicates that the OFS port is selected as an active port.	{true} (Lowercase)

Element	Description	Return Value
neighbors	Information of the OFS that is an OFS subdomain gateway port and connected to the neighbor OFS	neighbors type (For details, refer to "Table 3-66 Description of Elements in neighbors Type (page 80)" below.)

Table 3-73 Description of Elements in configofses Type

Element	Description	Return Value
dp_id	datapath ID of the OFS registered as a subdomain gateway	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
port_cnt	Number of ports of the OFSs registered as an OFS subdomain gateway	Decimal number or all all: Indicates that all OFS ports belong to the OFS subdomain gateway.
active_cnt	Number of selected active ports of OFS subdomain gateway	Decimal number
priority	Priority of the OFS port registered as an OFS subdomain gateway	Decimal number (0 to 255)
ports	Information of the OFS port registered as an OFS subdomain gateway	ports type (For details, refer to "Table 3-74 Description of Elements in ports Type (page 83)" below.)

Table 3-74 Description of Elements in ports Type

Element	Description	Return Value
port_name	Port name of OFS registered in the OFS subdomain	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
status	Connection status of the OFS port registered as an OFS port belonging to the OFS subdomain gateway	{connected|config} (Lowercase)
active	Indicates that the OFS port is selected as an active port.	{true} (Lowercase)
priority	Priority of the OFS port registered as an OFS subdomain gateway	Decimal number (0 to 255) If the priority is not set when specifying the OFS port setting, 128 is assumed.

Table 3-75 Description of Elements in split Type

Element	Description	Return Value
memberofses	List of the OFSs which are registered as an OFS subdomain gateway but disconnected from the OFS subdomain gateway ports	memberofses type (For details, refer to "Table 3-76 Description of Elements in memberofses Type (page 84)" below.)

Table 3-76 Description of Elements in memberofses Type

Element	Description	Return Value
dp_id	datapath ID of the OFS registered as a subdomain gateway	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
port_cnt	Number of ports of the OFSs registered as an OFS subdomain gateway	Decimal number or all all: Indicates that all OFS ports belong to the OFS subdomain gateway.

Element	Description	Return Value
status	Connection status of the OFS port registered as an OFS subdomain gateway	{connected|disconnected|config} (Lowercase)
ports	Information of the OFS port registered as an OFS subdomain gateway	ports type (For details, refer to "Table 3-77 Description of Elements in ports Type (page 84)" below.)

Table 3-77 Description of Elements in ports Type

Element	Description	Return Value
port_name	Port name of OFS registered in the OFS subdomain	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
status	Connection status of the OFS port registered as an OFS port belonging to the OFS subdomain gateway	{connected|config} (Lowercase)

Table 3-78 Description of Elements in vtnsubdomainmapping Type

Element	Description	Return Value
mapping	Indicates whether the VTN information is mapped.	{yes|no} (Lowercase)
vtns	VTN information	vtns type (For details, refer to "Table 3-79 Description of Elements in vtns Type (page 84)" below.)

Table 3-79 Description of Elements in vtns Type

Element	Description	Return Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
mappings	VTN mapping information	mappings type (For details, refer to "Table 3-80 Description of Elements in mappings Type (page 84)" below.)

Table 3-80 Description of Elements in mappings Type

Element	Description	Return Value
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
vlan_id	VLAN ID corresponding to vBridge	Decimal number (1 to 4095)

Table 3-81 Description of Elements in ofsdomainstatistics Type

Element	Description	Return Value
ofsdomaincount	OFS domain statistical information	ofsdomaincount type (For details, refer to "Table 3-82 Description of Elements in ofsdomaincount, ofssubdomaincount, gatewaycount Type (page 85)" below.)
ofssubdomaincount	OFS subdomain statistical information	ofssubdomaincount type (For details, refer to "Table 3-82 Description of Elements in ofsdomaincount, ofssubdomaincount, gatewaycount Type (page 85)" below.)
gatewaycount	OFS subdomain gateway statistical information	gatewaycount type (For details, refer to "Table 3-82 Description of Elements in ofsdomaincount, ofssubdomaincount, gatewaycount Type (page 85)" below.)

Table 3-82 Description of Elements in ofsdomaincount, ofssubdomaincount, gatewaycount Type

Element	Description	Return Value
config_count	Number of OFSs that have only been registered, but have not yet connected to the secure channel	Decimal number
connected_count	Number of OFSs to which secure channel is connected	Decimal number
disconnected_count	Number of OFSs from which the secure channel is disconnected	Decimal number

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On failure

  "Get ofsdomain **ofs_domain_name** detail of ofssubdomain **ofs_subdomain_name** of real-network failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

3.8 Operate VTN Station Information

3.8.1 Show VTN stations

Following describes how to acquire the VTN station information.

Processing request

Method

GET

request URI

- XML format

  `/vtnstations.xml`
- JSON format

  `/vtnstations.json`
- Parameter

The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

`macaddr=`**`macaddr`**

`ipaddr=`**`ipaddr`**

`dp_id=`**`dp_id`**

`port_name=`**`port_name`**

`vlan_id=`**`vlan_id`**

`vtn_name=`**`vtn_name`**

Table 3-83 Description of Parameters in URI

Element	Description	Valid Value
macaddr	MACS address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
ipaddr	IP address	Decimal number (1 to 4095)
dp_id	datapath ID	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number) It is not permitted to specify F's for all of them.
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Remember

If port_name contains the following symbols, they must be percent-encoded (converted to "%xx") on the Web client.

/ < > % , { }

Remember

The parameter in the query character string can be omitted.

Remember

Specify dp_id when specifying port_name.

Remember

Specify dp_id when specifying vlan_id.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vtnstations>
  [
  <vtnstation stationid="stationid"
    createdtime="createdtime"
    macaddr="macaddr"
    ipaddr="[ipaddr]"
    ipv6addr="[ipv6addr]"
    maptype="maptype"
    mapstatus="mapstatus"
    vtn_name="vtn_name"
    vex_name="vex_name"
    vexternalstatus="vexternalstatus"
    dp_id="dp_id"
    port_name="port_name"
    vlan_id="vlan_id"/>
  ]
</vtnstations>
```

- JSON format

```
{
  "vtnstations" : [
    {
      "stationid" : "stationid",
      "createdtime" : "createdtime",
      "macaddr" : "macaddr",
      "ipaddr" : [ "ipaddr" ],
      "ipv6addr" : [ "ipv6addr" ],
      "maptype" : "maptype",
      "mapstatus" : "mapstatus",
      "vtn_name" : "vtn_name",
      "vex_name" : "vex_name",
      "vexternalstatus" : "vexternalstatus",
      "dp_id" : "dp_id",
      "port_name" : "port_name",
      "vlan_id" : "vlan_id"
    }
  ]
}
```

- On error
  - Insufficient information is returned from pfcshell (VTN station ID)

    `"*** Error:Internal error. (no vtn station ID)"`
  - Insufficient information is returned from pfcshell (Created time)

    `"*** Error:Internal error. (no created time)"`

Table 3-84 Description of Elements in vtnstations

Element	Description	Return Value
stationid	StationID	Decimal number (1 to 4095)
createdtime	Generation time	YYYY-MM-DD hh:mm:ss format
macaddr	MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipv6addr	IPv6 address	The IPv6 format conforms to RFC 5952.
maptype	vExternal mapping type	{ofs-map|vlan-map}
mapstatus	Whether mapping is enabled or disabled	{valid|invalid}

Element	Description	Return Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
vexternalstatus	vExternal status	{up|down}
dp_id	datapath ID	hhhh-hhhh-hhhh-hhhh format (h: Hexadecimal number)
port_name	port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On failure

  `"Get vtn station info failed. [Invalid argument(parameter subdivision)](<Error information from pfcshell>)"`

Remark

None

3.8.2 Show VTN stations details

Following describes how to acquire detailed information including statistical information about the VTN station.

Processing request

Method

GET

request URI

- XML format

  `/vtnstations/detail.xml`

- JSON format

  `/vtnstations/detail.json`

- Parameter

The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

`macaddr=`**`macaddr`**

`ipaddr=`**`ipaddr`**

`dp_id=`**`dp_id`**

`port_name=`**`port_name`**

`vlan_id=`**`vlan_id`**

`vtn_name=`**`vtn_name`**

Table 3-85 Description of Parameters in URI

Element	Description	Valid Value
macaddr	MACS Address	hhhh.hhhh.hhhh format (h: hexadecimal number) "0000.0000.0000," "ffff.ffff.ffff," or any multicast MAC address cannot be specified.
ipaddr	IP Address	IPv4 dot-separated format (Example: 192.168.1.1)
dp_id	datapath ID	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Remember

If port_name contains the following symbols, they must be percent-encoded (converted to "%xx") on the Web client.

/ < > % , { }

Remember

The parameter in the query character string can be omitted.

Remember

Specify dp_id when specifying port_name.

Remember

Specify dp_id when specifying vlan_id.

Settings of request body

none

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vtnstations>
  [
  <vtnstation stationid="stationid"
    createdtime="createdtime"
    macaddr="macaddr"
    ipaddr="[ipaddr]"
    ipv6addr="[ipv6addr]"
    maptype="maptype"
    mapstatus="mapstatus"
    vtn_name="vtn_name"
    vex_name="vex_name"
    vexternalstatus="vexternalstatus"
    dp_id="dp_id"
    port_name="port_name"
    vlan_id="vlan_id">
    <statistics>
      <openflowcontroller>
        <allrx packets="packets" octets="octets"/>
        <alltx packets="packets" octets="octets"/>
      </openflowcontroller>
      <openflownw>
        <allrx packets="packets" octets="octets"/>
        <alltx packets="packets" octets="octets"/>
        <existingrx packets="packets" octets="octets"/>
        <expiredrx packets="packets" octets="octets"/>
        <existingtx packets="packets" octets="octets"/>
        <expiredtx packets="packets" octets="octets"/>
        <alldroprx packets="packets" octets="octets"/>
        <existingdroprx packets="packets" octets="octets"/>
        <expireddroprx packets="packets" octets="octets"/>
      </openflownw>
    </statistics>
  </vtnstation>
  ]
</vtnstations>
```

- JSON format

```
{
  "vtnstations" : [
    {
      "stationid" : "stationid",
      "createdtime" : "createdtime",
      "macaddr" : "macaddr",
      "ipaddr" : ["ipaddr"],
      "ipv6addr" : ["ipv6addr"],
      "maptype" : "maptype",
      "mapstatus" : "mapstatus",
      "vtn_name" : "vtn_name",
      "vex_name" : "vex_name",
      "vexternalstatus" : "vexternalstatus",
      "dp_id" : "dp_id",
      "port_name" : "port_name",
      "vlan_id" : "vlan_id",
      "statistics" : {
        "openflowcontroller" : {
          "allrx" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "alltx" : {
            "packets" : "packets",
            "octets" : "octets"
          }
        },
        "openflownw" : {
          "allrx" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "alltx" : {
```

```
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingrx" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expiredrx" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingtx" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expiredtx" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "alldroprx" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingdroprx" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expireddroprx" : {
            "packets" : "packets",
            "octets" : "octets"
          }
        }
      }
    }
  ]
}
```

- On error
  - Insufficient information is returned from pfcshell (VTN station ID)

    `"*** Error:Internal error. (no vtn station ID)"`
  - Insufficient information is returned from pfcshell (Created time)

    `"*** Error:Internal error. (no created time)"`

Table 3-86 Description of Elements in vtnstations

Element	Description	Return Value
stationid	StationID	Decimal number (1 to 4095)
createdtime	Generation time	YYYY-MM-DD hh:mm:ss format
macaddr	MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipv6addr	IPv6 address	The IPv6 format conforms to RFC 5952.
maptype	vExternal mapping type	{ofs-map|vlan-map}
mapstatus	Whether mapping is enabled or disabled	{valid|invalid}
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Return Value
vexternalstatus	vExternal status	{up|down}
dp_id	datapath ID	hhhh.hhhh.hhhh.hhhh format (h: Hexadecimal number)
port_name	port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
statistics	Statistical information list	statistics (For details, refer to "Table 3-87 Description of Elements in statistics (page 92)".)

Table 3-87 Description of Elements in statistics

Element	Description	Valid Value
openflowcontroller	openflowcontroller statistical information	openflowcontroller type (For details, refer to "Table 3-88 Description of Elements in openflowcontroller (page 92)" below.)
openflownw	openflownw statistical information	openflownw type (For details, refer to "Table 3-89 Description of Elements in openflownw (page 92)")

Table 3-88 Description of Elements in openflowcontroller

Element	Description	Valid Value
allrx	Total number of receptions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")
alltx	Total number of transmissions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")

Table 3-89 Description of Elements in openflownw

Element	Description	Valid Value
allrx	Total number of receptions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")
alltx	Total number of transmissions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")
existingrx	Number of existing receptions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")
expiredrx	Number of expired receptions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")
existingtx	Number of existing transmissions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")
expiredtx	Number of expired transmissions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")
alldroprx	Total number of packets dropped at receptions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")
existingdroprx	Number of existing dropped receptions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")
expireddroprx	Number of expired dropped receptions	statistics type (For details, refer to "Table 3-90 Description of Elements in statistics (page 93)")

Table 3-90 Description of Elements in statistics

Element	Description	Valid Value
octets	Number of bytes	Counter
packets	Number of packets	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On failure

  `"Get vtn station detail info failed. [Invalid argument(parameter sub division)](<Error information from pfcshell>)"`

Remark

None

3.8.3 Show count of VTN stations

Following describes how to acquire the number of VTN stations.

Processing request

Method

GET

request URI

- XML format

  `/vtnstations/count.xml`
- JSON format

  `/vtnstations/count.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<count value="value"/>
```

- JSON format

```
{
  "count" : {
    "value" : "value"
  }
}
```

- On error
  - Insufficient information is returned from pfcshell (VTN station count)

    `"*** Error:Internal error. (no vtn station count)"`

Table 3-91 Description of Elements in count

Element	Description	Return Value
value	Number of VTNStation	Decimal number (1 to 524287)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.9 Operate Trunk Port

3.9.1 List trunk ports

Following describes how to acquire the trunk port list.

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/trunkports.xml`

- JSON format

  `/realnetwork/trunkports.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<trunkports>
  [
  <trunkport trunk_port_name="trunk_port_name"/>
  ]
</trunkports>
```

- JSON format

```
{
  "trunkports" : [
    {
      "trunk_port_name" : "trunk_port_name"
    }
  ]
}
```

Table 3-92 Description of Elements in trunkports

Element	Description	Return Value
trunk_port_name	trunkport name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.9.2 Show trunk port

Following describes how to acquire the trunk port information.

Processing request

Method

GET

request URI

- XML format

  /realnetwork/trunkports/**trunk_port_name**.xml

- JSON format

  /realnetwork/trunkports/**trunk_port_name**.json

Table 3-93 Description of Parameters in URI

Element	Description	Valid Value
trunk_port_name	Trunkport name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<trunk_port trunk_port_name="trunk_port_name">
  <member_ports>
    [
    <member_port dp_id="dp_id"
      port_name="port_name"/>
    ]
  </member_ports>
  <stack_link_ports>
    [
    <stack_link_port dp_id="dp_id"
      port_name="port_name"
      direction="direction"/>
    ]
  </stack_link_ports>
</trunk_port>
```

- JSON format

```
  "trunk_port" : {
    "trunk_port_name" : "trunk_port_name",
    "member_ports" : [
      {
        "dp_id" : "dp_id",
        "port_name" : "port_name"
      }
    ],
    "stack_link_ports" : [
      {
        "dp_id" : "dp_id",
        "port_name" : "port_name",
        "direction" : "direction"
```

```
        }
      ]
    }
  }
```

- On error
  - Insufficient information is returned from pfcshell (trunk-port)

    `"*** Error:Internal error. (no trunk-port)"`

Table 3-94 Description of Elements in trunk_port

Element	Description	Return Value
trunk_port_name	Trunkport name	Up to 31 characters including one-byte alphanumeric characters and underscores
member_ports	OFS physical port	member_port Type (For details, refer to "Table 3-95 Description of Elements in member_ports (page 98)" below.)
stack_link_ports	stack link OFS port	stack_link_port Type (For details, refer to "Table 3-96 Description of Elements in stack_link_ports Type (page 98)" below.)

Table 3-95 Description of Elements in member_ports

Element	Description	Return Value
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

Table 3-96 Description of Elements in stack_link_ports Type

Element	Description	Return Value
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
direction	Data transfer direction of stack link	{east|west}

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.10 Operate Trunk Port Group

3.10.1 List trunk port groups

Following describes how to acquire the trunk port group list.

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/trunkportgroups.xml`

- JSON format

  `/realnetwork/trunkportgroups.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<trunk_port_groups>
  [
    <trunk_port_group trunk_port_group_name="trunk_port_group_name"/>
  ]
</trunk_port_groups>
```

- JSON format

```
{
  "trunk_port_groups" : [
    {
      "trunk_port_group_name" : "trunk_port_group_name"
    }
  ]
}
```

Table 3-97 Description of Elements in trunk_port_groups

Element	Description	Return Value
trunk_port_group_name	trunk port group name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

3.10.2 Show trunk port group

Following describes how to acquire the trunk port group information.

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/trunkportgroups/`**`trunk_port_group_name`**`.xml`

- JSON format

  `/realnetwork/trunkportgroups/`**`trunk_port_group_name`**`.json`

Table 3-98 Description of Parameters in URI

Element	Description	Valid Value
trunk_port_group_name	Trunkport group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<trunk_port_group trunk_port_group_name="trunk_port_group_name">
  <member_ofses>
    [
    <member_ofs
      ofs_index="ofs_index"
      dp_id="dp_id"/>
    ]
```

```
  </member_ofses>
  <shared_stack_link_ports>
    [
    <shared_stack_link_port
      ofs_index="ofs_index"
      east="east"
      west="west"/>
    ]
  </shared_stack_link_ports>
  <trunk_ports>
    [
    <trunk_port
      trunk_port_name="trunk_port_name"
      load_balance="load_balance">
      <member_ports>
        [
        <member_port
          ofs_index="ofs_index"
          port_name="port_name"/>
        ]
      </member_ports>
    </trunk_port>
    ]
  </trunk_ports>
</trunk_port_group>
```

- JSON format

```
{
  "trunk_port_group" : {
    "trunk_port_group_name": "trunk_port_group_name",
    "member_ofses" : [
      {
        "ofs_index" : "ofs_index",
        "dp_id" : "dp_id"
      }
    ],
    "shared_stack_link_ports" : [
      {
        "ofs_index" : "ofs_index",
        "east" : "east",
        "west" : "west"
      }
    ],
    "trunk_ports" : [
      {
        "trunk_port_name" : "trunk_port_name",
        "load_balance" : "load_balance",
        "member_ports" : [
          {
            "ofs_index" : "ofs_index",
            "port_name" : "port_name"
          }
        ]
      }
    ]
  }
}
```

Table 3-99 Description of Elements in trunk_port_group

Element	Description	Return Value
trunk_port_group_name	Trunkport group name	Up to 31 characters including one-byte alphanumeric characters and underscores
member_ofses	list of member OFS	member_ofses type (For details, refer to "Table 3-100 Description of Elements in member_ofses type (page 101)" below.)

Element	Description	Return Value
shared_stack_link_ports	list of shared stack link port	shared_stack_link_ports type (For details, refer to "Table 3-101 Description of Elements in stack_link_ports type (page 101)" below.)
trunk_ports	list of trunk port	trunk_ports type (For details, refer to "Table 3-102 Description of Elements in trunk_ports type (page 102)" below.)

Table 3-100 Description of Elements in member_ofses type

Element	Description	Return Value
ofs_index	index of OFS	{ofs1|ofs2}
dp_id	OFS datapath ID	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number)

Table 3-101 Description of Elements in stack_link_ports type

Element	Description	Return Value
ofs_index	index of OFS	{ofs1|ofs2}
east	shared stack link port for east direction	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
west	shared stack link port for west direction	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

Table 3-102 Description of Elements in trunk_ports type

Element	Description	Return Value
trunk_port_name	trunk port name	Up to 31 characters including one-byte alphanumeric characters and underscores
load_balance	load balancing method for the trunk port	{hash|round-robin}
member_ports	list of member port	member_ports type (For details, refer to "Table 3-103 Description of Elements in member_ports (page 102)" below.)

Table 3-103 Description of Elements in member_ports

Element	Description	Return Value
ofs_index	index of OFS	{ofs1|ofs2}
port_name	port name registered to trunk port	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

Chapter 4. WebAPI (L2-Related) Reference

Following describes the URI, parameters, request/response data and other details about the L2-related API.

Tip

The WebAPI uses structure data in the XML or JSON format as API input/output. In this chapter, for each API input/output data unit, the data elements on the same structure level is summarized in a table and the element name is written in bold like **"data."** If a lower level has data, the relevant level name is written in plain text like "data." The reference table number and other details about this data are shown in the "Return value" column.

4.1 Operate VTN

4.1.1 List VTNs

Following describes how to acquire the list of VTNs.

Processing request

Method

GET

request URI

- XML format

  `/vtns.xml`

- JSON format

  `/vtns.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vtns>
  [
  <vtn vtn_name="vtn_name"/>
  ]
</vtns>
```

- JSON format

```
{
  "vtns": [
    {
      "vtn_name": "vtn_name"
    }
  ]
}
```

Table 4-1 Description of Elements in vtns

Element	Description	Return Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.1.2 Create VTN

Following describes how to create a VTN.

Processing request

Method

POST

request URI

- XML format

  `/vtns.xml`

- JSON format

  `/vtns.json`

Settings of request body

- XML format

```
<vtn vtn_name="vtn_name"/>
```

- JSON format

```
{
  "vtn" : {
    "vtn_name" : "vtn_name"
  }
}
```

Table 4-2 Description of Elements in vtn

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Create VTN vtn_name succeeded."`
- On failure

  `"Create VTN vtn_name failed. (<Error information from pfcshell>)"`

Remark

None

4.1.3 Show VTN

Following describes how to acquire the information on VTNs.

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name.xml`
- JSON format

```
/vtns/vtn_name.json
```

Table 4-3 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vtn vtn_name="vtn_name"
  createdtime="createdtime"
  lastcommittedtime="lastcommittedtime"/>
```

- JSON format

```
{
  "vtn": {
    "vtn_name": "vtn_name",
    "createdtime": "createdtime",
    "lastcommittedtime": "lastcommittedtime"
  }
}
```

Table 4-4 Description of Elements in vtn

Element	Description	Return Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
createdtime	VTN creation time	YYYY-MM-DD hh:mm:ss format
lastcommittedtime	Last VTN update time	YYYY-MM-DD hh:mm:ss format

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.1.4 Show VTN details

Following describes how to acquire the detailed information on the VTN including statistical information.

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/detail.xml`

- JSON format

  `/vtns/vtn_name/detail.json`

Table 4-5 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vtn vtn_name="vtn_name"
  status="status"
  faultinfo="faultinfo"
  createdtime="createdtime"
  lastcommittedtime="lastcommittedtime">
  <statistics>
    <rx>
      <all packets="packets" octets="octets"/>
      <multicast packets="packets" octets="octets"/>
      <broadcast packets="packets" octets="octets"/>
    </rx>
    <tx>
      <all packets="packets" octets="octets"/>
      <multicast packets="packets" octets="octets"/>
      <broadcast packets="packets" octets="octets"/>
    </tx>
    <drop>
      <all packets="packets" octets="octets"/>
      <vtnbusy packets="packets" octets="octets"/>
      <vtnrxdisabled packets="packets" octets="octets"/>
```

```
      <vtntxdisabled packets="packets" octets="octets"/>
      <vnodestatusdown packets="packets" octets="octets"/>
      <interfacestatusdown packets="packets" octets="octets"/>
      <vpacketloopcounter packets="packets" octets="octets"/>
      <routenotfound packets="packets" octets="octets"/>
      <flowfilter packets="packets" octets="octets"/>
      <congestioncontrol packets="packets" octets="octets"/>
      <otherreason packets="packets" octets="octets"/>
    </drop>
  </statistics>
</vtn>
```

- JSON format

```
{
  "vtn": {
    "vtn_name": "vtn_name",
    "status": "status",
    "faultinfo": "faultinfo",
    "createdtime": "createdtime",
    "lastcommittedtime": "lastcommittedtime",
    "statistics": {
      "rx": {
        "all": {
          "packets": "packets",
          "octets": "octets"
        },
        "multicast": {
          "packets": "packets",
          "octets": "octets"
        },
        "broadcast": {
          "packets": "packets",
          "octets": "octets"
        }
      },
      "tx": {
        "all": {
          "packets": "packets",
          "octets": "octets"
        },
        "multicast": {
          "packets": "packets",
          "octets": "octets"
        },
        "broadcast": {
          "packets": "packets",
          "octets": "octets"
        }
      },
      "drop": {
        "all": {
          "packets": "packets",
          "octets": "octets"
        },
        "vtnbusy": {
          "packets": "packets",
          "octets": "octets"
        },
        "vtnrxdisabled": {
          "packets": "packets",
          "octets": "octets"
        },
        "vtntxdisabled": {
          "packets": "packets",
          "octets": "octets"
        },
        "vnodestatusdown": {
          "packets": "packets",
          "octets": "octets"
        },
        "interfacestatusdown": {
```

```
          "packets": "packets",
          "octets": "octets"
        },
        "vpacketloopcounter": {
          "packets": "packets",
          "octets": "octets"
        },
        "routenotfound": {
          "packets": "packets",
          "octets": "octets"
        },
        "flowfilter": {
          "packets": "packets",
          "octets": "octets"
        },
        "congestioncontrol": {
          "packets": "packets",
          "octets": "octets"
        },
        "otherreason": {
          "packets": "packets",
          "octets": "octets"
        }
      }
    }
  }
}
```

Table 4-6 Description of Elements in vtn

Element	Description	Return Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	VTN status	{enable|disable} enable: Enable disable: Disable
faultinfo	VTN faultInfo	{fault|active} fault: Failed. active: Not failed.
createdtime	VTN creation time	YYYY-MM-DD hh:mm:ss format
lastcommittedtime	VTN last update time	YYYY-MM-DD hh:mm:ss format
statistics	Statistical information list	statistics (For details, refer to "Table 4-7 Description of Elements in statistics (page 109)")

Table 4-7 Description of Elements in statistics

Element	Description	Return Value
rx	rx list	rx type (For details, refer to "Table 4-8 Description of Elements in rx (page 109)")
tx	tx list	tx type (For details, refer to "Table 4-9 Description of Elements in tx (page 110)")
drop	drop list	drop type (For details, refer to "Table 4-10 Description of Elements in drop (page 110)")

Table 4-8 Description of Elements in rx

Element	Description	Return Value
all	Total number of received packets	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")

Element	Description	Return Value
multicast	Number of received multicast packets	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
broadcast	Number of received broadcast packets	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")

Table 4-9 Description of Elements in tx

Element	Description	Return Value
all	Total number of transmissions	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
multicast	Number of multicast transmissions	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
broadcast	Number of broadcast transmissions	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")

Table 4-10 Description of Elements in drop

Element	Description	Return Value
all	Total number of discarded packets	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
vtnbusy	Number of packets discarded due to VTNBusy	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
vtnrxdisabled	Number of packets discarded because VTN reception is disabled	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
vtntxdisabled	Number of packets discarded because VTN transmission is disabled	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
vnodestatusdown	Number of packets discarded due to VTN Node Down	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
interfacestatusdown	Number of packets discarded due to Interface statusDown	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
vpacketloopcounter	Number of packets discarded due to VPacket LoopCounter	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
routenotfound	Number of packets discarded because there is no routing table	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
flowfilter	Number of packets discarded due to FlowFilter	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")

Element	Description	Return Value
congestioncontrol	Number of packets discarded due to congestion control	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")
otherreason	Number of packets discarded due to some other reason	statistics type (For details, refer to "Table 4-11 Description of Elements in statistics Type (page 111)")

Table 4-11 Description of Elements in statistics Type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of bytes	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.1.5 Delete VTN

Following describes how to delete a VTN.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`.xml`
- JSON format

  `/vtns/`**`vtn_name`**`.json`

Table 4-12 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Delete VTN vtn_name succeeded."`
- On failure

  `"Delete VTN vtn_name failed.(<Error information from pfcshell>)"`

Remark

None

4.2 Operate VTN Flow

4.2.1 Show data-flows for VTN

Following describes how to acquire the VTN flow information.

Processing request

Method

GET

request URI

- XML format

/vtns/**vtn_name**/dataflows.xml

- JSON format

/vtns/**vtn_name**/dataflows.json

Table 4-13 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataflows>
  [
  <dataflow flowid="flowid"
    createdtime="createdtime"
    egressportname="egressportname"
    egressvnode="egressvnode"
    hardtimeout="hardtimeout"
    idletimeout="idletimeout"
    ingressportname="ingressportname"
    ingressvnode="ingressvnode"
    ingressofsdpid="ingressofsdpid"
    egressofsdpid="egressofsdpid"/>
  ]
</dataflows>
```

- JSON format

```
{
  "dataflows": [
    {
      "flowid": "flowid",
      "createdtime": "createdtime",
      "egressportname": "egressportname",
      "egressvnode": "egressvnode",
      "hardtimeout": "hardtimeout",
      "idletimeout": "idletimeout",
      "ingressportname": "ingressportname",
      "ingressvnode": "ingressvnode",
      "ingressofsdpid": "ingressofsdpid",
      "egressofsdpid": "egressofsdpid"
    }
  ]
}
```

Table 4-14 Description of Elements in dataflows

Element	Description	Return Value
flowid	Flow ID	Decimal number (1 to 4294967295)
createdtime	Flow generation time	YYYY-MM-DD hh:mm:ss format

Element	Description	Return Value
egressportname	EgressPort name	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
egressvnode	EgressVNode name	Up to 31 characters including one-byte alphanumeric characters and underscores
hardtimeout	Hard timeout time	Decimal number (1 to 65536)
idletimeout	Idle timeout time	Decimal number (1 to 65536)
ingressportname	IngressPort name	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
ingressvnode	IngressVNode name	Up to 31 characters including one-byte alphanumeric characters and underscores
ingressofsdpid	OFS datapath ID for Ingress	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
egressofsdpid	OFS datapath ID for Egress	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.2.2 Show data-flows details for VTN

Following describes how to acquire the detailed VTN flow information including statistical information.

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/dataflows/detail.xml`

- JSON format

  /vtns/**vtn_name**/dataflows/detail.json

Table 4-15 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<dataflows>
  [
  <dataflow flowid="flowid"
    createdtime="createdtime"
    egressportname="egressportname"
    egressvnode="egressvnode"
    hardtimeout="hardtimeout"
    idletimeout="idletimeout"
    ingressportname="ingressportname"
    ingressvnode="ingressvnode"
    ingressofsdpid="ingressofsdpid"
    egressofsdpid="egressofsdpid">
    <match dstmac="dstmac"
      inport="inport"
      srcmac="srcmac"
      vlan_id="vlan_id"
      vlanpriority="vlanpriority"
      ethernettype="ethernettype"
      srcip="srcip"
      dstip="dstip"
      srcipv6="srcipv6"
      dstipv6="dstipv6"
      iptos="iptos"
      ipprotocol="ipprotocol"
      l4srcport_icmptype="l4srcport_icmptype"
      l4dstport_icmpcode="l4dstport_icmpcode"/>
    <action drop="drop"
      outport="outport"
      setvlan_id="setvlan_id"
      setvlanpriority="setvlanpriority"
      stripvlanheader="stripvlanheader"
      setsrcmac="setsrcmac"
      setdstmac="setdstmac"
      setsrcip="setsrcip"
      setdstip="setdstip"
      srcipv6="srcipv6"
      dstipv6="dstipv6"
      setiptos="setiptos"
      setl4srcport_icmptype="setl4srcport_icmptype"
      setl4dstport_icmpcode="setl4dstport_icmpcode"
      setenqueue="setenqueue"/>
    <statistics packets="packets"
      octets="octets"
      duration="duration"/>
  </dataflow>
```

```
  ]
</dataflows>
```

- JSON format

```
{
  "dataflows": [
    {
      "flowid": "flowid",
      "createdtime": "createdtime",
      "egressportname": "egressportname",
      "egressvnode": "egressvnode",
      "hardtimeout": "hardtimeout",
      "idletimeout": "idletimeout",
      "ingressportname": "ingressportname",
      "ingressvnode": "ingressvnode",
      "ingressofsdpid": "ingressofsdpid",
      "egressofsdpid": "egressofsdpid",
      "match": {
        "dstmac": "dstmac",
        "inport": "inport",
        "srcmac": "srcmac",
        "vlan_id": "vlan_id",
        "vlanpriority": "vlanpriority",
        "ethernettype": "ethernettype",
        "srcip": "srcip",
        "dstip": "dstip",
        "srcipv6": "srcipv6",
        "dstipv6": "dstipv6",
        "iptos": "iptos",
        "ipprotocol": "ipprotocol",
        "l4srcport_icmptype": "l4srcport_icmptype",
        "l4dstport_icmpcode": "l4dstport_icmpcode"
      },
      "action": {
        "drop": "drop",
        "outport": "outport",
        "setvlan_id": "setvlan_id",
        "setvlanpriority": "setvlanpriority",
        "stripvlanheader": "stripvlanheader",
        "setsrcmac": "setsrcmac",
        "setdstmac": "setdstmac",
        "setsrcip": "setsrcip",
        "setdstip": "setdstip",
        "srcipv6": "srcipv6",
        "dstipv6": "dstipv6",
        "setiptos": "setiptos",
        "setl4srcport_icmptype": "setl4srcport_icmptype",
        "setl4dstport_icmpcode": "setl4dstport_icmpcode",
        "setenqueue": "setenqueue"
      },
      "statistics": {
        "packets": "packets",
        "octets": "octets",
        "duration": "duration"
      }
    }
  ]
}
```

Table 4-16 Description of Elements in dataflows

Element	Description	Return Value
flowid	Flow ID	Decimal number (1 to 4294967295)
createdtime	Flow creation time	YYYY-MM-DD hh:mm:ss format
egressportname	EgressPort name	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)

Element	Description	Return Value
egressvnode	EgressVNode name	Up to 31 characters including one-byte alphanumeric characters and underscores
hardtimeout	Hard timeout	Decimal number (1 to 65536) ""0" indicates unlimited time.
idletimeout	Idle timeout	Decimal number (1 to 65536) ""0" indicates unlimited time.
ingressportname	IngressPort name	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
ingressvnode	IngressVNode name	Up to 31 characters including one-byte alphanumeric characters and underscores
ingressofsdpid	OFS datapath ID for Ingress	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
egressofsdpid	OFS datapath ID for Egress	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
match	Match	match type (For details, refer to "Table 4-17 Description of Elements in match Type (page 117)")
action	Action	action type (For details, refer to "Table 4-18 Description of Elements in action Type (page 118)")
statistics	Statistical information list	statistics type (For details, refer to "Table 4-19 Description of Elements in statistics Type (page 119)")

Table 4-17 Description of Elements in match Type

Element	Description	Return Value
dstmac	Destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
inport	Received OFS port number	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
srcmac	Source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
vlan_id	VLAN ID	Decimal number (1 to 4095)
vlanpriority	VLAN priority	Up to one character
ethernettype	Ether type	Hexadecimal format including "0x"
srcip	Source IP address / Mask length	IPAddress/Prefix IPAddress: IPv4 dot-separated format (Example: 192.168.1.1) Prefix: 1 to 32
dstip	Destination IP address / Mask length	IPAddress/Prefix IPAddress: IPv4 dot-separated format (Example: 192.168.1.1) Prefix: 1 to 32
srcipv6	Source IPv6 address	IPv6Address/Prefix IPv6Address: The IPv6 address format conforms to RFC 5952. Prefix: 1 to 128

Element	Description	Return Value
dstipv6	Destination IPv6 address	IPv6Address/Prefix IPv6Address: The IPv6 address format conforms to RFC 5952. Prefix: 1 to 128
iptos	TOS value	Hexadecimal format including "0x"
ipprotocol	Protocol field of IPv4 header or Next Header field of IPv6 header	Up to three characters (1 to 255)
l4srcport_icmptype	L4 source port/ICMPtype	Source port number or ICMP type (10x)(0x(16x)
l4dstport_icmpcode	L4 destination port/ICMPcode	Destination port number or ICMP code(10x)(0x(16x)

Table 4-18 Description of Elements in action Type

Element	Description	Return Value
drop	Drop information	Discard packets.
outport	(ALL/ CONTROLLER/ LOCAL/INPORT/ NORMAL/FLOOD/ Port number)	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
setvlan_id	VLAN ID	Decimal number (1 to 4095)
setvlanpriority	Priority	PCP field of the VLAN tab
stripvlanheader	Strip VLAN header	Delete the VLAN tag from the packet.
setsrcmac	Source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
setdstmac	Destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
setsrcip	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
setdstip	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
srcipv6	Source IPv6 address	The IPv6 address format conforms to RFC 5952.
dstipv6	Destination IPv6 address	The IPv6 address format conforms to RFC 5952.
setiptos	ToS value	Hexadecimal format including "0x"
setl4srcport_icmptype	L4 source port/ICMPtype	Source port number or ICMP type (10x)(0x(16x)
setl4dstport_icmpcode	L4 destination port/ICMP code	Return Value: Destination port number or ICMP code(10x)(0x(16x)
setenqueue	Port number of queuing destination	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)

Table 4-19 Description of Elements in statistics Type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of bytes	Counter
duration	Duration time	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.2.3 Show data-flows count for VTN

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/dataflows/count.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/dataflows/count.json`

Table 4-20 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataflow count="count"/>
```

- JSON format

```
{
  "dataflow" : {
    "count" : "count"
  }
}
```

- On error
  - Insufficient information is returned from pfcshell (data-flow count)

    `"*** Error:Internal error. (no data-flow count)"`

Table 4-21 Description of Elements in dataflow

Element	Description	Return Value
count	Number of dataflows	Decimal number (Up to 20 characters)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.3 Operate OFS Map Information (VTN Mode)

4.3.1 Show ofs-maps on VTN

Following describes how to acquire the VTN physical-virtual mapping information.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/ofsmaps.xml

- JSON format

  /vtns/**vtn_name**/ofsmaps.json

Table 4-22 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ofsmaps>
  [
  <ofsmap vnodename="vnodename"
    type="type"
    maptype="maptype"
    dp_id="dp_id"
    port_name="port_name"
    trunk_port_name="trunk_port_name"
    vlan_id="vlan_id"
    vlan_tag="vlan_tag"/>
  ]
</ofsmaps>
```

- JSON format

```
{
  "ofsmaps" : [
    {
      "vnodename" : "vnodename",
      "type" : "type",
      "maptype" : "maptype",
      "dp_id" : "dp_id",
      "port_name" : "port_name",
      "trunk_port_name" : "trunk_port_name",
      "vlan_id" : "vlan_id",
      "vlan_tag" : "vlan_tag"
    }
  ]
}
```

Table 4-23 Description of Elements in ofsmaps

Element	Description	Return Value
vtnnode	vNode name	Up to 31 characters including one-byte alphanumeric characters and underscores
type	Mapping type	{dynamic|static} static: Displayed for the mapping explicitly specified using ofs-map dynamic: Displayed for the mapping dynamically learned using vlan-map
maptype	Map type	{ofs-map|vlan-map|mac-map}
dp_id	datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number) *This element is not returned if the trunk interface is mapped.
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?) *This element is not returned if the trunk interface is mapped.
trunk_port_name	trunkport name	Up to 31 characters including one-byte alphanumeric characters and underscores *This element is not returned if the Physics interface is mapped.
vlan_id	VLAN ID	Decimal number (1 to 4095)
vlan_tag	Whether the VLAN tag exists	{tagged|untagged} tagged: Sends and receives packets with VLAN tag. untagged: Sends and receives packets without VLAN tag.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.4 Operate OFS Domain (VTN Mode)

4.4.1 Set OFS Domain (VTN Mode)

Following describes how to set OFS domain (VTN mode) configuration.

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/ofsdomain.xml

- JSON format

  /vtns/**vtn_name**/ofsdomain.json

Table 4-24 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<ofs_domain ofs_domain_name="ofs_domain_name"/>
```

- JSON format

```
{
  "ofs_domain" : {
    "ofs_domain_name" : "ofs_domain_name"
  }
}
```

Table 4-25 Description of Elements in ofs_domain

Element	Description	Valid Value
ofs_domain_name	OFS domain name	Up to 31 characters including one-byte alphanumeric characters and underscores

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Create OFSDomain **ofs_domain_name** for VTN **vtn_name**succeeded."

- On failure

  "Create OFSDomain **ofs_domain_name** for VTN **vtn_name** failed.([invalid argument(OFSDomain)])(<Error information from pfcshell>)"

Remark

None

4.4.2 Show OFS Domain (VTN Mode)

Following describes how to acquire the OFS domain (VTN mode) configuration.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/ofsdomain.xml

- JSON format

  /vtns/**vtn_name**/ofsdomain.json

Table 4-26 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ofs_domain ofs_domain_name="ofs_domain_name"/>
```

- JSON format

```
{
  "ofs_domain" : {
```

```
    "ofs_domain_name" : "ofs_domain_name"
  }
}
```

Table 4-27 Description of Elements in ofs_domain

Element	Description	Return Value
ofs_domain_name	OFS domain name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.4.3 Unset OFS Domain (VTN Mode)

Following describes how to unset the OFS domain (VTN mode) configuration.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/vtn_name/ofsdomain/ofs_domain_name.xml`
- JSON format

  `/vtns/vtn_name/ofsdomain/ofs_domain_name.json`

Table 4-28 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
ofs_domain_name	OFS domain name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Delete OFSDomain ofs_domain_name of VTN vtn_name succeeded."`

- On failre

  `"Delete OFSDomain ofs_domain_name of VTN vtn_name failed. (<Error in formation from pfcshell>)"`

Remark

None

4.5 Operate vBridge

4.5.1 List vBridges

Following describes how to acquire the list of vBridges under VTN.

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/vbridges.xml`

- JSON format

  `/vtns/vtn_name/vbridges.json`

Table 4-29 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vbridges>
  [
  <vbridge vbr_name="vbr_name"/>
  ]
</vbridges>
```

- JSON format

```
{
  "vbridges" : [
    {
      "vbr_name" : "vbr_name"
    }
  ]
}
```

Table 4-30 Description of Elements in vbridges

Element	Description	Return Value
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.5.2 Create vBridge

Following describes how to create a new vBridge node.

Processing request

Method

POST

request URI

- XML format

  `/vtns/vtn_name/vbridges.xml`

- JSON format

  `/vtns/vtn_name/vbridges.json`

Table 4-31 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<vbridge vbr_name="vbr_name"/>
```

- JSON format

```
{
  "vbridge" : {
    "vbr_name" : "vbr_name"
  }
}
```

Table 4-32 Description of Elements in vbridge

Element	Description	Valid Value
vtn_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Create vBridge **vbr_name** to VTN **vtn_name** succeeded."

- On failure

  "Create vBridge **vbr_name** to VTN **vtn_name** failed.(<Error information from pfcshell>)"

Remark

None

4.5.3 Show vBridge

Following describes how to acquire the information on vBridges.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**.json

Table 4-33 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vbridge vbr_name="vbr_name">
  <interfaces>
    [
```

```
    <interface if_name="if_name"/>
    ]
  </interfaces>
</vbridge>
```

- JSON format

```
{
  "vbridge" : {
    "vbr_name" : "vbr_name",
    "interfaces" : [
      {
        "if_name" : "if_name"
      }
    ]
  }
}
```

Table 4-34 Description of Elements in vbridge

Element	Description	Return Value
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.5.4 Show vBridge details

Following describes how to acquire the detailed vBridge information including statistical information.

Processing request

Method

GET

request URI

- XML format

/vtns/**vtn_name**/vbridges/**vbr_name**/detail.xml

- JSON format

/vtns/**vtn_name**/vbridges/**vbr_name**/detail.json

Table 4-35 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vbridge  vbr_name="vbr_name"
  status="status">
  <interfaces>
    [
    <interface if_name="if_name"
      status="status">
      <rx>
        <all packets="packets" octets="octets"/>
      </rx>
      <tx>
        <all packets="packets" octets="octets"/>
        <multibroad packets="packets" octets="octets"/>
        <unicast packets="packets" octets="octets"/>
        <flooding packets="packets" octets="octets"/>
        <host packets="packets" octets="octets"/>
      </tx>
    </interface>
    ]
  </interfaces>
  <flooding>
    <unknownunicast count="count"/>
    <multibroad count="count"/>
  </flooding>
  <drop>
    <all packets="packets" octets="octets"/>
    <noroute packets="packets" octets="octets"/>
    <timeout packets="packets" octets="octets"/>
    <otherreason packets="packets" octets="octets"/>
  </drop>
</vbridge>
```

- JSON format

```
{
  "vbridge" : {
    "vbr_name" : "vbr_name",
    "status" : "status",
    "interfaces" : [
      {
        "if_name" : "if_name",
        "status" : "status",
```

```
        "rx" : {
          "all" : {
            "packets" : "packets",
            "octets" : "octets"
          }
        },
        "tx" : {
          "all" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "multibroad" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "unicast" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "flooding" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "host" : {
            "packets" : "packets",
            "octets" : "octets"
          }
        }
      }
    ],
    "flooding" : {
      "unknownunicast" : {
        "count" : "count"
      },
      "multibroad" : {
        "count" : "count"
      }
    },
    "drop" : {
      "all" : {
        "packets" : "packets",
        "octets" : "octets"
      },
      "noroute" : {
        "packets" : "packets",
        "octets" : "octets"
      },
      "timeout" : {
        "packets" : "packets",
        "octets" : "octets"
      },
      "otherreason" : {
        "packets" : "packets",
        "octets" : "octets"
      }
    }
  }
}
```

Table 4-36 Description of Elements in vbridge

Element	Description	Return Value
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	Status of vBridge	{up|down}
interfaces	Interface list	interface type (For details, refer to "Table 4-37 Description of Elements in interface (page 133)")

Element	Description	Return Value
flooding	flooding information	flooding type (For details, refer to "Table 4-40 Description of Elements in flooding type (page 133)")
drop	Discarded information	drop type (For details, refer to "Table 4-42 Description of Elements in drop (page 134)")

Table 4-37 Description of Elements in interface

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	Interface status of vBridge	{up|down}
rx	Reception	rx type (For details, refer to "Table 4-38 Description of Elements in rx (page 133)")
tx	Transmission	tx type (For details, refer to "Table 4-39 Description of Elements in tx (page 133)")

Table 4-38 Description of Elements in rx

Element	Description	Return Value
all	Total number of transmission on vBridge	statistics type (For details, refer to "Table 4-43 Description of Elements in statistics (page 134)")

Table 4-39 Description of Elements in tx

Element	Description	Return Value
all	Total number of transmission on vBridge	statistics type (For details, refer to "Table 4-43 Description of Elements in statistics (page 134)")
multibroad	Number of multicast and broad-cast transmissions on vBridge	statistics type (For details, refer to "Table 4-43 Description of Elements in statistics (page 134)")
unicast	Number of unicast transmission on vBridge	statistics type (For details, refer to "Table 4-43 Description of Elements in statistics (page 134)")
flooding	Number of flooded transmissions on vBridge	statistics type (For details, refer to "Table 4-43 Description of Elements in statistics (page 134)")
host	Number of transmissions to the host on vBridge	statistics type (For details, refer to "Table 4-43 Description of Elements in statistics (page 134)")

Table 4-40 Description of Elements in flooding type

Element	Description	Return Value
unknownunicast	Flooding processing count (unknown unicast)	count type (For details, refer to "Table 4-41 Description of Elements in count type (page 133)")
multibroad	Flooding processing count (Multicast/ Broadcast)	count type (For details, refer to "Table 4-41 Description of Elements in count type (page 133)")

Table 4-41 Description of Elements in count type

Element	Description	Return Value
count	Number of bytes	counter

Table 4-42 Description of Elements in drop

Element	Description	Return Value
all	Total number of packets discarded when going through vBridge	statistics type (For details, refer to "Table 4-43 Description of Elements in statistics (page 134)")
noroute	Number of packets discarded because there is no routing table when going through vBridge	statistics type (For details, refer to "Table 4-43 Description of Elements in statistics (page 134)")
timeout	Number of packets discarded due to the timeout of a lower-level protocol when going through vBridge	statistics type (For details, refer to "Table 4-43 Description of Elements in statistics (page 134)")
otherreason	Number of packets discarded due to some other reason when going through vBridge	statistics type (For details, refer to "Table 4-43 Description of Elements in statistics (page 134)")

Table 4-43 Description of Elements in statistics

Element	Description	Return Value
packets	Number of packets	counter
octets	Number of bytes	counter

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.5.5 Delete vBridge

Following describes how to delete a vBridge.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/vtn_name/vbridges/vbr_name.xml`

- JSON format

  `/vtns/vtn_name/vbridges/vbr_name.json`

Table 4-44 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Delete vBridge vbr_name of VTN vtn_name succeeded."`

- On failure

  `"Delete vBridge vbr_name of VTN vtn_name failed.(<Error information from pfcshell>)"`

Remark

None

4.6 Operate vBridge IP Address

4.6.1 Register IP address to vBridge

Processing request

Method

PUT

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/ipaddress.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/ipaddress.json

Table 4-45 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format
  - When prefix is specified

```
<hostaddress ipaddr="ipaddr" prefix="prefix"/>
```

  - When netmask is specified

```
<hostaddress ipaddr="ipaddr" netmask="netmask"/>
```

- JSON format
  - When prefix is specified

```
{
  "hostaddress" : {
    "ipaddr" : "ipaddr",
    "prefix" : "prefix"
  }
}
```

  - When netmask is specified

```
{
  "hostaddress" : {
    "ipaddr" : "ipaddr",
    "netmask" : "netmask"
  }
}
```

Table 4-46 Description of Elements in hostaddress

Element	Description	Return Value
ipaddr	IP Address	IPAddress: IPv4 dot-separated format (Example: 192.168.1.1)
prefix	Prefix of IP address	Decimal number (1 to 30)
netmask	Subnet mask	IPAddress: IPv4 dot-separated format (128.0.0.0 to 255.255.255.252)

Remember

ipaddr is required.

Remember

Either prefix or netmask must be always specified.

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Register hostaddress ipaddr **ipaddr** for vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Register hostaddress ipaddr of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(prefix, netmask)](<Error information from pfc shell>)"

  "Register hostaddress ipaddr of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

4.6.2 Show IP address of vBridge

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/ipaddress.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/ipaddress.json

Table 4-47 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<hostaddress ipaddr="ipaddr" netmask="netmask"/>
```

- JSON format

```
{
  "hostaddress" : {
    "ipaddr" : "ipaddr",
    "netmask" : "netmask"
  }
}
```

Table 4-48 Description of Elements in hostaddress

Element	Description	Return Value
ipaddr	IP Address	IPAddress: IPv4 dot-separated format (Example: 192.168.1.1)
netmask	Subnet mask	IPAddress: IPv4 dot-separated format (128.0.0.0 to 255.255.255.252)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.6.3 Unregister IP address from vBridge

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/ipaddress.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/ipaddress.json`
- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `ipaddr=`**`ipaddr`**

  `prefix=`**`prefix`**

  `netmask=`**`netmask`**

Table 4-49 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
ipaddr	IP Address	IPAddress: IPv4 dot-separated format (Example: 192.168.1.1)
prefix	Prefix of IP address	Decimal number (1 to 30)

Element	Description	Valid Value
netmask	Subnet mask	IPAddress: IPv4 dot-separated format (128.0.0.0 to 255.255.255.252)

Remember

ipaddr is required.

Remember

Either prefix or netmask must be always specified.

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Unregister hostaddress ipaddr **ipaddr** for vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Unregister hostaddress ipaddr of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(prefix or netmask)](<Error information from pfcshell>)"

  "Unregister hostaddress ipaddr of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)]"

Remark

None

4.7 Operate Interface (vBridge Mode)

4.7.1 List interfaces for vBridge

Following describes how to acquire the vBridge virtual interface list.

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/vbridges/vbr_name/interfaces.xml`

- JSON format

  `/vtns/vtn_name/vbridges/vbr_name/interfaces.json`

Table 4-50 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interfaces>
  [
  <interface if_name="if_name"/>
  ]
</interfaces>
```

- JSON format

```
{
  "interfaces" : [
    {
      "if_name" : "if_name"
    }
  ]
}
```

Table 4-51 Description of Elements in interfaces

Element	Description	Valid Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Important

When VLAN mapping (vlan-map command) is specified, the resources that were dynamically generated by this function are prefixed with a question mark (?) and returned to if_name. These names cannot be specified for the WebAPI URI path.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.7.2 Create interface for vBridge

Following describes how to create a new virtual interface for the vBridge.

Processing request

Method

POST

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/interfaces.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/interfaces.json`

Table 4-52 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<interface if_name="if_name"/>
```

- JSON format

```
{
  "interface" : {
    "if_name" : "if_name"
  }
}
```

Table 4-53 Description of Elements in interface

Element	Description	Valid Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Remember

if_name can be omitted.

Remember

If if_name is omitted, an interface is automatically created under a name in the IF_WA_{number} format. The range for number is between 1 and 2147483647.

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"/>
```

- JSON format

```
{
  "interface" : {
    "if_name" : "if_name"
  }
}
```

Table 4-54 Description of Elements in interface

Element	Description	Valid Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Create interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Create interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(if_name)](<Error information from pfcshell>)"

Remark

None

4.7.3 Show interface for vBridge

Following describes how to acquire the vBridge virtual interface information.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**.json

Table 4-55 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"
  ifindex="ifindex"
  mtu="mtu"/>
```

- JSON format

```
{
  "interface" : {
    "if_name" : "if_name",
    "ifindex" : "ifindex",
    "mtu" : "mtu"
  }
}
```

Table 4-56 Description of Elements in interface

Element	Description	Valid Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ifindex	index	Decimal number
mtu	MTU	Decimal number (1 to 16000)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On failure

  "Get interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(if_name)](<Error information from pfcshell>)"

Remark

None

4.7.4 Show interface details for vBridge

Following describes how to acquire the vBridge virtual interface information (details).

Processing request

Method

GET

request URI

- XML format

/vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/detail.xml

- JSON format

/vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/detail.json

Table 4-57 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"
  ifindex="ifindex"
  adminstatus="adminstatus"
  operstatus="operstatus"
  mtu="mtu"/>
```

- JSON format

```
{
  "interface" : {
    "if_name" : "if_name",
    "ifindex" : "ifindex",
    "adminstatus" : "adminstatus",
    "operstatus" : "operstatus",
    "mtu" : "mtu"
  }
}
```

Table 4-58 Description of Elements in interface

Element	Description	Valid Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ifindex	Interface index	Decimal number
adminstatus	Port status	{up|down}
operstatus	Link status	{up|down}
mtu	Maximum data size that can be transferred at a time	Decimal number (1 to 16000)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.7.5 Delete interface for vBridge

Following describes how to delete a virtual interface under vBridge.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/interfaces/`**`if_name`**`.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/interfaces/`**`if_name`**`.json`

Table 4-59 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name under vBridge	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Delete interface` **`if_name`** `of VTN` **`vtn_name`** `vBridge` **`vbr_name`** `succeeded."`

- On failure

  `"Delete interface` **`if_name`** `of VTN` **`vtn_name`** `vBridge` **`vbr_name`** `failed.(<Error information from pfcshell>)"`

Remark

None

4.8 Operate MAC entry

4.8.1 Show MAC entries

Following describes how to acquire MAC entry information.

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/macentries.xml`

- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/macentries.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `type=`**`type`**

  `if_name=`**`if_name`**

Table 4-60 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
type	type	{dynamic|static}
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Remember

type, if_name can be omitted.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<macentries>
  [
  <macentry macaddr="macaddr"
    type="type"
    port_name="port_name"
    ifkind="ifkind"/>
  ]
  <totalcount value="value" />
</macentries>
```

- JSON format

```
{
  "macentries" : [
    {
      "macaddr" : "macaddr",
      "type" : "type",
      "port_name" : "port_name",
      "ifkind" : "ifkind"

    }
  ]
  "totalcount" : {
      "value" : "value"
  }
}
```

Table 4-61 Description of Elements in macentries

Element	Description	Valid Value
macentry	MAC entry	macentry type (For details, refer to "Table 4-62 Description of Elements in macentry (page 150)" below.)
totalcount	Number of MAC entries	totalcount type (For details, refer to "Table 4-63 Description of Elements in total count (page 150)" below.)

Table 4-62 Description of Elements in macentry

Element	Description	Valid Value
macaddr	MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).
type	type	{static|dynamic} dynamic: The learning type is dynamic static: The learning type is static.
port_name	Port name	Up to 31 characters including one-byte alphanumeric characters and underscores
ifkind	Interface type	{trunk| } trunk: Trunk interface Blank: Interface

Table 4-63 Description of Elements in total count

Element	Description	Valid Value
value	Number of MAC entries	Decimal number (0 to 65535)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On failure

  "Get mac entry of vbridge **vbr_name** of vtn **vtn_name** with type **type** interface **if_name** failed. [Invalid argument(**type** or **if_name**)](<Error information from pfcshell>)"

Remark

None

4.8.2 Show count of MAC entries

Following describes how to acquire the number of MAC entries.

Processing request

Method

GET

request URI

- XML format

/vtns/**vtn_name**/vbridges/**vbr_name**/macentries/count.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/macentries/count.json

- Parameter

  The following parameters are passed in the URI query character string (in the format of ?param1=***&param2=***).

  type=**type**

  if_name=**if_name**

Table 4-64 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTNname	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
type	type	{dynamic|static}
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Remember

type, if_name can be omitted.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<macentry totalcount="totalcount" />
```

- JSON format

```
{
  "macentry" : {
       "totalcount" : "totalcount"
  }
}
```

Table 4-65 Description of Elements in macentry

Element	Description	Return Value
totalcount	Number of MAC entries	Decimal number (0 to 65535)

HTTP status code for response

- On success

200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On failure

  "Get mac entry count of vBridge **vbr_name** of VTN **vtn_name** with type **type** interface **if_name** failed. [Invalid argument(type or if_name)](<Error information from pfcshell>)"

Remark

None

4.9 Operate VLAN Map

4.9.1 Create VLAN Map

Following describes how to create a new virtual interface for the vBridge.

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/vlanmaps.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/vlanmaps.json

Table 4-66 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format
  - When dp_id is specified

```
<vlanmap dp_id="dp_id" vlan_id="vlan_id"/>
```

- When ofs_subdomain_name is specified

```
<vlanmap ofs_subdomain_name="ofs_subdomain_name" vlan_id="vlan_id"/>
```

- When both dp_id and ofs_subdomain_name are omitted
  * When vlan_id is specified

```
<vlanmap vlan_id="vlan_id"/>
```

  * When no_vlan_id is specified

```
<vlanmap no_vlan_id="no_vlan_id"/>
```

• JSON format
- When dp_id is specified

```
{
  "vlanmap" : {
    "dp_id" : "dp_id",
    "vlan_id" : "vlan_id"
  }
}
```

- When ofs_subdomain_name is specified

```
{
  "vlanmap" : {
    "ofs_subdomain_name" : "ofs_subdomain_name",
    "vlan_id" : "vlan_id"
  }
}
```

- When both dp_id and ofs_subdomain_name are omitted
  * When vlan_id is specified

```
{
  "vlanmap" : {
    "vlan_id" : "vlan_id"
  }
}
```

  * When no_vlan_id is specified

```
{
  "vlanmap" : {
    "no_vlan_id" : "no_vlan_id"
  }
}
```

Table 4-67 Description of Elements in vlanmap

Element	Description	Valid Value
dp_id	OFS datapath ID	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number). It is not allowed to specify F's to all digits.
ofs_subdomain_name	OFS subdomain name	Up to 31 characters including one-byte alphanumeric characters and underscores
vlan_id	VLAN ID	Decimal number (1 to 4095)

Element	Description	Valid Value
no_vlan_id	Frames without VLAN ID are handled.	{true} (Lowercase)

Remember

Only dp_id or ofs_subdomain_name can be specified.

If both dp_id and ofs_subdomain_name are omitted, it is assumed that all OFSs are handled (dp_id indicates ffff-ffff-ffff-ffff showing all OFSs).

Remember

Only vlan_id or no_vlan_id can be specified.

Remember

Before specifying dp_id, enter the "vlan-connect enable" command with pfcshell. When specifying no_vlan_id, the "vlan-connect enable" command must be not entered state.

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vlanmap vlanmap_id="vlanmap_id"/>
```

- JSON format

```
{
  "vlanmap" : {
    "vlanmap_id" : "vlanmap_id"
  }
}
```

Table 4-68 Description of Elements in vlanmap

Element	Description	Valid Value
vlanmap_id	VLAN map ID	**dp_id**-*HHHH* format (H: hexadecimal) if dp_id is specified **ofs_subdomain_name**-*HHHH* format (H: hexadecimal) if ofs_subdomain_name is specified ffff-ffff-ffffffff-*HHHH* format (H: hexadecimal) if neither dp_id or ofs_subdomain_name is specified

Remember

vlanmap_id is created with the fixed form from the values included in a request, but the form will possibly be changed without notice.

HTTP status code for response

- On success

  200 (OK)

- On failure

400, 500

Miscellaneous

Log

- When dp_id is specified
  - On success

    "Map VLAN **vlan_id** to VTN **vtn_name** vBridge **vbr_name** DPID **dp_id** succeeded."
  - On failure

    "Map VLAN **vlan_id** to VTN **vtn_name** vBridge **vbr_name** DPID **dp_id** failed. (<Error information from pfcshell>)"
- When ofs_subdomain_name is specified
  - On success

    "Map VLAN **vlan_id** to VTN **vtn_name** vBridge **vbr_name** OFS SubDomain **ofs_subdomain_name** succeeded."
  - On failure

    "Map VLAN **vlan_id** to VTN **vtn_name** vBridge **vbr_name** OFS SubDomain **ofs_subdomain_name** failed. (<Error information from pfcshell>)"
- If neither dp_id or ofs_subdomain_name is specified
  - When vlan_id is specified
    * On success

      "Map VLAN **vlan_id** to VTN **vtn_name** vBridge **vbr_name** succeeded."
    * On failure

      "Map VLAN to VTN **vtn_name** vBridge **vbr_name** failed. [Invalid argument(**parameter subdivision**)] (<Error information from pfcshell>)"
  - When no_vlan_id is specified
    * On success

      "Map VLAN without vlantag to VTN **vtn_name** vBridge **vbr_name** succeeded."
    * On failure

      "Map VLAN to VTN **vtn_name** vBridge **vbr_name** failed. [Invalid argument(**parameter subdivision**)] (<Error information from pfcshell>)"

Remark

None

4.9.2 Show VLAN map

Following describes how to acquire the VLAN map information.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/vlanmaps.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/vlanmaps.json

Table 4-69 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format
  - When the VLAN map is registered with ofs_subdomain_name specified

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vlanmaps>
  [
  <vlanmap vlanmap_id="vlanmap_id"
    ofs_subdomain_name="ofs_subdomain_name"
    vlan_id="vlan_id"/>
  ]
</vlanmaps>
```

  - When the VLAN map is registered with dp_id specified

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vlanmaps>
  [
  <vlanmap vlanmap_id="vlanmap_id"
    dp_id="dp_id"
    vlan_id="vlan_id"/>
  ]
</vlanmaps>
```

  - When the VLAN map is registered without specifying ofs_subdomain_name or dp_id

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vlanmaps>
  [
  <vlanmap vlanmap_id="vlanmap_id"
    vlan_id="vlan_id"/>
  ]
</vlanmaps>
```

- When the VLAN map is registered with no-vlan-id specified

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vlanmaps>
  <vlanmap no_vlan_id="no_vlan_id"/>
</vlanmaps>
```

- If the VLAN map is not registered

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vlanmaps/>
```

• JSON format

- When the VLAN map is registered with ofs_subdomain_name specified

```
{
  "vlanmaps" : [
    {
      "vlanmap_id" : "vlanmap_id",
      "ofs_subdomain_name" : "ofs_subdomain_name",
      "vlan_id" : "vlan_id"
    }
  ]
}
```

- When the VLAN map is registered with dp_id specified

```
{
  "vlanmaps" : [
    {
      "vlanmap_id" : "vlanmap_id",
      "dp_id" : "dp_id",
      "vlan_id" : "vlan_id"
    }
  ]
}
```

- When the VLAN map is registered without specifying ofs_subdomain_name or dp_id

```
{
  "vlanmaps" : [
    {
      "vlanmap_id=" : "vlanmap_id"
      "vlan_id" : "vlan_id"
    }
  ]
}
```

- When the VLAN map is registered with no-vlan-id specified

```
{
  "vlanmaps" : [
    {
      "no_vlan_id" : "no_vlan_id"
    }
  ]
}
```

- If the VLAN map is not registered

```
{
  "vlanmaps" : [

  ]
}
```

Table 4-70 Description of Elements in vlanmap

Element	Description	Valid Value
vlanmap_id	vlan-map ID	**ofs_subdomain_name**-HHHH format (H: hexadecimal) when the VLAN map is registered with ofs_subdomain_name specified HHHH-HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number) if dp_id is specified
ofs_subdomain_name	OFS subdomain name	Up to 31 characters including one-byte alphanumeric characters and underscores
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number).
vlan_id	VLAN ID	Decimal number (1 to 4095)
no_vlan_id	VLAN ID flag	{true} *Always true

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.9.3 Unregister VLAN Map

Following describes how to deregister a VLAN map in vBridge.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/vlanmaps/`**`vlanmap_id`**`.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/vlanmaps/`**`vlanmap_id`**`.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `df_id=`**`df_id`**

  `ofs_subdomain_name=`**`ofs_subdomain_name`**

  `vlan_id=`**`vlan_id`**

  `no_vlan_id=`**`no_vlan_id`**

Table 4-71 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
vlanmap_id	VLAN map ID	**dp_id**-HHHH or **ofs_subdomain_name**-HHHH format (H: hexadecimal)
dp_id	OFS datapath ID	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal)
ofs_subdomain_name	OFS subdomain name	Up to 31 characters including one-byte alphanumeric characters and underscores
vlan_id	VLAN ID	Decimal number (1 to 4095)
no_vlan_id	Frames without VLAN ID are handled.	{true} (Lowercase)

Remember

vlanmap_id specified in the URI and dp_id, vlan_id, and no_vlan_id specified in the query character string must match those specified at "4.9.1 Create VLAN Map (page 152)". You can confirm those values in the response body described in "4.9.2 Show VLAN map (page 156)". .

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

When dp_id is specified

- On success

  "Unmap VLAN **vlan_id** to VTN **vtn_name** vBridge **vbr_name** DPID **dp_id** succeeded."

- On failure

  "Unmap VLAN **vlan_id** to VTN **vtn_name** vBridge **vbr_name** DPID **dp_id** failed. (<Error information from pfcshell>)"

When ofs_subdomain_name is specified

- On success

  "Unmap VLAN **vlan_id** to VTN **vtn_name** vBridge **vbr_name** OFS SubDomain **ofs_subdomain_name** succeeded."

- On failure

  "Unmap VLAN **vlan_id** to VTN **vtn_name** vBridge **vbr_name** OFS SubDomain **ofs_subdomain_name** failed. (<Error information from pfcshell>)"

If neither dp_id or ofs_subdomain_name is specified

- When vlan_id is specified
  - On success

    "Unmap VLAN **vlan_id** to VTN **vtn_name** vBridge **vbr_name** succeeded."

  - On failure

    "Unmap VLAN to VTN **vtn_name** of vBridge **vbr_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

- When no_vlan_id is specified
  - On success

    "Unmap VLAN without vlantag to VTN **vtn_name** vBridge **vbr_name** succeeded."

  - On failure

    "Unmap VLAN to VTN **vtn_name** of vBridge **vbr_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

4.9.4 Deleting dynamic mapping information from the VLAN map

Following describes how to delete mapping that has been dynamically learned from the VLAN map function.

Processing request

Method

DELETE

request URI

- XML format

  `/vlanmapinfo.xml`

- JSON format

  `/vlanmapinfo.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `dp_id=`**`dp_id`**

  `port_name=`**`port_name`**

  `vlan_id=`**`vlan_id`**

  `no_vlan_id=`**`no_vlan_id`**

Table 4-72 Description of Parameters in URI

Element	Description	Valid Value
dp_id	datapath ID	HHHH-HHHH-HHHH-HHHH format (H: hexadecimal number). It is not allowed to specify F's to all digits.
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
no_vlan_id	Frames without VLAN ID are handled.	{true} (Lowercase)

Remember

port_name can be omitted. If omitted, the handling of all ports is assumed.

Only when port_name is specified, vlan_id or no_vlan_id can be specified. If omitted, it is assumed that all VLAN IDs are handled (including those frames without VLAN ID).

Only vlan_id or no_vlan_id can be specified

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On success

  "Delete vlan-map of datapath-id(**dp_id**) port-name(**port_name**) vlan-id(**vlan_id**) succeeded."
- On failure

  "Delete vlan-map of datapath-id **dp_id** port-name **port_name** vlan-id **vlan_id** or no_vlan_id **no_vlan_id** failed.(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

4.10 Operate OFS Map Information (vExternal Mode)

4.10.1 Map OFS port to vExternal

Following describes how to register the OFS map information for vExternal.

Processing request

Method

PUT

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/ofsmap.xml
- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/ofsmap.json

Table 4-73 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<ofsmap dp_id="dp_id"
  trunk_port_name="trunk_port_name"
  port_name="port_name"
  vlan_id="vlan_id"
  tagged="tagged"/>
```

- JSON format

```
{
  "ofsmap" : {
    "dp_id" : "dp_id",
    "trunk_port_name" : "trunk_port_name",
    "port_name" : "port_name",
    "vlan_id" : "vlan_id",
    "tagged" : "tagged"
  }
}
```

Table 4-74 Description of Elements in ofsmap

Element	Description	Return Value
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: Hexadecimal number)
trunk_port_name	Trunk port name	Up to 31 characters including one-byte alphanumeric characters and underscores
port_name	Port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
tagged	VLAN Tag	{true|false}

Remember

If port_name contains the following symbols, they must be percent-encoded (converted to "%xx") on the Web client.

/ < > % , { }

Remember

A combination of dp_id and port_name cannot be specified with trunk_port_name at the same time.

Remember

vlan_id and tagged can be omitted.

Remember

If the PFC shell command, vlan-connect enable, is input, vlan_id must be specified to execute this API.

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On success
  - When trunk_port_name is not specified

    "Map OFS port to VTN **vtn_name** vExternal **vex_name** DPID **dp_id** Port Name **port_name** VLANID **vlan_id** **tagged** succeeded."
  - When trunk_port_name is specified

    "Map OFS port to VTN **vtn_name** vExternal **vex_name** TrunkPort **trunk_port_name** VLANID **vlan_id** **tagged** succeeded."
- On failure
  - "Map OFS port from VTN **vtn_name** vExternal **vex_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

4.10.2 Show ofs-map on vExternal

Following describes how to acquire the vExternal physical-virtual mapping information.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/ofsmap.xml
- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/ofsmap.json

Table 4-75 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ofsmap type="type"
  maptype="maptype"
  dp_id="dp_id"
  port_name="port_name"
  trunk_port_name="trunk_port_name"
  vlan_id="vlan_id"
  vlan_tag="vlan_tag"/>
```

- JSON format

```
{
  "ofsmap" : {
    "type" : "type",
    "maptype" : "maptype",
    "dp_id" : "dp_id",
    "port_name" : "port_name",
    "trunk_port_name" : "trunk_port_name",
    "vlan_id" : "vlan_id",
    "vlan_tag" : "vlan_tag"
  }
}
```

Table 4-76 Description of Elements in ofsmap

Element	Description	Return Value
type	Type	{dynamic|static}
maptype	Map type	{ofs-map|vlan-map|mac-map}
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: Hexadecimal number) *This element is not returned if the trunk interface is mapped.
port_name	Trunk port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?) *This element is not returned if the trunk interface is mapped.
trunk_port_name	Port name	Up to 31 characters including one-byte alphanumeric characters and underscores *This element is not returned if the Physics interface is mapped.
vlan_id	VLAN ID	Decimal number (1 to 4095)
vlan_tag	Whether the VLAN tag exists	{tagged|untagged}

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.10.3 Unmap OFS port from vExternal

Following describes how to deregister the virtual external boundary point, which indicates the physical network position information set to vExternal.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/ofsmap.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/ofsmap.json`
- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `dp_id=`**`dp_id`**

  `port_name=`**`port_name`**

  `trunk_port_name=`**`trunk_port_name`**

  `vlan_id=`**`vlan_id`**

  `tagged=`**`tagged`**

Remember

A query string can be omitted.

Table 4-77 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: Hexadecimal number)
trunk_port_name	Port name	Up to 31 characters including one-byte alphanumeric characters and underscores
port_name	Trunk port name	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
tagged	VLAN Tag	{true|false}

Remember

If port_name contains the following symbols, they must be percent-encoded (converted to "%xx") on the Web client.

/ < > % , { }

Remember

A combination of dp_id and port_name cannot be specified with trunk_port_name at the same time.

Remember

vlan_id and tagged can be omitted.

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On success

- When trunk_port_name is not specified

  "Unmap OFS port from VTN **vtn_name** vExternal **vex_name** DPID **dp_id** PortName **port_name** VLANID **vlan_id** **tagged** succeeded."

- When trunk_port_name is specified

  "Unmap OFS port from VTN **vtn_name** vExternal **vex_name** TrunkPort **trunk_port_name** VLANID **vlan_id** **tagged** succeeded."

- On failure
  - "UnMap OFS port from VTN **vtn_name** vExternal **vex_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

4.11 Operate vLink

4.11.1 List vLinks

Following describes how to acquire the list of vLinks under VTN.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vlinks.xml

- JSON format

  /vtns/**vtn_name**/vlinks.json

Table 4-78 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vlinks>
```

```
  [
  <vlink vlk_name="vlk_name"/>
  ]
</vlinks>
```

- JSON format

```
{
  "vlinks" : [
    {
      "vlk_name" : "vlk_name"
    }
  ]
}
```

- On error
  - Insufficient information is returned from pfcshell (vlink name)

    `"*** Error:Internal error. (no vlink)"`
  - Insufficient information is returned from pfcshell (VTN node | interface)

    `"*** Error:Internal error. (no vtn node or interface)"`

Table 4-79 Description of Elements in vlinks

Element	Description	Return Value
vlk_name	vLink name	Up to 31 characters including one-byte alphanumeric characters and underscores

Important

When VLAN mapping (vlan-map command) is specified, the resources that were dynamically generated by this function are prefixed with a question mark (?) and returned to vlk_name. These names cannot be specified for the WebAPI URI path.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

No data is displayed if association with the virtual node is not completed

4.11.2 Create vLink for vBridge

Following describes how to create a new virtual link.

This section also describes how to associate the vBridge node with the virtual node (vRouter, vExternal) and set up the created virtual link.

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/vlinks.xml

- JSON format

  /vtns/**vtn_name**/vlinks.json

Table 4-80 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<vlink vlk_name="vlk_name"
  vbr_name="vbr_name"
  if_name="if_name"
  vtnnode_name="vtnnode_name"
  vtnnode_if_name="vtnnode_if_name"/>
```

- JSON format

```
{
  "vlink" : {
    "vlk_name" : "vlk_name",
    "vbr_name" : "vbr_name",
    "if_name" : "if_name",
    "vtnnode_name" : "vtnnode_name",
    "vtnnode_if_name" : "vtnnode_if_name"
  }
}
```

Table 4-81 Description of Elements in vlink

Element	Description	Return Value
vlk_name	vLink name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name of vBridge	Up to 31 characters including one-byte alphanumeric characters and underscores
vtnnode_name	Virtual node name to be linked	Up to 31 characters including one-byte alphanumeric characters and underscores
vtnnode_if_name	Virtual interface name of virtual node	Up to 31 characters including one-byte alphanumeric characters and underscores

Remember

vlk_name can be omitted. If vlk_name is omitted, an interface is automatically created under a name in the VLink_WA_{**number**} format. The range for **number** is between 1 and 2147483647.

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vlink vlk_name="vlk_name"/>
```

- JSON format

```
{
  "vlink" : {
    "vlk_name" : "vlk_name"
  }
}
```

Table 4-82 Description of Elements in vlink

Element	Description	Return Value
vlk_name	vLink name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Create vLink to VTN **vtn_name** vBridge **vbr_name** succeeded."

- On failure

  "Create vLink to VTN **vtn_name** vBridge **vbr_name** failed. (<Error information from pfcshell>)"

Remark

None

4.11.3 Show vLink

Following describes how to acquire the vLink information.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vlinks/**vlk_name**.xml

- JSON format

  /vtns/**vtn_name**/vlinks/**vlk_name**.json

Table 4-83 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vlk_name	vLink name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vlink vlk_name="vlk_name"
  vnode1name="vnode1name"
  interface1name="interface1name"
  vnode2name="vnode2name"
  interface2name="interface2name"/>
```

- JSON format

```
{
  "vlink" : {
    "vlk_name " : "vlk_name",
    "vnode1name" : "vnode1name",
    "interface1name" : "interface1name",
    "vnode2name" : "vnode2name",
    "interface2name" : "interface2name"
  }
}
```

- On error
  - Insufficient information is returned from pfcshell (VTN node | interface)

    "*** Error:Internal error. (no vtn node or interface)"

Table 4-84 Description of Parameters in vlink

Element	Description	Valid Value
vlk_name	vLink name	Up to 31 characters including one-byte alphanumeric characters and underscores
vnode1name	vNode1 name	Up to 31 characters including one-byte alphanumeric characters and underscores
interface1name	interface1 name	Up to 31 characters including one-byte alphanumeric characters and underscores
vnode2name	vNode2 name	Up to 31 characters including one-byte alphanumeric characters and underscores
interface2name	interface2 name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.11.4 Show vLink details

Following describes how to acquire detailed information including statistical information and link status about the vLink.

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vlinks/`**`vlk_name`**`/detail.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vlinks/`**`vlk_name`**`/detail.json`

Table 4-85 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vlk_name	vLink name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<vlink vlk_name="vlk_name"
  status="status"
  vnode1name="vnode1name"
  interface1name="interface1name"
  vnode2name="vnode2name"
  interface2name="interface2name">
  <statistics>
    <vnode1tovnode2 packets="packets"
      octets="octets"/>
    <vnode2tovnode1 packets="packets"
      octets="octets"/>
  </statistics>
</vlink>
```

- JSON format

```
{
  "vlink" : {
    "vlk_name" : "vlk_name",
    "status" : "status",
    "vnode1name" : "vnode1name",
    "interface1name" : "interface1name",
    "vnode2name" : "vnode2name",
    "interface2name" : "interface2name",
    "statistics" : {
      "vnode1tovnode2" : {
        "packets" : "packets",
        "octets" : "octets"
      },
      "vnode2tovnode1" : {
        "packets" : "packets",
        "octets" : "octets"
      }
    }
  }
}
```

- On error
  - Insufficient information is returned from pfcshell (vlink name)

    `"*** Error:Internal error. (no vlink)"`

  - Insufficient information is returned from pfcshell (Status)

    `"*** Error:Internal error. (no vlink status)"`

- Insufficient information is returned from pfcshell (VTN node | interface)

```
"*** Error:Internal error. (no vtn node or interface)"
```

Table 4-86 Description of Elements in vlink

Element	Description	Return Value
vlk_name	vLink name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	vLink status	{up|down}
vnode1name	vNode1 name	Up to 31 characters including one-byte alphanumeric characters and underscores
interface1name	interface1 name	Up to 31 characters including one-byte alphanumeric characters and underscores
vnode2name	vNode2 name	Up to 31 characters including one-byte alphanumeric characters and underscores
interface2name	interface2 name	Up to 31 characters including one-byte alphanumeric characters and underscores
statistics	vLink Statistical information	statistics (For details, refer to "Table 4-87 Description of Elements in statistics (page 175)")

Table 4-87 Description of Elements in statistics

Element	Description	Return Value
vnode1tovnode2	Communication information from vNode1 to vNode2	statistics type (For details, refer to "Table 4-88 Description of Elements in statistics (page 175)")
vnode2tovnode1	Communication information from vNode2 to vNode1	statistics type (For details, refer to "Table 4-88 Description of Elements in statistics (page 175)")

Table 4-88 Description of Elements in statistics

Element	Description	Return Value
packets	Number of packets	counter
octets	Number of bytes	counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

4.11.5 Delete vLink

Following describes how to delete a vLink.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vlinks/`**`vlk_name`**`.xml`

- JSON format

  `/vtns/`**`vtn_name`**`/vlinks/`**`vlk_name`**`.json`

Table 4-89 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vlk_name	vLink name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Delete vLink` **`vlk_name`** `of VTN` **`vtn_name`** `succeeded."`

- On failure

  "Delete vLink **vlk_name** of VTN **vtn_name** failed.(<Error information from pfcshell>)"

Remark

None

4.12 Operate L2 Domain Configuration Information

4.12.1 Show L2 domains

Following describes how to acquire the L2 domain configuration information.

Processing request

Method

GET

request URI

- XML format

  /l2domains.xml

- JSON format

  /l2domains.json

- Parameter

  The following parameters are passed in the URI query character string (in the format of ?param1=***&param2=***).

  vtn_name=**vtn_name**

  vbr_name=**vbr_name**

Table 4-90 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<l2domains count="count">
  [
  <l2domain domain_id="domain_id">
    <vtninfo vtn_name="vtn_name" vbr_name="vbr_name"/>
    <realnetworkinfos>
      [
      <realnetworkinfo dp_id="dp_id" vlan_id="vlan_id"/>
      ]
    </realnetworkinfos>
  </l2domain>
  ]
</l2domains>
```

- JSON format

```
{
  "count" : "count",
  "l2domains" : [
    {
      "domain_id" : "domain_id",
      "vtninfo" : {
        "vtn_name" : "vtn_name",
        "vbr_name" : "vbr_name"
      },
      "realnetworkinfos" : [
        {
          "dp_id" : "dp_id",
          "vlan_id" : "vlan_id"
        }
      ]
    }
  ]
}
```

Table 4-91 Description of Elements in l2domains

Element	Description	Return Value
count	Number of L2Domain	Decimal number (1 to 65536)
domain_id	L2Domain ID	Decimal number (1 to 65536)
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
dp_id	OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number).
vlan_id	VLAN ID	Decimal number (1 to 4095)

HTTP status code for response

- On success

  200 (OK)

- On failure

400, 500

Miscellaneous

Log

- On failure

  "Get l2domain failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

Chapter 5. WebAPI (L2/L3-Related) Reference

Following describes the URI, parameters, request/response data and other details about the L2/L3-related API.

Tip

The WebAPI uses structure data in the XML or JSON format as API input/output. In this chapter, for each API input/output data unit, the data elements on the same structure level is summarized in a table and the element name is written in bold like **"data."** If a lower level has data, the relevant level name is written in plain text like "data." The reference table number and other details about this data are shown in the "Return value" column.

5.1 Operate VTN Topology

5.1.1 Show topologies for VTN

Following describes how to acquire the information on the topology built in the VTN.

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/topologies.xml`

- JSON format

  `/vtns/`**`vtn_name`**`/topologies.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `vbr_name=`**`vbr_name`**

  `vrt_name=`**`vrt_name`**

  `vex_name=`**`vex_name`**

  `if_name=`**`if_name`**

Remember

Only one of vbr_name, vrt_name, and vex_name can be specified.

if_name can be specified only when vbr_name or vrt_name is specified.

Table 5-1 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<topologies>
  [
  <topology vtnnode="vtnnode"
    type="type"
    status="status"/>
  <interfaces>
    [
    <interface if_name="if_name"
      status="status"
      connected_vnode="connected_vnode"
      connected_vnode_interface="connected_vnode_interface"
      dp_id="dp_id"
      port_name="port_name"
      trunk_port_name="trunk_port_name"
      vlan_id="vlan_id"
      vlantag="vlantag"/>
    ]
  </interfaces>
  ]
</topologies>
```

- JSON format

```
{
  "topologies": [
    {
      "vtnnode": "vtnnode",
      "type": "type",
      "status": "status",
      "interfaces": [
        {
          "if_name": "if_name",
          "status": "status",
          "connected_vnode": "connected_vnode",
          "connected_vnode_interface": "connected_vnode_interface",
          "dp_id": "dp_id",
```

```
            "port_name": "port_name",
            "trunk_port_name": "trunk_port_name",
            "vlan_id": "vlan_id",
            "vlantag": "vlantag"
          }
        ]
      }
    ]
}
```

Table 5-2 Description of Elements in topologies

Element	Description	Return Value
vtnnode	Node name	Up to 31 characters including one-byte alphanumeric characters and underscores
type	vNode type	{vbridge|vexternal|vrouter}
status	vNode status	{up|down}
interfaces	Interface list	interface type (For details, refer to "Table 5-3 Description of Elements in interface Type (page 182)")

Table 5-3 Description of Elements in interface Type

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	Interface status	{up|down}
connected_vnode	Connected vNode name	Up to 31 characters including one-byte alphanumeric characters and underscores
connected_vnode_interface	Connected vNode interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
dp_id	Destination OFS datapath ID	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number) * Not returned if the trunk interface is mapped.
port_name	Destination port name	Up to 16 characters including ascii alphanumeric characters except for a question mark (?) * Not returned if the trunk interface is mapped.
trunk_port_name	Trunk port name	Up to 31 characters including one-byte alphanumeric characters and underscores * Not returned if the trunk interface is mapped.
vlan_id	VLAN ID	Decimal number (1 to 4095)
vlantag	Whether the VLAN tag exists	{tagged|untagged}

Important

When VLAN mapping (vlan-map command) is specified, the resources that were dynamically generated by this function are prefixed with a question mark (?) and returned to vtnnode and if_name. These names cannot be specified for the WebAPI URI path.

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On failure

  "Get topology failed. [Invalid argument](<Error information from pfc shell>)"

  "Get topology failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

5.2 Operate vExternal

5.2.1 List vExternals

Following describes how to acquire the list of vExternals under the VTN.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals.xml

- JSON format

  /vtns/**vtn_name**/vexternals.json

Table 5-4 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric

Element	Description	Valid Value
		characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vexternals>
  [
  <vexternal vex_name="vex_name"/>
  ]
</vexternals>
```

- JSON format

```
{
  "vexternals": [
    {
      "vex_name": "vex_name"
    }
  ]
}
```

Table 5-5 Description of Elements in vexternals

Element	Description	Return Value
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Important

When VLAN mapping (vlan-map command) is specified, the resources that were dynamically generated by this function are prefixed with a question mark (?) and returned to vtnnode and if_name. These names cannot be specified for the WebAPI URI path.

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

5.2.2 Create vExternal

Following describes how to create a new vExternal node.

Processing request

Method

POST

request URI

- XML format

  `/vtns/vtn_name/vexternals.xml`

- JSON format

  `/vtns/vtn_name/vexternals.json`

Table 5-6 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<vexternal vex_name="vex_name"/>
```

- JSON format

```
{
  "vexternal" : {
    "vex_name" : "vex_name"
  }
}
```

Table 5-7 Description of Elements in vexternal

Element	Description	Valid Value
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Processing result

Details of response body

None

HTTP status code for response

- On success

200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Create vExternal vex_name to VTN vtn_name succeeded."`

- On failure

  `"Create vExternal vex_name to VTN vtn_name failed.(<Error information from pfcshell>)"`

Remark

None

5.2.3 Show vExternal

Following describes how to acquire the information on vExternals.

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/vexternals/vex_name.xml`

- JSON format

  `/vtns/vtn_name/vexternals/vex_name.json`

Table 5-8 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vexternal vex_name="vex_name">
  <interface if_name="if_name"/>
</vexternal>
```

- JSON format

```
{
  "vexternal": {
    "vex_name": "vex_name",
    "interface": {
      "if_name": "if_name"
    }
  }
}
```

Table 5-9 Description of Elements in vexternal

Element	Description	Return Value
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

5.2.4 Show vExternal details

Following describes how to acquire the detailed information on vExternals.

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/vexternals/vex_name/detail.xml`

- JSON format

  `/vtns/vtn_name/vexternals/vex_name/detail.json`

Table 5-10 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<vexternal vex_name="vex_name" status="status">
  <interface if_name="if_name"
    status="status">
    <statistics
      rx_packets="rx_packets"
      tx_packets="tx_packets"
      rx_octets="rx_octets"
      tx_octets="tx_octets"/>
  </interface>
</vexternal>
```

- JSON format

```
{
  "vexternal": {
    "vex_name": "vex_name",
    "status": "status",
    "interface": {
      "if_name": "if_name",
      "status": "status",
      "statistics": {
        "rx_packets": "rx_packets",
        "tx_packets": "tx_packets",
        "rx_octets": "rx_octets",
        "tx_octets": "tx_octets"
      }
    }
  }
}
```

Table 5-11 Description of Elements in vexternal

Element	Description	Return Value
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Return Value
status	vExternal status	{up|down}
interface	Virtual interface set to vExternal	interface type (For details, refer to "Table 5-12 Description of Elements in interface Type (page 189)")

Table 5-12 Description of Elements in interface Type

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	Interface status	{up|down}
rx_packets	Number of received packets	Counter
tx_packets	Number of transferred packets	Counter
rx_octets	Number of received bytes	Counter
tx_octets	Number of transferred bytes	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

5.2.5 Delete vExternal

Following describes how to delete the vExternal.

Processing request

Method

DELETE

request URI

- XML format

/vtns/**vtn_name**/vexternals/**vex_name**.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**.json

Table 5-13 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Delete vExternal **vex_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete vExternal **vex_name** of VTN **vtn_name** failed.(<Error informatio n from pfcshell>)"

Remark

None

5.2.6 Acquire the vExternal additional information list

Following describes how to acquire the additional information set to vExternal.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/additional_infos.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/additional_infos.json

Table 5-14 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<additionals>
  [
  <additional additional_info="additional_info"/>
  ]
</additionals>
```

- JSON format

```
{
  "additionals" : [
    {
      "additional_info" : "additional_info"
    }
  ]
}
```

Table 5-15 Description of Elements in additionals

Element	Description	Return Value
additional_info	Additional information set to vExternal	Up to 127 characters including ascii alphanumeric characters except for a question mark (?)

HTTP status code for response

- On success

200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

5.2.7 Add or delete the vExternal additional information

Following describes how to add or delete the additional information to/from vExternal.

Processing request

Method

PUT

request URI

- XML format

  `/vtns/vtn_name/vexternals/vex_name/additional_infos.xml`

- JSON format

  `/vtns/vtn_name/vexternals/vex_name/additional_infos.json`

Table 5-16 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<additional
  op="op"
  additional_info="additional_info"/>
```

- JSON format

```
{
  "additional" : {
    "op" : "op",
    "additional_info" : "additional_info"
  }
}
```

Table 5-17 Description of Elements in additional

Element	Description	Valid Value
op	op status information	{add|delete}
additional_info	Additional information set to vExternal	Up to 127 characters including ascii alphanumeric characters except for a question mark (?)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Update(op) additional_info "additional_info" for vExternal vex_name of VTN vtn_namesucceeded."`
- On failure

  `"Update(op) additional_info for vExternal vex_name of VTN vtn_name failed. [Invalid argument(additional_info)](<Error information from pfcshell>)"`

Remark

None

5.3 Operate Interface (vExternal Mode)

5.3.1 List interfaces for vExternal

Following describes how to acquire the list of vExternal virtual interfaces.

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/vexternals/vex_name/interfaces.xml`

- JSON format

  `/vtns/vtn_name/vexternals/vex_name/interfaces.json`

Table 5-18 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"/>
```

- JSON format

```
{
  "interface": {
    "if_name": "if_name"
  }
}
```

Table 5-19 Description of Elements in interface

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

5.3.2 Create interface for vExternal

Following describes how to create a vExternal virtual interface.

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces.json

Table 5-20 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<interface if_name="if_name"/>
```

- JSON format

```
{
  "interface" : {
    "if_name" : "if_name"
  }
}
```

Table 5-21 Description of Elements in interface

Element	Description	Valid Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Remember

if_name can be omitted.

Remember

If if_name is omitted, an interface is automatically created under a name in the IF_WA_{**number**} format. The range for **number** is between 1 and 2147483647.

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"/>
```

- JSON format

```
{
  "interface" : {
    "if_name" : "if_name"
  }
}
```

Table 5-22 Description of Elements in interface

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Create interface **if_name** of vExternal **vex_name**of VTN **vtn_name**succeeded."

- On failure

  "Create interface of vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument(if_name)](<Error information from pfcshell>)"

Remark

None

5.3.3 Show interface for vExternal

Following describes how to acquire the information on the vExternal virtual interface.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**.json

Table 5-23 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"/>
```

- JSON format

```
{
  "interface": {
    "if_name": "if_name"
  }
}
```

Table 5-24 Description of Elements in interface

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

400, 500

Miscellaneous

Log

None

Remark

None

5.3.4 Show interface details for vExternal

Following describes how to acquire the (detailed) information on the vExternal virtual interface.

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/interfaces/`**`if_name`**`/detail.xml`

- JSON format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/interfaces/`**`if_name`**`/detail.json`

Table 5-25 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"
  status="status">
  <statistics
    rx_packets="rx_packets"
    tx_packets="tx_packets"
```

```
    rx_octets="rx_octets"
    tx_octets="tx_octets"/>
</interface>
```

- JSON format

```
{
  "interface": {
    "if_name": "if_name",
    "status": "status",
    "statistics": {
      "rx_packets": "rx_packets",
      "tx_packets": "tx_packets",
      "rx_octets": "rx_octets",
      "tx_octets": "tx_octets"
    }
  }
}
```

Table 5-26 Description of Elements in interface

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	vExternal status	{up|down}
statistics	Statistical information	statistics type (For details, refer to "Table 5-27 Description of Elements in statistics Type (page 199)")

Table 5-27 Description of Elements in statistics Type

Element	Description	Return Value
rx_packets	Number of received packets	Counter
tx_packets	Number of transferred packets	Counter
rx_octets	Number of received bytes	Counter
tx_octets	Number of transferred bytes	Counter

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On failure

  "Get interface **if_name** detail info for vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument(if_name)](<Error information from pfcshell>)"

Remark

None

5.3.5 Delete interface for vExternal

Following describes how to delete a virtual interface under vExternal.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/vtn_name/vexternals/vex_name/interfaces/if_name.xml`

- JSON format

  `/vtns/vtn_name/vexternals/vex_name/interfaces/if_name.json`

Table 5-28 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Delete interface if_name of VTN vtn_name vExternal vex_name succeeded."`

- On failure

  "Delete interface **if_name** of VTN **vtn_name** vExternal **vex_name** failed. (<Error information from pfcshell>)"

Remark

None

Chapter 6. WebAPI (L3-Related) Reference

Following describes the URI, parameters, request/response data and other details about the L3-related API.

Tip

The WebAPI uses structure data in the XML or JSON format as API input/output. In this chapter, for each API input/output data unit, the data elements on the same structure level is summarized in a table and the element name is written in bold like **" data."** If a lower level has data, the relevant level name is written in plain text like "data." The reference table number and other details about this data are shown in the "Return value" column.

6.1 Operate vRouter

6.1.1 List vRouters

Following describes how to acquire the list of vRouters under the VTN.

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters.xml`

- JSON format

  `/vtns/`**`vtn_name`**`/vrouters.json`

Table 6-1 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vrouters>
  [
```

```
  <vrouter vrt_name="vrt_name"/>
  ]
</vrouters>
```

- JSON format

```
{
  "vrouters": [
    {
      "vrt_name": "vrt_name"
    }
  ]
}
```

Table 6-2 Description of Elements in vrouters

Element	Description	Return Value
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.1.2 Create vRouter

Following describes how to create a vRouter node on the VTN.

Processing request

Method

POST

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters.xml`

- JSON format

  `/vtns/`**`vtn_name`**`/vrouters.json`

Table 6-3 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<vrouter vrt_name="vrt_name"/>
```

- JSON format

```
{
  "vrouter" : {
    "vrt_name" : "vrt_name"
  }
}
```

Table 6-4 Description of Elements in vrouter

Element	Description	Valid Value
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Create vRouter **vrt_name** for VTN **vtn_name** succeeded."

- On failure

  "Create vRouter for VTN **vtn_name** failed. [Invalid argument(vrt_name)](<Error information from pfcshell>)"

Remark

None

6.1.3 Show vRouter

Following describes how to acquire the information on vRouters.

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/vrouters/vrt_name.xml`

- JSON format

  `/vtns/vtn_name/vrouters/vrt_name.json`

Table 6-5 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vrouter vrt_name="vrt_name">
  <interfaces>
    [
    <interface if_name="if_name"/>
    ]
  </interfaces>
</vrouter>
```

- JSON format

```
{
  "vrouter": {
    "vrt_name": "vrt_name",
    "interfaces": [
      {
        "if_name": "if_name"
      }
    ]
  }
}
```

Table 6-6 Description of Elements in vrouter

Element	Description	Return Value
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
interfaces	Interface list	interface type (For details, refer to "Table 6-7 Description of Elements in interface Type (page 206)")

Table 6-7 Description of Elements in interface Type

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.1.4 Show vRouter details

Following describes how to acquire the detailed information on the vRouter including statistical information.

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/detail.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/detail.json`

Table 6-8 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vrouter vrt_name="vrt_name"
  status="status">
  <interfaces>
    [
    <interface if_name="if_name"
      status="status">
      <rx>
        <all packets="packets" octets="octets"/>
      </rx>
      <tx>
        <all packets="packets" octets="octets"/>
        <forwarding packets="packets" octets="octets"/>
        <host packets="packets" octets="octets"/>
      </tx>
    </interface>
    ]
  </interfaces>
  <drop>
    <all packets="packets" octets="octets"/>
    <noroute packets="packets" octets="octets"/>
    <timeout packets="packets" octets="octets"/>
    <otherreason packets="packets" octets="octets"/>
  </drop>
</vrouter>
```

- JSON format

```
{
  "vrouter": {
    "vrt_name": "vrt_name",
    "status": "status",
    "interfaces": [
      {
        "if_name": "if_name",
        "status": "status",
        "rx": {
          "all": {
            "packets": "packets",
            "octets": "octets"
          }
        },
        "tx": {
          "all": {
            "packets": "packets",
            "octets": "octets"
          },
```

```
          "forwarding": {
            "packets": "packets",
            "octets": "octets"
          },
          "host": {
            "packets": "packets",
            "octets": "octets"
          }
        }
      }
    ],
    "drop": {
      "all": {
        "packets": "packets",
        "octets": "octets"
      },
      "noroute": {
        "packets": "packets",
        "octets": "octets"
      },
      "timeout": {
        "packets": "packets",
        "octets": "octets"
      },
      "otherreason": {
        "packets": "packets",
        "octets": "octets"
      }
    }
  }
}
```

Table 6-9 Description of Elements in vrouter

Element	Description	Return Value
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	vRouter status	{up|down}
interfaces	Interface list	interface type (For details, refer to "Table 6-10 Description of Elements in interface Type (page 208)")
drop	Drop information	drop type (For details, refer to "Table 6-13 Description of Elements in drop Type (page 209)")

Table 6-10 Description of Elements in interface Type

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	Interface status	{up|down}
rx	Reception	rx type (For details, refer to "Table 6-11 Description of Elements in rx Type (page 208)")
tx	Transmission	tx type (For details, refer to "Table 6-12 Description of Elements in tx Type (page 209)")

Table 6-11 Description of Elements in rx Type

Element	Description	Return Value
all	Total number of transmission on vRouter	statisticsType (For details, refer to "Table 6-14 Description of Elements in statistics Type (page 209)")

Table 6-12 Description of Elements in tx Type

Element	Description	Return Value
all	Total number of transmission on vRouter	statistics type (For details, refer to "Table 6-14 Description of Elements in statistics Type (page 209)")
forwarding	Number of transmissions forwarded on vRouter	statistics type (For details, refer to "Table 6-14 Description of Elements in statistics Type (page 209)")
host	Number of transmissions forwarded to the host on vRouter	statistics type (For details, refer to "Table 6-14 Description of Elements in statistics Type (page 209)")

Table 6-13 Description of Elements in drop Type

Element	Description	Return Value
all	Total number of packets discarded when going through vRouter	statistics type (For details, refer to "Table 6-14 Description of Elements in statistics Type (page 209)")
noroute	Number of packets discarded because there is no routing table when going through vRouter	statistics type (For details, refer to "Table 6-14 Description of Elements in statistics Type (page 209)")
timeout	Number of packets discarded due to the timeout of a lower-level protocol when going through vRouter	statistics type (For details, refer to "Table 6-14 Description of Elements in statistics Type (page 209)")
otherreason	Number of packets discarded due to some other reason when going through vRouter	statistics type (For details, refer to "Table 6-14 Description of Elements in statistics Type (page 209)")

Table 6-14 Description of Elements in statistics Type

Element	Description	Return Value
octets	Number of bytes	Counter
packets	Number of packets	Counter

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.1.5 Delete vRouter

Following describes how to delete the vRouter.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/vtn_name/vrouters/vrt_name.xml`

- JSON format

  `/vtns/vtn_name/vrouters/vrt_name.json`

Table 6-15 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Delete vRouter vrt_name of VTN vtn_name succeeded."`

- On failure

  `"Delete vRouter vrt_name of VTN vtn_name failed.(<Error information from pfcshell>)"`

Remark

None

6.2 Operate DHCP Relay

6.2.1 Show DHCP relay

Following describes how to acquire the status of DHCP relay.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/dhcprelay.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/dhcprelay.json

Table 6-16 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dhcprelay dhcp_relay_status="dhcp_relay_status"/>
```

- JSON format

```
{
  "dhcprelay": {
    "dhcp_relay_status": "dhcp_relay_status"
  }
}
```

Table 6-17 Description of Elements in dhcprelay

Element	Description	Return Value
dhcp_relay_status	dhcprelay status	{active|inactive}

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.2.2 Enable/Disable DHCP relay

Following describes how to enable or disable the DHCP relay.

Processing request

Method

PUT

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/dhcprelay.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/dhcprelay.json`

Table 6-18 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<dhcprelay dhcp_relay_status="dhcp_relay_status"/>
```

- JSON format

```
{
  "dhcprelay" : {
    "dhcp_relay_status" : "dhcp_relay_status"
  }
}
```

Table 6-19 Description of Elements in dhcprelay

Element	Description	Valid Value
dhcp_relay_status	DHCP relay status	{enable|disable}

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- When dhcp_relay_status is enable
  - On success

    `"Enable dhcp-realy for vRouter` **`vrt_name`** `of VTN` **`vtn_name`** `succeeded."`
  - On error

    `"Enable dhcp-realy for vRouter` **`vrt_name`** `of VTN` **`vtn_name`** `failed."`

    (<Error information from pfcshell>)"
- When dhcp_relay_status is disable
  - On success

    `"Disable dhcp-realy for vRouter` **`vrt_name`** `of VTN` **`vtn_name`** `succeeded."`
  - On error

    `"Disable dhcp-realy for vRouter` **`vrt_name`** `of VTN` **`vtn_name`** `failed."`

    (<Error information from pfcshell>)"

Remark

None

6.2.3 List interface for DHCP relay

Following describes how to acquire the list of DHCP relay interfaces.

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/vrouters/vrt_name/dhcprelay/interfaces.xml`

- JSON format

  `/vtns/vtn_name/vrouters/vrt_name/dhcprelay/interfaces.json`

Table 6-20 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interfaces>
  [
  <interface if_name="if_name"/>
  ]
</interfaces>
```

- JSON format

```
{
  "interfaces": [
    {
      "if_name": "if_name"
    }
  ]
}
```

Table 6-21 Description of Elements in interfaces

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric

Element	Description	Return Value
		characters and underscores

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.2.4 Register interface for DHCP relay

Following describes how to register the interface that accepts requests from the DHCP client with vRouter.

Processing request

Method

POST

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/dhcprelay/interfaces.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/dhcprelay/interfaces.json`

Table 6-22 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<interface if_name="if_name"/>
```

- JSON format

```
{
  "interface" : {
    "if_name" : "if_name"
  }
}
```

Table 6-23 Description of Elements in interface

Element	Description	Valid Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Register interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** for DHCP relay succeeded."

- On failure

  "Register interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** for DHCP relay failed. (<Error information from pfcshell>)"

Remark

None

6.2.5 Unregister interface for DHCP relay

Following describes how to deregister the DHCP relay interface set to the vRouter.

Processing request

Method

DELETE

request URI

- XML format

/vtns/**vtn_name**/vrouters/**vrt_name**/dhcprelay/interfaces/**if_name**.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/dhcprelay/interfaces/**if_name**.json

Table 6-24 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Unregister interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** fo r relay succeeded"

- On failure

  "Unregister interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** fo r relay failed.(<Error information from pfcshell>)"

Remark

None

6.2.6 List Servers for DHCP relay

Following describes how to acquire the list of DHCP servers used by the DHCP relay.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/dhcprelay/servers.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/dhcprelay/servers.json

Table 6-25 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<servers>
  [
  <server ipaddr="ipaddr"/>
  ]
</servers>
```

- JSON format

```
{
  "servers": [
    {
      "ipaddr": "ipaddr"
    }
  ]
}
```

Table 6-26 Description of Elements in servers

Element	Description	Return Value
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)

HTTP status code for response

- On success

200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.2.7 Register Server for DHCP relay

Following describes how to register the IP address of the DHCP server used by the DHCP relay with the vRouter.

Processing request

Method

POST

request URI

- XML format

  `/vtns/vtn_name/vrouters/vrt_name/dhcprelay/servers.xml`

- JSON format

  `/vtns/vtn_name/vrouters/vrt_name/dhcprelay/servers.json`

Table 6-27 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<server ipaddr="ipaddr"/>
```

- JSON format

```
{
  "server" : {
    "ipaddr" : "ipaddr"
  }
}
```

Table 6-28 Description of Elements in server

Element	Description	Valid Value
ipaddr	Server IP address	IPv4 dot-separated format (Example: 192.168.1.1)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Register server **ipaddr** of vRouter **vrt_name** of VTN **vtn_name** for DHCP relay succeeded."

- On failure

  "Register server **ipaddr** of vRouter **vrt_name** of VTN **vtn_name** for DHCP relay failed. (<Error information from pfcshell>)"

Remark

None

6.2.8 Unregister Server for DHCP relay

Following describes how to deregister the IP address of the DHCP server used by the DHCP relay set to the vRouter.

Processing request

Method

DELETE

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/dhcprelay/servers/**ipaddr**.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/dhcprelay/servers/**ipaddr**.json

Table 6-29 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On success

  "Unregister server **ipaddr** of vRouter **vrt_name** of VTN **vtn_name** for relay succeeded."
- On failure

  "Unregister server **ipaddr** of vRouter **vrt_name** of VTN **vtn_name** for relay failed.(<Error information from pfcshell>)"

Remark

None

6.3 Operate Interface (vRouter Mode)

6.3.1 List interfaces for vRouter

Following describes how to acquire the list of vRouter interfaces.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces.json

Table 6-30 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interfaces>
  [
  <interface if_name="if_name"/>
  ]
</interfaces>
```

- JSON format

```
{
  "interfaces": [
    {
      "if_name": "if_name"
    }
  ]
}
```

Table 6-31 Description of Elements in interfaces

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.3.2 Create interface for vRouter

Following describes how to create a virtual interface on the vRouter.

Processing request

Method

POST

request URI

- XML format

  `/vtns/vtn_name/vrouters/vrt_name/interfaces.xml`

- JSON format

  `/vtns/vtn_name/vrouters/vrt_name/interfaces.json`

Table 6-32 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<interface if_name="if_name"/>
```

- JSON format

```
{
  "interface" : {
    "if_name" : "if_name"
  }
}
```

Table 6-33 Description of Elements in interface

Element	Description	Valid Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Remember

if_name can be omitted.

Remember

If if_name is omitted, an interface is automatically created with a name in IF_WA_{number}. **number** is in the range from 1 to 2147483647.

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"/>
```

- JSON format

```
{
  "interface" : [
    {
      "if_name" : "if_name"
    }
  ]
}
```

Table 6-34 Description of Elements in interface

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Create interface **if_name** for vRouter **vrt_name** succeeded."

- On failure

```
"Create interface for vRouter vrt_name for VTN vrn_name failed. [Inv
alid argument(if_name)](<Error information from pfcshell>)"
```

Remark

None

6.3.3 Show interface for vRouter

Following describes how to acquire the vRouter interface information.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**.json

Table 6-35 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"
  ifindex="ifindex"
  ipaddr="ipaddr"
  macaddr="macaddr"
  netmask="netmask"
  mtu="mtu"/>
```

- JSON format

```
{
  "interface": {
    "if_name": "if_name",
    "ifindex": "ifindex",
    "ipaddr": "ipaddr",
    "macaddr": "macaddr",
    "netmask": "netmask",
    "mtu": "mtu"
  }
}
```

Table 6-36 Description of Elements in interface

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ifindex	Interface index	Decimal number
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)
macaddr	MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
netmask	Subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)
mtu	Maximum data size that can be transferred at a time	Decimal number (1 to 16000)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.3.4 Show interface details for vRouter

Following describes how to acquire the (detailed) vRouter interface information.

Processing request

Method

GET

request URI

- XML format

/vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/detail.xml

- JSON format

/vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/detail.json

Table 6-37 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"
  ifindex="ifindex"
  adminstatus="adminstatus"
  operstatus="operstatus"
  ipaddr="ipaddr"
  macaddr="macaddr"
  netmask="netmask"
  mtu="mtu"/>
```

- JSON format

```
{
  "interface": {
    "if_name": "if_name",
    "ifindex": "ifindex",
    "adminstatus": "adminstatus",
    "operstatus": "operstatus",
    "ipaddr": "ipaddr",
    "macaddr": "macaddr",
    "netmask": "netmask",
    "mtu": "mtu"
  }
}
```

Table 6-38 Description of Elements in interface

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ifindex	Interface index	Decimal number
adminstatus	Port status	{up|down}
operstatus	Link status	{up|down}
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)

Element	Description	Return Value
macaddr	MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
netmask	Subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)
mtu	Maximum data size that can be transferred at a time	Decimal number (1 to 16000)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.3.5 Delete interface for vRouter

Following describes how to delete a virtual interface under vRouter.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/interfaces/`**`if_name`**`.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/interfaces/`**`if_name`**`.json`

Table 6-39 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Delete interface **if_name** for vRouter **vrt_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete interface **if_name** for vRouter **vrt_name** of VTN **vtn_name** failed.(<Error information from pfcshell>)"

Remark

None

6.3.6 Show IP address to interface

Following describes how to acquire the IP address of the interface defined in the vRouter.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/ipaddress.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/ipaddress.json

Table 6-40 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"
  ipaddr="ipaddr"
  netmask="netmask"/>
```

- JSON format

```
{
  "interface": {
    "if_name": "if_name",
    "ipaddr": "ipaddr",
    "netmask": "netmask"
  }
}
```

Table 6-41 Description of Elements in interface

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)
netmask	Subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.3.7 Register IP address to interface

Following describes how to register the IP address with the vRouter interface.

Processing request

Method

PUT

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/ipaddress.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/ipaddress.json

Table 6-42 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format
  - When specifying prefix

```
<interface ipaddr="ipaddr" prefix="prefix"/>
```

  - When specifying netmask

```
<interface ipaddr="ipaddr" netmask="netmask"/>
```

- JSON format
  - When specifying prefix

```
{
  "interface":{
    "ipaddr" : "ipaddr",
    "prefix" : "prefix"
  }
}
```

  - When specifying netmask

```
{
  "interface" : {
    "ipaddr" : "ipaddr",
    "netmask" : "netmask"
  }
}
```

Table 6-43 Description of Elements in interface

Element	Description	Valid Value
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)
prefix	IP address prefix	Decimal number (1 to 30)
netmask	Subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)

Remember

ipaddr is required.

Remember

Either prefix or netmask must be always specified.

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On success

  "Register IP address **ipaddr** to interface **if_name** of VTN **vtn_name** vRouter **vrt_name** succeeded."
- On failure

  "Register IP address of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(netmask or prefix)](<Error information from pfcshell>)"

Remark

None

6.3.8 Unregister IP address from interface (vRouter mode)

Following describes how to deregister the IP address of the virtual interface set under the vRouter.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/interfaces/`**`if_name`**`/ipaddress.xml`

- JSON format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/interfaces/`**`if_name`**`/ipaddress.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `ipaddr=`**`ipaddr`**

  `prefix=`**`prefix`**

  `netmask=`**`netmask`**

Table 6-44 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)
prefix	IP address prefix	Decimal number (1 to 30)
netmask	Subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)

Remember

ipaddr is required.

Remember

Either prefix or netmask must be always specified.

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On success

  "Unregister IP address **ipaddr** from interface **if_name** of VTN **vtn_name** vRouter **vrt_name** succeeded."
- On failure

  "Unregister IP address of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(netmask or prefix)](<Error information from pfcshell>)"

Remark

None

6.3.9 Show MAC address to interface (vRouter mode)

Following describes how to acquire the MAC address of the interface defined in the vRouter.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/macaddress.xml
- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/macaddress.json

Table 6-45 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Valid Value
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<interface if_name="if_name"
  macaddr="macaddr"/>
```

- JSON format

```
{
  "interface": {
    "if_name": "if_name",
    "macaddr": "macaddr"
  }
}
```

Table 6-46 Description of Elements in interface

Element	Description	Return Value
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
macaddr	MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.3.10 Register MAC address to interface (vRouter mode)

Following describes how to register the MAC address with the vRouter interface.

Processing request

Method

PUT

request URI

- XML format

  `/vtns/vtn_name/vrouters/vrt_name/interfaces/if_name/macaddress.xml`

- JSON format

  `/vtns/vtn_name/vrouters/vrt_name/interfaces/if_name/macaddress.json`

Table 6-47 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<interface macaddr="macaddr"/>
```

- JSON format

```
{
  "interface" : {
    "macaddr" : "macaddr"
  }
}
```

Table 6-48 Description of Elements in interface

Element	Description	Valid Value
macaddr	MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number) "0000.0000.0000," "ffff.ffff.ffff," or any multicast MAC address cannot be specified.

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  `"Register MAC address macaddr to interface if_name of VTN vtn_name vRouter vrt_name succeeded."`

- On error

  `"Register MAC address macaddr to interface if_name of VTN vtn_name vRouter vrt_name failed.""` (<Error information from pfcshell>)"

Remark

None

6.3.11 Unregister MAC address from interface (vRouter mode)

Following describes how to deregister the MAC address of the virtual interface set under the vRouter.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/vtn_name/vrouters/vrt_name/interfaces/if_name/macaddress.xml`

- JSON format

  `/vtns/vtn_name/vrouters/vrt_name/interfaces/if_name/macaddress.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `macaddr=macaddr`

Table 6-49 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Valid Value
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
macaddr	MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number) "0000.0000.0000," "ffff.ffff.ffff," or any multicast MAC address cannot be specified.

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

- On success

  "Unregister MAC address **macaddr** from interface **if_name** of VTN **vtn_name** vRouter **vrt_name** succeeded."
- On failure

  "Unregister MAC address **macaddr** from interface **if_name** of VTN **vtn_name** vRouter **vrt_name** failed.(<Error information from pfcshell>)"

Remark

None

6.4 Operate IP Routing Information

6.4.1 Show IP routes

Following describes how to acquire the path information defined for the vRouter.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/iproutes.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/iproutes.json

Table 6-50 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<iproutes>
  [
  <iproute
    dstaddr="dstaddr"
    gateway="gateway"
    netmask="netmask"
    flags="flags"
    metric="metric"
    use="use"
    if_name="if_name"
    nw="nw"/>
  ]
</iproutes>
```

- JSON format

```
{
  "iproutes": [
    {
      "dstaddr": "dstaddr",
      "gateway": "gateway",
      "netmask": "netmask",
      "flags": "flags",
      "metric": "metric",
      "use": "use",
      "if_name": "if_name",
      "nw": "nw"
    }
```

```
  ]
}
```

Table 6-51 Description of Elements in iproutes

Element	Description	Return Value
dstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
gateway	Gateway IP address	IPv4 dot-separated format (Example: 192.168.1.1)
netmask	Subnet mask of the destination IP address	IPv4 dot-separated format (Example: 255.255.255.0)
flags	Route status	U (route is up) H (target is a host) G (use gateway) R (reinstate route for dynamic routing) D (dynamically installed by daemon or redirect) M (modified from routing daemon or redirect) A (installed by addrconf) S (static route) C (cache entry) ! (reject route)
metric	Priority order for multiple paths	Decimal number (0 to 65535)
use	Number of lookups on the route	Decimal number; up to eight digits
if_name	Interface that receives packets through this route	Up to 31 characters including one-byte alphanumeric characters and underscores
nw	Flag indicating whether the net monitoring is specified	["Y" |"N"]

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.4.2 Show IP routes details

Following describes how to acquire detailed information on the path defined in the vRouter.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/iproutes/detail.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/iproutes/detail.json

Table 6-52 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<iproutes>
  [
  <iproute
    dstaddr="dstaddr"
    gateway="gateway"
    netmask="netmask"
    flags="flags"
    metric="metric"
    use="use"
    if_name="if_name"
    netgroup_name="netgroup_name"
    groupmetric="groupmetric"/>
  ]
</iproutes>
```

- JSON format

```
{
  "iproutes": [
    {
      "dstaddr": "dstaddr",
      "gateway": "gateway",
      "netmask": "netmask",
      "flags": "flags",
      "metric": "metric",
      "use": "use",
      "if_name": "if_name",
      "netgroup_name": "netgroup_name",
```

```
        "groupmetric": "groupmetric"
      }
    ]
}
```

Table 6-53 Description of Elements in iproutes

Element	Description	Return Value
dstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
gateway	Gateway IP address	IPv4 dot-separated format (Example: 192.168.1.1)
netmask	Subnet mask of the destination IP address	IPv4 dot-separated format (Example: 255.255.255.0)
flags	Routing flag	U (route is up) H (target is a host) G (use gateway) R (reinstate route for dynamic routing) D (dynamically installed by daemon or redirect) M (modified from routing daemon or redirect) A (installed by addrconf) S (static route) C (cache entry) ! (reject route)
metric	Priority order for multiple paths	Decimal number (0 to 65535)
use	Number of lookups on the route	Decimal number; up to eight digits
if_name	Interface that receives packets through this route	Up to 31 characters including one-byte alphanumeric characters and underscores
netgroup_name	Network group name to be monitored if the net monitoring function is used	One-byte characters (alphanumeric characters, underbar, and "-" (hyphen) (Up to 31 characters) * "-" (hyphen) is displayed if the net monitoring function is not used.
groupmetric	Priority order for multiple paths if the net monitoring function is used	Decimal number within the range of 1 to 65535 and "-" (hyphen) * "-" (hyphen) is displayed if the net monitoring function is not used.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.4.3 Show static IP routes

Following describes how to acquire the static routing information that is defined for vRouter.

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/static_iproutes.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/static_iproutes.json

Table 6-54 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<static_iproutes>
  [
  <static_iproute
    static_iproute_id="static_iproute_id"
    ipaddr="ipaddr"
    nexthopaddr="nexthopaddr"
    netmask="netmask"/>
  ]
</static_iproutes>
```

- JSON format

```
{
  "static_iproutes" : [
    {
      "static_iproute_id" : "static_iproute_id",
      "ipaddr" : "ipaddr",
      "nexthopaddr" : "nexthopaddr",
```

```
      "netmask" : "netmask"
    }
  ]
}
```

Table 6-55 Description of Elements in static_iproutes

Element	Description	Return Value
static_iproute_id	Path information ID	Up to 47 characters
ipaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
nexthopaddr	IP address of the router that is a destination to which packets for ipaddr are sent.	IPv4 dot-separated format (Example: 192.168.1.1)
netmask	Destination IP address subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

A user can use the returned **static_iproute_id** only as a single ID. It is not allowed to interpret **static_iproute_id**.

6.4.4 Set static IP route to vRouter

Following describes how to register the static routing information to vRouter.

Processing request

Method

POST

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/static_iproutes.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/static_iproutes.json`

Table 6-56 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<static_iproute
  ipaddr="ipaddr"
  prefix="prefix"
  netmask="netmask"
  nexthopaddr="nexthopaddr"/>
```

- JSON format

```
{
  "static_iproute " : {
    "ipaddr" : "ipaddr",
    "prefix" : "prefix",
    "netmask" : "netmask",
    "nexthopaddr" : "nexthopaddr"
  }
}
```

Table 6-57 Description of Elements in static_iproute

Element	Description	Valid Value
ipaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
prefix	Destination IP address prefix length	0 to 32
netmask	Destination IP address subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)
nexthopaddr	IP address of the router that is a destination to which packets for ip addr are sent.	IPv4 dot-separated format (Example: 192.168.1.1)

Remember

prefix and netmask cannot be specified at the same time.

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<static_iproute static_iproute_id="static_iproute_id"/>
```

- JSON format

```
{
  "static_iproute" : {
    "static_iproute_id" : "static_iproute_id"
  }
}
```

Table 6-58 Description of Elements in static_iproute

Element	Description	Return Value
static_iproute_id	Path information ID	Up to 47 characters

Remember

static_iproute_id is created with the fixed form from the values included in a request, but the form will possibly be changed without notice.

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Create static_iproute of vRouter **vrt_name** of VTN **vtn_name** succeeded."

- On failure

  "Create static_iproute of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfc shell>)"

  "Create static_iproute of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(prefix,netmask)](<Error information from pfcshell>)"

Remark

A user can use the returned static_iproute_id only as a single ID. It is not allowed to interpret static_iproute_id.

6.4.5 Delete static IP route from vRouter

Following describes how to deregister the static routing information that is defined for vRouter.

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/static_iproutes/`**`static_iproute_id`**`.xml`

- JSON format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/static_iproutes/`**`static_iproute_id`**`.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `ipaddr=`**`ipaddr`**

  `prefix=`**`prefix`**

  `netmask=`**`netmask`**

  `nexthopaddr=`**`nexthopaddr`**

Table 6-59 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
static_iproute_id	Path information ID	Up to 47 characters
ipaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
prefix	Destination IP address prefix length	0 to 32
netmask	Destination IP address subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)
nexthopaddr	IP address of the router that is a destination to which packets for ipaddr are sent.	IPv4 dot-separated format (Example: 192.168.1.1)

Remember

prefix and netmask cannot be specified at the same time.

Remember

The specified static_iproute_idin URI and the specified ipaddr, prefix, netmask, and nexthopaddr in the query character string must be equal to the values returned in response body of "6.4.3 Show static IP routes (page 243)" and "6.4.4 Set static IP route to vRouter (page 244)".

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

- On success

  "Delete static IP route **static_iproute_id** from vRouter **vrt_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete static IP route **static_iproute_id** of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(prefix,netmask)](<Error information from pfcshell>)"

  "Delete static IP route **static_iproute_id** of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

6.5 Operate ARP entry

6.5.1 Show ARP entry

Following describes how to acquire the ARP entry information of the vRouter.

Processing request

Method

GET

request URI

- XML format

/vtns/**vtn_name**/vrouters/**vrt_name**/arpentries.xml

- JSON format

/vtns/**vtn_name**/vrouters/**vrt_name**/arpentries.json

Table 6-60 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<arpentries>
  [<arpentry ipaddr="ipaddr"
               macaddr="macaddr"
               type="type"
               if_name="if_name"/>]
  <totalcount  value="value"/>
</arpentries>
```

- JSON format

```
{
   "arpentries" : [{
        "ipaddr" : "ipaddr",
        "macaddr" : "macaddr",
        "type" : "type",
        "if_name" : "if_name"
   }],
   "totalcount" : {
        "value" : "value"
   }
}
```

Table 6-61 Description of Elements in arpentries Type

Element	Description	Return Value
arpentry	ARP entry	arpentry type (For details, refer to "Table 6-62 Description of Elements in arpentry Type (page 249)" below.)
totalcount	Number of ARP entries	interfacetype (For details, refer to "Table 6-63 Description of Elements in total count Type (page 250)" below.)

Table 6-62 Description of Elements in arpentry Type

Element	Description	Return Value
ipaddr	IP address	IPv4 dot-separated format (Example: 192.168.1.1)
macaddr	MAC address	hhhh.hhhh.hhhh format (H: hexadecimal number)

Element	Description	Return Value
type	ARP entry type	{dynamic|static} (Lowercase)
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Table 6-63 Description of Elements in total count Type

Element	Description	Return Value
value	Number of ARP entries	Decimal number (0 to 65535)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

6.5.2 Show count of ARP entry

Following describes how to acquire the number of ARP entries of the vRouter.

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/arpentries/count.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/arpentries/count.json`

Table 6-64 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<arpentry totalcount="totalcount" />
```

- JSON format

```
{
  "arpentry" : {
       "totalcount" : "totalcount"
  }
}
```

Table 6-65 Description of Elements in arpentry

Element	Description	Return Value
totalcount	Number of ARP entries	Decimal number (0 to 65535)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400, 500

Miscellaneous

Log

None

Remark

None

Chapter 7. WebAPI (Policy Management-Related) Reference

Following describes the URI, parameters, request/response data and other details about the policy-related API.

Tip

The WebAPI uses structure data in the XML or JSON format as API input/output. In this chapter, for each API input/output data unit, the data elements on the same structure level is summarized in a table and the element name is written in bold like **"data."** If a lower level has data, the relevant level name is written in plain text like "data." The reference table number and other details about this data are shown in the "Return value" column.

7.1 Operate Flowlist

7.1.1 List flowlists

Processing request

Method

GET

request URI

- XML format

  `/flowlists.xml`

- JSON format

  `/flowlists.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowlists>
  [
  <flowlist fl_name="fl_name"/>
  ]
</flowlists>
```

- JSON format

```
{
  "flowlists" : [
    {
      "fl_name" : "fl_name"
    }
  ]
}
```

Table 7-1 Description of Elements in flowlists

Element	Description	Return Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.1.2 Create flowlist

Processing request

Method

POST

request URI

- XML format

  `/flowlists.xml`
- JSON format

  `/flowlists.json`

Settings of request body

- XML format

```
<flowlist fl_name="fl_name" ip_version="ip_version">
  <restrict
    srcmac="srcmac"
    dstmac="dstmac"
    srcip="srcip"
```

```
    dstip="dstip"
    ipproto="ipproto"
    srcport="srcport"
    dstport="dstport"/>
</flowlist>
```

- JSON format

```
{
  "flowlist" : {
    "fl_name" : "fl_name",
    "ip_version" : "ip_version",
    "restrict" : {
      "srcmac" : "srcmac",
      "dstmac" : "dstmac",
      "srcip" : "srcip",
      "dstip" : "dstip",
      "ipproto" : "ipproto",
      "srcport" : "srcport",
      "dstport" : "dstport"
    }
  }
}
```

Table 7-2 Description of Elements in flowlist

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
ip_version	IP version	{ip|ipv6} (Lowercase)
restrict	Restriction on flowlist	restrict

Table 7-3 Description of Elements in restrict

Element	Description	Valid Value
srcmac	Flag indicating that the source IP address is set as a restriction.	{true} *Always true
dstmac	Flag indicating that the destination IP address is set as a restriction.	{true} *Always true
srcip	Flag indicating that the source IP address is set as a restriction.	{true} *Always true
dstip	Flag indicating that the destination IP address is set as a restriction.	{true} *Always true
ipproto	Flag indicating that the IP protocol is set as a restriction.	{true} *Always true
srcport	Flag indicating that the source TCP/UDP port number is set as a restriction.	{true} *Always true
dstport	Flag indicating that the destination TCP/UDP port number is set as a restriction.	{true} *Always true

Remember

fl_name must be specified.

Remember

ip_version can be omitted.

Remember

restrict and elements under **restrict** can be omitted. When specifying **restrict**, one or more restrictions must be specified.

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  `"Create flow-list fl_namesucceeded."`

- On failure

  `"Create flow-list fl_name failed. [Invalid argument](<Error information from pfcshell>)"`

Remark

None

7.1.3 Show flowlist

Processing request

Method

GET

request URI

- XML format

  `/flowlists/fl_name.xml`

- JSON format

  `/flowlists/fl_name.json`

Table 7-4 Description of Parameters in URI

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowlist fl_name="fl_name" ip_version="ip_version">
  <restrict
    srcmac="srcmac"
    dstmac="dstmac"
    srcip="srcip"
    dstip="dstip"
    ipproto="ipproto"
    srcport="srcport"
    dstport="dstport"/>
  <flowlistentries>
    [
    <flowlistentry
      seqnum="seqnum"
      macdstaddr="macdstaddr"
      macsrcaddr="macsrcaddr"
      macethertype="macethertype"
      macvlanpriority="macvlanpriority"
      ipdstaddr="ipdstaddr"
      ipsrcaddr="ipsrcaddr"
      ipv6dstaddr="ipv6dstaddr"
      ipv6srcaddr="ipv6srcaddr"
      ipdstaddrprefix="ipdstaddrprefix"
      ipsrcaddrprefix="ipsrcaddrprefix"
      ipv6dstaddrprefix="ipv6dstaddrprefix"
      ipv6srcaddrprefix="ipv6srcaddrprefix"
      ipproto="ipproto"
      ipdscp="ipdscp"
      l4dstport="l4dstport"
      l4dstendport="l4dstendport"
      l4srcport="l4srcport"
      l4srcendport="l4srcendport"
      icmptypenum="icmptypenum"
      icmpcodenum="icmpcodenum"
      ipv6icmptypenum="ipv6icmptypenum"
      ipv6icmpcodenum="ipv6icmpcodenum"/>
     ]
  </flowlistentries>
</flowlist>
```

- JSON format

```
{
  "flowlist" : {
    "fl_name" : "fl_name",
    "ip_version" : "ip_version",
    "restrict" : {
      "srcmac" : "srcmac",
      "dstmac" : "dstmac",
      "srcip" : "srcip",
      "dstip" : "dstip",
```

```
      "ipproto" : "ipproto",
      "srcport" : "srcport",
      "dstport" : "dstport"
    },
    "flowlistentries" : [
      {
        "seqnum" : "seqnum",
        "macdstaddr" : "macdstaddr",
        "macsrcaddr" : "macsrcaddr",
        "macethertype" : "macethertype",
        "macvlanpriority" : "macvlanpriority",
        "ipdstaddr" : "ipdstaddr",
        "ipsrcaddr" : "ipsrcaddr",
        "ipv6dstaddr" : "ipv6dstaddr",
        "ipv6srcaddr" : "ipv6srcaddr",
        "ipdstaddrprefix" : "ipdstaddrprefix",
        "ipsrcaddrprefix" : "ipsrcaddrprefix",
        "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
        "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
        "ipproto" : "ipproto",
        "ipdscp" : "ipdscp",
        "l4dstport" : "l4dstport",
        "l4dstendport" : "l4dstendport",
        "l4srcport" : "l4srcport",
        "l4srcendport" : "l4srcendport",
        "icmptypenum" : "icmptypenum",
        "icmpcodenum" : "icmpcodenum",
        "ipv6icmptypenum" : "ipv6icmptypenum",
        "ipv6icmpcodenum" : "ipv6icmpcodenum"
      }
    ]
  }
}
```

Table 7-5 Description of Elements in flowlist

Element	Description	Return Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
ip_version	IP version	{ip|ipv6} (Lowercase)
restrict	Restriction on flowlist	restrict
flowlistentries	List of flowlist entries	flowlistentries type (For details, refer to "Table 7-7 Description of Elements in flowlistentries Type (page 258)" later.)

Table 7-6 Description of Elements in restrict

Element	Description	Valid Value
srcmac	Flag indicating that the source IP address is set as a restriction.	{true} *Always true
dstmac	Flag indicating that the destination IP address is set as a restriction.	{true} *Always true
srcip	Flag indicating that the source IP address is set as a restriction.	{true} *Always true
dstip	Flag indicating that the destination IP address is set as a restriction.	{true} *Always true
ipproto	Flag indicating that the IP protocol is set as a restriction.	{true} *Always true
srcport	Flag indicating that the source TCP/UDP port number is set as a restriction.	{true} *Always true

Element	Description	Valid Value
dstport	Flag indicating that the destination TCP/UDP port number is set as a restriction.	{true} *Always true

Table 7-7 Description of Elements in flowlistentries Type

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipdstaddrprefix	Destination IP address prefix length	Decimal integer (1 to 32)
ipsrcaddrprefix	Source IP address prefix length	Decimal number (1 to 32)
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal number (1 to 128)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal number (1 to 128)
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (1 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6icmptypenum	ICMPv6 type	Decimal number (0 to 255)

Element	Description	Return Value
ipv6icmpcodenum	ICMPv6 code	Decimal number (0 to 255)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.1.4 Delete flowlist

Processing request

Method

DELETE

request URI

- XML format

  `/flowlists/`**`fl_name`**`.xml`
- JSON format

  `/flowlists/`**`fl_name`**`.json`
- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***`).

  `ip_version=`**`ip_version`**

Table 7-8 Description of Parameters in URI

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6} (Lowercase)

Remember

ip_version can be omitted.

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete flow-list **fl_name** succeeded."
- On failure

  "Delete flow-list **fl_name** failed. [Invalid argument(ip_version)](<Error information from pfcshell>)"

Remark

None

7.2 Operate Flowlist Entry

7.2.1 List sequence numbers of flowlist entry

Processing request

Method

GET

request URI

- XML format

  /flowlists/**fl_name**/flowlistentries.xml
- JSON format

  /flowlists/**fl_name**/flowlistentries.json

Table 7-9 Description of Parameters in URI

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowlistentries>
  [
  <flowlistentry seqnum="seqnum"/>
  ]
</flowlistentries>
```

- JSON format

```
{
  "flowlistentries" : [
    {
      "seqnum" : "seqnum"
    }
  ]
}
```

Table 7-10 Description of Elements in flowlistentries

Element	Description	Return Value
seqnum	Sequence number of path map entry	Decimal number (1 to 65535)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.2.2 Create flowlist entry

Processing request

Method

POST

request URI

- XML format

  `/flowlists/fl_name/flowlistentries.xml`

- JSON format

  `/flowlists/fl_name/flowlistentries.json`

Table 7-11 Description of Parameters in URI

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.

Settings of request body

- XML format

```
<flowlistentry
  seqnum="seqnum"
  macdstaddr="macdstaddr"
  macsrcaddr="macsrcaddr"
  macethertype="macethertype"
  macvlanpriority="macvlanpriority"
  ipdstaddr="ipdstaddr"
  ipdstaddrprefix="ipdstaddrprefix"
  ipsrcaddr="ipsrcaddr"
  ipsrcaddrprefix="ipsrcaddrprefix"
  ipv6dstaddr="ipv6dstaddr"
  ipv6srcaddr="ipv6srcaddr"
  ipv6dstaddrprefix="ipv6dstaddrprefix"
  ipv6srcaddrprefix="ipv6srcaddrprefix"
  ipproto="ipproto"
  ipdscp="ipdscp"
  l4dstport="l4dstport"
  l4dstendport="l4dstendport"
  l4srcport="l4srcport"
  l4srcendport="l4srcendport"
  icmptypenum="icmptypenum"
  icmpcodenum="icmpcodenum"
  ipv6icmptypenum="ipv6icmptypenum"
  ipv6icmpcodenum="ipv6icmpcodenum"/>
```

- JSON format

```
{
  "flowlistentry" : {
    "seqnum" : "seqnum",
    "macdstaddr" : "macdstaddr",
    "macsrcaddr" : "macsrcaddr",
    "macethertype" : "macethertype",
    "macvlanpriority" : "macvlanpriority",
    "ipdstaddr" : "ipdstaddr",
    "ipdstaddrprefix" : "ipdstaddrprefix",
    "ipsrcaddr" : "ipsrcaddr",
```

```
    "ipsrcaddrprefix" : "ipsrcaddrprefix",
    "ipv6dstaddr" : "ipv6dstaddr",
    "ipv6srcaddr" : "ipv6srcaddr",
    "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
    "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
    "ippproto" : "ippproto",
    "ipdscp" : "ipdscp",
    "l4dstport" : "l4dstport",
    "l4dstendport" : "l4dstendport",
    "l4srcport" : "l4srcport",
    "l4srcendport" : "l4srcendport",
    "icmptypenum" : "icmptypenum",
    "icmpcodenum" : "icmpcodenum",
    "ipv6icmptypenum" : "ipv6icmptypenum",
    "ipv6icmpcodenum" : "ipv6icmpcodenum"
  }
}
```

Table 7-12 Description of Elements in flowlistentry

Element	Description	Valid Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal integer notation (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal integer (1 to 32)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddrprefix	Source IP address prefix length	Decimal integer (1 to 32)
ippproto	IP protocol number	Decimal integer (1 to 255)
ipdscp	DSCP value	Decimal integer notation (0 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (1 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (1 to 65535)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer (1 to 128)

Element	Description	Valid Value
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer (1 to 128)
icmptypenum	ICMP type	Decimal integer (0 to 255)
icmpcodenum	ICMP code	Decimal integer (0 to 255)
ipv6icmptypenum	ICMPv6 type	Decimal integer (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal integer (0 to 255)

Remember

Only seqnum is required.

Remember

For details on the restrictions on element combinations etc., see the Command Reference.

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create flowlistentry with sequence number **seqnum** for flow-list **fl_name** succeeded."

- On failure

  "Create flowlistentry for flow-list **fl_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

7.2.3 Show flowlist entry

Processing request

Method

GET

request URI

- XML format

  `/flowlists/fl_name/flowlistentries/seqnum.xml`

- JSON format

  `/flowlists/fl_name/flowlistentries/seqnum.json`

Table 7-13 Description of Parameters in URI

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
seqnum	Flowlist sequence number	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowlistentry
  seqnum="seqnum"
  macdstaddr="macdstaddr"
  macsrcaddr="macsrcaddr"
  macethertype="macethertype"
  macvlanpriority="macvlanpriority"
  ipdstaddr="ipdstaddr"
  ipsrcaddr="ipsrcaddr"
  ipv6dstaddr="ipv6dstaddr"
  ipv6srcaddr="ipv6srcaddr"
  ipdstaddrprefix="ipdstaddrprefix"
  ipsrcaddrprefix="ipsrcaddrprefix"
  ipv6dstaddrprefix="ipv6dstaddrprefix"
  ipv6srcaddrprefix="ipv6srcaddrprefix"
  ipproto="ipproto"
  ipdscp="ipdscp"
  l4dstport="l4dstport"
  l4dstendport="l4dstendport"
  l4srcport="l4srcport"
  l4srcendport="l4srcendport"
  icmptypenum="icmptypenum"
  icmpcodenum="icmpcodenum"
  ipv6icmptypenum="ipv6icmptypenum"
  ipv6icmpcodenum="ipv6icmpcodenum"/>
```

- JSON format

```
{
  "flowlistentry" : {
      "seqnum" : "seqnum",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr",
      "macethertype" : "macethertype",
      "macvlanpriority" : "macvlanpriority",
      "ipdstaddr" : "ipdstaddr",
      "ipsrcaddr" : "ipsrcaddr",
      "ipv6dstaddr" : "ipv6dstaddr",
```

```
        "ipv6srcaddr" : "ipv6srcaddr",
        "ipdstaddrprefix" : "ipdstaddrprefix",
        "ipsrcaddrprefix" : "ipsrcaddrprefix",
        "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
        "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
        "ipproto" : "ipproto",
        "ipdscp" : "ipdscp",
        "l4dstport" : "l4dstport",
        "l4dstendport" : "l4dstendport",
        "l4srcport" : "l4srcport",
        "l4srcendport" : "l4srcendport",
        "icmptypenum" : "icmptypenum",
        "icmpcodenum" : "icmpcodenum",
        "ipv6icmptypenum" : "ipv6icmptypenum",
        "ipv6icmpcodenum" : "ipv6icmpcodenum"
    }
}
```

Table 7-14 Description of Elements in flowlistentries Type

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipdstaddrprefix	Destination IP address prefix length	Decimal integer (1 to 32)
ipsrcaddrprefix	Source IP address prefix length	Decimal integer (1 to 32)
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer (1 to 128)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer (1 to 128)
ipproto	IP protocol number	Decimal integer (1 to 255)
ipdscp	DSCP value	Decimal integer notation (0 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)

Element	Description	Return Value
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
icmptypenum	ICMP type	Decimal integer (1 to 255)
icmpcodenum	ICMP code	Decimal integer (1 to 255)
ipv6icmptypenum	ICMPv6 type	Decimal integer (1 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal integer (1 to 255)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get flowlistentry with sequence number **seqnum** for flow-list **fl_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.2.4 Update flowlist entry

Processing request

Method

PUT

request URI

- XML format

  /flowlists/**fl_name**/flowlistentries/**seqnum**.xml

- JSON format

  /flowlists/**fl_name**/flowlistentries/**seqnum**.json

Table 7-15 Description of Parameters in URI

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.

Element	Description	Valid Value
seqnum	Flowlist sequence number	Decimal integer (1 to 65535)

Settings of request body

- XML format

```
<flowlistentry
  op="op"
  macdstaddr="macdstaddr"
  macsrcaddr="macsrcaddr"
  macethertype="macethertype"
  macvlanpriority="macvlanpriority"
  ipdstaddr="ipdstaddr"
  ipsrcaddr="ipsrcaddr"
  ipv6dstaddr="ipv6dstaddr"
  ipv6srcaddr="ipv6srcaddr"
  ipdstaddrprefix="ipdstaddrprefix"
  ipsrcaddrprefix="ipsrcaddrprefix"
  ipv6dstaddrprefix="ipv6dstaddrprefix"
  ipv6srcaddrprefix="ipv6srcaddrprefix"
  ipproto="ipproto"
  ipdscp="ipdscp"
  l4dstport="l4dstport"
  l4dstendport="l4dstendport"
  l4srcport="l4srcport"
  l4srcendport="l4srcendport"
  icmptypenum="icmptypenum"
  icmpcodenum="icmpcodenum"
  ipv6icmptypenum="ipv6icmptypenum"
  ipv6icmpcodenum="ipv6icmpcodenum"/>
```

- JSON format

```
{
  "flowlistentry" : {
    "op" : "op",
    "macdstaddr" : "macdstaddr",
    "macsrcaddr" : "macsrcaddr",
    "macethertype" : "macethertype",
    "macvlanpriority" : "macvlanpriority",
    "ipdstaddr" : "ipdstaddr",
    "ipsrcaddr" : "ipsrcaddr",
    "ipv6dstaddr" : "ipv6dstaddr",
    "ipv6srcaddr" : "ipv6srcaddr",
    "ipdstaddrprefix" : "ipdstaddrprefix",
    "ipsrcaddrprefix" : "ipsrcaddrprefix",
    "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
    "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
    "ipproto" : "ipproto",
    "ipdscp" : "ipdscp",
    "l4dstport" : "l4dstport",
    "l4dstendport" : "l4dstendport",
    "l4srcport" : "l4srcport",
    "l4srcendport" : "l4srcendport",
    "icmptypenum" : "icmptypenum",
    "icmpcodenum" : "icmpcodenum",
    "ipv6icmptypenum" : "ipv6icmptypenum",
    "ipv6icmpcodenum" : "ipv6icmpcodenum"
  }
}
```

Table 7-16 Description of Elements in flowlistentry Type

Element	Description	Valid Value
op	Update type	{add|delete} (Lowercase)

Element	Description	Valid Value
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipdstaddrprefix	Destination IP address prefix length	Decimal integer (1 to 32)
ipsrcaddrprefix	Source IP address prefix length	Decimal integer (1 to 32)
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer (1 to 128)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer (1 to 128)
ipproto	IP protocol number	Decimal integer (1 to 255)
ipdscp	DSCP value	Decimal integer notation (0 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
icmptypenum	ICMP type	Decimal integer (1 to 255)
icmpcodenum	ICMP code	Decimal integer (1 to 255)
ipv6icmptypenum	ICMPv6 type	Decimal integer (1 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal integer (1 to 255)

Remember

For details on the restrictions on element combinations etc., see the Command Reference.

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- When **op** is **add**
  - On success

    "Add flowlistentry with sequence number **seqnum** for flow-list **fl_name** succeeded."
  - On failure

    "Update flowlistentry with sequence number **seqnum** for flow-list **fl_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

    "Update flowlistentry with sequence number **seqnum** for flow-list **fl_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"
- When **op** is **delete**
  - On success

    "Delete flowlistentry with sequence number **seqnum** for flow-list **fl_name** succeeded."
  - On failure

    "Update flowlistentry with sequence number **seqnum** for flow-list **fl_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

    "Update flowlistentry with sequence number **seqnum** for flow-list **fl_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

7.2.5 Delete flowlist entry

Processing request

Method

DELETE

request URI

- XML format

  `/flowlists/fl_name/flowlistentries/seqnum.xml`

- JSON format

  `/flowlists/fl_name/flowlistentries/seqnum.json`

Table 7-17 Description of Parameters in URI

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
seqnum	Flowlist sequence number	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  `"Delete flowlistentry with sequence number seqnum of flow-list fl_name succeeded."`

- On failure

  `"Delete flowlistentry with sequence number seqnum of flow-list fl_name failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"`

Remark

None

7.3 Operate Flow Filter (vBridge Interface)

7.3.1 List flow filters for vBridge interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters.json

Table 7-18 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilters>
  [
  <flowfilter ff_type="ff_type"/>]
</flowfilters>
```

- JSON format

```
{
  "flowfilters" : [
    {
      "ff_type" : "ff_type"
    }
  ]
}
```

Table 7-19 Description of Elements in flowfilters

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.3.2 Create flow filter for vBridge interface

Processing request

Method

POST

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/interfaces/`**`if_name`**`/flowfilters.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/interfaces/`**`if_name`**`/flowfilters.json`

Table 7-20 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.

Settings of request body

- XML format

```
<flowfilter ff_type="ff_type"/>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type"
  }
}
```

Table 7-21 Description of Elements in flowfilters

Element	Description	Valid Value
ff_type	Flow filter application direction	{in|out}

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create flowfilter **ff_type** for interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Create flowfilter **ff_type** for interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.3.3 Show flow filter for vBridge interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**.json

Table 7-22 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentries>
    [
    <flowfilterentry
       seqnum="seqnum"
       fl_name="fl_name"
       action_type="action_type"
       nmg_name="nmg_name"/>
    ]
  </flowfilterentries>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentries" : [
      {
        "seqnum" : "seqnum",
        "fl_name" : "fl_name",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name"
      }
    ]
  }
}
```

Table 7-23 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}
flowfilterentries	Flow filter entry list	flowfilterentries type

Table 7-24 Description of Elements in flowfilterentries

Element	Description	Return Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.3.4 Delete flow filter for vBridge interface

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/interfaces/`**`if_name`**`/flowfilters/`**`ff_type`**`.xml`
- JSON format

/vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**.json

Table 7-25 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete flowfilter with flowfilter type **ff_type** for interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete flowfilter with flowfilter type **ff_type** for interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** failed.(<Error information from pfcshell>)"

Remark

None

7.3.5 Show flow filter details for vBridge interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/detail.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/detail.json

Table 7-26 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentries>
    [
    <flowfilterentry seqnum="seqnum"
      action_type="action_type" nmg_name="nmg_name">
      <redirectdst vnode_name="vnode_name"
        if_name="if_name"
        macdstaddr="macdstaddr"
        macsrcaddr="macsrcaddr"/>
      <statistics>
        <software packets="packets" octets="octets"/>
        <existingflow packets="packets" octets="octets"/>
        <expiredflow packets="packets" octets="octets"/>
        <total packets="packets" octets="octets"/>
      </statistics>
      <set priority="priority" dscp="dscp"/>
      <flowlist fl_name="fl_name" ip_version="ip_version">
```

```
        <flowlistentries>
          [
          <flowlistentry seqnum="seqnum"
            macdstaddr="macdstaddr"
            macsrcaddr="macsrcaddr"
            macethertype="macethertype"
            macvlanpriority="macvlanpriority"
            ipdstaddr="ipdstaddr"
            ipsrcaddr="ipsrcaddr"
            ipdstaddrprefix="ipdstaddrprefix"
            ipsrcaddrprefix="ipsrcaddrprefix"
            ipv6dstaddr="ipv6dstaddr"
            ipv6srcaddr="ipv6srcaddr"
            ipv6dstaddrprefix="ipv6dstaddrprefix"
            ipv6srcaddrprefix="ipv6srcaddrprefix"
            ipproto="ipproto"
            ipdscp="ipdscp"
            l4dstport="l4dstport"
            l4dstendport="l4dstendport"
            l4srcport="l4srcport"
            l4srcendport="l4srcendport"
            icmptypenum="icmptypenum"
            icmpcodenum="icmpcodenum"
            ipv6icmptypenum="ipv6icmptypenum"
            ipv6icmpcodenum="ipv6icmpcodenum">

            <statistics>
              <software packets="packets" octets="octets"/>
              <existingflow packets="packets" octets="octets"/>
              <expiredflow packets="packets" octets="octets"/>
              <total packets="packets" octets="octets"/>
            </statistics>
          </flowlistentry>
          ]
        </flowlistentries>
      </flowlist>
    </flowfilterentry>
    ]
  </flowfilterentries>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentries" : [
      {
        "seqnum" : "seqnum",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name",
        "redirectdst" : {
          "vnode_name" : "vnode_name",
          "if_name" : "if_name",
          "macdstaddr" : "macdstaddr",
          "macsrcaddr" : "macsrcaddr"
        },
        "statistics" : {
          "software" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expiredflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "total" : {
            "packets" : "packets",
```

```
            "octets" : "octets"
          }
        },
        "set" : {
          "priority" : "priority",
          "dscp" : "dscp"
        },
        "flowlist" : {
          "fl_name" : "fl_name",
          "ip_version" : "ip_version",
          "flowlistentries" : [
            {
              "seqnum" : "seqnum",
              "macdstaddr" : "macdstaddr",
              "macsrcaddr" : "macsrcaddr",
              "macethertype" : "macethertype",
              "macvlanpriority" : "macvlanpriority",
              "ipdstaddr" : "ipdstaddr",
              "ipsrcaddr" : "ipsrcaddr",
              "ipdstaddrprefix" : "ipdstaddrprefix",
              "ipsrcaddrprefix" : "ipsrcaddrprefix",
              "ipv6dstaddr" : "ipv6dstaddr",
              "ipv6srcaddr" : "ipv6srcaddr",
              "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
              "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
              "ipproto" : "ipproto",
              "ipdscp" : "ipdscp",
              "l4dstport" : "l4dstport",
              "l4dstendport" : "l4dstendport",
              "l4srcport" : "l4srcport",
              "l4srcendport" : "l4srcendport",
              "icmptypenum" : "icmptypenum",
              "icmpcodenum" : "icmpcodenum",
              "ipv6icmptypenum" : "ipv6icmptypenum",
              "ipv6icmpcodenum" : "ipv6icmpcodenum",
              "statistics" : {
                "software" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "existingflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "expiredflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "total" : {
                  "packets" : "packets",
                  "octets" : "octets"
                }
              }
            }
          ]
        }
      }
    ]
  }
}
```

Table 7-27 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter type	{in|out}
flowfilterentries	Flow filter entry	flowfilterentries type

Table 7-28 Description of Elements in flowfilterentries

Element	Description	Return Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores.
redirectdst	Redirect destination information	redirectdst type
statistics	Flowfilter entry statistical information	statistics type
set	Settings based on the matching flow	set type
flowlist	Flowlist information	flowlist type

Table 7-29 Description of Elements in redirectdst type of flowfilterentries

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores.
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)

Table 7-30 Description of Elements in statistics type of flowfilterentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-31 Description of Elements in set type

Element	Description	Return Value
priority	Packet transfer priority	Decimal notation (0 to 7)
dscp	DSCP value	Decimal notation (0 to 63)

Table 7-32 Description of Elements in flowlist type

Element	Description	Return Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
ip_version	IP version	{ip|ipv6} (Lowercase)

Element	Description	Return Value
flowlistentries	Flowlist entry information	flowlistentries type

Table 7-33 Description of Elements in flowlistentries Type

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal integer (1 to 32)
ipsrcaddrprefix	Source IP address prefix length	Decimal integer (1 to 32)
ipproto	IP protocol number	Decimal integer (1 to 255)
ipdscp	DSCP value	Decimal integer notation (0 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
icmptypenum	ICMP type	Decimal integer (0 to 255)
icmpcodenum	ICMP code	Decimal integer (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer (0 to 128)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer (0 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal integer (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal integer (0 to 255)
statistics	Flowlist entry statistical information	statistics type

Table 7-34 Description of Elements in statistics type of flowlistentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-35 Description of Elements in statistics type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of octets	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.4 Operate Flow Filter entry (vBridge Interface)

7.4.1 List sequence numbers of flow filter entry for vBridge interface

Processing request

Method

GET

request URI

- XML format

/vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.xml

- JSON format

/vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.json

Table 7-36 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilterentries>
  [
  <flowfilterentry seqnum="seqnum"/>
  ]
</flowfilterentries>
```

- JSON format

```
{
  "flowfilterentries" : [
    {
      "seqnum" : "seqnum"
    }
  ]
}
```

Table 7-37 Description of Elements in flowfilterentries

Element	Description	Return Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

HTTP status code for response

- On success

  200 (OK)

- On failure

400,500

Miscellaneous

Log

None

Remark

None

7.4.2 Create flow filter entry for vBridge interface

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.json

Table 7-38 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
ff_type	Flow filter application direction	{in|out}

Settings of request body

- XML format

```
<flowfilterentry seqnum="seqnum"
  fl_name="fl_name"
  action_type="action_type"
  nmg_name="nmg_name">
  <set priority="priority" dscp="dscp"/>
  <redirectdst vnode_name="vnode_name"
    if_name="if_name"
    macdstaddr="macdstaddr"
    macsrcaddr="macsrcaddr"/>
</flowfilterentry>
```

- JSON format

```
  {"flowfilterentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "nmg_name" : "nmg_name",
    "set" : {
      "priority" : "priority",
      "dscp" : "dscp"
    },
    "redirectdst" : {
      "vnode_name" : "vnode_name",
      "if_name" : "if_name",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr"
    }
  }
}
```

Table 7-39 Description of Elements in flowfilterentry

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores.
set	Settings based on the matching flow	set type
redirectdist	Redirect destination information	redirectdst type

Table 7-40 Description of Elements in set

Element	Description	Valid Value
priority	Packet transfer priority	Decimal number (1 to 7)
dscp	DSCP value	Decimal number (1 to 63)

Table 7-41 Description of Elements in redirectdst

Element	Description	Valid Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Create flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.4.3 Show flow filter entry for vBridge interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-42 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentry seqnum="seqnum"
    fl_name="fl_name"
    action_type="action_type"
    nmg_name="nmg_name"/>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentry" : {
        "seqnum" : "seqnum",
        "fl_name" : "fl_name",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name"
     }
  }
}
```

Table 7-43 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
action_type	Action for the matched ethernet frame	{pass|drop|redirect } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores.

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vNode **vbr_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.4.4 Update flow filter entry for vBridge interface

Processing request

Method

PUT

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-44 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

- XML format

```
<flowfilterentry
  fl_name="fl_name"
  action_type="action_type"
  nmg_name="nmg_name"
  op="op">
  <set priority="priority" dscp="dscp"/>
  <redirectdst
    vnode_name="vnode_name"
    if_name="if_name"
    macdstaddr="macdstaddr"
    macsrcaddr="macsrcaddr"/>
</flowfilterentry>
```

- JSON format

```
{
  "flowfilterentry" : {
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "nmg_name" : "nmg_name",
    "op" : "op",
    "set" : {
      "priority" : "priority",
      "dscp" : "dscp"
    },
    "redirectdst" : {
      "vnode_name" : "vnode_name",
      "if_name" : "if_name",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr"
    }
  }
}
```

Table 7-45 Description of Elements in flowfilterentry

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores.
op	op state information	{add|delete}
set	Settings based on the matching flow	set type
redirectdst	Redirect destination information	redirectdst type

Table 7-46 Description of Elements in set

Element	Description	Valid Value
priority	Packet transfer priority	Decimal number (1 to 7)
dscp	DSCP value	Decimal number (1 to 63)

Table 7-47 Description of Elements in redirectdst

Element	Description	Valid Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- When **op** is **add**
  - On success

    "Add flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** succeeded."
  - On failure

    "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

    "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"
- When **op** is **delete**
  - On success

    "Delete flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

  "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

7.4.5 Delete flow filter entry for vBridge interface

Processing request

Method

DELETE

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-48 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vNode **vbr_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.4.6 Show flow filter entry details for vBridge interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**/detail.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**/detail.json

Table 7-49 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores.
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores.
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores.
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentry seqnum="seqnum"
    action_type="action_type" nmg_name="nmg_name">
    <redirectdst vnode_name="vnode_name"
      if_name="if_name"
      macdstaddr="macdstaddr"
      macsrcaddr="macsrcaddr"/>
    <statistics>
      <software packets="packets" octets="octets"/>
      <existingflow packets="packets" octets="octets"/>
      <expiredflow packets="packets" octets="octets"/>
      <total packets="packets" octets="octets"/>
    </statistics>
    <set priority="priority" dscp="dscp"/>
    <flowlist fl_name="fl_name" ip_version="ip_version">
      <flowlistentries>
        [
        <flowlistentry seqnum="seqnum"
          macdstaddr="macdstaddr"
          macsrcaddr="macsrcaddr"
          macethertype="macethertype"
          macvlanpriority="macvlanpriority"
          ipdstaddr="ipdstaddr"
          ipsrcaddr="ipsrcaddr"
          ipdstaddrprefix="ipdstaddrprefix"
          ipsrcaddrprefix="ipsrcaddrprefix"
          ipv6dstaddr="ipv6dstaddr"
          ipv6srcaddr="ipv6srcaddr"
          ipv6dstaddrprefix="ipv6dstaddrprefix"
          ipv6srcaddrprefix="ipv6srcaddrprefix"
          ipproto="ipproto"
          ipdscp="ipdscp"
          l4dstport="l4dstport"
          l4dstendport="l4dstendport"
          l4srcport="l4srcport"
          l4srcendport="l4srcendport"
          icmptypenum="icmptypenum"
          icmpcodenum="icmpcodenum"
          ipv6icmptypenum="ipv6icmptypenum"
```

```
          ipv6icmpcodenum="ipv6icmpcodenum">
          <statistics>
            <software packets="packets" octets="octets"/>
            <existingflow packets="packets" octets="octets"/>
            <expiredflow packets="packets" octets="octets"/>
            <total packets="packets" octets="octets"/>
          </statistics>
        </flowlistentry>
        ]
      </flowlistentries>
    </flowlist>
  </flowfilterentry>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentry" : {
      "seqnum" : "seqnum",
      "action_type" : "action_type",
      "nmg_name" : "nmg_name",
      "redirectdst" : {
        "vnode_name" : "vnode_name",
        "if_name" : "if_name",
        "macdstaddr" : "macdstaddr",
        "macsrcaddr" : "macsrcaddr"
      },
      "statistics" : {
        "software" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "existingflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "expiredflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "total" : {
          "packets" : "packets",
          "octets" : "octets"
        }
      },
      "set" : {
        "priority" : "priority",
        "dscp" : "dscp"
      },
      "flowlist" : {
        "fl_name" : "fl_name",
        "ip_version" : "ip_version",
        "flowlistentries" : [
          {
            "seqnum" : "seqnum",
            "macdstaddr" : "macdstaddr",
            "macsrcaddr" : "macsrcaddr",
            "macethertype" : "macethertype",
            "macvlanpriority" : "macvlanpriority",
            "ipdstaddr" : "ipdstaddr",
            "ipsrcaddr" : "ipsrcaddr",
            "ipdstaddrprefix" : "ipdstaddrprefix",
            "ipsrcaddrprefix" : "ipsrcaddrprefix",
            "ipv6dstaddr" : "ipv6dstaddr",
            "ipv6srcaddr" : "ipv6srcaddr",
            "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
            "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
            "ipproto" : "ipproto",
            "ipdscp" : "ipdscp",
            "l4dstport" : "l4dstport",
```

```
            "l4dstendport" : "l4dstendport",
            "l4srcport" : "l4srcport",
            "l4srcendport" : "l4srcendport",
            "icmptypenum" : "icmptypenum",
            "icmpcodenum" : "icmpcodenum",
            "ipv6icmptypenum" : "ipv6icmptypenum",
            "ipv6icmpcodenum" : "ipv6icmpcodenum",
            "statistics" : {
              "software" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "existingflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "expiredflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "total" : {
                "packets" : "packets",
                "octets" : "octets"
              }
            }
          }
        ]
      }
    }
  }
}
```

Table 7-50 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{"in"|"out"}
flowfilterentry	Flow filter entry	flowfilterentry type

Table 7-51 Description of Elements in flowfilterentry

Element	Description	Return Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
redirectdst	Redirect destination information	redirectdst type
statistics	Flowfilter entry statistical information	statistics type
set	Settings based on the matching flow	set type
flowlist	Flowlist information	flowlist type

Table 7-52 Description of redirectdst Type Elements in flowfilterentry

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Return Value
if_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)

Table 7-53 Description of statistics Type Elements in flowfilterentry

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-54 Description of Elements in set Type

Element	Description	Return Value
priority	Packet transfer priority	Decimal notation (0 to 7)
dscp	DSCP value	Decimal notation (0 to 63)

Table 7-55 Description of Elements in flowlist Type

Element	Description	Return Value
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry information	flowlistentries type

Table 7-56 Description of Elements in flowlistentries Type

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal number (1 to 32)

Element	Description	Return Value
ipsrcaddrprefix	Source IP address prefix length	Decimal number (1 to 32)
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (1 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer notation (1 to 128)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer notation (1 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal integer notation (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal integer notation (0 to 255)
statistics	Flow filter entry statistical information	statistics

Table 7-57 Description of Elements in statistics type of flowlistentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-58 Description of Elements in statistics type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of octets	Counter

HTTP status code for response

- On success

200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get flowfilterentry detail with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vNode **vbr_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

3.29.6.4 Remark

None

7.5 Operate Flow Filter (vExternal Interface)

7.5.1 List flow filters for vExternal interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters.json

Table 7-59 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilters>
  [
  <flowfilter ff_type="ff_type"/>
  ]
</flowfilters>
```

- JSON format

```
{
  "flowfilters" : [
    {
      "ff_type" : "ff_type"
    }
  ]
}
```

Table 7-60 Description of Elements in flowfilters

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.5.2 Create flow filter for vExternal interface

Processing request

Method

POST

request URI

- XML format

/vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters.xml

- JSON format

/vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters.json

Table 7-61 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<flowfilter ff_type="ff_type"/>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type"
  }
}
```

Table 7-62 Description of Elements in flowfilters

Element	Description	Valid Value
ff_type	Flow filter application direction	{in|out}

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

```
"Create flowfilter ff_type for interface if_name of vExternal vex_name of VTN vtn_name succeeded."
```

- On failure

```
"Create flowfilter ff_type for interface if_name of vExternal vex_name of VTN vtn_name failed. [Invalid argument](<Error information from pfcshell>)"
```

Remark

None

7.5.3 Show flow filter for vExternal interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**.json

Table 7-63 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
```

```
  <flowfilterentries>
    [
    <flowfilterentry seqnum="seqnum"
      fl_name="fl_name"
      action_type="action_type"
      nmg_name="nmg_name"/>
    ]
  </flowfilterentries>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentries" : [
      {
        "seqnum" : "seqnum",
        "fl_name" : "fl_name",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name"
      }
    ]
  }
}
```

Table 7-64 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{pass|drop|redirect } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.5.4 Delete flow filter for vExternal interface

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/vtn_name/vexternals/vex_name/interfaces/if_name/flowfilters/ff_type.xml`

- JSON format

  `/vtns/vtn_name/vexternals/vex_name/interfaces/if_name/flowfilters/ff_type.json`

Table 7-65 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

```
"Delete flowfilter with flowfilter type ff_type for interface if_nam
e of vExternal vex_name of VTN vtn_name succeeded."
```

- On failure

```
"Delete flowfilter with flowfilter type ff_type for interface if_nam
e of vExternal vex_name of VTN vtn_name failed.(<Error information f
rom pfcshell>)"
```

Remark

None

7.5.5 Show flow filter details for vExternal interface

Processing request

Method

GET

request URI

- XML format

```
/vtns/vtn_name/vexternals/vex_name/interfaces/if_name/flowfilters/ff
_type/detail.xml
```

- JSON format

```
/vtns/vtn_name/vexternals/vex_name/interfaces/if_name/flowfilters/ff
_type/detail.json
```

Table 7-66 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
```

```
  <flowfilterentries>
    [
    <flowfilterentry seqnum="seqnum"
      action_type="action_type" nmg_name="nmg_name">
      <redirectdst vnode_name="vnode_name"
        if_name="if_name"
        macdstaddr="macdstaddr"
        macsrcaddr="macsrcaddr"/>
      <statistics>
        <software packets="packets" octets="octets"/>
        <existingflow packets="packets" octets="octets"/>
        <expiredflow packets="packets" octets="octets"/>
        <total packets="packets" octets="octets"/>
      </statistics>
      <set priority="priority" dscp="dscp"/>
      <flowlist fl_name="fl_name" ip_version="ip_version">
        <flowlistentries>
          [
          <flowlistentry seqnum="seqnum"
            macdstaddr="macdstaddr"
            macsrcaddr="macsrcaddr"
            macethertype="macethertype"
            macvlanpriority="macvlanpriority"
            ipdstaddr="ipdstaddr"
            ipsrcaddr="ipsrcaddr"
            ipdstaddrprefix="ipdstaddrprefix"
            ipsrcaddrprefix="ipsrcaddrprefix"
            ipv6dstaddr="ipv6dstaddr"
            ipv6srcaddr="ipv6srcaddr"
            ipv6dstaddrprefix="ipv6dstaddrprefix"
            ipv6srcaddrprefix="ipv6srcaddrprefix"
            ipproto="ipproto"
            ipdscp="ipdscp"
            l4dstport="l4dstport"
            l4dstendport="l4dstendport"
            l4srcport="l4srcport"
            l4srcendport="l4srcendport"
            icmptypenum="icmptypenum"
            icmpcodenum="icmpcodenum"
            ipv6icmptypenum="ipv6icmptypenum"
            ipv6icmpcodenum="ipv6icmpcodenum">
            <statistics>
              <software packets="packets" octets="octets"/>
              <existingflow packets="packets" octets="octets"/>
              <expiredflow packets="packets" octets="octets"/>
              <total packets="packets" octets="octets"/>
            </statistics>
          </flowlistentry>
          ]
        </flowlistentries>
      </flowlist>
    </flowfilterentry>
    ]
  </flowfilterentries>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentries" : [
      {
        "seqnum" : "seqnum",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name",
        "redirectdst" : {
          "vnode_name" : "vnode_name",
          "if_name" : "if_name",
          "macdstaddr" : "macdstaddr",
          "macsrcaddr" : "macsrcaddr"
        },
```

```
        "statistics" : {
          "software" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expiredflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "total" : {
            "packets" : "packets",
            "octets" : "octets"
          }
        },
        "set" : {
          "priority" : "priority",
          "dscp" : "dscp"
        },
        "flowlist" : {
          "fl_name" : "fl_name",
          "ip_version" : "ip_version",
          "flowlistentries" : [
            {
              "seqnum" : "seqnum",
              "macdstaddr" : "macdstaddr",
              "macsrcaddr" : "macsrcaddr",
              "macethertype" : "macethertype",
              "macvlanpriority" : "macvlanpriority",
              "ipdstaddr" : "ipdstaddr",
              "ipsrcaddr" : "ipsrcaddr",
              "ipdstaddrprefix" : "ipdstaddrprefix",
              "ipsrcaddrprefix" : "ipsrcaddrprefix",
              "ipv6dstaddr" : "ipv6dstaddr",
              "ipv6srcaddr" : "ipv6srcaddr",
              "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
              "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
              "ipproto" : "ipproto",
              "ipdscp" : "ipdscp",
              "l4dstport" : "l4dstport",
              "l4dstendport" : "l4dstendport",
              "l4srcport" : "l4srcport",
              "l4srcendport" : "l4srcendport",
              "icmptypenum" : "icmptypenum",
              "icmpcodenum" : "icmpcodenum",
              "ipv6icmptypenum" : "ipv6icmptypenum",
              "ipv6icmpcodenum" : "ipv6icmpcodenum",
              "statistics" : {
                "software" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "existingflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "expiredflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "total" : {
                  "packets" : "packets",
                  "octets" : "octets"
                }
              }
            }
          ]
        }
      }
    ]
```

```
    }
}
```

Table 7-67 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{"in"|"out"}
flowfilterentries	Flow filter entry	flowfilterentries type

Table 7-68 Description of Elements in flowfilterentries Type

Element	Description	Return Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
redirectdst	Redirect destination information	redirectdst type
statistics	Flowfilter entry statistical information	statistics type
set	Settings based on the matching flow	set type
flowlist	Flowlist information	flowlist type

Table 7-69 Description of redirectdst Type Elements in flowfilterentries

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)

Table 7-70 Description of statistics Type Elements in flowfilterentry

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-71 Description of Elements in set Type

Element	Description	Return Value
priority	Packet transfer priority	Decimal notation (0 to 7)
dscp	DSCP value	Decimal notation (0 to 63)

Table 7-72 Description of Elements in flowlist Type

Element	Description	Return Value
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry information	flowlistentries type

Table 7-73 Description of Elements in flowlistentries Type

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal number (1 to 32)
ipsrcaddrprefix	Source IP address prefix length	Decimal number (1 to 32)
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (1 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer notation (1 to 128)

Element	Description	Return Value
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer notation (1 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal integer notation (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal integer notation (0 to 255)
statistics	Flowlist entry statistical information	statistics type

Table 7-74 Description of Elements in statistics type of flowlistentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-75 Description of Elements in statistics type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of octets	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.6 Operate Flow Filter Entry (vExternal Interface)

7.6.1 List sequence numbers of flow filter entry for vExternal interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.json

Table 7-76 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilterentries>
  [
  <flowfilterentry seqnum="seqnum"/>
  ]
</flowfilterentries>
```

- JSON format

```
{
  "flowfilterentries" : [
    {
```

```
        "seqnum" : "seqnum"
      }
    ]
}
```

Table 7-77 Description of Elements in flowlistentries

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.6.2 Create flow filter entry for vExternal interface

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.xml
- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.json

Table 7-78 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Valid Value
ff_type	Flow filter application direction	{in|out}

Settings of request body

- XML format

```
<flowfilterentry
  seqnum="seqnum"
  fl_name="fl_name"
  action_type="action_type"
  nmg_name="nmg_name">
  <set priority="priority" dscp="dscp"/>
  <redirectdst
    vnode_name="vnode_name"
    if_name="if_name"
    macdstaddr="macdstaddr"
    macsrcaddr="macsrcaddr"/>
</flowfilterentry>
```

- JSON format

```
{  "flowfilterentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "nmg_name" : "nmg_name",
    "set" : {
      "priority" : "priority",
      "dscp" : "dscp"
    },
    "redirectdst" : {
      "vnode_name" : "vnode_name",
      "if_name" : "if_name",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr"
    }
  }
}
```

Table 7-79 Description of Elements in flowfilterentry

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{pass|drop|redirect } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
set	Settings based on the matching flow	set type
redirectdst	Redirect destination information	redirectdst type

Table 7-80 Description of Elements in set Type

Element	Description	Return Value
priority	Packet transfer priority	Decimal notation (0 to 7)

Element	Description	Return Value
dscp	DSCP value	Decimal notation (0 to 63)

Table 7-81 Description of Elements in flowlist Type

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Create flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.6.3 Show flow filter entry for vExternal interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-82 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentry seqnum="seqnum"
    fl_name="fl_name"
    action_type="action_type"
    nmg_name="nmg_name"/>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentry" : {
```

```
        "seqnum" : "seqnum",
        "fl_name" : "fl_name",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name"
      }
    }
  }
```

Table 7-83 Description of Elements in flowfilterentry

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{pass|drop|redirect } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vNode **vex_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.6.4 Update flow filter entry for vExternal interface

Processing request

Method

PUT

request URI

- XML format

/vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

/vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-84 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

- XML format

```
<flowfilterentry
  fl_name="fl_name"
  action_type="action_type"
  nmg_name="nmg_name"
  op="op">
  <set priority="priority" dscp="dscp"/>
  <redirectdst vnode_name="vnode_name"
    if_name="if_name"
    macdstaddr="macdstaddr"
    macsrcaddr="macsrcaddr"/>
</flowfilterentry>
```

- JSON format

```
{
  "flowfilterentry" : {
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "nmg_name" : "nmg_name",
    "op" : "op",
    "set" : {
      "priority" : "priority",
      "dscp" : "dscp"
    },
    "redirectdst" : {
      "vnode_name" : "vnode_name",
      "if_name" : "if_name",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr"
    }
  }
}
```

Table 7-85 Description of Elements in flowfilterentry

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
op	op state information	{add|delete}
set	Settings based on the matching flow	set type
redirectdst	Redirect destination information	redirectdst type

Table 7-86 Description of Elements in set

Element	Description	Return Value
priority	Packet transfer priority	Decimal notation (0 to 7)
dscp	DSCP value	Decimal notation (0 to 63)

Table 7-87 Description of Elements in redirectdst

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- When **op** is **add**

- On success

  "Add flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vExternal **vex_name** of VTN **vtn_name** succeeded."

- On failure

  "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

  "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

- When **op** is **delete**
  - On success

    "Delete flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vExternal **vex_name** of VTN **vtn_name** succeeded."

  - On failure

    "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

    "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

7.6.5 Delete flow filter entry for vExternal interface

Processing request

Method

DELETE

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

/vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-88 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vExternal **vex_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vNode **vex_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.6.6 Show flow filter entry details for vExternal interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**/detail.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**/detail.json

Table 7-89 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentry seqnum="seqnum"
    action_type="action_type" nmg_name="nmg_name">
    <redirectdst vnode_name="vnode_name"
      if_name="if_name"
      macdstaddr="macdstaddr"
      macsrcaddr="macsrcaddr"/>
    <statistics>
      <software packets="packets" octets="octets"/>
      <existingflow packets="packets" octets="octets"/>
      <expiredflow packets="packets" octets="octets"/>
      <total packets="packets" octets="octets"/>
    </statistics>
    <set priority="priority" dscp="dscp"/>
```

```
    <flowlist fl_name="fl_name" ip_version="ip_version">
      <flowlistentries>
        [
        <flowlistentry seqnum="seqnum"
          macdstaddr="macdstaddr"
          macsrcaddr="macsrcaddr"
          macethertype="macethertype"
          macvlanpriority="macvlanpriority"
          ipdstaddr="ipdstaddr"
          ipsrcaddr="ipsrcaddr"
          ipdstaddrprefix="ipdstaddrprefix"
          ipsrcaddrprefix="ipsrcaddrprefix"
          ipv6dstaddr="ipv6dstaddr"
          ipv6srcaddr="ipv6srcaddr"
          ipv6dstaddrprefix="ipv6dstaddrprefix"
          ipv6srcaddrprefix="ipv6srcaddrprefix"
          ipproto="ipproto"
          ipdscp="ipdscp"
          l4dstport="l4dstport"
          l4dstendport="l4dstendport"
          l4srcport="l4srcport"
          l4srcendport="l4srcendport"
          icmptypenum="icmptypenum"
          icmpcodenum="icmpcodenum"
          ipv6icmptypenum="ipv6icmptypenum"
          ipv6icmpcodenum="ipv6icmpcodenum">
          <statistics>
            <software packets="packets" octets="octets"/>
            <existingflow packets="packets" octets="octets"/>
            <expiredflow packets="packets" octets="octets"/>
            <total packets="packets" octets="octets"/>
          </statistics>
        </flowlistentry>
        ]
      </flowlistentries>
    </flowlist>
  </flowfilterentry>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentry" : {
      "seqnum" : "seqnum",
      "action_type" : "action_type",
      "nmg_name" : "nmg_name",
      "redirectdst" : {
        "vnode_name" : "vnode_name",
        "if_name" : "if_name",
        "macdstaddr" : "macdstaddr",
        "macsrcaddr" : "macsrcaddr"
      },
      "statistics" : {
        "software" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "existingflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "expiredflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "total" : {
          "packets" : "packets",
          "octets" : "octets"
        }
      },
```

```
      "set" : {
        "priority" : "priority",
        "dscp" : "dscp"
      },
      "flowlist" : {
        "fl_name" : "fl_name",
        "ip_version" : "ip_version",
        "flowlistentries" : [
          {
            "seqnum" : "seqnum",
            "macdstaddr" : "macdstaddr",
            "macsrcaddr" : "macsrcaddr",
            "macethertype" : "macethertype",
            "macvlanpriority" : "macvlanpriority",
            "ipdstaddr" : "ipdstaddr",
            "ipsrcaddr" : "ipsrcaddr",
            "ipdstaddrprefix" : "ipdstaddrprefix",
            "ipsrcaddrprefix" : "ipsrcaddrprefix",
            "ipv6dstaddr" : "ipv6dstaddr",
            "ipv6srcaddr" : "ipv6srcaddr",
            "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
            "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
            "ipproto" : "ipproto",
            "ipdscp" : "ipdscp",
            "l4dstport" : "l4dstport",
            "l4dstendport" : "l4dstendport",
            "l4srcport" : "l4srcport",
            "l4srcendport" : "l4srcendport",
            "icmptypenum" : "icmptypenum",
            "icmpcodenum" : "icmpcodenum",
            "ipv6icmptypenum" : "ipv6icmptypenum",
            "ipv6icmpcodenum" : "ipv6icmpcodenum",
            "statistics" : {
              "software" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "existingflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "expiredflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "total" : {
                "packets" : "packets",
                "octets" : "octets"
              }
            }
          }
        ]
      }
    }
  }
}
```

Table 7-90 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}
flowfilterentry	Flow filter entry	flowfilterentry type

Table 7-91 Description of Elements in flowfilterentry Type

Element	Description	Return Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Element	Description	Return Value
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 32 characters including one-byte alphanumeric characters and underscores
redirectdst	Redirect destination information	redirectdst type
statistics	Flow filter entry Statistical information	statistics type
set	Settings based on the matching flow	set type
flowlist	Flowlist information	flowlist type

Table 7-92 Description of Elements in redirectdst type of flowfilterentry

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)

Table 7-93 Description of Elements in statistics type of flowfilterentry

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-94 Description of Elements in set type

Element	Description	Return Value
priority	Packet transfer priority	Decimal notation (0 to 7)
dscp	DSCP value	Decimal notation (0 to 63)

Table 7-95 Description of Elements in flowlist type

Element	Description	Return Value
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry information	flowlistentries type

Table 7-96 Description of Elements in flowlistentries Type

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal integer (1 to 32)
ipsrcaddrprefix	Source IP address prefix length	Decimal integer (1 to 32)
ipproto	IP protocol number	Decimal integer (1 to 255)
ipdscp	DSCP value	Decimal integer notation (0 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
icmptypenum	ICMP type	Decimal integer (0 to 255)
icmpcodenum	ICMP code	Decimal integer (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer (0 to 128)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer (0 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal integer (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal integer (0 to 255)
statistics	Flowlist entry Statistical information	statistics type

Table 7-97 Description of Elements in statistics type of flowfilterentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type

Element	Description	Return Value
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-98 Description of Elements in statistics type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of octets	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- On failure

```
"Get flowfilterentry detail with sequence number seqnum for flowfilt
er ff_type of interface if_name of vNode vex_name of VTN vtn_name fa
iled. [Invalid argument(seqnum)](<Error information from pfcshell>)"
```

Remark

None

7.7 Operate Flow Filter (vRouter Interface)

7.7.1 List flow filters for vRouter interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters.xml
- JSON format

/vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters.json

Table 7-99 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilters>
  [
  <flowfilter ff_type="ff_type"/>
  ]
</flowfilters>
```

- JSON format

```
{
  "flowfilters" : [
    {
      "ff_type" : "ff_type"
    }
  ]
}
```

Table 7-100 Description of Elements in flowfilters

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.7.2 Create flow filter for vRouter interface

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters.json

Table 7-101 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<flowfilter ff_type="ff_type"/>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type"
  }
}
```

Table 7-102 Description of Elements in flowfilters

Element	Description	Valid Value
ff_type	Flow filter application direction	{in|out}

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  `"Create flowfilter ff_type for interface if_name of vRouter vrt_name of VTN vtn_name succeeded."`

- On failure

  `"Create flowfilter ff_type for interface if_name of vRouter vrt_name of VTN vtn_name failed. [Invalid argument](<Error information from pfcshell>)"`

Remark

None

7.7.3 Show flow filter for vRouter interface

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/vrouters/vrt_name/interfaces/if_name/flowfilters/ff_type.xml`

- JSON format

  `/vtns/vtn_name/vrouters/vrt_name/interfaces/if_name/flowfilters/ff_type.json`

Table 7-103 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Valid Value
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentries>
    [
    <flowfilterentry
      seqnum="seqnum"
      fl_name="fl_name"
      action_type="action_type"
      nmg_name="nmg_name"/>
    ]
  </flowfilterentries>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentries" : [
      {
        "seqnum" : "seqnum",
        "fl_name" : "fl_name",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name"
      }
    ]
  }
}
```

Table 7-104 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}
flowfilterentries	Flow filter entry	flowfilterentries type

Table 7-105 Description of Elements in flowfilterentries Type

Element	Description	Return Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{pass|drop|redirect } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.7.4 Delete flow filter for vRouter interface

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/interfaces/`**`if_name`**`/flowfilters/`**`ff_type`**`.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/interfaces/`**`if_name`**`/flowfilters/`**`ff_type`**`.json`

Table 7-106 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete flowfilter with flowfilter type **ff_type** for interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete flowfilter with flowfilter type **ff_type** for interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** failed.(<Error information from pfcshell>)"

Remark

None

7.7.5 Show flow filter details for vRouter interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/detail.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/detail.json

Table 7-107 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentries>
    [
    <flowfilterentry seqnum="seqnum"
      action_type="action_type" nmg_name="nmg_name">
      <redirectdst vnode_name="vnode_name"
        if_name="if_name"
        macdstaddr="macdstaddr"
        macsrcaddr="macsrcaddr"/>
      <statistics>
        <software packets="packets" octets="octets"/>
        <existingflow packets="packets" octets="octets"/>
        <expiredflow packets="packets" octets="octets"/>
        <total packets="packets" octets="octets"/>
      </statistics>
      <set priority="priority" dscp="dscp"/>
      <flowlist fl_name="fl_name" ip_version="ip_version">
        <flowlistentries>
          [
          <flowlistentry seqnum="seqnum"
            macdstaddr="macdstaddr"
            macsrcaddr="macsrcaddr"
            macethertype="macethertype"
            macvlanpriority="macvlanpriority"
            ipdstaddr="ipdstaddr"
            ipsrcaddr="ipsrcaddr"
            ipdstaddrprefix="ipdstaddrprefix"
            ipsrcaddrprefix="ipsrcaddrprefix"
            ipv6dstaddr="ipv6dstaddr"
            ipv6srcaddr="ipv6srcaddr"
            ipv6dstaddrprefix="ipv6dstaddrprefix"
            ipv6srcaddrprefix="ipv6srcaddrprefix"
            ipproto="ipproto"
            ipdscp="ipdscp"
            l4dstport="l4dstport"
            l4dstendport="l4dstendport"
            l4srcport="l4srcport"
            l4srcendport="l4srcendport"
            icmptypenum="icmptypenum"
            icmpcodenum="icmpcodenum"
            ipv6icmptypenum="ipv6icmptypenum"
            ipv6icmpcodenum="ipv6icmpcodenum">
```

```
            <statistics>
              <software packets="packets" octets="octets"/>
              <existingflow packets="packets" octets="octets"/>
              <expiredflow packets="packets" octets="octets"/>
              <total packets="packets" octets="octets"/>
            </statistics>
          </flowlistentry>
          ]
        </flowlistentries>
      </flowlist>
    </flowfilterentry>
    ]
  </flowfilterentries>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentries" : [
      {
        "seqnum" : "seqnum",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name",
        "redirectdst" : {
          "vnode_name" : "vnode_name",
          "if_name" : "if_name",
          "macdstaddr" : "macdstaddr",
          "macsrcaddr" : "macsrcaddr"
        },
        "statistics" : {
          "software" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expiredflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "total" : {
            "packets" : "packets",
            "octets" : "octets"
          }
        },
        "set" : {
          "priority" : "priority",
          "dscp" : "dscp"
        },
        "flowlist" : {
          "fl_name" : "fl_name",
          "ip_version" : "ip_version",
          "flowlistentries" : [
            {
              "seqnum" : "seqnum",
              "macdstaddr" : "macdstaddr",
              "macsrcaddr" : "macsrcaddr",
              "macethertype" : "macethertype",
              "macvlanpriority" : "macvlanpriority",
              "ipdstaddr" : "ipdstaddr",
              "ipsrcaddr" : "ipsrcaddr",
              "ipdstaddrprefix" : "ipdstaddrprefix",
              "ipsrcaddrprefix" : "ipsrcaddrprefix",
              "ipv6dstaddr" : "ipv6dstaddr",
              "ipv6srcaddr" : "ipv6srcaddr",
              "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
              "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
              "ipproto" : "ipproto",
```

```
                "ipdscp" : "ipdscp",
                "l4dstport" : "l4dstport",
                "l4dstendport" : "l4dstendport",
                "l4srcport" : "l4srcport",
                "l4srcendport" : "l4srcendport",
                "icmptypenum" : "icmptypenum",
                "icmpcodenum" : "icmpcodenum",
                "ipv6icmptypenum" : "ipv6icmptypenum",
                "ipv6icmpcodenum" : "ipv6icmpcodenum",
                "statistics" : {
                  "software" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  },
                  "existingflow" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  },
                  "expiredflow" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  },
                  "total" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  }
                }
              }
            ]
          }
        }
      ]
    }
}
```

Table 7-108 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{"in"|"out"}
flowfilterentries	Flow filter entry	flowfilterentries type

Table 7-109 Description of Elements in flowfilterentries Type

Element	Description	Return Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
redirectdst	Redirect destination information	redirectdst type
statistics	Flowfilter entry statistical information	statistics type
set	Settings based on the matching flow	set type
flowlist	Flowlist information	flowlist type

Table 7-110 Description of Elements in redirectdst type of flowfilterentries

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)

Table 7-111 Description of Elements in statistics type of flowfilterentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-112 Description of Elements in set Type

Element	Description	Return Value
priority	Packet transfer priority	Decimal notation (0 to 7)
dscp	DSCP value	Decimal notation (0 to 63)

Table 7-113 Description of Elements in flowlist type

Element	Description	Return Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry information	flowlistentries type

Table 7-114 Description of Elements in flowlistentries Type

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)

Element	Description	Return Value
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal integer (1 to 32)
ipsrcaddrprefix	Source IP address prefix length	Decimal integer (1 to 32)
ipproto	IP protocol number	Decimal integer (1 to 255)
ipdscp	DSCP value	Decimal integer notation (0 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
icmptypenum	ICMP type	Decimal integer notation (0 to 255)
icmpcodenum	ICMP code	Decimal integer notation (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer (0 to 128)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer (0 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal integer (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal integer (0 to 255)
statistics	Statistical information	statistics type

Table 7-115 Description of Elements in statistics type of flowfilterentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-116 Description of Elements in statistics Type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of bytes	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.8 Operate Flow Filter Entry (vRouter Interface)

7.8.1 List sequence numbers of flow filter entry for vRouter interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.xml
- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.json

Table 7-117 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilterentries>
  [
  <flowfilterentry seqnum="seqnum"/>
  ]
</flowfilterentries>
```

- JSON format

```
{
  "flowfilterentries" : [
    {
      "seqnum" : "seqnum"
    }
  ]
}
```

Table 7-118 Description of Elements in flowfilterentries

Element	Description	Return Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.8.2 Create flow filter entry for vRouter interface

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries.json

Table 7-119 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

- XML format

```
<flowfilterentry
  seqnum="seqnum"
  fl_name="fl_name"
  action_type="action_type"
  nmg_name="nmg_name">
  <set priority="priority" dscp="dscp"/>
  <redirectdst vnode_name="vnode_name"
    if_name="if_name"
    macdstaddr="macdstaddr"
    macsrcaddr="macsrcaddr"/>
</flowfilterentry>
```

- JSON format

```
{  "flowfilterentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "nmg_name" : "nmg_name",
    "set" : {
      "priority" : "priority",
      "dscp" : "dscp"
    },
    "redirectdst" : {
      "vnode_name" : "vnode_name",
      "if_name" : "if_name",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr"
    }
  }
}
```

Table 7-120 Description of Elements in flowfilterentry

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{pass|drop|redirect } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
set	Settings based on the matching flow	set type
redirectdst	Redirect destination information	redirectdst type

Table 7-121 Description of Elements in set Type

Element	Description	Return Value
priority	Packet transfer priority	Decimal notation (0 to 7)
dscp	DSCP value	Decimal notation (0 to 63)

Table 7-122 Description of Elements in flowlist type

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- On success

  "Create flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** succeeded."

- On failure

  "Create flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.8.3 Show flow filter entry for vRouter interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-123 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentry seqnum="seqnum"
    fl_name="fl_name"
    action_type="action_type"
    nmg_name="nmg_name"/>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentry" : {
      "seqnum" : "seqnum",
      "fl_name" : "fl_name",
      "action_type" : "action_type",
      "nmg_name" : "nmg_name"
    }
  }
}
```

Table 7-124 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vNode **vrt_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.8.4 Update flow filter entry for vRouter interface

Processing request

Method

PUT

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-125 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

- XML format

```
<flowfilterentry
  fl_name="fl_name"
  action_type="action_type"
  nmg_name="nmg_name"
  op="op">
  <set priority="priority" dscp="dscp"/>
  <redirectdst
    vnode_name="vnode_name"
    if_name="if_name"
    macdstaddr="macdstaddr"
    macsrcaddr="macsrcaddr"/>
</flowfilterentry>
```

- JSON format

```
{
  "flowfilterentry" : {
```

```
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "nmg_name" : "nmg_name",
    "op" : "op",
    "set" : {
      "priority" : "priority",
      "dscp" : "dscp"
    },
    "redirectdst" : {
      "vnode_name" : "vnode_name",
      "if_name" : "if_name",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr"
    }
  }
}
```

Table 7-126 Description of Elements in flowfilterentry

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
op	op state information	{add|delete}
set	Settings based on the matching flow	set type
redirectdst	Redirect destination information	redirectdst type

Table 7-127 Description of Elements in set

Element	Description	Return Value
priority	Packet transfer priority	Decimal notation (0 to 7)
dscp	DSCP value	Decimal notation (0 to 63)

Table 7-128 Description of Elements in redirectdst

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number).

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- When **op** is **add**
  - On success

    "Add flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** succeeded."

  - On failure

    "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

    "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

- When **op** is **delete**
  - On success

    "Delete flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** succeeded."

  - On failure

    "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

    "Update flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

7.8.5 Delete flow filter entry for vRouter interface

Processing request

Method

DELETE

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-129 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vRouter **vrt_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete sequence number **seqnum** for flowfilter **ff_type** of interface **if_name** of vNode **vrt_name** of VTN **vtn_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.8.6 Show flow filter entry for vRouter interface

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**/detail.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/interfaces/**if_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**/detail.json

Table 7-130 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentry seqnum="seqnum"
    action_type="action_type" nmg_name="nmg_name">
    <redirectdst vnode_name="vnode_name"
      if_name="if_name"
      macdstaddr="macdstaddr"
      macsrcaddr="macsrcaddr"/>
    <statistics>
      <software packets="packets" octets="octets"/>
      <existingflow packets="packets" octets="octets"/>
      <expiredflow packets="packets" octets="octets"/>
      <total packets="packets" octets="octets"/>
    </statistics>
    <set priority="priority" dscp="dscp"/>
    <flowlist fl_name="fl_name" ip_version="ip_version">
      <flowlistentries>
        [
        <flowlistentry seqnum="seqnum"
          macdstaddr="macdstaddr"
          macsrcaddr="macsrcaddr"
          macethertype="macethertype"
          macvlanpriority="macvlanpriority"
          ipdstaddr="ipdstaddr"
          ipsrcaddr="ipsrcaddr"
          ipdstaddrprefix="ipdstaddrprefix"
          ipsrcaddrprefix="ipsrcaddrprefix"
          ipv6dstaddr="ipv6dstaddr"
          ipv6srcaddr="ipv6srcaddr"
          ipv6dstaddrprefix="ipv6dstaddrprefix"
          ipv6srcaddrprefix="ipv6srcaddrprefix"
          ipproto="ipproto"
          ipdscp="ipdscp"
          l4dstport="l4dstport"
          l4dstendport="l4dstendport"
          l4srcport="l4srcport"
          l4srcendport="l4srcendport"
          icmptypenum="icmptypenum"
          icmpcodenum="icmpcodenum"
          ipv6icmptypenum="ipv6icmptypenum"
          ipv6icmpcodenum="ipv6icmpcodenum">
          <statistics>
            <software packets="packets" octets="octets"/>
            <existingflow packets="packets" octets="octets"/>
            <expiredflow packets="packets" octets="octets"/>
            <total packets="packets" octets="octets"/>
          </statistics>
        </flowlistentry>
        ]
      </flowlistentries>
    </flowlist>
  </flowfilterentry>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentry" : {
      "seqnum" : "seqnum",
      "action_type" : "action_type",
      "nmg_name" : "nmg_name",
      "redirectdst" : {
```

```
        "vnode_name" : "vnode_name",
        "if_name" : "if_name",
        "macdstaddr" : "macdstaddr",
        "macsrcaddr" : "macsrcaddr"
      },
      "statistics" : {
        "software" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "existingflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "expiredflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "total" : {
          "packets" : "packets",
          "octets" : "octets"
        }
      },
      "set" : {
        "priority" : "priority",
        "dscp" : "dscp"
      },
      "flowlist" : {
        "fl_name" : "fl_name",
        "ip_version" : "ip_version",
        "flowlistentries" : [
          {
            "seqnum" : "seqnum",
            "macdstaddr" : "macdstaddr",
            "macsrcaddr" : "macsrcaddr",
            "macethertype" : "macethertype",
            "macvlanpriority" : "macvlanpriority",
            "ipdstaddr" : "ipdstaddr",
            "ipsrcaddr" : "ipsrcaddr",
            "ipdstaddrprefix" : "ipdstaddrprefix",
            "ipsrcaddrprefix" : "ipsrcaddrprefix",
            "ipv6dstaddr" : "ipv6dstaddr",
            "ipv6srcaddr" : "ipv6srcaddr",
            "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
            "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
            "ipproto" : "ipproto",
            "ipdscp" : "ipdscp",
            "l4dstport" : "l4dstport",
            "l4dstendport" : "l4dstendport",
            "l4srcport" : "l4srcport",
            "l4srcendport" : "l4srcendport",
            "icmptypenum" : "icmptypenum",
            "icmpcodenum" : "icmpcodenum",
            "ipv6icmptypenum" : "ipv6icmptypenum",
            "ipv6icmpcodenum" : "ipv6icmpcodenum",
            "statistics" : {
              "software" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "existingflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "expiredflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "total" : {
                "packets" : "packets",
                "octets" : "octets"
              }
            }
```

```
                }
            ]
        }
    }
  }
}
```

Table 7-131 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{"in"|"out"}
flowfilterentry	Flow filter entry	flowfilterentry type

Table 7-132 Description of Elements in flowfilterentry Type

Element	Description	Return Value
seqnum	Sequence number of flow filter entry	Decimal integer (1 to 65535)
action_type	Action for the matched ethernet frame	{"pass"|"drop"|"redirect" } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
redirectdst	Redirect destination information	redirectdst type
statistics	Flowfilter entry statistical information	statistics type
set	Settings based on the matching flow	set type
flowlist	Flowlist information	flowlist type

Table 7-133 Description of Elements in redirectdst type of flowfilterentry

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)

Table 7-134 Description of Elements in statistics type of flowfilterentry

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-135 Description of Elements in set type

Element	Description	Return Value
priority	Packet transfer priority	Decimal notation (0 to 7)
dscp	DSCP value	Decimal notation (0 to 63)

Table 7-136 Description of Elements in flowlist type

Element	Description	Return Value
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry information	flowlistentries type

Table 7-137 Description of Elements in flowlistentries Type

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal integer (1 to 32)
ipsrcaddrprefix	Source IP address prefix length	Decimal integer (1 to 32)
ipproto	IP protocol number	Decimal integer (1 to 255)
ipdscp	DSCP value	Decimal integer notation (0 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal integer notation (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer notation (0 to 65535)
icmptypenum	ICMP type	Decimal integer notation (0 to 255)
icmpcodenum	ICMP code	Decimal integer notation (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddr	Source IPv6 address	IPv6 address

Element	Description	Return Value
		The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer notation (0 to 128)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer notation (0 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal integer notation (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal integer notation (0 to 255)
statistics	Flowlist entry Statistical information	statistics type

Table 7-138 Description of Elements in statistics type of flowfilterentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-139 Description of Elements in statistics Type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of bytes	Counter

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

```
"Get flowfilterentry detail with sequence number seqnum for flowfilt
er ff_type of interface if_name of vNode vrt_name of VTN vtn_name fa
iled. [Invalid argument(seqnum)](<Error information from pfcshell>)"
```

Remark

None

7.9 Operate Network Monitor Group

7.9.1 List network monitor groups

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups.json

Table 7-140 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<netmongroups>
  [
  <netmongroup name="name"/>
  ]
</netmongroups>
```

- JSON format

```
  {
  "netmongroups" : [
    {
      "name" : "name"
    }
  ]
}
```

Table 7-141 Description of Elements in netmongroups

Element	Description	Valid Value
name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

3.34.1.3.1 Log

None

Remark

None

7.9.2 Create network monitor group

Processing request

Method

POST

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/netmongroups.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/netmongroups.json`

Table 7-142 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<netmongroup name="name"/>
```

- JSON format

```
{
  "netmongroup" : {
    "name" : "name"
  }
}
```

Table 7-143 Description of Elements in netmongroups

Element	Description	Valid Value
name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create netmongroup **name** for vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Create netmongroup **name** for vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(netmongroup name)](<Error information from pfcshell>)"

Remark

None

7.9.3 Show network monitor group

Processing request

Method

GET

request URI

- XML format

/vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**.xml

- JSON format

/vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**.json

Table 7-144 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<netmongroup name="name">
  <hosts>
    [
    <host ipaddr="ipaddr"
      health_interval="health_interval"
      recovery_interval="recovery_interval"
      wait_time="wait_time"
      failure_counts="failure_counts"
      recovery_counts="recovery_counts"/>
     ]
  </hosts>
</netmongroup>
```

- JSON format

```
{
  "netmongroup" : {
    "name" : "name",
    "hosts" : [
      {
        "ipaddr" : "ipaddr",
        "health_interval" : "health_interval",
        "recovery_interval" : "recovery_interval",
        "wait_time" : "wait_time",
        "failure_counts" : "failure_counts",
        "recovery_counts" : "recovery_counts"
      }
    ]
  }
}
```

Table 7-145 Description of Elements in netmongroups

Element	Description	Valid Value
name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Valid Value
ipaddr	IP address of the Ping destination host	IPv4 dot-separated format (Example: 192.168.1.1)
health_interval	Ping health check transmission interval (sec.)	Decimal integer notation (5 to 600)
recovery_interval	Ping recovery check transmission interval (sec.)	Decimal integer notation (5 to 600)
wait_time	Ping response waiting time (sec.)	Decimal integer notation (1 to 60)
failure_counts	Count after which a failure is detected	Decimal integer notation (1 to 10)
recovery_counts	Count after which recovery is detected	Decimal integer notation (1 to 10)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.9.4 Enable/Disable network monitor group

Processing request

Method

PUT

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/netmongroups/`**`nmg_name`**`.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/netmongroups/`**`nmg_name`**`.json`

Table 7-146 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<netmongroup status="status"/>
```

- JSON format

```
{
  "netmongroup" : {
    "status" : "status"
  }
}
```

Table 7-147 Description of Elements in netmongroup

Element	Description	Valid Value
status	Network monitor group name	{enable|disable} (Lowercase)

Remember

The IP address must be specified under vbr_name by using the host address command before the API can be executed.

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- When **status** is **enable**
  - On success

"Add status **status** for netmongroup **nmg_name** of vBridge **vbr_name** VTN **vtn_name** succeeded."

- On failure

"Add status **status** for netmongroup **nmg_name** of vBridge **vbr_name** VTN **vtn_name** failed.(<Error information from pfcshell>)"

- When **status** is **disable**
  - On success

"Delete status **status** for netmongroup **nmg_name** of vBridge **vbr_name** VTN **vtn_name** succeeded."

  - On failure

"Update status **status** for netmongroup **nmg_name** of vBridge **vbr_name** VTN **vtn_name** failed. [Invalid argument(status)](<Error information from pfcshell>)"

Remark

None

7.9.5 Delete network monitor group

Processing request

Method

DELETE

request URI

- XML format

/vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**.xml

- JSON format

/vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**.json

Table 7-148 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete netmongroup **nmg_name** of vBridge **vbr_name** VTN **vtn_name** succeeded."

- On failure

  "Delete netmongroup **nmg_name** of vBridge **vbr_name** VTN **vtn_name** failed.(<Error information from pfcshell>)"

Remark

None

7.9.6 Show network monitor group details

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**/detail.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**/detail.json

Table 7-149 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Valid Value
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<netmongroup name="name"
  status="status">
  <hosts>
    [
    <host ipaddr="ipaddr"
      status="status"
      pingsend="pingsend"
      pingrecv="pingrecv"
      pingerr="pingerr"
      pingtrbl="pingtrbl"/>
    ]
  </hosts>
</netmongroup>
```

- JSON format

```
{
  "netmongroup" : {
    "name" : "name",
    "status" : "status",
    "hosts" : [
      {
        "ipaddr" : "ipaddr",
        "status" : "status",
        "pingsend" : "pingsend",
        "pingrecv" : "pingrecv",
        "pingerr" : "pingerr",
        "pingtrbl" : "pingtrbl"
      }
    ]
  }
}
```

Table 7-150 Description of Elements in netmongroups

Element	Description	Valid Value
name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	Network monitor group status	{health|trouble|disable} (Lowercase)
ipaddr	IP address of the Ping destination host	IPv4 dot-separated format (Example: 192.168.1.1)
status	Status of monitoring host (for display)	{health|trouble|-} (Lowercase)
pingsend	Number of pings sent to the monitoring host	Counter
pingrecv	Number of responses received for the pings sent to the monitoring host	Counter

Element	Description	Valid Value
pingerr	Number of ping errors for the monitoring host	Counter
pingtrbl	Number of ping failures for the monitoring host	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.10 Operate monitoring host

7.10.1 List monitor hosts in network monitor group

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/netmongroups/`**`nmg_name`**`/moniphosts.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/netmongroups/`**`nmg_name`**`/moniphosts.json`

Table 7-151 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Valid Value
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<hosts>
  [
  <host ipaddr="ipaddr"/>
  ]
</hosts>
```

- JSON format

```
{
  "hosts" : [
    {
      "ipaddr" : "ipaddr"
    }
  ]
}
```

Table 7-152 Description of Elements in hosts

Element	Description	Valid Value
ipaddr	IP address of the Ping destination host	IPv4 dot-separated format (Example: 192.168.1.1)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.10.2 Register monitor host to network monitor group

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**/moniphosts.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**/moniphosts.json

Table 7-153 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<host ipaddr="ipaddr"
  health_interval="health_interval"
  recovery_interval="recovery_interval"
  wait_time="wait_time"
  failure_counts="failure_counts"
  recovery_counts="recovery_counts="/>
```

- JSON format

```
{
  "host":{
    "ipaddr" : "ipaddr",
    "health_interval" : "health_interval",
    "recovery_interval" : "recovery_interval",
    "wait_time" : "wait_time",
    "failure_counts" : "failure_counts",
    "recovery_counts" : "recovery_counts"
  }
}
```

Remember

ipaddr must be specified though other items can be omitted.

Table 7-154 Description of Parameter in host

Element	Description	Valid Value
ipaddr	IP address of the Ping destination host	IPv4 dot-separated format (Example: 192.168.1.1)
health_interval	Ping health check transmission interval (sec.)	Decimal integer notation (5 to 600)
recovery_interval	Ping recovery check transmission interval (sec.)	Decimal integer notation (5 to 600)
wait_time	Ping response waiting time (sec.)	Decimal integer notation (1 to 60)
failure_counts	Count after which a failure is detected	Decimal integer notation (1 to 10)
recovery_counts	Count after which recovery is detected	Decimal integer notation (1 to 10)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create moniphost with ip address **ipaddr** for netmongroup **nmg_name** of vBridge **vbr_name** VTN **vtn_name** succeeded."

- On failure

  "Create moniphost with ip address **ipaddr** for netmongroup **nmg_name** of vBridge **vbr_name** VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.10.3 Show monitor host in network monitor group

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**/moniphosts/**ipaddr**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**/moniphosts/**ipaddr**.json

Table 7-155 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
ipaddr	IP address of the Ping destination host	IPv4 dot-separated format (Example: 192.168.1.1)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<host ipaddr="ipaddr"
  health_interval="health_interval"
  recovery_interval="recovery_interval"
  wait_time="wait_time"
  failure_counts="failure_counts"
  recovery_counts="recovery_counts"/>
```

- JSON format

```
{
  "host" : {
    "ipaddr" : "ipaddr",
    "health_interval" : "health_interval",
    "recovery_interval" : "recovery_interval",
    "wait_time" : "wait_time",
    "failure_counts" : "failure_counts",
```

```
    "recovery_counts" : "recovery_counts"
  }
}
```

Table 7-156 Description of Elements in host

Element	Description	Valid Value
ipaddr	IP address of the Ping destination host	IPv4 dot-separated format (Example: 192.168.1.1)
health_interval	Ping health check transmission interval (sec.)	Decimal integer notation (5 to 600)
recovery_interval	Ping recovery check transmission interval (sec.)	Decimal integer notation (5 to 600)
wait_time	Ping response waiting time (sec.)	Decimal integer notation (1 to 60)
failure_counts	Count after which a failure is detected	Decimal integer notation (1 to 10)
recovery_counts	Count after which recovery is detected	Decimal integer notation (1 to 10)

Remember

ipaddr must be specified though other items can be omitted.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.10.4 Unregister monitor host from network monitor group

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/netmongroups/`**`nmg_name`**`/moniphosts/`**`ipaddr`**`.xml`
- JSON format

/vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**/moniphosts/**ipaddr**.json

- Parameter

  The following parameters are passed in the URI query character string (in the format of ?param1=***&param2=***).

  health_interval=**health_interval**

  recovery_interval=**recovery_interval**

  wait_time=**wait_time**

  failure_counts=**failure_counts**

  recovery_counts=**recovery_counts**

Table 7-157 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
ipaddr	IP address of the Ping destination host	IPv4 dot-separated format (Example: 192.168.1.1)
health_interval	Ping health check transmission interval (sec.)	Decimal integer notation (5 to 600)
recovery_interval	Ping recovery check transmission interval (sec.)	Decimal integer notation (5 to 600)
wait_time	Ping response waiting time (sec.)	Decimal integer notation (1 to 60)
failure_counts	Count after which a failure is detected	Decimal integer notation (1 to 10)
recovery_counts	Count after which recovery is detected	Decimal integer notation (1 to 10)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete moniphost with ip address **ipaddr** for netmongroup **nmg_name** of vBridge **vbr_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete moniphost with ip address **ipaddr** for netmongroup **nmg_name** of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.10.5 Show monitor host details in network monitor group

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**/moniphosts/**ipaddr**/detail.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/netmongroups/**nmg_name**/moniphosts/**ipaddr**/detail.json

Table 7-158 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
ipaddr	IP address of the Ping destination host	IPv4 dot-separated format (Example: 192.168.1.1)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<netmongroup name="name"
  status="status">
  <host ipaddr="ipaddr"
    status="status"
    pingsend="pingsend"
    pingrecv="pingrecv"
    pingerr="pingerr"
    pingtrbl="pingtrbl"/>
</netmongroup>
```

- JSON format

```
{
  "netmongroup" : {
    "name" : "name",
    "status" : "status",
    "host" : {
      "ipaddr" : "ipaddr",
      "status" : "status",
      "pingsend" : "pingsend",
      "pingrecv" : "pingrecv",
      "pingerr" : "pingerr",
      "pingtrbl" : "pingtrbl"
    }
  }
}
```

Table 7-159 Description of Elements in netmongroup

Element	Description	Valid Value
name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
status	Network monitor group status	{health|trouble|disable} (Lowercase)
host	Host to be monitored	host type

Table 7-160 Description of Elements in host Type

Element	Description	Valid Value
ipaddr	IP address of the Ping destination host	IPv4 dot-separated format (Example: 192.168.1.1)
status	Status of monitoring host	{health|trouble|-} (Lowercase)
pingsend	Number of pings sent to the monitoring host	Counter
pingrecv	Number of responses received for the pings sent to the monitoring host	Counter
pingerr	Number of ping errors for the monitoring host	Counter
pingtrbl	Number of ping failures for the monitoring host	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.11 Operate Global Path Map

7.11.1 Show global path-map

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/pathmap.xml`
- JSON format

  `/realnetwork/pathmap.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathmap>
  <pathmapentries>
    [
    <pathmapentry  seqnum="seqnum"
      fl_name="fl_name"
      ip_version="ip_version"
      ppol_idx="ppol_idx"
      ppol_type="ppol_type"/>
```

```
    ]
  </pathmapentries>
</pathmap>
```

- JSON format

```
{
  "pathmap" : {
    "pathmapentries" : [
      {
        "seqnum" : "seqnum",
        "fl_name" : "fl_name",
        "ip_version" : "ip_version",
        "ppol_idx" : "ppol_idx",
        "ppol_type" : "ppol_type"
      }]
    }
}
```

Table 7-161 Description of Elements in pathmapentries

Element	Description	Valid Value
seqnum	Sequence number of path map entry	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6|-} (Lowercase)
ppol_idx	Index of path policy to be applied	0 to 4 or "-" (hyphen)
ppol_type	Type of path policy to be applied	{link-weights|-} (Lowercase)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.11.2 Show global path-map details

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/pathmap/detail.xml`

- JSON format

  `/realnetwork/pathmap/detail.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathmap>
  <pathmapentries>
    [
    <pathmapentry seqnum="seqnum"
      ppol_idx="ppol_idx"
      ppol_type="ppol_type">
      <statistics>
        <software packets="packets" octets="octets"/>
        <existingflow packets="packets" octets="octets"/>
        <expiredflow packets="packets" octets="octets"/>
        <total packets="packets" octets="octets"/>
      </statistics>
      <flowlist fl_name="fl_name" ip_version="ip_version">
        <flowlistentries>
          [
          <flowlistentry seqnum="seqnum"
            macdstaddr="macdstaddr"
            macsrcaddr="macsrcaddr"
            macethertype="macethertype"
            macvlanpriority="macvlanpriority"
            ipdstaddr="ipdstaddr"
            ipdstaddrprefix="ipdstaddrprefix"
            ipsrcaddr="ipsrcaddr"
            ipsrcaddrprefix="ipsrcaddrprefix"
            ipv6dstaddr="ipv6dstaddr"
            ipv6dstaddrprefix="ipv6dstaddrprefix"
            ipv6srcaddr="ipv6srcaddr"
            ipv6srcaddrprefix="ipv6srcaddrprefix"
            ipproto="ipproto"
            ipdscp="ipdscp"
            l4dstport="l4dstport"
            l4dstendport="l4dstendport"
            l4srcport="l4srcport"
            l4srcendport="l4srcendport"
            icmptypenum="icmptypenum"
            icmpcodenum="icmpcodenum"
            ipv6icmptypenum="ipv6icmptypenum"
            ipv6icmpcodenum="ipv6icmpcodenum">
            <statistics>
              <software packets="packets" octets="octets"/>
              <existingflow packets="packets" octets="octets"/>
              <expiredflow packets="packets" octets="octets"/>
              <total packets="packets" octets="octets"/>
            </statistics>
          </flowlistentry>
          ]
        </flowlistentries>
      </flowlist>
```

```
      <linkweights>
        [
        <linkweight dp_id="dp_id"
          port_name="port_name"
          weight="weight"/>
        ]
      </linkweights>
    </pathmapentry>
    ]
  </pathmapentries>
</pathmap>
```

- JSON format

```
{
  "pathmap" : {
    "pathmapentries" : [
      {
        "seqnum" : "seqnum",
        "ppol_idx" : "ppol_idx",
        "ppol_type" : "ppol_type",
        "statistics" : {
          "software" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expiredflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "total" : {
            "packets" : "packets",
            "octets" : "octets"
          }
        },
        "flowlist" : {
          "fl_name" : "fl_name",
          "ip_version" : "ip_version",
          "flowlistentries" : [
            {
              "seqnum" : "seqnum",
              "macdstaddr" : "macdstaddr",
              "macsrcaddr" : "macsrcaddr",
              "macethertype" : "macethertype",
              "macvlanpriority" : "macvlanpriority",
              "ipdstaddr" : "ipdstaddr",
              "ipdstaddrprefix" : "ipdstaddrprefix",
              "ipsrcaddr" : "ipsrcaddr",
              "ipsrcaddrprefix" : "ipsrcaddrprefix",
              "ipv6dstaddr" : "ipv6dstaddr",
              "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
              "ipv6srcaddr" : "ipv6srcaddr",
              "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
              "ipproto" : "ipproto",
              "ipdscp" : "ipdscp",
              "l4dstport" : "l4dstport",
              "l4dstendport" : "l4dstendport",
              "l4srcport" : "l4srcport",
              "l4srcendport" : "l4srcendport",
              "icmptypenum" : "icmptypenum",
              "icmpcodenum" : "icmpcodenum",
              "ipv6icmptypenum" : "ipv6icmptypenum",
              "ipv6icmpcodenum" : "ipv6icmpcodenum",
              "statistics" : {
                "software" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
```

```
                "existingflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "expiredflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "total" : {
                  "packets" : "packets",
                  "octets" : "octets"
                }
              }
            }
          ]
        },
        "linkweights" : [
          {
            "dp_id" : "dp_id",
            "port_name" : "port_name",
            "weight" : "weight"
          }
        ]
      }
    ]
  }
}
```

Table 7-162 Description of Elements in pathmapentries

Element	Description	Return Value
seqnum	Sequence number of path map entry	Decimal integer (1 to 65535)
ppol_idx	Index of path policy to be applied	0 to 4 or "-" (hyphen)
ppol_type	Type of path policy to be applied	{link-weights|-} (Lowercase)
statistics	Path map entry entry statistical information	Statistics
flowlist	Flowlist	Flowlist type
linkweight	Settings for linkweight between OFSs	Linkweight type

Table 7-163 Description of Elements in statistics

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	Statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	Statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	Statistics type
total	Total number of the flows that match the entry	Statistics type

Table 7-164 Description of Elements in flowlist

Element	Description	Return Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry	flowlistentries type

Table 7-165 Description of flowlistentries Type Elements in flowlist

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	0x0000 to 0xffff in hexadecimal
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	ddd.ddd.ddd.ddd format (d: decimal number)
ipdstaddrprefix	Destination IP address prefix length	Decimal number (1 to 32)
ipsrcaddr	Source IP address	ddd.ddd.ddd.ddd format (d: decimal number)
ipsrcaddrprefix	Source IP address prefix length	Decimal number (1 to 32)
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (1 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6dstaddr	Destination IPv6 address	ddd.ddd.ddd.ddd format (d: decimal number)
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal number (1 to 128)
ipv6srcaddr	Source IPv6 address	ddd.ddd.ddd.ddd format (d: decimal number)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal number (1 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal number (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal number (0 to 255)
statistics	Flowlist entry statistical information	statistics

Table 7-166 Description of statistics Type Elements in flowlistentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	Statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	Statistics type

Element	Description	Return Value
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	Statistics type
total	Total number of the flows that match the entry	Statistics type

Table 7-167 Description of Elements in Statistics Type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of octets	Counter

Table 7-168 Description of Elements in linkweights

Element	Description	Return Value
dp_id	DPID of the OFS to which linkweight is set	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number) * It is not allowed to specify F's to all digits.
port_name	Port name of the OFS to which linkweight is set	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
weight	linkweight between OFSs	Decimal number (1 to 4294967295)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.11.3 List sequence numbers of global path-map entry

Processing request

Method

GET

request URI

- XML format

/realnetwork/pathmap/pathmapentries.xml

- JSON format

  /realnetwork/pathmap/pathmapentries.json

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathmapentries>
  [
  <pathmapentry seqnum="seqnum"/>
  ]
</pathmapentries>
```

- JSON format

```
{
  "pathmapentries" : [
    {
      "seqnum" : "seqnum"
    }
  ]
}
```

Table 7-169 Description of Elements in pathmapentries

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.11.4 Create global path-map entry

Processing request

Method

POST

request URI

- XML format

  `/realnetwork/pathmap/pathmapentries.xml`

- JSON format

  `/realnetwork/pathmap/pathmapentries.json`

Settings of request body

- XML format

```
<pathmapentry seqnum="seqnum"
  fl_name="fl_name"
  ppol_idx="ppol_idx"
  ageout_time="ageout_time"/>
```

- JSON format

```
{
  "pathmapentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "ppol_idx" : "ppol_idx",
    "ageout_time" : "ageout_time"
  }
}
```

Remember

seqnum must be specified though other items can be omitted.

Table 7-170 Description of Elements in pathmapentry

Element	Description	Valid Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ppol_idx	Index of path policy to be applied	Decimal number (0 to 4)
ageout_time	ageout time to be applied (sec.)	Decimal number (0 to 65535)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  `"Create pathmap-entry with sequence number seqnum of path-map of real-network succeeded."`

- On failure

  `"Create pathmap-entry with sequence number seqnum of path-map of real-network or VTN failed. [Invalid argument] (<Error information from pfcshell>)"`

Remark

None

7.11.5 Show global path-map entry

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/pathmap/pathmapentries/seqnum.xml`

- JSON format

  `/realnetwork/pathmap/pathmapentries/seqnum.json`

Table 7-171 Description of Parameter in URI

Element	Description	Valid Value
seqnum	Sequence number of path map entry	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathmapentry seqnum="seqnum"
  fl_name="fl_name"
  ip_version="ip_version"
  ppol_idx="ppol_idx"
  ppol_type="ppol_type"/>
```

- JSON format

```
{
  "pathmapentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "ip_version" : "ip_version",
    "ppol_idx" : "ppol_idx",
    "ppol_type" : "ppol_type"
  }
}
```

Table 7-172 Description of Elements in pathmapentry

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6|-} (Lowercase)
ppol_idx	Index of path policy to be applied	Decimal number (0 to 4)
ppol_type	Type of path policy to be applied	{link-weights|-} (Lowercase)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get pathmap-entry info with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.11.6 Update global path-map entry

Processing request

Method

PUT

request URI

- XML format

  `/realnetwork/pathmap/pathmapentries/`**`seqnum`**`.xml`

- JSON format

  `/realnetwork/pathmap/pathmapentries/`**`seqnum`**`.json`

Table 7-173 Description of Parameter in URI

Element	Description	Valid Value
seqnum	Sequence number of path map entry	Decimal number (1 to 65535)

Settings of request body

- XML format

```
<pathmapentry fl_name="fl_name"
  ppol_idx="ppol_idx"
  ageout_time="ageout_time"
  op="op"/>
```

- JSON format

```
{
  "pathmapentry" : {
    "fl_name" : "fl_name",
    "ppol_idx" : "ppol_idx",
    "ageout_time" : "ageout_time",
    "op" : "op"
  }
}
```

Table 7-174 Description of Elements in pathmapentry

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ppol_idx	Index of path policy to be applied	Decimal number (0 to 4)
ageout_time	ageout time to be applied (sec.)	Decimal number (0 to 65535)
op	Operation type to be distinguished	{add|delete}

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- When **op** is **add**
  - On success

    "Create pathmap-entry with sequence number **seqnum** of path-map of real-network succeed."
  - On failure

    "Update pathmap-entry with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(seqnum)](<Error in formation from pfcshell>)"

    "Update pathmap-entry with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(**parameter subdivis ion**)](<Error information from pfcshell>)"
- When **op** is **delete**
  - On success

    "Delete pathmap-entry with sequence number **seqnum** of path-map of real-network succeed."
  - On failure

    "Update pathmap-entry with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(seqnum)](<Error in formation from pfcshell>)"

    "Update pathmap-entry with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(**parameter subdivis ion**)](<Error information from pfcshell>)"

Remark

None

7.11.7 Delete global path-map entry

Processing request

Method

DELETE

request URI

- XML format

  `/realnetwork/pathmap/pathmapentries/`**`seqnum`**`.xml`

- JSON format

  `/realnetwork/pathmap/pathmapentries/`**`seqnum`**`.json`

Table 7-175 Description of Parameter in URI

Element	Description	Valid Value
seqnum	Sequence number of path map entry	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Remember

If any path map entry no longer exists, the path-map mode which path map entries belong to is deleted from running-configuration.

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  `"Delete pathmap-entry with sequence number` **`seqnum`** `of path-map of real-network succeeded."`

- On failure

  `"Delete pathmap-entry with sequence number` **`seqnum`** `of path-map of real-network or VTN failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"`

Remark

None

7.11.8 Show global path-map entry details

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/pathmap/pathmapentries/`**`seqnum`**`/detail.xml`

- JSON format

  `/realnetwork/pathmap/pathmapentries/`**`seqnum`**`/detail.json`

Table 7-176 Description of Parameter in URI

Element	Description	Valid Value
seqnum	Sequence number of path map entry	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathmapentry seqnum="seqnum"
  ppol_idx="ppol_idx"
  ppol_type="ppol_type">
  <statistics>
    <software packets="packets" octets="octets"/>
    <existingflow packets="packets" octets="octets"/>
    <expiredflow packets="packets" octets="octets"/>
    <total packets="packets" octets="octets"/>
  </statistics>
  <flowlist fl_name="fl_name" ip_version="ip_version">
    <flowlistentries>
      [
      <flowlistentry seqnum="seqnum"
        macdstaddr="macdstaddr"
        macsrcaddr="macsrcaddr"
        macethertype="macethertype"
        macvlanpriority="macvlanpriority"
        ipdstaddr="ipdstaddr"
        ipdstaddrprefix="ipdstaddrprefix"
        ipsrcaddr="ipsrcaddr"
        ipsrcaddrprefix="ipsrcaddrprefix"
        ipv6dstaddr="ipv6dstaddr"
        ipv6dstaddrprefix="ipv6dstaddrprefix"
        ipv6srcaddr="ipv6srcaddr"
        ipv6srcaddrprefix="ipv6srcaddrprefix"
        ipproto="ipproto"
        ipdscp="ipdscp"
        l4dstport="l4dstport"
        l4dstendport="l4dstendport"
        l4srcport="l4srcport"
```

```
        l4srcendport="l4srcendport"
        icmptypenum="icmptypenum"
        icmpcodenum="icmpcodenum"
        ipv6icmptypenum="ipv6icmptypenum"
        ipv6icmpcodenum="ipv6icmpcodenum">
        <statistics>
          <software packets="packets" octets="octets"/>
          <existingflow packets="packets" octets="octets"/>
          <expiredflow packets="packets" octets="octets"/>
          <total packets="packets" octets="octets"/>
        </statistics>
      </flowlistentry>
      ]
    </flowlistentries>
  </flowlist>
  <linkweights>
    [
    <linkweight dp_id="dp_id"
      port_name="port_name"
      weight="weight"/>
    ]
  </linkweights>
</pathmapentry>
```

- JSON format

```
{
  "pathmapentry" : {
    "seqnum" : "seqnum",
    "ppol_idx" : "ppol_idx",
    "ppol_type" : "ppol_type",
    "statistics" : {
      "software" : {
        "packets" : "packets",
        "octets" : "octets"
      },
      "existingflow" : {
        "packets" : "packets",
        "octets" : "octets"
      },
      "expiredflow" : {
        "packets" : "packets",
        "octets" : "octets"
      },
      "total" : {
        "packets" : "packets",
        "octets" : "octets"
      }
    },
    "flowlist" : {
      "fl_name" : "fl_name",
      "ip_version" : "ip_version",
      "flowlistentries" : [
        {
          "seqnum" : "seqnum",
          "macdstaddr" : "macdstaddr",
          "macsrcaddr" : "macsrcaddr",
          "macethertype" : "macethertype",
          "macvlanpriority" : "macvlanpriority",
          "ipdstaddr" : "ipdstaddr",
          "ipdstaddrprefix" : "ipdstaddrprefix",
          "ipsrcaddr" : "ipsrcaddr",
          "ipsrcaddrprefix" : "ipsrcaddrprefix",
          "ipv6dstaddr" : "ipv6dstaddr",
          "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
          "ipv6srcaddr" : "ipv6srcaddr",
          "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
          "ipproto" : "ipproto",
          "ipdscp" : "ipdscp",
          "l4dstport" : "l4dstport",
          "l4dstendport" : "l4dstendport",
          "l4srcport" : "l4srcport",
```

```
          "l4srcendport" : "l4srcendport",
          "icmptypenum" : "icmptypenum",
          "icmpcodenum" : "icmpcodenum",
          "ipv6icmptypenum" : "ipv6icmptypenum",
          "ipv6icmpcodenum" : "ipv6icmpcodenum",
          "statistics" : {
            "software" : {
              "packets" : "packets",
              "octets" : "octets"
            },
            "existingflow" : {
              "packets" : "packets",
              "octets" : "octets"
            },
            "expiredflow" : {
              "packets" : "packets",
              "octets" : "octets"
            },
            "total" : {
              "packets" : "packets",
              "octets" : "octets"
            }
          }
        }
      ]
    },
    "linkweights" : [
      {
        "dp_id" : "dp_id",
        "port_name" : "port_name",
        "weight" : "weight"
      }
    ]
  }
}
```

Table 7-177 Description of Elements in pathmapentry

Element	Description	Return Value
seqnum	Sequence number of path map entry	Decimal number (1 to 65535)
ppol_idx	Index of path policy to be applied	Decimal number (0 to 4)
ppol_type	Type of path policy to be applied	{link-weights|-} (Lowercase)
statistics	Path map entry statistical information	statistics type
flowlist	Flowlist	Flowlist type
linkweight	Settings for linkweight between OFSs	Linkweight type

Table 7-178 Description of Elements in statistics

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	Statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	Statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	Statistics type
total	Total number of the flows that match the entry	Statistics type

Table 7-179 Description of Elements in flowlist

Element	Description	Return Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry	flowlistentries type

Table 7-180 Description of flowlistentries Type Elements in flowlist

Element	Description	Return Value
seqnum	Flowlist entry sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	0x0000 to 0xffff
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format
ipdstaddrprefix	Destination IP address prefix length	Decimal number (1 to 32)
ipsrcaddr	Source IP address	IPv4 dot-separated format
ipsrcaddrprefix	Source IP address prefix length	Decimal number (1 to 32)
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (1 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal number (1 to 128)
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal number (1 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal number (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal number (0 to 255)

Element	Description	Return Value
statistics	Flowlist entry statistical information	statistics

Table 7-181 Description of statistics Type Elements in flowlistentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	Statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	Statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	Statistics type
total	Total number of the flows that match the entry	Statistics type

Table 7-182 Description of Elements in Statistics Type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of octets	Counter

Table 7-183 Description of Elements in linkweights

Element	Description	Return Value
dp_id	DPID of the OFS to which linkweight is set	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
port_name	Port name of the OFS to which linkweight is set	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
weight	Linkweight between OFSs	Decimal number (1 to 4294967295)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- On failure

```
"Get pathmap-entry detail info with sequence number seqnum of path-m
ap of real-network or VTNfailed. [Invalid argument(seqnum)](<Error i
nformation from pfcshell>)"
```

Remark

None

7.12 Operate Path Policy

7.12.1 List path-policies

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/pathpolicies.xml`

- JSON format

  `/realnetwork/pathpolicies.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathpolicies>
  [
  <pathpolicy ppol_idx="ppol_idx"/>
  ]
</pathpolicies>
```

- JSON format

```
{
  "pathpolicies" : [
    {
      "ppol_idx" : "ppol_idx"
    }
  ]
}
```

Table 7-184 Description of Elements in pathpolicy

Element	Description	Valid Value
ppol_idx	Index of path policy to be applied	Decimal number (1 to 7)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.12.2 Create path-policy

Processing request

Method

POST

request URI

- XML format

  `/realnetwork/pathpolicies.xml`

- JSON format

  `/realnetwork/pathpolicies.json`

Settings of request body

- XML format

```
<pathpolicy ppol_idx="ppol_idx"/>
```

- JSON format

```
{
  "pathpolicy" : {
    "ppol_idx" : "ppol_idx"
  }
}
```

Table 7-185 Description of Elements in pathpolicy

Element	Description	Valid Value
ppol_idx	Index of path policy to be applied	Decimal number (1 to 3)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  `"Create pathpolicy ppol_idx to realnetwork succeeded."`

- On failure

  `"Create pathpolicy ppol_idx of realnetwork failed. [Invalid argumen t(ppol_idx)](<Error information from pfcshell>)"`

Remark

None

7.12.3 Show path-policy

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/pathpolicies/ppol_idx.xml`

- JSON format

  `/realnetwork/pathpolicies/ppol_idx.json`

Table 7-186 Description of Parameter in URI

Element	Description	Valid Value
ppol_idx	Index of path policy to be applied	Decimal number (1 to 7)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathpolicy ppol_idx="ppol_idx">
  <linkweights>
    [
    <linkweight dp_id="dp_id"
      port_name="port_name"
```

```
      weight="weight"/>
    ]
  </linkweights>
</pathpolicy>
```

- JSON format

```
{
  "pathpolicy" : {
    "ppol_idx" : "ppol_idx",
    "linkweights" : [
      {
        "dp_id" : "dp_id",
        "port_name" : "port_name",
        "weight" : "weight"
      }
    ]
  }
}
```

Table 7-187 Description of Elements in pathpolicy

Element	Description	Return Value
ppol_idx	Index of path policy to be applied	Decimal number (1 to 7)
linkweight	Settings for linkweight between OFSs	Linkweight type

Table 7-188 Description of Elements in linkweights

Element	Description	Return Value
dp_id	DPID of the OFS to which linkweight is set	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
port_name	Port name of the OFS to which linkweight is set	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
weight	linkweight between OFSs	Decimal number (1 to 4294967295)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Delete pathpolicy **ppol_idx** of realnetwork failed. [Invalid argumen t(ppol_idx)](<Error information from pfcshell>)"

Remark

None

7.12.4 Delete path-policy

Processing request

Method

DELETE

request URI

- XML format

  `/realnetwork/pathpolicies/`**`ppol_idx`**`.xml`

- JSON format

  `/realnetwork/pathpolicies/`**`ppol_idx`**`.json`

Table 7-189 Description of Parameter in URI

Element	Description	Valid Value
ppol_idx	Index of path policy to be applied	Decimal number (1 to 7)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  `"Delete pathpolicy` **`ppol_idx`** `to realnetwork succeeded."`

- On failure

  `"Delete pathpolicy` **`ppol_idx`** `of realnetwork failed. [Invalid argumen t(ppol_idx)](<Error information from pfcshell>)"`

Remark

None

7.13 Operate Linkweight

7.13.1 List Link weights

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/pathpolicies/`**`ppol_idx`**`/linkweights.xml`

- JSON format

  `/realnetwork/pathpolicies/`**`ppol_idx`**`/linkweights.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `dp_id=`**`dp_id`**

  `port_name=`**`port_name`**

Table 7-190 Description of Parameter in URI

Element	Description	Valid Value
ppol_idx	Index of path policy to be applied	Decimal number (1 to 7)
dp_id	DPID of the OFS to which linkweight is set	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
port_name	Port name of the OFS to which linkweight is set	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<linkweights>
  [
  <linkweight dp_id="dp_id"
    port_name="port_name"
    weight="weight"/>
  ]
</linkweights>
```

- JSON format

```
{
  "linkweights" : [
    {
      "dp_id" : "dp_id",
      "port_name" : "port_name",
      "weight" : "weight"
    }]
  }
```

Table 7-191 Description of Elements in linkweights

Element	Description	Return Value
dp_id	DPID of the OFS to which linkweight is set	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
port_name	Port name of the OFS to which linkweight is set	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
weight	linkweight between OFSs	Decimal number (1 to 4294967295)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get linkweight of pathpolicy **ppol_idx** failed. [Invalid argument(ppol_idx)](<Error information from pfcshell>)"

  "Get linkweight of pathpolicy **ppol_idx** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

7.13.2 Add/Delete Link weights

Processing request

Method

PUT

request URI

- XML format

  /realnetwork/pathpolicies/**ppol_idx**/linkweights.xml
- JSON format

```
/realnetwork/pathpolicies/ppol_idx/linkweights.json
```

Table 7-192 Description of Parameter in URI

Element	Description	Valid Value
ppol_idx	Index of path policy to be applied	Decimal number (1 to 7)

Settings of request body

- XML format

```
<linkweight
  dp_id="dp_id"
  port_name="port_name"
  weight="weight"
  op="op"/>
```

- JSON format

```
{
  "linkweight" : {
    "dp_id" : "dp_id",
    "port_name" : "port_name",
    "weight" : "weight",
    "op" : "op"
  }
}
```

Table 7-193 Description of Elements in linkweight

Element	Description	Valid Value
dp_id	DPID of the OFS to which linkweight is set	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
port_name	Port name of the OFS to which linkweight is set	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
weight	linkweight between OFSs	Decimal number (1 to 4294967295)
op	Setting type	{add|delete} (Lowercase)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- When **op** is **add**
  - On success

    "Add linkweight to pathpolicy **ppol_idx** succeeded."
  - On failure

    "Update linkweight of pathpolicy **ppol_idx** failed.([Invalid argument(ppol_idx)])(<Error information from pfcshell>)"

    "Update linkweight of pathpolicy **ppol_idx** failed.([Invalid argument(**parameter subdivision**)])(<Error information from pfcshell>)"
- When **op** is **delete**
  - On success

    "Delete linkweight to pathpolicy **ppol_idx** succeeded."
  - On failure

    "Update linkweight of pathpolicy **ppol_idx** failed.([Invalid argument(ppol_idx)])(<Error information from pfcshell>)"

    "Update linkweight of pathpolicy **ppol_idx** failed.([Invalid argument(**parameter subdivision**)])(<Error information from pfcshell>)"

Remark

None

7.14 Operate VTN Path Map

7.14.1 Show VTN path-map

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/pathmap.xml
- JSON format

  /vtns/**vtn_name**/pathmap.json

Table 7-194 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathmap>
  <pathmapentries>
    [
    <pathmapentry seqnum="seqnum"
      fl_name="fl_name"
      ip_version="ip_version"
      ppol_idx="ppol_idx"
      ppol_type="ppol_type"/>
    ]
  </pathmapentries>
</pathmap>
```

- JSON format

```
{
  "pathmap" : {
    "pathmapentries" : [
      {
        "seqnum" : "seqnum",
        "fl_name" : "fl_name",
        "ip_version" : "ip_version",
        "ppol_idx" : "ppol_idx",
        "ppol_type" : "ppol_type"
      }]
    }
}
```

Table 7-195 Description of Elements in pathmapentry

Element	Description	Return Value
seqnum	Sequence number of path map entry	Decimal number (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6|-} (Lowercase)
ppol_idx	Index of path policy to be applied	0 to 4 or "-" (hyphen)
ppol_type	Type of path policy to be applied	{link-weights|-} (Lowercase)

HTTP status code for response

- On success

  200 (OK)

- On failure

400,500

Miscellaneous

Log

None

Remark

None

7.14.2 Show VTN path-map details

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/pathmap/detail.xml

- JSON format

  /vtns/**vtn_name**/pathmap/detail.json

Table 7-196 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathmap>
  <pathmapentries>
    [
    <pathmapentry seqnum="seqnum"
      ppol_idx="ppol_idx"
      ppol_type="ppol_type">
      <statistics>
        <software packets="packets" octets="octets"/>
        <existingflow packets="packets" octets="octets"/>
        <expiredflow packets="packets" octets="octets"/>
        <total packets="packets" octets="octets"/>
      </statistics>
      <flowlist fl_name="fl_name" ip_version="ip_version">
```

```
        <flowlistentries>
          [
          <flowlistentry seqnum="seqnum"
            macdstaddr="macdstaddr"
            macsrcaddr="macsrcaddr"
            macethertype="macethertype"
            macvlanpriority="macvlanpriority"
            ipdstaddr="ipdstaddr"
            ipdstaddrprefix="ipdstaddrprefix"
            ipsrcaddr="ipsrcaddr"
            ipsrcaddrprefix="ipsrcaddrprefix"
            ipv6dstaddr="ipv6dstaddr"
            ipv6dstaddrprefix="ipv6dstaddrprefix"
            ipv6srcaddr="ipv6srcaddr"
            ipv6srcaddrprefix="ipv6srcaddrprefix"
            ipproto="ipproto"
            ipdscp="ipdscp"
            l4dstport="l4dstport"
            l4dstendport="l4dstendport"
            l4srcport="l4srcport"
            l4srcendport="l4srcendport"
            icmptypenum="icmptypenum"
            icmpcodenum="icmpcodenum"
            ipv6icmptypenum="ipv6icmptypenum"
            ipv6icmpcodenum="ipv6icmpcodenum">
            <statistics>
              <software packets="packets" octets="octets"/>
              <existingflow packets="packets" octets="octets"/>
              <expiredflow packets="packets" octets="octets"/>
              <total packets="packets" octets="octets"/>
            </statistics>
          </flowlistentry>
          ]
        </flowlistentries>
      </flowlist>
      <linkweights>
        [
        <linkweight dp_id="dp_id"
          port_name="port_name"
          weight="weight"/>
        ]
      </linkweights>
    </pathmapentry>
    ]
  </pathmapentries>
</pathmap>
```

- JSON format

```
{
  "pathmap" : {
    "pathmapentries" : [
      {
        "seqnum" : "seqnum",
        "ppol_idx" : "ppol_idx",
        "ppol_type" : "ppol_type",
        "statistics" : {
          "software" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expiredflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "total" : {
            "packets" : "packets",
            "octets" : "octets"
```

```
          }
        },
        "flowlist" : {
          "fl_name" : "fl_name",
          "ip_version" : "ip_version",
          "flowlistentries" : [
            {
              "seqnum" : "seqnum",
              "macdstaddr" : "macdstaddr",
              "macsrcaddr" : "macsrcaddr",
              "macethertype" : "macethertype",
              "macvlanpriority" : "macvlanpriority",
              "ipdstaddr" : "ipdstaddr",
              "ipdstaddrprefix" : "ipdstaddrprefix",
              "ipsrcaddr" : "ipsrcaddr",
              "ipsrcaddrprefix" : "ipsrcaddrprefix",
              "ipv6dstaddr" : "ipv6dstaddr",
              "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
              "ipv6srcaddr" : "ipv6srcaddr",
              "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
              "ipproto" : "ipproto",
              "ipdscp" : "ipdscp",
              "l4dstport" : "l4dstport",
              "l4dstendport" : "l4dstendport",
              "l4srcport" : "l4srcport",
              "l4srcendport" : "l4srcendport",
              "icmptypenum" : "icmptypenum",
              "icmpcodenum" : "icmpcodenum",
              "ipv6icmptypenum" : "ipv6icmptypenum",
              "ipv6icmpcodenum" : "ipv6icmpcodenum",
              "statistics" : {
                "software" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "existingflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "expiredflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "total" : {
                  "packets" : "packets",
                  "octets" : "octets"
                }
              }
            }
          ]
        },
        "linkweights" : [
          {
            "dp_id" : "dp_id",
            "port_name" : "port_name",
            "weight" : "weight"
          }
        ]
      }
    ]
  }
}
```

Table 7-197 Description of Elements in pathmapentries

Element	Description	Return Value
seqnum	Sequence number of path map entry	Decimal integer (1 to 65535)
ppol_idx	Index of path policy to be applied	0 to 4 or "-" (hyphen)
ppol_type	Type of path policy to be applied	{link-weights|-} (Lowercase)

Element	Description	Return Value
statistics	Path map entry statistical information	Statistics
flowlist	Flowlist	Flowlist type
linkweight	Settings for linkweight between OFSs	Linkweight type

Table 7-198 Description of Elements in statistics

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-199 Description of Elements in flowlist

Element	Description	Return Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores (The underbar cannot be specified at the beginning.)
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentry	Flowlist entry	flowlistentry type

Table 7-200 Description of flowlistentries Type Elements in flowlist

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	0x0000 to 0xffff
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	ddd.ddd.ddd.ddd format (d: decimal number)
ipdstaddrprefix	Destination IP address prefix length	Decimal number (1 to 32)
ipsrcaddr	Source IP address	ddd.ddd.ddd.ddd format (d: decimal number)
ipsrcaddrprefix	Source IP address prefix length	Decimal number (1 to 32)
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (1 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)

Element	Description	Return Value
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6Address format (conforming to RFC 5952)
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal number (1 to 128)
ipv6srcaddr	Source IPv6 address	IPv6Address format (conforming to RFC 5952)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal number (1 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal number (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal number (0 to 255)
statistics	Flowlit entry statistical information	statistics

Table 7-201 Description of statistics Type Elements in flowfilterentry

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-202 Description of Elements in Statistics Type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of octets	Counter

Table 7-203 Description of Elements in linkweights

Element	Description	Return Value
dp_id	DPID of the OFS to which linkweight is set	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number) * It is not allowed to specify F's to all digits.
port_name	Port name of the OFS to which linkweight is set	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
weight	linkweight between OFSs	Decimal number (1 to 4294967295)

HTTP status code for response

- On success

200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.14.3 List sequence numbers of VTN path-map entry

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/pathmap/pathmapentries.xml`

- JSON format

  `/vtns/vtn_name/pathmap/pathmapentries.json`

Table 7-204 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathmapentries>
   [<pathmapentry seqnum="seqnum"/>]
</pathmapentries>
```

- JSON format

```
{
  "pathmapentries" : [{
```

```
    "seqnum" : "seqnum"
  }]
}
```

Table 7-205 Description of Elements in pathmapentries

Element	Description	Return Value
seqnum	Sequence number of path map entry	Decimal number (1 to 65535)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.14.4 Create VTN path-map entry

Processing request

Method

POST

3.39.4.1.2 request URI

- XML format

  `/vtns/vtn_name/pathmap/pathmapentries.xml`

- JSON format

  `/vtns/vtn_name/pathmap/pathmapentries.json`

Table 7-206 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<pathmapentry seqnum="seqnum"
  fl_name="fl_name"
```

```
  ppol_idx="ppol_idx"
  ageout_time="ageout_time"/>
```

- JSON format

```
{
  "pathmapentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "ppol_idx" : "ppol_idx",
    "ageout_time" : "ageout_time"
    }
}
```

Remember

seqnum must be specified though other items can be omitted.

Table 7-207 Description of Elements in pathmapentry

Element	Description	Valid Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ppol_idx	Index of path policy to be applied	Decimal number (0 to 4)
ageout_time	ageout time to be applied (sec.)	Decimal number (0 to 65535)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create pathmap-entry with sequence number **seqnum** of path-map of VTN **vtn_name** succeeded."

- On failure

  "Create pathmap-entry with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.14.5 Show VTN path-map entry

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/pathmap/pathmapentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/pathmap/pathmapentries/**seqnum**.json

Table 7-208 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
seqnum	Path map entry sequence number	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathmapentry seqnum="seqnum"
  fl_name="fl_name"
  ip_version="ip_version"
  ppol_idx="ppol_idx"
  ppol_type="ppol_type"/>
```

- JSON format

```
{
  "pathmapentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "ip_version" : "ip_version",
    "ppol_idx" : "ppol_idx",
    "ppol_type" : "ppol_type"
  }
}
```

Table 7-209 Description of Elements in pathmapentry

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ip_version	IP version	{ip|ipv6} (Lowercase)
ppol_idx	Index of path policy to be applied	Decimal number (0 to 4)
ppol_type	Type of path policy to be applied	{link-weights|-} (Lowercase)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get pathmap-entry info with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.14.6 Update VTN path-map entry

Processing request

Method

PUT

request URI

- XML format

  /vtns/**vtn_name**/pathmap/pathmapentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/pathmap/pathmapentries/**seqnum**.json

Table 7-210 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
seqnum	Path map entry sequence number	Decimal number (1 to 65535)

Settings of request body

- XML format

```
<pathmapentry fl_name="fl_name"
  ppol_idx="ppol_idx"
  ageout_time="ageout_time"
  op="op"/>
```

- JSON format

```
{
  "pathmapentry" : {
    "fl_name" : "fl_name",
    "ppol_idx" : "ppol_idx",
    "ageout_time" : "ageout_time",
    "op" : "op"
  }
}
```

Table 7-211 Description of Elements in pathmapentry

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
ppol_idx	Index of path policy to be applied	Decimal number (0 to 4)
ageout_time	ageout time to be applied (sec.)	Decimal number (0 to 65535)
op	Type of operation to be indentified	{add|delete} (Lowercase)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- When **op** is **add**

- On success

  "Create pathmap-entry with sequence number **seqnum** of path-map of VTN **vtn_name** succeeded."

- On failure

  "Update pathmap-entry with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(seqnum)](<Error in formation from pfcshell>)"

  "Update pathmap-entry with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

- When **op** is **delete**
  - On success

    "Delete pathmap-entry with sequence number **seqnum** of path-map of VTN **vtn_name** succeeded."

  - On failure

    "Update pathmap-entry with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(seqnum)](<Error in formation from pfcshell>)"

    "Update pathmap-entry with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

7.14.7 Delete VTN path-map entry

Processing request

Method

DELETE

request URI

- XML format

  /vtns/**vtn_name**/pathmap/pathmapentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/pathmap/pathmapentries/**seqnum**.json

Table 7-212 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Element	Description	Valid Value
seqnum	Path map entry sequence number	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Remember

If any path map entry no longer exists, the path-map mode which path map entries belong to is deleted from running-configuration.

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete pathmap-entry with sequence number **seqnum** of path-map of real-network succeed."

- On failure

  "Delete pathmap-entry with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.14.8 Show VTN path-map entry details

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/pathmap/pathmapentries/seqnum/detail.xml`

- JSON format

  `/vtns/vtn_name/pathmap/pathmapentries/seqnum/detail.json`

Table 7-213 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
seqnum	Path map entry sequence number	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pathmapentry seqnum="seqnum"
  ppol_idx="ppol_idx"
  ppol_type="ppol_type">
  <statistics>
    <software packets="packets" octets="octets"/>
    <existingflow packets="packets" octets="octets"/>
    <expiredflow packets="packets" octets="octets"/>
    <total packets="packets" octets="octets"/>
  </statistics>
  <flowlist fl_name="fl_name" ip_version="ip_version">
    <flowlistentries>
      [
      <flowlistentry seqnum="seqnum"
        macdstaddr="macdstaddr"
        macsrcaddr="macsrcaddr"
        macethertype="macethertype"
        macvlanpriority="macvlanpriority"
        ipdstaddr="ipdstaddr"
        ipdstaddrprefix="ipdstaddrprefix"
        ipsrcaddr="ipsrcaddr"
        ipsrcaddrprefix="ipsrcaddrprefix"
        ipv6dstaddr="ipv6dstaddr"
        ipv6dstaddrprefix="ipv6dstaddrprefix"
        ipv6srcaddr="ipv6srcaddr"
        ipv6srcaddrprefix="ipv6srcaddrprefix"
        ipproto="ipproto"
        ipdscp="ipdscp"
        l4dstport="l4dstport"
        l4dstendport="l4dstendport"
        l4srcport="l4srcport"
        l4srcendport="l4srcendport"
        icmptypenum="icmptypenum"
        icmpcodenum="icmpcodenum"
        ipv6icmptypenum="ipv6icmptypenum"
        ipv6icmpcodenum="ipv6icmpcodenum">
        <statistics>
          <software packets="packets" octets="octets"/>
          <existingflow packets="packets" octets="octets"/>
```

```
          <expiredflow packets="packets" octets="octets"/>
          <total packets="packets" octets="octets"/>
        </statistics>
      </flowlistentry>
      ]
    </flowlistentries>
  </flowlist>
  <linkweights>
    [
    <linkweight dp_id="dp_id"
      port_name="port_name"
      weight="weight"/>
    ]
  </linkweights>
</pathmapentry>
```

- JSON format

```
{
  "pathmapentry" : {
    "seqnum" : "seqnum",
    "ppol_idx" : "ppol_idx",
    "ppol_type" : "ppol_type",
    "statistics" : {
      "software" : {
        "packets" : "packets",
        "octets" : "octets"
      },
      "existingflow" : {
        "packets" : "packets",
        "octets" : "octets"
      },
      "expiredflow" : {
        "packets" : "packets",
        "octets" : "octets"
      },
      "total" : {
        "packets" : "packets",
        "octets" : "octets"
      }
    },
    "flowlist" : {
      "fl_name" : "fl_name",
      "ip_version" : "ip_version",
      "flowlistentries" : [
        {
          "seqnum" : "seqnum",
          "macdstaddr" : "macdstaddr",
          "macsrcaddr" : "macsrcaddr",
          "macethertype" : "macethertype",
          "macvlanpriority" : "macvlanpriority",
          "ipdstaddr" : "ipdstaddr",
          "ipdstaddrprefix" : "ipdstaddrprefix",
          "ipsrcaddr" : "ipsrcaddr",
          "ipsrcaddrprefix" : "ipsrcaddrprefix",
          "ipv6dstaddr" : "ipv6dstaddr",
          "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
          "ipv6srcaddr" : "ipv6srcaddr",
          "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
          "ipproto" : "ipproto",
          "ipdscp" : "ipdscp",
          "l4dstport" : "l4dstport",
          "l4dstendport" : "l4dstendport",
          "l4srcport" : "l4srcport",
          "l4srcendport" : "l4srcendport",
          "icmptypenum" : "icmptypenum",
          "icmpcodenum" : "icmpcodenum",
          "ipv6icmptypenum" : "ipv6icmptypenum",
          "ipv6icmpcodenum" : "ipv6icmpcodenum",
          "statistics" : {
            "software" : {
              "packets" : "packets",
```

```
              "octets" : "octets"
            },
            "existingflow" : {
              "packets" : "packets",
              "octets" : "octets"
            },
            "expiredflow" : {
              "packets" : "packets",
              "octets" : "octets"
            },
            "total" : {
              "packets" : "packets",
              "octets" : "octets"
            }
          }
        }
      ]
    },
    "linkweights" : [
      {
        "dp_id" : "dp_id",
        "port_name" : "port_name",
        "weight" : "weight"
      }
    ]
  }
}
```

Table 7-214 Description of Elements in pathmapentry

Element	Description	Return Value
seqnum	Sequence number of path map entry	Decimal integer (1 to 65535)
ppol_idx	Index of path policy to be applied	0 to 4 or "-" (hyphen)
ppol_type	Type of path policy to be applied	{link-weights|-} (Lowercase)
statistics	Path map entry statistical information	Statistics
flowlist	Flowlist	Flowlist type
linkweight	Settings for linkweight between OFSs	Linkweight type

Table 7-215 Description of Elements in statistics

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	Statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	Statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	Statistics type
total	Total number of the flows that match the entry	Statistics type

Table 7-216 Description of Elements in flowlist

Element	Description	Return Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores (The underbar cannot be specified at the beginning.)
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentry	Flowlist entry	flowlistentry type

Table 7-217 Description of flowlistentries Type Elements in flowlist

Element	Description	Return Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
macethertype	Ether type of ethernet frame	0x0000 to 0xffff
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format
ipdstaddrprefix	Destination IP address prefix length	Decimal number (1 to 32)
ipsrcaddr	Source IP address	IPv4 dot-separated format
ipsrcaddrprefix	Source IP address prefix length	Decimal number (1 to 32)
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (1 to 63)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal number (0 to 65535)
l4srcendport	Source TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address format (conforming to RFC 5952)
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal number (1 to 128)
ipv6srcaddr	Source IPv6 address	IPv6 address format (conforming to RFC 5952)
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal number (1 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal number (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal number (0 to 255)
statistics	Flowlist entry statistical information	statistics

Table 7-218 Description of statistics Type Elements in flowlistentries

Element	Description	Return Value
software	Of the flows that match the entry, frames that passed through the VTN	Statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	Statistics type

Element	Description	Return Value
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	Statistics type
total	Total number of the flows that match the entry	Statistics type

Table 7-219 Description of Elements in Statistics Type

Element	Description	Return Value
packets	Number of packets	Counter
octets	Number of octets	Counter

Table 7-220 Description of Elements in linkweights

Element	Description	Return Value
dp_id	DPID of the OFS to which linkweight is set	hhhh-hhhh-hhhh-hhhh format (h: hexadecimal number)
port_name	Port name of the OFS to which linkweight is set	Up to 15 characters including ascii alphanumeric characters except for a question mark (?)
weight	Linkweight between OFSs	Decimal number (1 to 4294967295)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get pathmap-entry detail info with sequence number **seqnum** of path-map of real-network or VTN failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

7.15 Operate IP Routing Information Using Net Monitoring Function

7.15.1 Show IP routes set by network monitor

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vrouters/**vrt_name**/nmg_iproutes.xml

- JSON format

  /vtns/**vtn_name**/vrouters/**vrt_name**/nmg_iproutes.json

Table 7-221 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<nmg_iproutes>
  [
  <nmg_iproute
    nmg_iproute_id="nmg_iproute_id"
    ipaddr="ipaddr"
    nexthopaddr="nexthopaddr"
    netmask="netmask"/>
  ]
</nmg_iproutes>
```

- JSON format

```
{
  "nmg_iproutes" : [
    {
      "nmg_iproute_id" : "nmg_iproute_id",
      "ipaddr" : "ipaddr",
      "nexthopaddr" : "nexthopaddr",
      "netmask" : "netmask"
```

```
      }
    ]
}
```

Table 7-222 Description of Elements in nmg_iproutes

Element	Description	Return Value
nmg_iproute_id	Path information ID when using the network monitoring function	Up to 79 characters
ipaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
nexthopaddr	IP address of the router to which packets addressed to ipaddr are transferred	IPv4 dot-separated format (Example: 192.168.1.1)
netmask	Destination IP address subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.15.2 Set IP route by network monitor to vRouter

Processing request

Method

POST

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/nmg_iproutes.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vrouters/`**`vrt_name`**`/nmg_iproutes.json`

Table 7-223 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<nmg_iproute
  ipaddr="ipaddr"
  prefix="prefix"
  netmask="netmask"
  nexthopaddr="nexthopaddr"
  netgroup_name="netgroup_name"
  groupmetric="groupmetric"/>
```

- JSON format

```
{
  "nmg_iproute" : {
    "ipaddr" : "ipaddr",
    "prefix" : "prefix",
    "netmask" : "netmask",
    "nexthopaddr" : "nexthopaddr",
    "netgroup_name" : "netgroup_name",
    "groupmetric" : "groupmetric"
  }
}
```

Table 7-224 Description of Elements in nmg_iproute

Element	Description	Valid Value
ipaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
prefix	IP address prefix length	Decimal number (0 to 32)
netmask	Destination IP address subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)
nexthopaddr	IP address of the router to which packets addressed to ipaddr are transferred	IPv4 dot-separated format (Example: 192.168.1.1)
netgroup_name	Name of network group to be monitored when using the net monitoring function	Up to 31 characters including one-byte alphanumeric characters and underscores * "-" (hyphen) if the net monitoring function is not used.
groupmetric	Priorities for multiple paths	Decimal number (1 to 65535)

Remember

prefix and netmask cannot be specified at the same time.

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<nmg_iproute nmg_iproute_id="nmg_iproute_id"/>
```

- JSON format

```
{
  "nmg_iproute" : {
    "nmg_iproute_id" : "nmg_iproute_id"
  }
}
```

Table 7-225 Description of Elements in nmg_iproute

Element	Description	Return Value
nmg_iproute_id	Path information ID when using the network monitoring function	Up to 79 characters

Remember

nmg_iproute_id is created with the fixed form from the values included in a request, but the form will possibly be changed without notice.

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create iproute of vRouter **vrt_name** of VTN **vtn_name** succeeded."

- On failure

  "Create iproute of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(prefix,netmask)](<Error information from pfcshell>)"

  "Create iproute of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

7.15.3 Delete IP route set by network monitor from vRouter

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/vtn_name/vrouters/vrt_name/nmg_iproutes/nmg_iproute_id.xml`

- JSON format

  `/vtns/vtn_name/vrouters/vrt_name/nmg_iproutes/nmg_iproute_id.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `ipaddr=ipaddr`

  `prefix=prefix`

  `netmask=netmask`

  `nexthopaddr=nexthopaddr`

  `netgroup_name=netgroup_name`

  `groupmetric=groupmetric`

Table 7-226 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vrt_name	vRouter name	Up to 31 characters including one-byte alphanumeric characters and underscores
nmg_iproute_id	Path information ID when using the network monitoring function	Up to 79 characters
ipaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
prefix	IP address prefix length	Decimal number (0 to 32)
netmask	Destination IP address subnet mask	IPv4 dot-separated format (Example: 255.255.255.0)
nexthopaddr	IP address of the router to which packets addressed to ipaddr are transferred	IPv4 dot-separated format (Example: 192.168.1.1)
netgroup_name	Name of network group to be monitored when using the net monitoring function	Up to 31 characters including one-byte alphanumeric characters and underscores
groupmetric	Priorities for multiple paths	Decimal number (1 to 65535)

Remember

prefix and netmask cannot be specified at the same time.

Remember

The specified nmg_iproute_id in URI and the specified ipaddr, prefix, netmask, nexthopaddr, and netgroup_name in the query character string must be equal to the values returned in response body of "7.15.1 Show IP routes set by network monitor (page 419)" and "7.15.2 Set IP route by network monitor to vRouter (page 420)" .

Processing result

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete nmg_iproute **nmg_iproute_id** for vRouter **vrt_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete nmg_iproute of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(prefix,netmask)](<Error information from pfcshell>)"

  "Delete nmg_iproute of vRouter **vrt_name** of VTN **vtn_name** failed. [Invalid argument(**parameter subdivision**)](<Error information from pfcshell>)"

Remark

None

7.16 Operate Flow Filter (VTN)

7.16.1 List flow filters for VTN

Processing request

Method

GET

request URI

- XML format

/vtns/**vtn_name**/flowfilters.xml

- JSON format

/vtns/**vtn_name**/flowfilters.json

Table 7-227 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilters>
  [
  <flowfilter ff_type="ff_type"/>
  ]
</flowfilters>
```

- JSON format

```
{
  "flowfilters" : [
    {
      "ff_type" : "ff_type"
    }
  ]
}
```

Table 7-228 Description of Elements in flowfilters

Element	Description	Valid Value
ff_type	Flow filter application direction	{in|out}

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.16.2 Create flow filter for VTN

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/flowfilters.xml

- JSON format

  /vtns/**vtn_name**/flowfilters.json

Table 7-229 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<flowfilter ff_type="ff_type"/>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type"
  }
}
```

Table 7-230 Description of Elements in flowfilter

Element	Description	Valid Value
ff_type	Flow filter application direction	{in|out}

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

400,500

Miscellaneous

Log

- On success

  "Create flowfilter **ff_type** for VTN **vtn_name** succeeded. "

- On failure

  "Create flowfilter **ff_type** for VTN **vtn_name** failed. [Invalid argumen t(ff_type)](<Error information from pfcshell>)"

Remark

None

7.16.3 Show flow filter for VTN

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/flowfilters/**ff_type**.xml

- JSON format

  /vtns/**vtn_name**/flowfilters/**ff_type**.json

Table 7-231 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentries>
    [
```

```
    <flowfilterentry seqnum="seqnum"
      fl_name="fl_name"
      action_type="action_type"
      nmg_name="nmg_name"/>
    ]
  </flowfilterentries>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentries" : [
      {
        "seqnum" : "seqnum",
        "fl_name" : "fl_name",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name"
      }
    ]
  }
}
```

Table 7-232 Description of Elements in flowfilter

Element	Description	Valid Value
ff_type	Flow filter application direction	{in|out}
flowfilterentries	Flow filter entry list	flowfilterentries type

Table 7-233 Description of Elements in flowfilter

Element	Description	Valid Value
seqnum	Flow filter entry sequence number	Decimal integer (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matching Ethernet frame	{pass|-} (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.16.4 Delete flow filter for VTN

Processing request

Method

DELETE

request URI

- XML format

  /vtns/**vtn_name**/flowfilters/**ff_type**.xml

- JSON format

  /vtns/**vtn_name**/flowfilters/**ff_type**.json

Table 7-234 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 32 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete flowfilter with flowfilter type **ff_type** for VTN **vtn_name** succeeded. "

- On failure

  "Delete flowfilter with flowfilter type **ff_type** for VTN **vtn_name** failed. (<Error information from pfcshell>) "

Remark

None

7.16.5 Show flow filter details for VTN

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/flowfilters/**ff_type**/detail.xml

- JSON format

  /vtns/**vtn_name**/flowfilters/**ff_type**/detail.json

Table 7-235 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentries>
    [
    <flowfilterentry seqnum="seqnum"
      action_type="action_type" nmg_name="nmg_name">
      <statistics>
        <software packets="packets" octets="octets"/>
        <existingflow packets="packets" octets="octets"/>
        <expiredflow packets="packets" octets="octets"/>
        <total packets="packets" octets="octets"/>
      </statistics>
      <set priority="priority" dscp="dscp"/>
      <flowlist fl_name="fl_name" ip_version="ip_version">
        <flowlistentries>
          [
          <flowlistentry seqnum="seqnum"
            macdstaddr="macdstaddr"
            macsrcaddr="macsrcaddr"
            macethertype="macethertype"
            macvlanpriority="macvlanpriority"
            ipdstaddr="ipdstaddr"
```

```
            ipdstaddrprefix="ipdstaddrprefix"
            ipsrcaddr="ipsrcaddr"
            ipsrcaddrprefix="ipsrcaddrprefix"
            ipv6dstaddr="ipv6dstaddr"
            ipv6dstaddrprefix="ipv6dstaddrprefix"
            ipv6srcaddr="ipv6srcaddr"
            ipv6srcaddrprefix="ipv6srcaddrprefix"
            ipproto="ipproto"
            ipdscp="ipdscp"
            l4dstport="l4dstport"
            l4dstendport="l4dstendport"
            l4srcport="l4srcport"
            l4srcendport="l4srcendport"
            icmptypenum="icmptypenum"
            icmpcodenum="icmpcodenum"
            ipv6icmptypenum="ipv6icmptypenum"
            ipv6icmpcodenum="ipv6icmpcodenum">
            <statistics>
              <software packets="packets" octets="octets"/>
              <existingflow packets="packets" octets="octets"/>
              <expiredflow packets="packets" octets="octets"/>
              <total packets="packets" octets="octets"/>
            </statistics>
          </flowlistentry>
          ]
        </flowlistentries>
      </flowlist>
    </flowfilterentry>
    ]
  </flowfilterentries>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentries" : [
      {
        "seqnum" : "seqnum",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name",
        "statistics" : {
          "software" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expiredflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "total" : {
            "packets" : "packets",
            "octets" : "octets"
          }
        },
        "set" : {
          "priority" : "priority",
          "dscp" : "dscp"
        },
        "flowlist" : {
          "fl_name" : "fl_name",
          "ip_version" : "ip_version",
          "flowlistentries" : [
            {
              "seqnum" : "seqnum",
              "macdstaddr" : "macdstaddr",
              "macsrcaddr" : "macsrcaddr",
```

```
                "macethertype" : "macethertype",
                "macvlanpriority" : "macvlanpriority",
                "ipdstaddr" : "ipdstaddr",
                "ipdstaddrprefix" : "ipdstaddrprefix",
                "ipsrcaddr" : "ipsrcaddr",
                "ipsrcaddrprefix" : "ipsrcaddrprefix",
                "ipv6dstaddr" : "ipv6dstaddr",
                "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
                "ipv6srcaddr" : "ipv6srcaddr",
                "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
                "ipproto" : "ipproto",
                "ipdscp" : "ipdscp",
                "l4dstport" : "l4dstport",
                "l4dstendport" : "l4dstendport",
                "l4srcport" : "l4srcport",
                "l4srcendport" : "l4srcendport",
                "icmptypenum" : "icmptypenum",
                "icmpcodenum" : "icmpcodenum",
                "ipv6icmptypenum" : "ipv6icmptypenum",
                "ipv6icmpcodenum" : "ipv6icmpcodenum",
                "statistics" : {
                  "software" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  },
                  "existingflow" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  },
                  "expiredflow" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  },
                  "total" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  }
                }
              }
            ]
          }
        }
      ]
    }
}
```

Table 7-236 Description of Elements in flowfilter

Element	Description	Valid Value
ff_type	Flow filter type	{in|out}
flowfilterentries	Flow filter entry	flowfilterentries type

Table 7-237 Description of Elements in flowfilterentries type

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)
action_type	Action for the matched ethernet frame	{pass|-} (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
statistics	Statistical information of flow filter entry	statistics type
set	Settings based on the matching flow	set type
flowlist	Flowlist information	flowlist type

Table 7-238 Description of Elements of statistics type in flowfilterentries

Element	Description	Valid Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-239 Description of Elements in set type

Element	Description	Valid Value
priority	Packet transfer priority	Decimal number (0 to 7)
dscp	DSCP value	Decimal number (0 to 63)

Table 7-240 Description of Elements in flowlist type

Element	Description	Valid Value
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
ip_version	IP Version	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry information	flowlistentries type

Table 7-241 Description of Elements in flowlistentries type

Element	Description	Valid Value
seqnum	Flowlist entry sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macethertype	Ethernet frame type	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Prefix length of destination IP address	Decimal number (1 to 32)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddrprefix	Prefix length of source IP address	Decimal number (1 to 32)
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (0 to 63)
l4dstport	TCP/UDP destination port number (the first port number in the case of range specification)	Decimal number (1 to 65535)
l4dstendport	The last TCP/UDP port number in the case of range specification	Decimal number (1 to 65535)

Element	Description	Valid Value
l4srcport	TCP/UDP source port number (the first port number in the case of range specification)	Decimal number (1 to 65535)
l4srcendport	The last TCP/UDP port number in the case of range specification	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Prefix length of destination IP address	Decimal number (0 to 128)
ipv6srcaddr	Source IP address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddrprefix	Prefix length of source IP address	Decimal number (0 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal number (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal number (0 to 255)
statistics	Statistical information about flowlist entries	statistics type

Table 7-242 Description of Elements of statistics type in flowlistentries

Element	Description	Valid Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-243 Description of Elements in statistics type

Element	Description	Valid Value
packets	Number of packets	Counter
octets	Number of octets	Counter

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.17 Operate Flow Filter Entry (VTN)

7.17.1 List sequence numbers of flow filter entry for VTN

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/flowfilters/**ff_type**/flowfilterentries.xml

- JSON format

  /vtns/**vtn_name**/flowfilters/**ff_type**/flowfilterentries.json

Table 7-244 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilterentries>
  [
  <flowfilterentry seqnum="seqnum"/>
  ]
</flowfilterentries>
```

- JSON format

```
{
  "flowfilterentries" : [
    {
      "seqnum" : "seqnum"
    }
  ]
}
```

Table 7-245 Description of Elements in flowfilterentries

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.17.2 Create flow filter entry for VTN

Processing request

Method

POST

request URI

- XML format

  `/vtns/`**`vtn_name`**`/flowfilters/`**`ff_type`**`/flowfilterentries.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/flowfilters/`**`ff_type`**`/flowfilterentries.json`

Table 7-246 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}

Settings of request body

- XML format

```
<flowfilterentry seqnum="seqnum"
  fl_name="fl_name"
  action_type="action_type"
  nmg_name="nmg_name">
  <set priority="priority" dscp="dscp"/>
</flowfilterentry>
```

- JSON format

```
{
  "flowfilterentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "nmg_name" : "nmg_name",
    "set" : {
      "priority" : "priority",
      "dscp" : "dscp"
    }
  }
}
```

Table 7-247 Description of Elements in flowfilterentry

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{pass|-} (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Table 7-248 Description of Elements in set

Element	Description	Valid Value
priority	Packet transfer priority	Decimal number (0 to 7)
dscp	DSCP value	Decimal number (0 to 63)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of VTN **vtn_name** succeeded. "

- On failure

  "Create flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.17.3 Show flow filter entry for VTN

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-249 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentry seqnum="seqnum"
    fl_name="fl_name"
    action_type="action_type"
    nmg_name="nmg_name">
    <set priority="priority" dscp="dscp"/>
  </flowfilterentry>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentry" : {
      "seqnum" : "seqnum",
      "fl_name" : "fl_name",
      "action_type" : "action_type",
      "nmg_name" : "nmg_name",
      "set" : {
        "priority" : "priority",
        "dscp" : "dscp"
      }
    }
  }
}
```

Table 7-250 Description of Elements in flow_filter

Element	Description	Return Value
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{pass|-} (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Table 7-251 Description of Elements in set

Element	Description	Return Value
priority	Packet transfer priority	Decimal number (0 to 7)
dscp	DSCP value	Decimal number (0 to 63)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.17.4 Update flow filter entry for VTN

Processing request

Method

PUT

request URI

- XML format

  /vtns/**vtn_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-252 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)

Settings of request body

- XML format

```
<flowfilterentry fl_name="fl_name"
  action_type="action_type"
  nmg_name="nmg_name"
  op="op">
  <set priority="priority" dscp="dscp"/>
</flowfilterentry>
```

- JSON format

```
{
  "flowfilterentry" : {
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "nmg_name" : "nmg_name",
    "op" : "op",
    "set" : {
      "priority" : "priority",
      "dscp" : "dscp"
    }
  }
}
```

Table 7-253 Description of Elements in flowfilterentry

Element	Description	Valid Value
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matching Ethernet frame	{pass|-} (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
op	op status information	{add|delete}

Table 7-254 Description of Elements in set

Element	Description	Valid Value
priority	Packet transfer priority	Decimal number (0 to 7)
dscp	DSCP value	Decimal number (0 to 63)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Update(**op**) flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of VTN **vtn_name** succeeded. "

- On failure

  "Update(**op**) flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.17.5 Delete flow filter entry for VTN

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/vtn_name/flowfilters/ff_type/flowfilterentries/seqnum.xml`

- JSON format

  `/vtns/vtn_name/flowfilters/ff_type/flowfilterentries/seqnum.json`

Table 7-255 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  `"Delete sequence number seqnum for flowfilter ff_type of VTN vtn_name succeeded."`

- On failure

```
"Delete sequence number seqnum for flowfilter ff_type of VTN vtn_nam
e failed. (<Error information from pfcshell>) "
```

Remark

None

7.17.6 Show flow filter entry details for VTN

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**/detail.xml

- JSON format

  /vtns/**vtn_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**/detail.json

Table 7-256 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in|out}
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentry seqnum="seqnum"
    action_type="action_type" nmg_name="nmg_name">
    <statistics>
      <software packets="packets" octets="octets"/>
      <existingflow packets="packets" octets="octets"/>
      <expiredflow packets="packets" octets="octets"/>
      <total packets="packets" octets="octets"/>
    </statistics>
    <set priority="priority" dscp="dscp"/>
    <flowlist fl_name="fl_name" ip_version="ip_version">
```

```
      <flowlistentries>
        [
        <flowlistentry seqnum="seqnum"
          macdstaddr="macdstaddr"
          macsrcaddr="macsrcaddr"
          macethertype="macethertype"
          macvlanpriority="macvlanpriority"
          ipdstaddr="ipdstaddr"
          ipdstaddrprefix="ipdstaddrprefix"
          ipsrcaddr="ipsrcaddr"
          ipsrcaddrprefix="ipsrcaddrprefix"
          ipv6dstaddr="ipv6dstaddr"
          ipv6dstaddrprefix="ipv6dstaddrprefix"
          ipv6srcaddr="ipv6srcaddr"
          ipv6srcaddrprefix="ipv6srcaddrprefix"
          ipproto="ipproto"
          ipdscp="ipdscp"
          l4dstport="l4dstport"
          l4dstendport="l4dstendport"
          l4srcport="l4srcport"
          l4srcendport="l4srcendport"
          icmptypenum="icmptypenum"
          icmpcodenum="icmpcodenum"
          ipv6icmptypenum="ipv6icmptypenum"
          ipv6icmpcodenum="ipv6icmpcodenum">
          <statistics>
            <software packets="packets" octets="octets"/>
            <existingflow packets="packets" octets="octets"/>
            <expiredflow packets="packets" octets="octets"/>
            <total packets="packets" octets="octets"/>
          </statistics>
        </flowlistentry>
        ]
      </flowlistentries>
    </flowlist>
  </flowfilterentry>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentry" : {
      "seqnum" : "seqnum",
      "action_type" : "action_type",
      "nmg_name" : "nmg_name",
      "statistics" : {
        "software" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "existingflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "expiredflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "total" : {
          "packets" : "packets",
          "octets" : "octets"
        }
      },
      "set" : {
        "priority" : "priority",
        "dscp" : "dscp"
      },
      "flowlist" : {
        "fl_name" : "fl_name",
        "ip_version" : "ip_version",
```

```
        "flowlistentries" : [
          {
            "seqnum" : "seqnum",
            "macdstaddr" : "macdstaddr",
            "macsrcaddr" : "macsrcaddr",
            "macethertype" : "macethertype",
            "macvlanpriority" : "macvlanpriority",
            "ipdstaddr" : "ipdstaddr",
            "ipdstaddrprefix" : "ipdstaddrprefix",
            "ipsrcaddr" : "ipsrcaddr",
            "ipsrcaddrprefix" : "ipsrcaddrprefix",
            "ipv6dstaddr" : "ipv6dstaddr",
            "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
            "ipv6srcaddr" : "ipv6srcaddr",
            "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
            "ipproto" : "ipproto",
            "ipdscp" : "ipdscp",
            "l4dstport" : "l4dstport",
            "l4dstendport" : "l4dstendport",
            "l4srcport" : "l4srcport",
            "l4srcendport" : "l4srcendport",
            "icmptypenum" : "icmptypenum",
            "icmpcodenum" : "icmpcodenum",
            "ipv6icmptypenum" : "ipv6icmptypenum",
            "ipv6icmpcodenum" : "ipv6icmpcodenum",
            "statistics" : {
              "software" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "existingflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "expiredflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "total" : {
                "packets" : "packets",
                "octets" : "octets"
              }
            }
          }
        ]
      }
    }
  }
}
```

Table 7-257 Description of Elements in flowfilter

Element	Description	Valid Value
ff_type	Flow filter type	{in|out}
flowfilterentry	Flow filter entry	flowfilterentry type

Table 7-258 Description of Elements in flowfilterentry type

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)
action_type	Action for the matched ethernet frame	pass (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
statistics	Statistical information of flow filter entry	statistics type
set	Settings based on the matching flow	set type

Element	Description	Valid Value
flowlist	Flowlist information	flowlist type

Table 7-259 Description of Elements of statistics type in flowfilterentry

Element	Description	Valid Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-260 Description of Elements in set type

Element	Description	Valid Value
priority	Packet transfer priority	Decimal number (0 to 7)
dscp	DSCP value	Decimal number (0 to 63)

Table 7-261 Description of Elements in flowlist type

Element	Description	Valid Value
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
ip_version	IP VERSION	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry information	flowlistentries type

Table 7-262 Description of Elements in flowlistentries type

Element	Description	Valid Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal number (1 to 32)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddrprefix	Source IP address prefix length	Decimal number (1 to 32)
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (0 to 63)
l4dstport	TCP/UDP destination port number (the first port number in the case of range specification)	Decimal number (1 to 65535)

Element	Description	Valid Value
l4dstendport	The last TCP/UDP port number in the case of range specification	Decimal number (1 to 65535)
l4srcport	TCP/UDP source port number (the first port number in the case of range specification)	Decimal number (1 to 65535)
l4srcendport	The last TCP/UDP port number in the case of range specification	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IP address prefix length	Decimal number (0 to 128)
ipv6srcaddr	Source IP address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddrprefix	Source IP address prefix length	Decimal number (0 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal number (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal number (0 to 255)
statistics	Statistical information about flowlist entries	statistics type

Table 7-263 Description of Elements of statistics type in flowlistentries

Element	Description	Valid Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-264 Description of Elements in statistics type

Element	Description	Valid Value
packets	Number of packets	Counter
octets	Number of octets	Counter

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.18 Operate Flow Filter (vBridge)

7.18.1 List flow filters for vBridge

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters.json

Table 7-265 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilters>
  [
  <flowfilter ff_type="ff_type"/>
  ]
</flowfilters>
```

- JSON format

```
{
  "flowfilters" : [
    {
      "ff_type" : "ff_type"
    }
  ]
}
```

Table 7-266 Description of Elements in flowfilters

Element	Description	Valid Value
ff_type	Flow filter application direction	{in}

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.18.2 Create flow filter for vBridge

Processing request

Method

POST

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/flowfilters.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/flowfilters.json`

Table 7-267 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<flowfilter ff_type="ff_type"/>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type"
  }
}
```

Table 7-268 Description of Elements in flowfilter

Element	Description	Valid Value
ff_type	Flow filter application direction	{in}

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create flowfilter **ff_type** for vBridge **vbr_name** of VTN **vtn_name** succeeded. "

- On failure

  "Create flowfilter **ff_type** for vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument(ff_type)](<Error information from pfcshell>)"

Remark

None

7.18.3 Show flow filter for vBridge

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**.json

Table 7-269 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentries>
    [
    <flowfilterentry seqnum="seqnum"
      fl_name="fl_name"
      action_type="action_type"
      nmg_name="nmg_name"/>
    ]
  </flowfilterentries>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentries" : [
      {
        "seqnum" : "seqnum",
        "fl_name" : "fl_name",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name"
```

```
        }
      ]
    }
  }
```

Table 7-270 Description of Elements in flowfilter

Element	Description	Valid Value
ff_type	Flow filter application direction	{in}
flowfilterentries	List of flow filter entries	flowfilterentries type

Table 7-271 Description of Elements in flowfilterentries type

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{pass|drop|redirect|-} (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.18.4 Delete flow filter for vBridge

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/flowfilters/`**`ff_type`**`.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/flowfilters/`**`ff_type`**`.json`

Table 7-272 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in}

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete flowfilter with flowfilter type **ff_type** for vBridge **vbr_name** of VTN **vtn_name** succeeded. "

- On failure

  "Delete flowfilter with flowfilter type **ff_type** for vBridge **vbr_name** of VTN **vtn_name** failed. (<Error information from pfcshell>) "

Remark

None

7.18.5 Show flow filter details for vBridge

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/detail.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/detail.json

Table 7-273 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentries>
    [
    <flowfilterentry seqnum="seqnum"
      action_type="action_type" nmg_name="nmg_name">
      <redirectdst vnode_name="vnode_name"
        if_name="if_name"
        macdstaddr="macdstaddr"
        macsrcaddr="macsrcaddr"/>
      <statistics>
        <software packets="packets" octets="octets"/>
        <existingflow packets="packets" octets="octets"/>
        <expiredflow packets="packets" octets="octets"/>
        <total packets="packets" octets="octets"/>
      </statistics>
      <set priority="priority" dscp="dscp"/>
      <flowlist fl_name="fl_name" ip_version="ip_version">
        <flowlistentries>
          [
          <flowlistentry seqnum="seqnum"
            macdstaddr="macdstaddr"
            macsrcaddr="macsrcaddr"
            macethertype="macethertype"
            macvlanpriority="macvlanpriority"
            ipdstaddr="ipdstaddr"
            ipdstaddrprefix="ipdstaddrprefix"
            ipsrcaddr="ipsrcaddr"
            ipsrcaddrprefix="ipsrcaddrprefix"
            ipv6dstaddr="ipv6dstaddr"
            ipv6dstaddrprefix="ipv6dstaddrprefix"
            ipv6srcaddr="ipv6srcaddr"
            ipv6srcaddrprefix="ipv6srcaddrprefix"
            ipproto="ipproto"
            ipdscp="ipdscp"
            l4dstport="l4dstport"
```

```
            l4dstendport="l4dstendport"
            l4srcport="l4srcport"
            l4srcendport="l4srcendport"
            icmptypenum="icmptypenum"
            icmpcodenum="icmpcodenum"
            ipv6icmptypenum="ipv6icmptypenum"
            ipv6icmpcodenum="ipv6icmpcodenum">
            <statistics>
              <software packets="packets" octets="octets"/>
              <existingflow packets="packets" octets="octets"/>
              <expiredflow packets="packets" octets="octets"/>
              <total packets="packets" octets="octets"/>
            </statistics>
          </flowlistentry>
          ]
        </flowlistentries>
      </flowlist>
    </flowfilterentry>
    ]
  </flowfilterentries>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentries" : [
      {
        "seqnum" : "seqnum",
        "action_type" : "action_type",
        "nmg_name" : "nmg_name",
        "redirectdst" : {
          "vnode_name" : "vnode_name",
          "if_name" : "if_name",
          "macdstaddr" : "macdstaddr",
          "macsrcaddr" : "macsrcaddr"
        },
        "statistics" : {
          "software" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expiredflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "total" : {
            "packets" : "packets",
            "octets" : "octets"
          }
        },
        "set" : {
          "priority" : "priority",
          "dscp" : "dscp"
        },
        "flowlist" : {
          "fl_name" : "fl_name",
          "ip_version" : "ip_version",
          "flowlistentries" : [
            {
              "seqnum" : "seqnum",
              "macdstaddr" : "macdstaddr",
              "macsrcaddr" : "macsrcaddr",
              "macethertype" : "macethertype",
              "macvlanpriority" : "macvlanpriority",
              "ipdstaddr" : "ipdstaddr",
              "ipdstaddrprefix" : "ipdstaddrprefix",
```

```
                "ipsrcaddr" : "ipsrcaddr",
                "ipsrcaddrprefix" : "ipsrcaddrprefix",
                "ipv6dstaddr" : "ipv6dstaddr",
                "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
                "ipv6srcaddr" : "ipv6srcaddr",
                "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
                "ipproto" : "ipproto",
                "ipdscp" : "ipdscp",
                "l4dstport" : "l4dstport",
                "l4dstendport" : "l4dstendport",
                "l4srcport" : "l4srcport",
                "l4srcendport" : "l4srcendport",
                "icmptypenum" : "icmptypenum",
                "icmpcodenum" : "icmpcodenum",
                "ipv6icmptypenum" : "ipv6icmptypenum",
                "ipv6icmpcodenum" : "ipv6icmpcodenum",
                "statistics" : {
                  "software" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  },
                  "existingflow" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  },
                  "expiredflow" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  },
                  "total" : {
                    "packets" : "packets",
                    "octets" : "octets"
                  }
                }
              }
            ]
          }
        }
      ]
    }
}
```

Table 7-274 Description of Elements in flowfilter

Element	Description	Valid Value
ff_type	Flow filter type	{in}
flowfilterentries	Flow filter entry	flowfilterentries type

Table 7-275 Description of Elements in flowfilterentries type

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)
action_type	Action for the matched ethernet frame	{pass|drop|redirect|-} (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
redirectdst	Redirect destination information	redirectdst type
statistics	Statistical information of flow filter entry	statistics type
set	Settings based on the matching flow	set type
flowlist	Flowlist information	flowlist type

Table 7-276 Description of Elements in redirectdst type of flowfilterentries

Element	Description	Valid Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual interface of destination virtual node	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)

Table 7-277 Description of Elements in statistics type of flowfilterentries

Element	Description	Valid Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-278 Description of Elements in set type

Element	Description	Valid Value
priority	Packet transfer priority	Decimal number (0 to 7)
dscp	DSCP value	Decimal number (0 to 63)

Table 7-279 Description of Elements in flowlist type

Element	Description	Valid Value
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
ip_version	IP VERSION	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry information	flowlistentries type

Table 7-280 Description of Elements in flowlistentries type

Element	Description	Valid Value
seqnum	Flowlist sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macethertype	Ether type of ethernet frame	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal number (1 to 32)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddrprefix	Source IP address prefix length	Decimal number (1 to 32)

Element	Description	Valid Value
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (0 to 63)
l4dstport	TCP/UDP destination port number (the first port number in the case of range specification)	Decimal number (1 to 65535)
l4dstendport	The last TCP/UDP port number in the case of range specification	Decimal number (1 to 65535)
l4srcport	TCP/UDP source port number (the first port number in the case of range specification)	Decimal number (1 to 65535)
l4srcendport	The last TCP/UDP port number in the case of range specification	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IP address prefix length	Decimal number (0 to 128)
ipv6srcaddr	Source IP address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddrprefix	Source IP address prefix length	Decimal number (0 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal number (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal number (0 to 255)
statistics	Statistical information about flowlist entries	statistics type

Table 7-281 Description of Elements in statistics type of flowlistentries

Element	Description	Valid Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-282 Description of Elements in statistics type

Element	Description	Valid Value
packets	Number of packets	Counter
octets	Number of octets	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.19 Operate Flow Filter Entry (vBridge)

7.19.1 List sequence numbers of flow filter entry for vBridge

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/flowfilters/`**`ff_type`**`/flowfilterentries.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/flowfilters/`**`ff_type`**`/flowfilterentries.json`

Table 7-283 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in}

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilterentries>
  [
  <flowfilterentry seqnum="seqnum"/>
  ]
</flowfilterentries>
```

- JSON format

```
{
  "flowfilterentries" : [
    {
      "seqnum" : "seqnum"
    }
  ]
}
```

Table 7-284 Description of Elements in flowfilterentries

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.19.2 Create flow filter entry for vBridge

Processing request

Method

POST

request URI

- XML format

/vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/flowfilterentries.xml

- JSON format

/vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/flowfilterentries.json

Table 7-285 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in}

Settings of request body

- XML format

```
<flowfilterentry seqnum="seqnum"
  fl_name="fl_name"
  action_type="action_type"
  nmg_name="nmg_name">
  <set priority="priority" dscp="dscp"/>
  <redirectdst vnode_name="vnode_name"
    if_name="if_name"
    macdstaddr="macdstaddr"
    macsrcaddr="macsrcaddr"/>
</flowfilterentry>
```

- JSON format

```
{
  "flowfilterentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "nmg_name" : "nmg_name",
    "set" : {
      "priority" : "priority",
      "dscp" : "dscp"
    },
    "redirectdst" : {
      "vnode_name" : "vnode_name",
      "if_name" : "if_name",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr"
    }
  }
}
```

Table 7-286 Description of Elements in flowfilterentry

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores

Element	Description	Valid Value
action_type	Action for the matched ethernet frame	{pass|drop|redirect} (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)

Table 7-287 Description of Elements in set type

Element	Description	Valid Value
priority	Packet transfer priority	Decimal number (0 to 7)
dscp	DSCP value	Decimal number (0 to 63)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of vBridge **vbr_name** of VTN **vtn_name** succeeded. "

- On failure

  "Create flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of vBridge **vbr_name** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.19.3 Show flow filter entry for vBridge

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-288 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in}
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentry seqnum="seqnum"
    fl_name="fl_name"
    action_type="action_type"
    nmg_name="nmg_name">
    <set priority="priority" dscp="dscp"/>
  </flowfilterentry>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentry" : {
      "seqnum" : "seqnum",
```

```
      "fl_name" : "fl_name",
      "action_type" : "action_type",
      "nmg_name" : "nmg_name",
      "set" : {
        "priority" : "priority",
        "dscp" : "dscp"
      }
    }
  }
}
```

Table 7-289 Description of Elements in flowfilter

Element	Description	Return Value
ff_type	Flow filter application direction	{in}
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{pass|-} (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores

Table 7-290 Description of Elements in set

Element	Description	Return Value
priority	Packet transfer priority	Decimal number (0 to 7)
dscp	DSCP value	Decimal number (0 to 63)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.19.4 Update flow filter entry for vBridge

Processing request

Method

PUT

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-291 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in}
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)

Settings of request body

- XML format

```
<flowfilterentry fl_name="fl_name"
  action_type="action_type"
  nmg_name="nmg_name"
  op="op">
  <set priority="priority" dscp="dscp"/>
  <redirectdst vnode_name="vnode_name"
    if_name="if_name"
    macdstaddr="macdstaddr"
    macsrcaddr="macsrcaddr"/>
</flowfilterentry>
```

- JSON format

```
{
  "flowfilterentry" : {
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "nmg_name" : "nmg_name",
    "op" : "op",
    "set" : {
      "priority" : "priority",
      "dscp" : "dscp"
    },
    "redirectdst" : {
      "vnode_name" : "vnode_name",
      "if_name" : "if_name",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr"
    }
  }
}
```

Table 7-292 Description of Elements in flowfilterentry

Element	Description	Valid Value
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
action_type	Action for the matched ethernet frame	{pass|drop|redirect} (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
op	op state information	{add|delete}

Table 7-293 Description of Elements in set

Element	Description	Valid Value
priority	Packet transfer priority	Decimal number (0 to 7)
dscp	DSCP value	Decimal number (0 to 63)

Table 7-294 Description of Elements in redirectdst type of flowfilterentries

Element	Description	Valid Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual interface of destination virtual node	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- On success

  "Update(**op**) flowfilterentry with sequence number **seqnum** for flowfilter **ff_type** of vBridge **vbr_name** of VTN **vtn_name** succeeded. "
- On failure

```
"Update(op) flowfilterentry with sequence number seqnum for flowfilt
er ff_type of vBridge vbr_name of VTN vtn_name failed. [Invalid argu
ment](<Error information from pfcshell>)"
```

Remark

None

7.19.5 Delete flow filter entry for vBridge

Processing request

Method

DELETE

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**.json

Table 7-295 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in}
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

400,500

Miscellaneous

Log

- On success

  "Delete sequence number **seqnum** for flowfilter **ff_type** of vBridge **vbr_name** of VTN **vtn_name** succeeded. "

- On failure

  "Delete sequence number **seqnum** for flowfilter **ff_type** of vBridge **vbr_name** of VTN **vtn_name** failed. (<Error information from pfcshell>) "

Remark

None

7.19.6 Show flow filter entry details for vBridge

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**/detail.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/flowfilters/**ff_type**/flowfilterentries/**seqnum**/detail.json

Table 7-296 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
ff_type	Flow filter application direction	{in}
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfilter ff_type="ff_type">
  <flowfilterentry seqnum="seqnum"
    action_type="action_type" nmg_name="nmg_name">
    <redirectdst vnode_name="vnode_name"
      if_name="if_name"
      macdstaddr="macdstaddr"
      macsrcaddr="macsrcaddr"/>
    <statistics>
      <software packets="packets" octets="octets"/>
      <existingflow packets="packets" octets="octets"/>
      <expiredflow packets="packets" octets="octets"/>
      <total packets="packets" octets="octets"/>
    </statistics>
    <set priority="priority" dscp="dscp"/>
    <flowlist fl_name="fl_name" ip_version="ip_version">
      <flowlistentries>
        [
        <flowlistentry seqnum="seqnum"
          macdstaddr="macdstaddr"
          macsrcaddr="macsrcaddr"
          macethertype="macethertype"
          macvlanpriority="macvlanpriority"
          ipdstaddr="ipdstaddr"
          ipdstaddrprefix="ipdstaddrprefix"
          ipsrcaddr="ipsrcaddr"
          ipsrcaddrprefix="ipsrcaddrprefix"
          ipv6dstaddr="ipv6dstaddr"
          ipv6dstaddrprefix="ipv6dstaddrprefix"
          ipv6srcaddr="ipv6srcaddr"
          ipv6srcaddrprefix="ipv6srcaddrprefix"
          ipproto="ipproto"
          ipdscp="ipdscp"
          l4dstport="l4dstport"
          l4dstendport="l4dstendport"
          l4srcport="l4srcport"
          l4srcendport="l4srcendport"
          icmptypenum="icmptypenum"
          icmpcodenum="icmpcodenum"
          ipv6icmptypenum="ipv6icmptypenum"
          ipv6icmpcodenum="ipv6icmpcodenum">
          <statistics>
            <software packets="packets" octets="octets"/>
            <existingflow packets="packets" octets="octets"/>
            <expiredflow packets="packets" octets="octets"/>
            <total packets="packets" octets="octets"/>
          </statistics>
        </flowlistentry>
        ]
      </flowlistentries>
    </flowlist>
  </flowfilterentry>
</flowfilter>
```

- JSON format

```
{
  "flowfilter" : {
    "ff_type" : "ff_type",
    "flowfilterentry" : {
      "seqnum" : "seqnum",
      "action_type" : "action_type",
      "nmg_name" : "nmg_name",
      "redirectdst" : {
```

```
        "vnode_name" : "vnode_name",
        "if_name" : "if_name",
        "macdstaddr" : "macdstaddr",
        "macsrcaddr" : "macsrcaddr"
      },
      "statistics" : {
        "software" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "existingflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "expiredflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "total" : {
          "packets" : "packets",
          "octets" : "octets"
        }
      },
      "set" : {
        "priority" : "priority",
        "dscp" : "dscp"
      },
      "flowlist" : {
        "fl_name" : "fl_name",
        "ip_version" : "ip_version",
        "flowlistentries" : [
          {
            "seqnum" : "seqnum",
            "macdstaddr" : "macdstaddr",
            "macsrcaddr" : "macsrcaddr",
            "macethertype" : "macethertype",
            "macvlanpriority" : "macvlanpriority",
            "ipdstaddr" : "ipdstaddr",
            "ipdstaddrprefix" : "ipdstaddrprefix",
            "ipsrcaddr" : "ipsrcaddr",
            "ipsrcaddrprefix" : "ipsrcaddrprefix",
            "ipv6dstaddr" : "ipv6dstaddr",
            "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
            "ipv6srcaddr" : "ipv6srcaddr",
            "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
            "ipproto" : "ipproto",
            "ipdscp" : "ipdscp",
            "l4dstport" : "l4dstport",
            "l4dstendport" : "l4dstendport",
            "l4srcport" : "l4srcport",
            "l4srcendport" : "l4srcendport",
            "icmptypenum" : "icmptypenum",
            "icmpcodenum" : "icmpcodenum",
            "ipv6icmptypenum" : "ipv6icmptypenum",
            "ipv6icmpcodenum" : "ipv6icmpcodenum",
            "statistics" : {
              "software" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "existingflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "expiredflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "total" : {
                "packets" : "packets",
                "octets" : "octets"
              }
            }
```

```
                }
            ]
        }
    }
  }
}
```

Table 7-297 Description of Elements in flowfilter

Element	Description	Valid Value
ff_type	Flow filter type	{in}
flowfilterentry	Flow filter entry	flowfilterentry type

Table 7-298 Description of Elements in flowfilterentry type

Element	Description	Valid Value
seqnum	Sequence number of flow filter entry	Decimal number (1 to 65535)
action_type	Action for the matched ethernet frame	{pass|drop|redirect|- } (Lowercase)
nmg_name	Network monitor group name	Up to 31 characters including one-byte alphanumeric characters and underscores
redirectdst	Redirect destination information	redirectdst type
statistics	Statistical information of flow filter entry	statistics type
set	Settings based on the matching flow	set type
flowlist	Flowlist information	flowlist type

Table 7-299 Description of Elements in redirectdst type of flowfilterentry

Element	Description	Valid Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual interface of destination virtual node	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)

Table 7-300 Description of Elements in statistics type of flowfilterentry

Element	Description	Valid Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-301 Description of Elements in set type

Element	Description	Valid Value
priority	Packet transfer priority	Decimal number (0 to 7)
dscp	DSCP value	Decimal number (0 to 63)

Table 7-302 Description of Elements in flowlist type

Element	Description	Valid Value
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
ip_version	IP VERSION	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry information	flowlistentries type

Table 7-303 Description of Elements in flowlistentries type

Element	Description	Valid Value
seqnum	Flowlist entry sequence number	Decimal number (1 to 65535)
macdstaddr	Destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macsrcaddr	Source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal number)
macethertype	Ethernet frame type	Hexadecimal format including "0x" (0x0000 to 0xffff)
macvlanpriority	VLAN tag priority	Decimal number (0 to 7)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Prefix length of destination IP address	Decimal number (1 to 32)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddrprefix	Prefix length of source IP address	Decimal number (1 to 32)
ipproto	IP protocol number	Decimal number (1 to 255)
ipdscp	DSCP value	Decimal number (0 to 63)
l4dstport	TCP/UDP destination port number (the first port number in the case of range specification)	Decimal number (1 to 65535)
l4dstendport	The last TCP/UDP port number in the case of range specification	Decimal number (1 to 65535)
l4srcport	TCP/UDP source port number (the first port number in the case of range specification)	Decimal number (1 to 65535)
l4srcendport	The last TCP/UDP port number in the case of range specification	Decimal number (1 to 65535)
icmptypenum	ICMP type	Decimal number (0 to 255)
icmpcodenum	ICMP code	Decimal number (0 to 255)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Prefix length of destination IP address	Decimal number (0 to 128)
ipv6srcaddr	Source IP address	IPv6 address The IPv6 format conforms to RFC 5952.

Element	Description	Valid Value
ipv6srcaddrprefix	Prefix length of source IP address	Decimal number (0 to 128)
ipv6icmptypenum	ICMPv6 type	Decimal number (0 to 255)
ipv6icmpcodenum	ICMPv6 code	Decimal number (0 to 255)
statistics	Statistical information about flowlist entries	statistics type

Table 7-304 Description of Elements of statistics type in flowlistentries

Element	Description	Valid Value
software	Of the flows that match the entry, frames that passed through the VTN	statistics type
existingflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that is currently set to the OFS	statistics type
expiredflow	Of the flows that match the entry, frames that hard-transferred by the flow entry that was set to the OFS	statistics type
total	Total number of the flows that match the entry	statistics type

Table 7-305 Description of Elements in statistics type

Element	Description	Valid Value
packets	Number of packets	Counter
octets	Number of octets	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.20 Operate safe flow filter (vExternal interface)

7.20.1 Acquire a safe flow filter (vExternal interface)

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters.json

Table 7-306 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<safeflowfilters>
  [
  <safeflowfilter priority="priority"/>
  ]
</safeflowfilters>
```

- JSON format

```
{
  "safeflowfilters" : [
    {
      "priority" : "priority"
    }
  ]
}
```

Table 7-307 Description of Elements in safeflowfilters

Element	Description	Return Value
priority	Safe flow filter priority	Decimal integer (1, 2, or 3)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.20.2 Create a safe flow filter (vExternal interface)

Processing request

Method

POST

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/interfaces/`**`if_name`**`/safeflowfilters.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/interfaces/`**`if_name`**`/safeflowfilters.json`

Table 7-308 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<safeflowfilter priority="priority">
  <restrict
    srcmac="srcmac"
    dstmac="dstmac"
    srcip="srcip"
    dstip="dstip"
    ipproto="ipproto"
    srcport="srcport"
    dstport="dstport"
    allany="allany"/>
</safeflowfilter>
```

- JSON format

```
{
  "safeflowfilter" : {
    "priority" : "priority",
    "restrict" : {
      "srcmac" : "srcmac",
      "dstmac" : "dstmac",
      "srcip" : "srcip",
      "dstip" : "dstip",
      "ipproto" : "ipproto",
      "srcport" : "srcport",
      "dstport" : "dstport",
      "allany" : "allany"
    }
  }
}
```

Table 7-309 Description of Elements in safeflowfilter

Element	Description	Valid Value
priority	Safe flow filter priority	Decimal integer (1, 2, or 3)
srcmac	Flag indicating that the source IP address is set as a restriction.	{true} *Always true
dstmac	Flag indicating that the destination IP address is set as a restriction.	{true} *Always true
srcip	Flag indicating that the source IP address is set as a restriction.	{true} *Always true
dstip	Flag indicating that the destination IP address is set as a restriction.	{true} *Always true
ipproto	Flag indicating that the IP protocol is set as a restriction.	{true} *Always true
srcport	Flag indicating that the source TCP/UDP port number is set as a restriction.	{true} *Always true
dstport	Flag indicating that the destination TCP/UDP port number is set as a restriction.	{true} *Always true
allany	Flag indicating that **allany** is a match condition for a safe flow filter entry	{true} *Always true

Remember

priority must be specified. The larger the value is, the higher the priority is.

Remember

allany or at least one of **srcmac**, **dstmac**, **srcip**, **dstip**, **ipproto**, **srcport**, and **dstport** must be specified.

Remember

allany and any of **srcmac**, **dstmac**, **srcip**, **dstip**, **ipproto**, **srcport**, and **dstport** cannot be specified together.

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create priority **priority** for safeflowfilter for interface **if_name** for vExternal **vex_name** of VTN **vtn_name** succeeded."

- On failure

  "Create priority **priority** for safeflowfilter for interface **if_name** for vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.20.3 Acquire the information about the safe flow filter (vExternal interface)

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**.json

Table 7-310 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
priority	Safe flow filter priority	Decimal integer (1 and 2)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<safeflowfilter priority="priority">
  <safeflowfilterentries>
    [
    <safeflowfilterentry entryid="entryid"
      allany="allany"
      fl_name="fl_name"
      action_type="action_type"/>
    ]
  </safeflowfilterentries>
</safeflowfilter>
```

- JSON format

```
{
  "safeflowfilter" : {
    "priority" : "priority",
    "safeflowfilterentries" : [
      {
        "entryid" : "entryid",
        "allany" : "allany",
        "fl_name" : "fl_name",
        "action_type" : "action_type"
      }
    ]
  }
}
```

Table 7-311 Description of Elements in safeflowfilter

Element	Description	Return Value
priority	Safe flow filter priority	Decimal integer (1 and 2)
safeflowfilterentries	Safe flow filter entry	safeflowfilterentries type (For details, refer to "Table 7-312 Description of Elements in safeflowfilterentries (page 479)".)

Table 7-312 Description of Elements in safeflowfilterentries

Element	Description	Return Value
entryid	Safe flow filter entry ID	Decimal integer (1 to 3000)
allany	Flag indicating that all flows are a match target.	{true} *Always true
fl_name	Flowlist name to be used as a match condition	Up to 32 characters including one-byte alphanumeric characters and underscores.
action_type	Action for the matched ethernet frame	{pass|drop|redirect} (Lowercase)

Remember

Either **allany** or **fl_name** is returned.

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get safeflowfilter priority info with redirectdst **redirectdst** faile d. [Invalid argument(redirectdst)](<Error information from pfcshell>)"

Remark

None

7.20.4 Delete a safe flow filter (vExternal interface)

Processing request

Method

DELETE

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilter s/**priority**.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilter s/**priority**.json

Table 7-313 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
priority	Safe flow filter priority	Decimal integer (1 and 2)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete priority **priority** for safeflowfilter for interface **if_name** for vExternal **vex_name** of VTN **vtn_name** succeeded."

- On failure

  "Delete priority **priority** for safeflowfilter for interface **if_name** for vExternal **vex_name** of VTN **vtn_name** failed.(<Error information from pfcshell>)"

Remark

None

7.20.5 Acquire detailed information of safe flow filter (vExternal interface)

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**/detail.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**/detail.json

Table 7-314 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
priority	Safe flow filter priority	Decimal integer (1 and 2)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<safeflowfilter priority="priority">
  <restrict
    srcmac="srcmac"
    dstmac="dstmac"
    srcip="srcip"
    dstip="dstip"
    ipproto="ipproto"
    srcport="srcport"
    dstport="dstport"
    allany="allany"/>
  <safeflowfilterentries>
    [
    <safeflowfilterentry
      entryid="entryid"
      allany="allany"
      fl_name="fl_name"
```

```
      status="status"
      action_type="action_type">
      <redirectdst
        vnode_name="vnode_name"
        if_name="if_name"
        macdstaddr="macdstaddr"
        macsrcaddr="macsrcaddr"/>
      <statistics>
        <software packets="packets" octets="octets"/>
        <existingflow packets="packets" octets="octets"/>
        <expiredflow packets="packets" octets="octets"/>
        <total packets="packets" octets="octets"/>
      </statistics>
      <flowlist fl_name="fl_name" ip_version="ip_version">
        <flowlistentries>
          [
          <flowlistentry
            seqnum="seqnum"
            macdstaddr="macdstaddr"
            macsrcaddr="macsrcaddr"
            ipdstaddr="ipdstaddr"
            ipdstaddrprefix="ipdstaddrprefix"
            ipsrcaddr="ipsrcaddr"
            ipsrcaddrprefix="ipsrcaddrprefix"
            ipv6dstaddr="ipv6dstaddr"
            ipv6dstaddrprefix="ipv6dstaddrprefix"
            ipv6srcaddr="ipv6srcaddr"
            ipv6srcaddrprefix="ipv6srcaddrprefix"
            ipproto="ipproto"
            l4dstport="l4dstport"
            l4dstendport="l4dstendport"
            l4srcport="l4srcport"
            l4srcendport="l4srcendport">
            <statistics>
              <software packets="packets" octets="octets"/>
              <existingflow packets="packets" octets="octets"/>
              <expiredflow packets="packets" octets="octets"/>
              <total packets="packets" octets="octets"/>
            </statistics>
          </flowlistentry>
          ]
        </flowlistentries>
      </flowlist>
    </safeflowfilterentry>
    ]
  </safeflowfilterentries>
</safeflowfilter>
```

- JSON format

```
{
  "safeflowfilter" : {
    "priority" : "priority",
    "restrict" : {
      "srcmac" : "srcmac",
      "dstmac" : "dstmac",
      "srcip" : "srcip",
      "dstip" : "dstip",
      "ipproto" : "ipproto",
      "srcport" : "srcport",
      "dstport" : "dstport",
      "allany" : "allany"
    },
    "safeflowfilterentries" : [
      {
        "entryid" : "entryid",
        "allany" : "allany",
        "fl_name" : "fl_name",
        "status" : "status",
        "action_type" : "action_type",
        "redirectdst" : {
          "vnode_name" : "vnode_name",
```

```
          "if_name" : "if_name",
          "macdstaddr" : "macdstaddr",
          "macsrcaddr" : "macsrcaddr"
        },
        "statistics" : {
          "software" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "existingflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "expiredflow" : {
            "packets" : "packets",
            "octets" : "octets"
          },
          "total" : {
            "packets" : "packets",
            "octets" : "octets"
          }
        },
        "flowlist" : {
          "fl_name" : "fl_name",
          "ip_version" : "ip_version",
          "flowlistentries" : [
            {
              "seqnum" : "seqnum",
              "macdstaddr" : "macdstaddr",
              "macsrcaddr" : "macsrcaddr",
              "ipdstaddr" : "ipdstaddr",
              "ipdstaddrprefix" : "ipdstaddrprefix",
              "ipsrcaddr" : "ipsrcaddr",
              "ipsrcaddrprefix" : "ipsrcaddrprefix",
              "ipv6dstaddr" : "ipv6dstaddr",
              "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
              "ipv6srcaddr" : "ipv6srcaddr",
              "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
              "ipproto" : "ipproto",
              "l4dstport" : "l4dstport",
              "l4dstendport" : "l4dstendport",
              "l4srcport" : "l4srcport",
              "l4srcendport" : "l4srcendport",
              "statistics" : {
                "software" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "existingflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "expiredflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "total" : {
                  "packets" : "packets",
                  "octets" : "octets"
                }
              }
            }
          ]
        }
      }
    ]
  }
}
```

Table 7-315 Description of Elements in safeflowfilter

Element	Description	Return Value
priority	Safe flow filter priority	Decimal integer (1, 2, or 3)
srcmac	Flag indicating that the source IP address is set as a restriction.	{true} *Always true
dstmac	Flag indicating that the destination IP address is set as a restriction.	{true} *Always true
srcip	Flag indicating that the source IP address is set as a restriction.	{true} *Always true
dstip	Flag indicating that the destination IP address is set as a restriction.	{true} *Always true
ipproto	Flag indicating that the IP protocol is set as a restriction.	{true} *Always true
srcport	Flag indicating that the source TCP/UDP port number is set as a restriction.	{true} *Always true
dstport	Flag indicating that the destination TCP/UDP port number is set as a restriction.	{true} *Always true
allany	Flag indicating that allany is a match condition for a safe flow filter entry	{true} *Always true
safeflowfilterentries	Safe flow filter entry	safeflowfilterentries type (For details, refer to "Table 7-316 Description of Elements in safeflowfilterentries (page 484)".)

Table 7-316 Description of Elements in safeflowfilterentries

Element	Description	Return Value
entryid	Safe flow filter entry ID	Decimal integer (1 to 3000)
allany	Flag indicating that all flows are the match target	{true} *Always true
fl_name	Flowlist name to be used as a match condition	Up to 32 characters including one-byte alphanumeric characters and underscores.
status	Status of the safe flow filter entry	{active|down(details)}
action_type	Action for the matched ethernet frame	{pass|drop|redirect} (Lowercase)
redirectdst	Destination information when **action_type** is **redirect**	redirectdst type (For details, refer to "Table 7-317 Description of Elements in redirectdst (page 485)".)
statistics	Safe flow filter entry statistical information	statistics type (For details, refer to "Table 7-320 Description of Elements in statistics (page 486)".)
flowlist	Flowlist	flowlist type (For details, refer to "Table 7-318 Description of Elements in flowlist (page 485)".)

Remember

Either **allany** or **fl_name** is returned.

Table 7-317 Description of Elements in redirectdst

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual interface of virtual node of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal)

Table 7-318 Description of Elements in flowlist

Element	Description	Return Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry list	flowlistentries type (For details, refer to "Table 7-319 Description of Elements in flowlistentries (page 485)".)

Table 7-319 Description of Elements in flowlistentries

Element	Description	Return Value
seqnum	Flowlist entry sequence number	Decimal integer (1 to 65535)
macdstaddr	Destinaltion IP address	hhhh.hhhh.hhhh format (h: Hexadecimal)
macsrcaddr	Source IP address	hhhh.hhhh.hhhh format (h: Hexadecimal)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal integer (1 to 32)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddrprefix	Source IP address prefix length	Decimal integer (1 to 32)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer (1 to 128)
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer (1 to 128)
ipproto	IP protocol number	Decimal integer (1 to 255)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal integer (1 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer (1 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal integer (1 to 65535)

Element	Description	Return Value
l4srcendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer (1 to 65535)
statistics	Flowfilter entry statistical information	statistics type (For details, refer to "Table 7-320 Description of Elements in statistics (page 486)".)

Table 7-320 Description of Elements in statistics

Element	Description	Return Value
software	Number of packets that passed through VTN among the flows that matched the entry	Counter
existingflow	Number of packets that were hard-transferred by the flow entry currently set to OFS among the flows that matched the entry	Counter
expiredflow	Number of packets that were hard-transferred by the flow entry previously set to OFS in among the flows that matched the entry	Counter
total	Total number of packets of the flow that matched the entry	Counter

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.21 Operate safe flow filter entry (vExternal interface)

7.21.1 Acquire entry IDs of safe flow filter entry (vExternal interface)

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**/flowfilterentries.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**/flowfilterentries.json

Table 7-321 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
priority	Safe flow filter priority	Decimal integer (1, 2, or 3)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<safeflowfilterentries>
  [
  <safeflowfilterentry entryid="entryid"/>
  ]
</safeflowfilterentries>
```

- JSON format

```
{
  "safeflowfilterentries" : [
    {
      "entryid" : "entryid"
    }
  ]
}
```

Table 7-322 Description of Elements in safeflowfilterentries

Element	Description	Return Value
entryid	Safe flow filter entry ID	Decimal integer (1 to 3000)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

7.21.2 Create safe flow filter entry (vExternal interface)

Processing request

Method

POST

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**/flowfilterentries.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**/flowfilterentries.json

Table 7-323 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
priority	Safe flow filter priority	Decimal integer (1, 2, or 3)

Settings of request body

- XML format

```
<safeflowfilterentry
  entryid="entryid"
  allany="allany"
  fl_name="fl_name"
  action_type="action_type">
  <redirectdst
    vnode_name="vnode_name"
    if_name="if_name"
    macdstaddr="macdstaddr"
```

```
    macsrcaddr="macsrcaddr"/>
</safeflowfilterentry>
```

- JSON format

```
{
  "safeflowfilterentry" : {
    "entryid" : "entryid",
    "allany" : "allany",
    "fl_name" : "fl_name",
    "action_type" : "action_type",
    "redirectdst" : {
      "vnode_name" : "vnode_name",
      "if_name" : "if_name",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr"
    }
  }
}
```

Table 7-324 Description of Elements in safeflowfilterentry

Element	Description	Valid Value
entryid	Safe flow filter entry ID	Decimal integer (1 to 3000)
allany	Flag indicating that all flows are the match target	{true} *Always true
fl_name	Flowlist name to be used as a match condition	Up to 32 characters including one-byte alphanumeric characters and underscores.
action_type	Action for the matched ethernet frame	{pass|drop|redirect} (Lowercase)
redirectdst	Destination information when action_type is redirect	redirectdst type (For details, refer to "Table 7-325 Description of Elements in redirectdst (page 490)".)

Remember

You must specify either **allany** or **fl_name**.

Table 7-325 Description of Elements in redirectdst

Element	Description	Valid Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual interface of virtual node of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal)

Processing result

Details of response body

None

HTTP status code for response

- On success

200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create safeflowfilterentry with entryid **entryid** of priority **priority** for interface **if_name** for vExternal **vex_name** of VTN **vtn_name** succeeded."

- On failure

  "Create safeflowfilterentry with entryid **entryid** of priority **priority** for interface **if_name** for vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.21.3 Acquire information of safe flow filter entry (vExternal interface)

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**/flowfilterentries/**entryid**.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**/flowfilterentries/**entryid**.json

Table 7-326 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
priority	Safe flow filter priority	Decimal integer (1, 2, or 3)

Element	Description	Valid Value
entryid	Safe flow filter entry ID	Decimal integer (1 to 3000)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<safeflowfilter priority="priority">
  <safeflowfilterentry
    entryid="entryid"
    allany="allany"
    fl_name="fl_name"
    action_type="action_type">
    <redirectdst
      vnode_name="vnode_name"
      if_name="if_name"
      macdstaddr="macdstaddr"
      macsrcaddr="macsrcaddr"/>
  </safeflowfilterentry>
</safeflowfilter>
```

- JSON format

```
{
  "safeflowfilter" : {
    "priority" : "priority",
    "safeflowfilterentry" : {
      "entryid" : "entryid",
      "allany" : "allany",
      "fl_name" : "fl_name",
      "action_type" : "action_type",
      "redirectdst" : {
        "vnode_name" : "vnode_name",
        "if_name" : "if_name",
        "macdstaddr" : "macdstaddr",
        "macsrcaddr" : "macsrcaddr"
      }
    }
  }
}
```

Table 7-327 Description of Elements in safeflowfilter

Element	Description	Return Value
priority	Safe flow filter priority	Decimal integer (1 and 2)
safeflowfilterentry	Safe flow filter entry	safeflowfilterentry type (For details, refer to "Table 7-328 Description of Elements in safeflowfilterentry (page 492)".)

Table 7-328 Description of Elements in safeflowfilterentry

Element	Description	Return Value
entryid	Safe flow filter entry ID	Decimal integer (1 to 3000)
allany	Flag indicating that all flows are the match target	{true} *Always true

Element	Description	Return Value
fl_name	Flowlist name to be used as a match condition	Up to 32 characters including one-byte alphanumeric characters and underscores.
action_type	Action for the matched ethernet frame	{pass|drop|redirect} (Lowercase)
redirectdst	Destination information when action_type is redirect	redirectdst type (For details, refer to "Table 7-329 Description of Elements in redirectdst (page 492)".)

Remember

Either allany or fl_name is returned.

Table 7-329 Description of Elements in redirectdst

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual interface of virtual node of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

```
"Get safeflowfilterentry with entryid entryid of priority priority f
or interface if_name for vExternal vex_name of VTN vtn_name failed.
[Invalid argument(entryid)](<Error information from pfcshell>)"
```

Remark

None

7.21.4 Refresh safe flow filter entry (vExternal interface)

Processing request

Method

PUT

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**/flowfilterentries/**entryid**.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/interfaces/**if_name**/safeflowfilters/**priority**/flowfilterentries/**entryid**.json

Table 7-330 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
priority	Safe flow filter priority	Decimal integer (1, 2, or 3)
entryid	Safe flow filter entry ID	Decimal integer (1 to 3000)

Settings of request body

- XML format

```
<safeflowfilterentry
  op="op"
  action_type="action_type">
  <redirectdst
    vnode_name="vnode_name"
    if_name="if_name"
    macdstaddr="macdstaddr"
    macsrcaddr="macsrcaddr"/>
</safeflowfilterentry>
```

- JSON format

```
{
  "safeflowfilterentry" : {
    "op" : "op",
    "action_type" : "action_type",
    "redirectdst" : {
      "vnode_name" : "vnode_name",
      "if_name" : "if_name",
      "macdstaddr" : "macdstaddr",
      "macsrcaddr" : "macsrcaddr"
    }
  }
}
```

Table 7-331 Description of Elements in safeflowfilterentry

Element	Description	Valid Value
op	Operation type	{add|delete} (Lowercase)
action_type	Action for the matched ethernet frame	{pass|drop|redirect} (Lowercase)
redirectdst	Destination information when action_type is redirect	redirectdst type (For details, refer to "Table 7-332 Description of Elements in redirectdst (page 494)".)

Table 7-332 Description of Elements in redirectdst

Element	Description	Valid Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual interface of virtual node of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Update safeflowfilterentry with entryid **entryid** of priority **priority** for interface **if_name** for vExternal **vex_name** of VTN **vtn_name** succeeded."

- On failure

  "Update(**safeop**) safeflowfilterentry with entryid **entryid** of priority **priority** for interface **if_name** for vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

7.21.5 Delete safe flow filter entry (vExternal interface)

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/interfaces/`**`if_name`**`/safeflowfilters/`**`priority`**`/flowfilterentries/`**`entryid`**`.xml`

- JSON format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/interfaces/`**`if_name`**`/safeflowfilters/`**`priority`**`/flowfilterentries/`**`entryid`**`.json`

Table 7-333 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
priority	Safe flow filter priority	Decimal integer (1, 2, or 3)
entryid	Safe flow filter entry ID	Decimal integer (1 to 3000)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

```
"Delete safeflowfilterentry with entryid entryid of priority priorit
y for interface if_name for vExternal vex_name of VTN vtn_name succe
eded."
```

- On failure

```
"Delete safeflowfilterentry with entryid entryid of priority priorit
y for interface if_name for vExternal vex_name of VTN vtn_name faile
d. [Invalid argument(entryid)](<Error information from pfcshell>)"
```

Remark

None

7.21.6 Delete detailed information of safe flow filter entry (vExternal interface)

Processing request

Method

GET

request URI

- XML format

```
/vtns/vtn_name/vexternals/vex_name/interfaces/if_name/safeflowfilter
s/priority/flowfilterentries/entryid/detail.xml
```

- JSON format

```
/vtns/vtn_name/vexternals/vex_name/interfaces/if_name/safeflowfilter
s/priority/flowfilterentries/entryid/detail.json
```

Table 7-334 Description of Parameters in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Interface name	Up to 31 characters including one-byte alphanumeric characters and underscores
priority	Safe flow filter priority	Decimal integer (1, 2, or 3)
entryid	Safe flow filter entry ID	Decimal integer (1 to 3000)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<safeflowfilter priority="priority">
  <restrict
    srcmac="srcmac"
    dstmac="dstmac"
    srcip="srcip"
    dstip="dstip"
    ipproto="ipproto"
    srcport="srcport"
    dstport="dstport"
    allany="allany"/>
  <safeflowfilterentry
    entryid="entryid"
    allany="allany"
    fl_name="fl_name"
    status="status"
    action_type="action_type">
    <redirectdst
      vnode_name="vnode_name"
      if_name="if_name"
      macdstaddr="macdstaddr"
      macsrcaddr="macsrcaddr"/>
    <statistics>
      <software packets="packets" octets="octets"/>
      <existingflow packets="packets" octets="octets"/>
      <expiredflow packets="packets" octets="octets"/>
      <total packets="packets" octets="octets"/>
    </statistics>
    <flowlist fl_name="fl_name" ip_version="ip_version">
      <flowlistentries>
        [
        <flowlistentry
          seqnum="seqnum"
          macdstaddr="macdstaddr"
          macsrcaddr="macsrcaddr"
          ipdstaddr="ipdstaddr"
          ipdstaddrprefix="ipdstaddrprefix"
          ipsrcaddr="ipsrcaddr"
          ipsrcaddrprefix="ipsrcaddrprefix"
          ipv6dstaddr="ipv6dstaddr"
          ipv6dstaddrprefix="ipv6dstaddrprefix"
          ipv6srcaddr="ipv6srcaddr"
          ipv6srcaddrprefix="ipv6srcaddrprefix"
          ipproto="ipproto"
          l4dstport="l4dstport"
          l4dstendport="l4dstendport"
          l4srcport="l4srcport"
          l4srcendport="l4srcendport">
          <statistics>
            <software packets="packets" octets="octets"/>
            <existingflow packets="packets" octets="octets"/>
            <expiredflow packets="packets" octets="octets"/>
            <total packets="packets" octets="octets"/>
          </statistics>
        </flowlistentry>
        ]
      </flowlistentries>
    </flowlist>
  </safeflowfilterentry>
</safeflowfilter>
```

- JSON format

```
{
  "safeflowfilter" : {
    "priority" : "priority",
    "restrict" : {
      "srcmac" : "srcmac",
      "dstmac" : "dstmac",
      "srcip" : "srcip",
      "dstip" : "dstip",
      "ipproto" : "ipproto",
      "srcport" : "srcport",
      "dstport" : "dstport",
      "allany" : "allany"
    },
    "safeflowfilterentry" : {
      "entryid" : "entryid",
      "allany" : "allany",
      "fl_name" : "fl_name",
      "status" : "status",
      "action_type" : "action_type",
      "redirectdst" : {
        "vnode_name" : "vnode_name",
        "if_name" : "if_name",
        "macdstaddr" : "macdstaddr",
        "macsrcaddr" : "macsrcaddr"
      },
      "statistics" : {
        "software" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "existingflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "expiredflow" : {
          "packets" : "packets",
          "octets" : "octets"
        },
        "total" : {
          "packets" : "packets",
          "octets" : "octets"
        }
      },
      "flowlist" : {
        "fl_name" : "fl_name",
        "ip_version" : "ip_version",
        "flowlistentries" : [
          {
            "seqnum" : "seqnum",
            "macdstaddr" : "macdstaddr",
            "macsrcaddr" : "macsrcaddr",
            "ipdstaddr" : "ipdstaddr",
            "ipdstaddrprefix" : "ipdstaddrprefix",
            "ipsrcaddr" : "ipsrcaddr",
            "ipsrcaddrprefix" : "ipsrcaddrprefix",
            "ipv6dstaddr" : "ipv6dstaddr",
            "ipv6dstaddrprefix" : "ipv6dstaddrprefix",
            "ipv6srcaddr" : "ipv6srcaddr",
            "ipv6srcaddrprefix" : "ipv6srcaddrprefix",
            "ipproto" : "ipproto",
            "l4dstport" : "l4dstport",
            "l4dstendport" : "l4dstendport",
            "l4srcport" : "l4srcport",
            "l4srcendport" : "l4srcendport",
            "statistics" : {
              "software" : {
                "packets" : "packets",
                "octets" : "octets"
              },
              "existingflow" : {
                "packets" : "packets",
                "octets" : "octets"
              },
```

```
                "expiredflow" : {
                  "packets" : "packets",
                  "octets" : "octets"
                },
                "total" : {
                  "packets" : "packets",
                  "octets" : "octets"
                }
              }
            }
          ]
        }
      }
    }
  }
```

Table 7-335 Description of Elements in safeflowfilter

Element	Description	Return Value
priority	Safe flow filter priority	Decimal integer (1 and 2)
srcmac	Flag indicating that the source IP address is set as a restriction.	{true} *Always true
dstmac	Flag indicating that the destination IP address is set as a restriction.	{true} *Always true
srcip	Flag indicating that the source IP address is set as a restriction.	{true} *Always true
dstip	Flag indicating that the destination IP address is set as a restriction.	{true} *Always true
ipproto	Flag indicating that the IP protocol is set as a restriction.	{true} *Always true
srcport	Flag indicating that the source TCP/UDP port number is set as a restriction.	{true} *Always true
dstport	Flag indicating that the destination TCP/UDP port number is set as a restriction.	{true} *Always true
allany	Flag indicating that **allany** is set as a safe flow filter entry match condition	{true} *Always true
safeflowfilterentries	Safe flow filter entry	safeflowfilterentries type (For details, refer to "Table 7-336 Description of Elements in safeflowfilterentries (page 499)")

Table 7-336 Description of Elements in safeflowfilterentries

Element	Description	Return Value
entryid	Safe flow filter entry ID	Decimal integer (1 to 3000)
allany	Flag indicating that all flows are the match target	{true} *Always true
fl_name	Flowlist name to be used as a match condition	Up to 32 characters including one-byte alphanumeric characters and underscores.
status	Status of the afe flow filter entry	{active|down(*details*)}
action_type	Action for the matched ethernet frame	{pass|drop|redirect} (Lowercase)
redirectdst	Destination information when action_type is redirect	redirectdst type (For details, refer to "Table 7-337 Description of

Element	Description	Return Value
		Elements in redirectdst (page 500)".
statistics	Safe flow filter entry statistical information	statistics type (For details, refer to "Table 7-340 Description of Elements in statistics (page 501)"
flowlist	Flowlist	flowlist type (For details, refer to "Table 7-338 Description of Elements in flowlist (page 500)".

Remember

Either allany or fl_name is returned.

Table 7-337 Description of Elements in redirectdst

Element	Description	Return Value
vnode_name	Virtual node name of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
if_name	Virtual interface of virtual node of redirect destination	Up to 31 characters including one-byte alphanumeric characters and underscores
macdstaddr	New destination MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal)
macsrcaddr	New source MAC address	hhhh.hhhh.hhhh format (h: Hexadecimal)

Table 7-338 Description of Elements in flowlist

Element	Description	Return Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores.
ip_version	IP version	{ip|ipv6} (Lowercase)
flowlistentries	Flowlist entry list	flowlistentries type (For details, refer to "Table 7-339 Description of Elements in flowlistentries (page 500)".

Table 7-339 Description of Elements in flowlistentries

Element	Description	Return Value
seqnum	Flowlist entry sequence number	Decimal integer (1 to 65535)
macdstaddr	Destinaltion IP address	hhhh.hhhh.hhhh format (h: Hexadecimal)
macsrcaddr	Source IP address	hhhh.hhhh.hhhh format (h: Hexadecimal)
ipdstaddr	Destination IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipdstaddrprefix	Destination IP address prefix length	Decimal integer (1 to 32)
ipsrcaddr	Source IP address	IPv4 dot-separated format (Example: 192.168.1.1)
ipsrcaddrprefix	Source IP address prefix length	Decimal integer (1 to 32)
ipv6dstaddr	Destination IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6dstaddrprefix	Destination IPv6 address prefix length	Decimal integer (1 to 128)

Element	Description	Return Value
ipv6srcaddr	Source IPv6 address	IPv6 address The IPv6 format conforms to RFC 5952.
ipv6srcaddrprefix	Source IPv6 address prefix length	Decimal integer (1 to 128)
ipproto	IP protocol number	Decimal integer (1 to 255)
l4dstport	Destination TCP/UDP port number (First port number when a range is specified)	Decimal integer (1 to 65535)
l4dstendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer (1 to 65535)
l4srcport	Source TCP/UDP port number (First port number when a range is specified)	Decimal integer (1 to 65535)
l4srcendport	Destination TCP/UDP port number (Last TCP/UDP port number when a range is specified)	Decimal integer (1 to 65535)
statistics	Flowfilter entry statistical information	statistics type (For details, refer to "Table 7-340 Description of Elements in statistics (page 501)".

Table 7-340 Description of Elements in statistics

Element	Description	Return Value
software	Number of packets that passed through VTN within the flow that matched the entry	Counter
existingflow	Number of packets that were hard-transferred by the flow entry currently set to OFS among the flows that matched the entry	Counter
expiredflow	Number of packets that were hard-transferred by the flow entry previously set to OFS in among the flows that matched the entry	Counter
total	Total number of packets of the flow that matched the entry	Counter

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get safeflowfilterentry detail info with entryid **entryid** of priority **priority** for interface **if_name** for vExternal **vex_name** of VTN **vtn_name** failed. [Invalid argument(entryid)](<Error information from pfcshell>)"

Remark

None

Chapter 8. WebAPI (QoS-Related) Reference

Following describes the URI, parameters, request/response data and other details about the QoS-related API.

Tip

The WebAPI uses structure data in the XML or JSON format as API input/output. In this chapter, for each API input/output data unit, the data elements on the same structure level is summarized in a table and the element name is written in bold like **"data".** If a lower level has data, the relevant level name is written in plain text like "data". The reference table number and other details about this data are shown in the "Return value" column.

8.1 Operate Policing Profile

8.1.1 List policing profiles

Following describes how to acquire the policing profile list.

Processing request

Method

GET

request URI

- XML format

  `/policing/profiles.xml`

- JSON format

  `/policing/profiles.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<policing>
  <profiles>
    [
    <profile prf_name="prf_name"/>
    ]
  </profiles>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profiles" : [
      {
        "prf_name" : "prf_name"
      }
    ]
  }
}
```

Table 8-1 Description of Elements in profile

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.1.2 Create policing profile

Following describes how to create a policing profile.

Processing request

Method

POST

request URI

- XML format

  `/policing/profiles.xml`
- JSON format

  `/policing/profiles.json`

Settings of request body

- XML format

```
<policing>
  <profile prf_name="prf_name"/>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name"
    }
  }
}
```

Table 8-2 Description of Elements in profile

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores

Remember

If "flood" is specified for **prf_name**, a policing file that performs policing control for flooding packets.

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Create policing profile **prf_name** succeeded."

- On failure

  "Create policing profile **prf_name** failed. [Invalid argument(prf_name)](<Error information from pfcshell>)"

Remark

None

8.1.3 Show policing profile

Following describes how to acquire the policing file information.

Processing request

Method

GET

request URI

- XML format

  `/policing/profiles/prf_name.xml`

- JSON format

  `/policing/profiles/prf_name.json`

Table 8-3 Description of Parameter in URI

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<policing>
  <profile prf_name="prf_name">
    <profileentries>
      [
      <profileentry seqnum="seqnum"
        fl_name="fl_name">
        <tworatethreecolor>
          <meter rateunit="rateunit"
            cir="cir"
            cbs="cbs"
            pir="pir"
            pbs="pbs"/>
          <greenaction green_action="green_action"
            ga_priority="ga_priority"
            ga_dscp="ga_dscp"
            ga_drop_precedence="ga_drop_precedence"/>
          <yellowaction yellow_action="yellow_action"
            ya_priority="ya_priority"
            ya_dscp="ya_dscp"
            ya_drop_precedence="ya_drop_precedence"/>
          <redaction red_action="red_action"
            ra_priority="ra_priority"
            ra_dscp="ra_dscp"
            ra_drop_precedence="ra_drop_precedence"/>
        </tworatethreecolor>
      </profileentry>
      ]
    </profileentries>
  </profile>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name",
      "profileentries" : [
        {
          "seqnum" : "seqnum",
          "fl_name" : "fl_name",
          "tworatethreecolor" : {
            "meter" : {
              "rateunit" : "rate_unit",
              "cir" : "cir",
              "cbs" : "cbs",
              "pir" : "pir",
              "pbs" : "pbs"
            },
            "greenaction" : {
              "green_action" : "green_action",
              "ga_priority" : "ga_priority",
              "ga_dscp" : "ga_dscp",
              "ga_drop_precedence" : "ga_drop_precedence"
            },
            "yellowaction" : {
              "yellow_action" : "yellow_action",
              "ya_priority" : "ya_priority",
              "ya_dscp" : "ya_dscp",
              "ya_drop_precedence" : "ya_drop_precedence"
            },
            "redaction" : {
              "red_action" : "red_action",
              "ra_priority" : "ra_priority",
              "ra_dscp" : "ra_dscp",
              "ra_drop_precedence" : "ra_drop_precedence"
            }
          }
        }
      ]
    }
  }
}
```

Table 8-4 Description of Elements in profile

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores
profileentries	Policing profile entry list	profileentry

Table 8-5 Description of Elements in profileentry

Element	Description	Valid Value
seqnum	Policy sequence number	Decimal number (1 to 255)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
meter	Rate and burst size meter information	meter
greenaction	Green action	greenaction
yellowaction	Yellow action	yellowaction
redaction	Red action	redaction

Table 8-6 Description of Elements in meter

Element	Description	Valid Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-7 Description of Elements in greenaction

Element	Description	Valid Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-8 Description of Elements in yellowaction

Element	Description	Valid Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-9 Description of Elements in redaction

Element	Description	Valid Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.1.4 Delete policing profile

Following describes how to DELETE a policing profile.

Processing request

Method

DELETE

request URI

- XML format

  `/policing/profiles/`**`prf_name`**`.xml`

- JSON format

  `/policing/profiles/`**`prf_name`**`.json`

Table 8-10 Description of Parameter in URI

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete policing profile **prf_name** succeeded."

- On failure

  "Delete policing profile **prf_name** failed.(<Error information from pfcshell>)"

Remark

None

8.2 Operate Policing Profile Entry

8.2.1 List policing profile entries

Following describes how to acquire the policing profile entry sequence number list.

Processing request

Method

GET

request URI

- XML format

  /policing/profiles/**prf_name**/profileentries.xml

- JSON format

  /policing/profiles/**prf_name**/profileentries.json

Table 8-11 Description of Parameter in URI

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<profileentries>
  [
```

```
  <profileentry seqnum="seqnum"/>
  ]
</profileentries>
```

- JSON format

```
{
  "profileentries" : [
    {
      "seqnum" : "seqnum"
    }
  ]
}
```

Table 8-12 Description of Elements in profileentry

Element	Description	Valid Value
seqnum	Policy sequence number	Decimal number (1 to 255)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.2.2 Create policing profile entry

Following describes how to create a policing profile entry.

Processing request

Method

POST

request URI

- XML format

  `/policing/profiles/`**`prf_name`**`/profileentries.xml`

- JSON format

  `/policing/profiles/`**`prf_name`**`/profileentries.json`

Table 8-13 Description of Parameter in URI

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<profileentry seqnum="seqnum"
  fl_name="fl_name">
  <tworatethreecolor>
    <meter rateunit="rateunit"
      cir="cir"
      cbs="cbs"
      pir="pir"
      pbs="pbs"/>
    <greenaction green_action="green_action"
      ga_priority="ga_priority"
      ga_dscp="ga_dscp"
      ga_drop_precedence="ga_drop_precedence"/>
    <yellowaction yellow_action="yellow_action"
      ya_priority="ya_priority"
      ya_dscp="ya_dscp"
      ya_drop_precedence="ya_drop_precedence"/>
    <redaction red_action="red_action"
      ra_priority="ra_priority"
      ra_dscp="ra_dscp"
      ra_drop_precedence="ra_drop_precedence"/>
  </tworatethreecolor>
</profileentry>
```

- JSON format

```
{
  "profileentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "tworatethreecolor" : {
      "meter" : {
        "rateunit" : "rateunit",
        "cir" : "cir",
        "cbs" : "cbs",
        "pir" : "pir",
        "pbs" : "pbs"
      },
      "greenaction" : {
        "green_action" : "green_action",
        "ga_priority" : "ga_priority",
        "ga_dscp" : "ga_dscp",
        "ga_drop_precedence" : "ga_drop_precedence"
      },
      "yellowaction" : {
        "yellow_action" : "yellow_action",
        "ya_priority" : "ya_priority",
        "ya_dscp" : "ya_dscp",
        "ya_drop_precedence" : "ya_drop_precedence"
      },
      "redaction" : {
        "red_action" : "red_action",
        "ra_priority" : "ra_priority",
        "ra_dscp" : "ra_dscp",
        "ra_drop_precedence" : "ra_drop_precedence"
      }
    }
  }
}
```

Table 8-14 Description of Elements in profileentry

Element	Description	Valid Value
seqnum	Policy sequence number	Decimal number (1 to 255)
fl_name	Flowlist name	Up to 31 characters including one-byte alphanumeric characters and underscores
tworatethreecolor	2rate3color policing	tworatethreecolor

Table 8-15 Description of Elements in tworatethreecolor

Element	Description	Valid Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction
redaction	Red action (for display)	redaction

Table 8-16 Description of Elements in meter

Element	Description	Valid Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-17 Description of Elements in greenaction

Element	Description	Valid Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3) ("1" is discarded before "3".)

Table 8-18 Description of Elements in yellowaction

Element	Description	Valid Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3) ("1" is discarded before "3".)

Table 8-19 Description of Elements in redaction

Element	Description	Valid Value
red_action	Action	{pass|drop|penalty} (Lowercase)

Element	Description	Valid Value
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3) ("1" is discarded before "3".)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

- On success

  "Create sequence number **seqnum** of policing profile **prf_name** succeeded."
- On failure

  "Create sequence number **seqnum** of policing profile **prf_name** failed. [Invalid argument](<Error information from pfcshell>)"

Remark

None

8.2.3 Show policing profile entry

Processing request

Method

GET

request URI

- XML format

  /policing/profiles/**prf_name**/profileentries/**seqnum**.xml
- JSON format

  /policing/profiles/**prf_name**/profileentries/**seqnum**.json

Table 8-20 Description of Parameter in URI

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores
seqnum	Policy sequence number	Decimal number (1 to 255)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<profileentry seqnum="seqnum"
  fl_name="fl_name">
  <tworatethreecolor>
    <meter rateunit="rateunit"
      cir="cir"
      cbs="cbs"
      pir="pir"
      pbs="pbs"/>
    <greenaction green_action="green_action"
      ga_priority="ga_priority"
      ga_dscp="ga_dscp"
      ga_drop_precedence="ga_drop_precedence"/>
    <yellowaction yellow_action="yellow_action"
      ya_priority="ya_priority"
      ya_dscp="ya_dscp"
      ya_drop_precedence="ya_drop_precedence"/>
    <redaction red_action="red_action"
      ra_priority="ra_priority"
      ra_dscp="ra_dscp"
      ra_drop_precedence="ra_drop_precedence"/>
  </tworatethreecolor>
</profileentry>
```

- JSON format

```
{
  "profileentry" : {
    "seqnum" : "seqnum",
    "fl_name" : "fl_name",
    "tworatethreecolor" : {
      "meter" : {
        "rateunit" : "rateunit",
        "cir" : "cir",
        "cbs" : "cbs",
        "pir" : "pir",
        "pbs" : "pbs"
      },
      "greenaction" : {
        "green_action" : "green_action",
        "ga_priority" : "ga_priority",
        "ga_dscp" : "ga_dscp",
        "ga_drop_precedence" : "ga_drop_precedence"
      },
      "yellowaction" : {
        "yellow_action" : "yellow_action",
        "ya_priority" : "ya_priority",
        "ya_dscp" : "ya_dscp",
        "ya_drop_precedence" : "ya_drop_precedence"
```

```
      },
      "redaction" : {
        "red_action" : "red_action",
        "ra_priority" : "ra_priority",
        "ra_dscp" : "ra_dscp",
        "ra_drop_precedence" : "ra_drop_precedence"
      }
    }
  }
}
```

Table 8-21 Description of Elements in profileentry

Element	Description	Valid Value
seqnum	Policy sequence number	Decimal number (1 to 255)
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
tworatethreecolor	2rate3color policing	tworatethreecolor

Table 8-22 Description of Elements in tworatethreecolor

Element	Description	Valid Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction
redaction	Red action (for display)	redaction

Table 8-23 Description of Elements in meter

Element	Description	Valid Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-24 Description of Elements in greenaction

Element	Description	Valid Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-25 Description of Elements in yellowaction

Element	Description	Valid Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-26 Description of Elements in redaction

Element	Description	Valid Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3)

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On failure

  "Get sequence number **seqnum** of policing profile **prf_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

8.2.4 Update policing profile entry

Processing request

Method

PUT

request URI

- XML format

  /policing/profiles/**prf_name**/profileentries/**seqnum**.xml

- JSON format

  /policing/profiles/**prf_name**/profileentries/**seqnum**.json

Table 8-27 Description of Parameter in URI

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores
seqnum	Policy sequence number	Decimal number (1 to 255)

Settings of request body

- XML format

```
<profileentry op="op"
  fl_name="fl_name">
  <tworatethreecolor>
    <meter rateunit="rateunit"
      cir="cir"
      cbs="cbs"
      pir="pir"
      pbs="pbs"/>
    <greenaction green_action="green_action"
      ga_priority="ga_priority"
      ga_dscp="ga_dscp"
      ga_drop_precedence="ga_drop_precedence"/>
    <yellowaction yellow_action="yellow_action"
      ya_priority="ya_priority"
      ya_dscp="ya_dscp"
      ya_drop_precedence="ya_drop_precedence"/>
    <redaction red_action="red_action"
      ra_priority="ra_priority"
      ra_dscp="ra_dscp"
      ra_drop_precedence="ra_drop_precedence"/>
  </tworatethreecolor>
</profileentry>
```

- JSON format

```
{
  "profileentry" : {
    "op" : "op",
    "fl_name" : "fl_name",
    "tworatethreecolor" : {
      "meter" : {
        "rateunit" : "rateunit",
        "cir" : "cir",
        "cbs" : "cbs",
        "pir" : "pir",
        "pbs" : "pbs"
      },
      "greenaction" : {
        "green_action" : "green_action",
        "ga_priority" : "ga_priority",
        "ga_dscp" : "ga_dscp",
        "ga_drop_precedence" : "ga_drop_precedence"
      },
      "yellowaction" : {
        "yellow_action" : "yellow_action",
        "ya_priority" : "ya_priority",
        "ya_dscp" : "ya_dscp",
        "ya_drop_precedence" : "ya_drop_precedence"
      },
      "redaction" : {
        "red_action" : "red_action",
        "ra_priority" : "ra_priority",
        "ra_dscp" : "ra_dscp",
        "ra_drop_precedence" : "ra_drop_precedence"
      }
    }
  }
}
```

Table 8-28 Description of Elements in profileentry

Element	Description	Valid Value
op	Setting type	{add |delete} (Lowercase) This element must be specified.

Element	Description	Valid Value
fl_name	Flowlist name	Up to 32 characters including one-byte alphanumeric characters and underscores
tworatethreecolor	2rate3color policing	Tworatethreecolor

Table 8-29 Description of Elements in tworatethreecolor

Element	Description	Valid Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction
redaction	Red action (for display)	redaction

Table 8-30 Description of Elements in meter

Element	Description	Valid Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-31 Description of Elements in greenaction

Element	Description	Valid Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-32 Description of Elements in yellowaction

Element	Description	Valid Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-33 Description of Elements in redaction

Element	Description	Valid Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3)

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Update(**op**) sequence number **seqnum** of policing profile **prf_name** succeeded."

- On failure

  "Update sequence number **seqnum** of policing profile **prf_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

  "Update sequence number **seqnum** of policing profile **prf_name** failed. [Invalid argument(**parameter subdivision**))](<Error information from pfcshell>)"

Remark

None

8.2.5 Delete policing profile entry

Processing request

Method

DELETE

request URI

- XML format

  /policing/profiles/**prf_name**/profileentries/**seqnum**.xml

- JSON format

  /policing/profiles/**prf_name**/profileentries/**seqnum**.json

Table 8-34 Description of Parameter in URI

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores
seqnum	Policy sequence number	Decimal number (1 to 255)

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Delete sequence number **seqnum** of policing profile **prf_name** succeeded."

- On failure

  "Delete sequence number **seqnum** of policing profile **prf_name** failed. [Invalid argument(seqnum)](<Error information from pfcshell>)"

Remark

None

8.3 Operate Policing Setting Information for Flooding Packets

8.3.1 Show flood policing

Processing request

Method

GET

request URI

- XML format

  `/flooding/policing.xml`

- JSON format

  `/flooding/policing.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<policing>
  <profile prf_name="prf_name">
    <profileentry>
      <tworatethreecolor>
        <meter rateunit="rateunit"
          cir="cir"
          cbs="cbs"
          pir="pir"
          pbs="pbs"/>
        <greenaction green_action="green_action"
          ga_priority="ga_priority"
          ga_dscp="ga_dscp"
          ga_drop_precedence="ga_drop_precedence"/>
        <yellowaction yellow_action="yellow_action"
          ya_priority="ya_priority"
          ya_dscp="ya_dscp"
          ya_drop_precedence="ya_drop_precedence"/>
        <redaction red_action="red_action"
          ra_priority="ra_priority"
          ra_dscp="ra_dscp"
          ra_drop_precedence="ra_drop_precedence"/>
      </tworatethreecolor>
      <ofses>
        [
        <ofs dp_id="dp_id">
          <policer id="id"
            status="status"/>
        </ofs>
        ]
      </ofses>
    </profileentry>
  </profile>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name",
      "profileentry" : {
        "tworatethreecolor" : {
          "meter" : {
            "rateunit" : "rateunit",
            "cir" : "cir",
            "cbs" : "cbs",
            "pir" : "pir",
```

```
            "pbs" : "pbs"
          },
          "greenaction" : {
            "green_action" : "green_action",
            "ga_priority" : "ga_priority",
            "ga_dscp" : "ga_dscp",
            "ga_drop_precedence" : "ga_drop_precedence"
          },
          "yellowaction" : {
            "yellow_action" : "yellow_action",
            "ya_priority" : "ya_priority",
            "ya_dscp" : "ya_dscp",
            "ya_drop_precedence" : "ya_drop_precedence"
          },
          "redaction" : {
            "red_action" : "red_action",
            "ra_priority" : "ra_priority",
            "ra_dscp" : "ra_dscp",
            "ra_drop_precedence" : "ra_drop_precedence"
          }
        },
        "ofses" : [
          {
            "dp_id" : "dp_id",
            "policer" : {
              "id" : "id",
              "status" : "status"
            }
          }
        ]
      }
    }
  }
}
```

Table 8-35 Description of Elements in profile

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores
profileentry	Policing profile entry	profileentry

Table 8-36 Description of Elements in profileentry

Element	Description	Valid Value
tworatethreecolor	2rate3color policing	tworatethreecolor
ofses	OFS list	ofses

Table 8-37 Description of Elements in tworatethreecolor

Element	Description	Valid Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction
redaction	Red action (for display)	redaction

Table 8-38 Description of Elements in meter

Element	Description	Valid Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)

Element	Description	Valid Value
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-39 Description of Elements in greenaction

Element	Description	Valid Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3) ("1" is discarded before "3".)

Table 8-40 Description of Elements in yellowaction

Element	Description	Valid Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3) ("1" is discarded before "3".)

Table 8-41 Description of Elements in redaction

Element	Description	Valid Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3) ("1" is discarded before "3".)

Table 8-42 Description of Elements in ofses

Element	Description	Valid Value
ofs	OFS	ofs

Table 8-43 Description of Elements in ofs

Element	Description	Return Value
dp_id	DPID of the OFS registered as the policer	HHHH-HHHH-HHHH-HHHH format (H: Hexadecimal number). *It is not allowed to specify F's to all digits.
policer	Policer information	policer

Table 8-44 Description of Elements in policer

Element	Description	Return Value
id	ID of policer registered in the OFS	Decimal number (1 to 4294967295). *"N/A" is displayed if **status** is "down."
status	Setting status of policer registration	{active|down(detail)} (Lowercase) *The reason for "down" is displayed in detail.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.3.2 Show flood policing detail

Processing request

Method

GET

request URI

- XML format

  `/flooding/policing/detail.xml`
- JSON format

  `/flooding/policing/detail.json`

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<policing>
```

```
  <profile prf_name="prf_name">
    <profileentry>
      <tworatethreecolor>
        <meter rateunit="rateunit"
          cir="cir"
          cbs="cbs"
          pir="pir"
          pbs="pbs"/>
        <greenaction green_action="green_action"
          ga_priority="ga_priority"
          ga_dscp="ga_dscp"
          ga_drop_precedence="ga_drop_precedence"/>
        <yellowaction yellow_action="yellow_action"
          ya_priority="ya_priority"
          ya_dscp="ya_dscp"
          ya_drop_precedence="ya_drop_precedence"/>
        <redaction red_action="red_action"
          ra_priority="ra_priority"
          ra_dscp="ra_dscp"
          ra_drop_precedence="ra_drop_precedence"/>
      </tworatethreecolor>
      <statistics total_packets="total_packets"
        total_bytes="total_bytes"
        green_yellow_packets="green_yellow_packets"
        green_yellow_bytes="green_yellow_bytes"
        red_packets="red_packets"
        red_bytes="red_bytes"/>
      <ofses>
        [
        <ofs dp_id="dp_id">
          <policer id="id"
            status="status"/>
          <statistics total_packets="total_packets"
            total_bytes="total_bytes"
            green_yellow_packets="green_ywllow_packets"
            green_yellow_bytes="green_yellow_bytes"
            red_packets="red_packets"
            red_bytes="red_bytes"/>
        </ofs>
        ]
      </ofses>
    </profileentry>
  </profile>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name",
      "profileentry" : {
        "tworatethreecolor" : {
          "meter" : {
            "rateunit" : "rateunit",
            "cir" : "cir",
            "cbs" : "cbs",
            "pir" : "pir",
            "pbs" : "pbs"
          },
          "greenaction" : {
            "green_action" : "green_action",
            "ga_priority" : "ga_priority",
            "ga_dscp" : "ga_dscp",
            "ga_drop_precedence" : "ga_drop_precedence"
          },
          "yellowaction" : {
            "yellow_action" : "yellow_action",
            "ya_priority" : "ya_priority",
            "ya_dscp" : "ya_dscp",
            "ya_drop_precedence" : "ya_drop_precedence"
          },
```

```
          "redaction" : {
            "red_action" : "red_action",
            "ra_priority" : "ra_priority",
            "ra_dscp" : "ra_dscp",
            "ra_drop_precedence" : "ra_drop_precedence"
          }
        },
        "statistics" : {
          "total_packets" : "total_packets",
          "total_bytes" : "total_bytes",
          "green_yellow_packets" : "green_yellow_packets",
          "green_yellow_bytes" : "green_yellow_bytes",
          "red_packets" : "red_packets",
          "red_bytes" : "red_bytes"
        },
        "ofses" : [
          {
            "dp_id" : "dp_id",
            "policer" : {
              "id" : "id",
              "status" : "status"
            },
            "statistics" : {
              "total_packets" : "total_packets",
              "total_bytes" : "total_bytes",
              "green_yellow_packets" : "green_yellow_packets",
              "green_yellow_bytes" : "green_yellow_bytes",
              "red_packets" : "red_packets",
              "red_bytes" : "red_bytes"
            }
          }
        ]
      }
    }
  }
}
```

Table 8-45 Description of Elements in profile

Element	Description	Return Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores
profileentry	Policing profile entry	profileentry

Table 8-46 Description of Elements in profileentry

Element	Description	Return Value
tworatethreecolor	2rate3color policing	tworatethreecolor
statistics	Packet and byte information of all OFSs	statistics type
ofses	List of OFSs	ofses

Table 8-47 Description of Elements in tworatethreecolor

Element	Description	Return Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction
redaction	Red action (for display)	redaction

Table 8-48 Description of Elements in meter

Element	Description	Return Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-49 Description of Elements in greenaction

Element	Description	Return Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-50 Description of Elements in yellowaction

Element	Description	Return Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-51 Description of Elements in redaction

Element	Description	Return Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-52 Description of Elements in statistics type

Element	Description	Return Value
total_packets	Total number of packets	Counter
total_bytes	Total number of bytes	Counter
green_yellow_packets	Total number of packets which green or yellow actions were applied to	Counter
green_yellow_bytes	Total number of bytes which green or yellow actions were applied to	Counter
red_packets	Total number of packets which red actions were applied to	Counter
red_bytes	Total number of bytes which red actions were applied to	Counter

Table 8-53 Description of Elements in ofses

Element	Description	Return Value
ofs	OFS	ofs

Table 8-54 Description of Elements in ofs

Element	Description	Return Value
dp_id	DPID of the OFS registered as the policer	HHHH-HHHH-HHHH-HHHH format (H: Hexadecimal number). *It is not allowed to specify F's to all digits.
policer	Policer information	policer
statistics	Packet and byte information	statistics type

Table 8-55 Description of Elements in policer

Element	Description	Return Value
id	ID of policer registered in the OFS	Decimal number (1 to 4294967295). *"N/A" is displayed if status is "down."
status	Setting status of policer registration	{active|down(detail)} (Lowercase) *The reason for "down" is displayed in detail.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.4 Operate Policing Setting Information (VTN Mode)

8.4.1 Show policing profile (VTN mode)

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/policing.xml`

- JSON format

  `/vtns/vtn_name/policing.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `seqnum=seqnum`

Table 8-56 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
seqnum	Policy sequence number	Decimal number (1 to 255)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<policing>
  <profile prf_name="prf_name">
    <profileentries>
      [
      <profileentry seqnum="seqnum">
        <tworatethreecolor>
          <meter rateunit="rateunit"
            cir="cir"
            cbs="cbs"
            pir="pir"
            pbs="pbs"/>
          <greenaction green_action="green_action"
            ga_priority="ga_priority"
            ga_dscp="ga_dscp"
            ga_drop_precedence="ga_drop_precedence"/>
          <yellowaction yellow_action="yellow_action"
            ya_priority="ya_priority"
            ya_dscp="ya_dscp"
            ya_drop_precedence="ya_drop_precedence"/>
          <redaction red_action="red_action"
            ra_priority="ra_priority"
            ra_dscp="ra_dscp"
            ra_drop_precedence="ra_drop_precedence"/>
        </tworatethreecolor>
        <ofses>
          [
          <ofs dp_id="dp_id">
            <policer id="id"
              vexternal="vexternal"
              port="port"
              vlan_id="vlan_id"
```

```
              status="status"/>
          </ofs>
          ]
        </ofses>
      </profileentry>
      ]
    </profileentries>
  </profile>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name",
      "profileentries" : [
        {
          "seqnum" : "seqnum",
          "tworatethreecolor" : {
            "meter" : {
              "rateunit" : "rateunit",
              "cir" : "cir",
              "cbs" : "cbs",
              "pir" : "pir",
              "pbs" : "pbs"
            },
            "greenaction" : {
              "green_action" : "green_action",
              "ga_priority" : "ga_priority",
              "ga_dscp" : "ga_dscp",
              "ga_drop_precedence" : "ga_drop_precedence"
            },
            "yellowaction" : {
              "yellow_action" : "yellow_action",
              "ya_priority" : "ya_priority",
              "ya_dscp" : "ya_dscp",
              "ya_drop_precedence" : "ya_drop_precedence"
            },
            "redaction" : {
              "red_action" : "red_action",
              "ra_priority" : "ra_priority",
              "ra_dscp" : "ra_dscp",
              "ra_drop_precedence" : "ra_drop_precedence"
            }
          },
          "ofses" : [
            {
              "dp_id" : "dp_id",
              "policer" : {
                "id" : "id",
                "vexternal" : "vexternal",
                "port" : "port",
                "vlan_id" : "vlan_id",
                "status" : "status"
              }
            }
          ]
        }
      ]
    }
  }
}
```

Table 8-57 Description of Elements in profile

Element	Description	Return Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores
profileentries	Policing profile entry list	profileentries

Table 8-58 Description of Elements in profileentries

Element	Description	Return Value
profileentry	Policing profile entry	profileentry

Table 8-59 Description of Elements in profileentry

Element	Description	Return Value
seqnum	Policy secquence number	Hexadecimal number (1 to 255)
tworatethreecolor	2rate3color policing	tworatethreecolor
ofses	List of OFSs	ofses

Table 8-60 Description of Elements in tworatethreecolor

Element	Description	Return Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction
redaction	Red action (for display)	redaction

Table 8-61 Description of Elements in meter

Element	Description	Return Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-62 Description of Elements in greenaction

Element	Description	Return Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-63 Description of Elements in yellowaction

Element	Description	Return Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-64 Description of Elements in redaction

Element	Description	Return Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-65 Description of Elements in ofses

Element	Description	Return Value
ofs	OFS	ofs

Table 8-66 Description of Elements in ofs

Element	Description	Return Value
dp_id	DPID of the OFS registered as the policer	HHHH-HHHH-HHHH-HHHH format (H: Hexadecimal number). *It is not allowed to specify F's to all digits.
policer	Policer information	policer

Table 8-67 Description of Elements in policer

Element	Description	Return Value
id	ID of policer registered in the OFS	Decimal number (1 to 4294967295). *"N/A" is displayed if **status** is "down."
vexternal	vExternal name related to policer registration	Up to 31 characters including one-byte alphanumeric characters and underscores
port	Physical port which the policer settings are mapped to	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
status	Setting status of policer registration	{active|down(detail)} (Lowercase) *The reason for "down" is displayed in detail.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.4.2 Map policing profile (VTN mode)

Processing request

Method

PUT

request URI

- XML format

  /vtns/**vtn_name**/policing.xml

- JSON format

  /vtns/**vtn_name**/policing.json

Table 8-68 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<policing>
  <profile prf_name="prf_name"/>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name"
    }
  }
}
```

Table 8-69 Description of Elements in profile

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

400,500

Miscellaneous

Log

- On success

  `"Map policing profile prf_name to VTN vtn_name succeeded."`

- On failure

  `"Map policing profile prf_name of VTN vtn_name failed. [Invalid argument(prf_name)](<Error information from pfcshell>)"`

Remark

None

8.4.3 Unmap policing profile (VTN mode)

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/vtn_name/policing.xml`

- JSON format

  `/vtns/vtn_name/policing.json`

Table 8-70 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  `"Unmap policing profile from VTN vtn_name succeeded."`

- On failure

  `"Unmap policing profile from VTN vtn_name failed.(<Error information from pfcshell>)"`

Remark

None

8.4.4 Show policing profile detail (VTN mode)

Processing request

Method

GET

request URI

- XML format

  `/vtns/vtn_name/policing/detail.xml`

- JSON format

  `/vtns/vtn_name/policing/detail.json`

- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `seqnum=seqnum`

Table 8-71 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
seqnum	Policy sequence number	Decimal number (1 to 255)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<policing>
  <profile prf_name="prf_name">
    <profileentries>
      [
      <profileentry seqnum="seqnum">
        <tworatethreecolor>
          <meter rateunit="rateunit"
            cir="cir"
            cbs="cbs"
            pir="pir"
            pbs="pbs"/>
          <greenaction green_action="green_action"
            ga_priority="ga_priority"
            ga_dscp="ga_dscp"
            ga_drop_precedence="ga_drop_precedence"/>
          <yellowaction yellow_action="yellow_action"
            ya_priority="ya_priority"
            ya_dscp="ya_dscp"
            ya_drop_precedence="ya_drop_precedence"/>
          <redaction red_action="red_action"
            ra_priority="ra_priority"
            ra_dscp="ra_dscp"
            ra_drop_precedence="ra_drop_precedence"/>
        </tworatethreecolor>
        <statistics total_packets="total_packets"
          total_bytes="total_bytes"
          green_yellow_packets="green_yellow_packets"
          green_yellow_bytes="green_yellow_bytes"
          red_packets="red_packets"
          red_bytes="red_bytes"/>
        <ofses>
          [
          <ofs dp_id="dp_id">
            <policer id="id"
              vexternal="vexternal"
              port="port"
              vlan_id="vlan_id"
              status="status"/>
            <statistics total_packets="total_packets"
              total_bytes="total_bytes"
              green_yellow_packets="green_yellow_packets"
              green_yellow_bytes="green_yellow_bytes"
              red_packets="red_packets"
              red_bytes="red_bytes"/>
          </ofs>
          ]
        </ofses>
      </profileentry>
      ]
    </profileentries>
  </profile>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name",
      "profileentries" : [
        {
          "seqnum" : "seqnum",
          "tworatethreecolor" : {
```

```
            "meter" : {
              "rateunit" : "rateunit",
              "cir" : "cir",
              "cbs" : "cbs",
              "pir" : "pir",
              "pbs" : "pbs"
            },
            "greenaction" : {
              "green_action" : "green_action",
              "ga_priority" : "ga_priority",
              "ga_dscp" : "ga_dscp",
              "ga_drop_precedence" : "ga_drop_precedence"
            },
            "yellowaction" : {
              "yellow_action" : "yellow_action",
              "ya_priority" : "ya_priority",
              "ya_dscp" : "ya_dscp",
              "ya_drop_precedence" : "ya_drop_precedence"
            },
            "redaction" : {
              "red_action" : "red_action",
              "ra_priority" : "ra_priority",
              "ra_dscp" : "ra_dscp",
              "ra_drop_precedence" : "ra_drop_precedence"
            }
          },
          "statistics" : {
            "total_packets" : "total_packets",
            "total_bytes" : "total_bytes",
            "green_yellow_packets" : "green_yellow_packets",
            "green_yellow_bytes" : "green_yellow_bytes",
            "red_packets" : "red_packets",
            "red_bytes" : "red_bytes"
          },
          "ofses" : [
            {
              "dp_id" : "dp_id",
              "policer" : {
                "id" : "id",
                "vexternal" : "vexternal",
                "port" : "port",
                "vlan_id" : "vlan_id",
                "status" : "status"
              },
              "statistics" : {
                "total_packets" : "total_packets",
                "total_bytes" : "total_bytes",
                "green_yellow_packets" : "green_yellow_packets",
                "green_yellow_bytes" : "green_yellow_bytes",
                "red_packets" : "red_packets",
                "red_bytes" : "red_bytes"
              }
            }
          ]
        }
      ]
    }
  }
}
```

Table 8-72 Description of Elements in profile

Element	Description	Return Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores
profileentries	Policing profile entry list	profileentries

Table 8-73 Description of Elements in profileentries

Element	Description	Return Value
profileentry	Policing profile entry	profileentry

Table 8-74 Description of Elements in profileentry

Element	Description	Return Value
seqnum	Policy sequence number	Hexadecimal number (1 to 255)
tworatethreecolor	2rate3color policing	tworatethreecolor
statistics	Packet and byte information	statistics type
ofses	List of OFSs	ofses

Table 8-75 Description of Elements in tworatethreecolor

Element	Description	Return Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction
redaction	Red action (for display)	redaction

Table 8-76 Description of Elements in meter

Element	Description	Return Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-77 Description of Elements in greenaction

Element	Description	Return Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-78 Description of Elements in yellowaction

Element	Description	Return Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-79 Description of Elements in redaction

Element	Description	Return Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-80 Description of Elements in statistics type

Element	Description	Return Value
total_packets	Total number of packets	Counter
total_bytes	Total number of bytes	Counter
green_yellow_packets	Total number of packets which green or yellow actions were applied to	Counter
green_yellow_bytes	Total number of bytes which green or yellow actions were applied to	Counter
red_packets	Total number of packets which red actions were applied to	Counter
red_bytes	Total number of bytes which red actions were applied to	Counter

Table 8-81 Description of Elements in ofses

Element	Description	Return Value
ofs	OFS	ofs

Table 8-82 Description of Elements in ofs

Element	Description	Return Value
dp_id	DPID of the OFS registered as the policer	HHHH-HHHH-HHHH-HHHH format (H: Hexadecimal number). *It is not allowed to specify F's to all digits.
policer	Policer information	policer
statistics	Packet and byte information	statistics type

Table 8-83 Description of Elements in policer

Element	Description	Return Value
id	ID of policer registered in the OFS	Decimal number (1 to 4294967295). *"N/A" is displayed if **status** is "down."
vexternal	vExternal name related to policer registration	Up to 31 characters including one-byte alphanumeric characters and underscores
port	Physical port which the policer settings are mapped to	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
status	Setting status of policer registration	{active|down(detail)} (Lowercase) *The reason for "down" is displayed in detail.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.5 Operate Policing Setting Information (vBridge Mode)

8.5.1 Show policing profile (vBridge mode)

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/policing.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/policing.json`
- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `seqnum=`**`seqnum`**

Table 8-84 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
seqnum	Policy sequence number	Decimal number (1 to 255)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<policing>
  <profile prf_name="prf_name">
    <profileentries>
      [
      <profileentry seqnum="seqnum">
        <tworatethreecolor>
          <meter rateunit="rateunit"
            cir="cir"
            cbs="cbs"
            pir="pir"
            pbs="pbs"/>
          <greenaction green_action="green_action"
            ga_priority="ga_priority"
            ga_dscp="ga_dscp"
            ga_drop_precedence="ga_drop_precedence"/>
          <yellowaction yellow_action="yellow_action"
            ya_priority="ya_priority"
            ya_dscp="ya_dscp"
            ya_drop_precedence="ya_drop_precedence"/>
          <redaction red_action="red_action"
            ra_priority="ra_priority"
            ra_dscp="ra_dscp"
            ra_drop_precedence="ra_drop_precedence"/>
        </tworatethreecolor>
        <ofses>
          [
          <ofs dp_id="dp_id">
            <policer id="id"
              vexternal="vexternal"
              port="port"
              vlan_id="vlan_id"
              status="status"/>
          </ofs>
          ]
        </ofses>
      </profileentry>
      ]
    </profileentries>
  </profile>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name",
      "profileentries" : [
        {
          "seqnum" : "seqnum",
          "tworatethreecolor" : {
            "meter" : {
              "rateunit" : "rateunit",
              "cir" : "cir",
              "cbs" : "cbs",
              "pir" : "pir",
              "pbs" : "pbs"
            },
```

```
            "greenaction" : {
              "green_action" : "green_action",
              "ga_priority" : "ga_priority",
              "ga_dscp" : "ga_dscp",
              "ga_drop_precedence" : "ga_drop_precedence"
            },
            "yellowaction" : {
              "yellow_action" : "yellow_action",
              "ya_priority" : "ya_priority",
              "ya_dscp" : "ya_dscp",
              "ya_drop_precedence" : "ya_drop_precedence"
            },
            "redaction" : {
              "red_action" : "red_action",
              "ra_priority" : "ra_priority",
              "ra_dscp" : "ra_dscp",
              "ra_drop_precedence" : "ra_drop_precedence"
            }
          },
          "ofses" : [
            {
              "dp_id" : "dp_id",
              "policer" : {
                "id" : "id",
                "vexternal" : "vexternal",
                "port" : "port",
                "vlan_id" : "vlan_id",
                "status" : "status"
              }
            }
          ]
        }
      ]
    }
  }
}
```

Table 8-85 Description of Elements in profile

Element	Description	Return Value
prf_name	Policing profile name	Up to 31 characters including one-byte alphanumeric characters and underscores
profileentries	Policing profile entry list	profileentries

Table 8-86 Description of Elements in profileentries

Element	Description	Return Value
profileentry	Policing profile entry	profileentry

Table 8-87 Description of Elements in profileentry

Element	Description	Return Value
seqnum	Policy sequence number	Hexadecimal number (1 to 255)
tworatethreecolor	2rate3color policing	tworatethreecolor
ofses	List of OFSs	ofses

Table 8-88 Description of Elements in tworatethreecolor

Element	Description	Return Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction

Element	Description	Return Value
redaction	Red action (for display)	redaction

Table 8-89 Description of Elements in meter

Element	Description	Return Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-90 Description of Elements in greenaction

Element	Description	Return Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-91 Description of Elements in yellowaction

Element	Description	Return Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-92 Description of Elements in redaction

Element	Description	Return Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-93 Description of Elements in ofses

Element	Description	Return Value
ofs	OFS	ofs

Table 8-94 Description of Elements in ofs

Element	Description	Return Value
dp_id	DPID of the OFS registered as the policer	HHHH-HHHH-HHHH-HHHH format (H: Hexadecimal number). *It is not allowed to specify F's to all digits.

Element	Description	Return Value
policer	Policer information	policer

Table 8-95 Description of Elements in policer

Element	Description	Return Value
id	ID of policer registered in the OFS	Decimal number (1 to 4294967295). *"N/A" is displayed if **status** is "down."
vexternal	vExternal name related to policer registration	Up to 31 characters including one-byte alphanumeric characters and underscores
port	Physical port which the policer settings are mapped to	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
status	Setting status of policer registration	{active|down(detail)} (Lowercase) *The reason for "down" is displayed in detail.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.5.2 Map policing profile to vBridge

Processing request

Method

PUT

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/policing.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vbridges/`**`vbr_name`**`/policing.json`

Table 8-96 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<policing>
  <profile prf_name="prf_name"/>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name"
    }
  }
}
```

Table 8-97 Description of Elements in profile

Element	Description	Valid Value
prf_name	Policing profile name	Up to 31 characters including one-byte alphanumeric characters and underscores

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Map policing profile **prf_name** to VTN **vtn_name** vBridge **vbr_name** succeeded."

- On failure

```
"Map policing profile prf_name of vBridge vbr_name of VTN vtn_name f
ailed. [Invalid argument(prf_name)](<Error information from pfcshell
>)"
```

Remark

None

8.5.3 Unmap policing profile from vBridge

Processing request

Method

DELETE

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/policing.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/policing.json

Table 8-98 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Unmap policing profile from VTN **vtn_name** vBridge **vbr_name** succeeded."

- On failure

  "Unmap policing profile from VTN **vtn_name** vBridge **vbr_name** failed.(<Error information from pfcshell>)"

Remark

None

8.5.4 Show policing profile detail on vBridge

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vbridges/**vbr_name**/policing/detail.xml

- JSON format

  /vtns/**vtn_name**/vbridges/**vbr_name**/policing/detail.json

- Parameter

  The following parameters are passed in the URI query character string (in the format of ?param1=***&param2=***).

  seqnum=**seqnum**

Table 8-99 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vbr_name	vBridge name	Up to 31 characters including one-byte alphanumeric characters and underscores
seqnum	Policy sequence number	Decimal number (1 to 255)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<policing>
  <profile prf_name="prf_name">
    <profileentries>
      [
      <profileentry seqnum="seqnum">
        <tworatethreecolor>
          <meter rateunit="rateunit"
            cir="cir"
            cbs="cbs"
            pir="pir"
            pbs="pbs"/>
          <greenaction green_action="green_action"
            ga_priority="ga_priority"
            ga_dscp="ga_dscp"
            ga_drop_precedence="ga_drop_precedence"/>
          <yellowaction yellow_action="yellow_action"
            ya_priority="ya_priority"
            ya_dscp="ya_dscp"
            ya_drop_precedence="ya_drop_precedence"/>
          <redaction red_action="red_action"
            ra_priority="ra_priority"
            ra_dscp="ra_dscp"
            ra_drop_precedence="ra_drop_precedence"/>
        </tworatethreecolor>
        <statistics total_packets="total_packets"
          total_bytes="total_bytes"
          green_yellow_packets="green_yellow_packets"
          green_yellow_bytes="green_yellow_bytes"
          red_packets="red_packets"
          red_bytes="red_bytes"/>
        <ofses>
          [
          <ofs dp_id="dp_id">
            <policer id="id"
              vexternal="vexternal"
              port="port"
              vlan_id="vlan_id"
              status="status"/>
            <statistics total_packets="total_packets"
              total_bytes="total_bytes"
              green_yellow_packets="green_yellow_packets"
              green_yellow_bytes="green_yellow_bytes"
              red_packets="red_packets"
              red_bytes="red_bytes"/>
          </ofs>
          ]
        </ofses>
      </profileentry>
      ]
    </profileentries>
  </profile>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name",
      "profileentries" : [
        {
          "seqnum" : "seqnum",
          "tworatethreecolor" : {
```

```
            "meter" : {
              "rateunit" : "rateunit",
              "cir" : "cir",
              "cbs" : "cbs",
              "pir" : "pir",
              "pbs" : "pbs"
            },
            "greenaction" : {
              "green_action" : "green_action",
              "ga_priority" : "ga_priority",
              "ga_dscp" : "ga_dscp",
              "ga_drop_precedence" : "ga_drop_precedence"
            },
            "yellowaction" : {
              "yellow_action" : "yellow_action",
              "ya_priority" : "ya_priority",
              "ya_dscp" : "ya_dscp",
              "ya_drop_precedence" : "ya_drop_precedence"
            },
            "redaction" : {
              "red_action" : "red_action",
              "ra_priority" : "ra_priority",
              "ra_dscp" : "ra_dscp",
              "ra_drop_precedence" : "ra_drop_precedence"
            }
          },
          "statistics" : {
            "total_packets" : "total_packets",
            "total_bytes" : "total_bytes",
            "green_yellow_packets" : "green_yellow_packets",
            "green_yellow_bytes" : "green_yellow_bytes",
            "red_packets" : "red_packets",
            "red_bytes" : "red_bytes"
          },
          "ofses" : [
            {
              "dp_id" : "dp_id",
              "policer" : {
                "id" : "id",
                "vexternal" : "vexternal",
                "port" : "port",
                "vlan_id" : "vlan_id",
                "status" : "status"
              },
              "statistics" : {
                "total_packets" : "total_packets",
                "total_bytes" : "total_bytes",
                "green_yellow_packets" : "green_yellow_packets",
                "green_yellow_bytes" : "green_yellow_bytes",
                "red_packets" : "red_packets",
                "red_bytes" : "red_bytes"
              }
            }
          ]
        }
      ]
    }
  }
}
```

Table 8-100 Description of Elements in profile

Element	Description	Return Value
prf_name	Policing profile name	Up to 31 characters including one-byte alphanumeric characters and underscores
profileentries	Policing profile entry list	profileentries

Table 8-101 Description of Elements in profileentries

Element	Description	Return Value
profileentry	Policing profile entry	profileentry

Table 8-102 Description of Elements in profileentry

Element	Description	Return Value
seqnum	Policy sequence number	Hexadecimal number (1 to 255)
tworatethreecolor	2rate3color policing	tworatethreecolor
statistics	Packet and byte information	statistics type
ofses	List of OFSs	ofses

Table 8-103 Description of Elements in tworatethreecolor

Element	Description	Return Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction
redaction	Red action (for display)	redaction

Table 8-104 Description of Elements in meter

Element	Description	Return Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-105 Description of Elements in greenaction

Element	Description	Return Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-106 Description of Elements in yellowaction

Element	Description	Return Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-107 Description of Elements in redaction

Element	Description	Return Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-108 Description of Elements in statistics type

Element	Description	Return Value
total_packets	Total number of packets	Counter
total_bytes	Total number of bytes	Counter
green_yellow_packets	Total number of packets which green or yellow actions were applied to	Counter
green_yellow_bytes	Total number of bytes which green or yellow actions were applied to	Counter
red_packets	Total number of packets which red actions were applied to	Counter
red_bytes	Total number of bytes which red actions were applied to	Counter

Table 8-109 Description of Elements in ofses

Element	Description	Return Value
ofs	OFS	ofs

Table 8-110 Description of Elements in ofs

Element	Description	Return Value
dp_id	DPID of the OFS registered as the policer	HHHH-HHHH-HHHH-HHHH format (H: Hexadecimal number). *It is not allowed to specify F's to all digits.
policer	Policer information	policer
statistics	Packet and byte information	statistics type

Table 8-111 Description of Elements in policer

Element	Description	Return Value
id	ID of policer registered in the OFS	Decimal number (1 to 4294967295). *"N/A" is displayed if **status** is "down."
vexternal	vExternal name related to policer registration	Up to 31 characters including one-byte alphanumeric characters and underscores
port	Physical port which the policer settings are mapped to	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
status	Setting status of policer registration	{active|down(detail)} (Lowercase) *The reason for "down" is displayed in detail.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.6 Operate Policing Setting Information (vExternal Mode)

8.6.1 Show policing profile on vExternal

Processing request

Method

GET

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/policing.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/policing.json`
- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `seqnum=`**`seqnum`**

Table 8-112 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
seqnum	Policy sequence number	Decimal number (1 to 255)

Settings of request body

None.

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<policing>
  <profile prf_name="prf_name">
    <profileentries>
      [
      <profileentry seqnum="seqnum">
        <tworatethreecolor>
          <meter rateunit="rateunit"
            cir="cir"
            cbs="cbs"
            pir="pir"
            pbs="pbs"/>
          <greenaction green_action="green_action"
            ga_priority="ga_priority"
            ga_dscp="ga_dscp"
            ga_drop_precedence="ga_drop_precedence"/>
          <yellowaction yellow_action="yellow_action"
            ya_priority="ya_priority"
            ya_dscp="ya_dscp"
            ya_drop_precedence="ya_drop_precedence"/>
          <redaction red_action="red_action"
            ra_priority="ra_priority"
            ra_dscp="ra_dscp"
            ra_drop_precedence="ra_drop_precedence"/>
        </tworatethreecolor>
        <ofses>
          [
          <ofs dp_id="dp_id">
            <policer id="id"
              vexternal="vexternal"
              port="port"
              vlan_id="vlan_id"
              status="status"/>
          </ofs>
          ]
        </ofses>
      </profileentry>
      ]
    </profileentries>
  </profile>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name",
      "profileentries" : [
        {
          "seqnum" : "seqnum",
          "tworatethreecolor" : {
            "meter" : {
              "rateunit" : "rateunit",
              "cir" : "cir",
              "cbs" : "cbs",
              "pir" : "pir",
              "pbs" : "pbs"
            },
```

```
            "greenaction" : {
              "green_action" : "green_action",
              "ga_priority" : "ga_priority",
              "ga_dscp" : "ga_dscp",
              "ga_drop_precedence" : "ga_drop_precedence"
            },
            "yellowaction" : {
              "yellow_action" : "yellow_action",
              "ya_priority" : "ya_priority",
              "ya_dscp" : "ya_dscp",
              "ya_drop_precedence" : "ya_drop_precedence"
            },
            "redaction" : {
              "red_action" : "red_action",
              "ra_priority" : "ra_priority",
              "ra_dscp" : "ra_dscp",
              "ra_drop_precedence" : "ra_drop_precedence"
            }
          },
          "ofses" : [
            {
              "dp_id" : "dp_id",
              "policer" : {
                "id" : "id",
                "vexternal" : "vexternal",
                "port" : "port",
                "vlan_id" : "vlan_id",
                "status" : "status"
              }
            }
          ]
        }
      ]
    }
  }
}
```

Table 8-113 Description of Elements in profile

Element	Description	Return Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores
profileentries	Policing profile entry list	profileentries

Table 8-114 Description of Elements in profileentries

Element	Description	Return Value
profileentry	Policing profile entry	profileentry

Table 8-115 Description of Elements in profileentry

Element	Description	Return Value
seqnum?	Policy sequence number	Decimal number (1 to 255)
tworatethreecolor	2rate3color policing	tworatethreecolor
ofses	List of OFSs	ofses

Table 8-116 Description of Elements in tworatethreecolor

Element	Description	Return Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction

Element	Description	Return Value
redaction	Red action (for display)	redaction

Table 8-117 Description of Elements in meter

Element	Description	Return Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-118 Description of Elements in greenaction

Element	Description	Return Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-119 Description of Elements in yellowaction

Element	Description	Return Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-120 Description of Elements in redaction

Element	Description	Return Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-121 Description of Elements in ofses

Element	Description	Return Value
ofs	OFS	ofs

Table 8-122 Description of Elements in ofs

Element	Description	Return Value
dp_id	DPID of the OFS registered as the policer	HHHH-HHHH-HHHH-HHHH format (H: Hexadecimal number). *It is not allowed to specify F's to all digits.

Element	Description	Return Value
policer	Policer information	policer

Table 8-123 Description of Elements in policer

Element	Description	Return Value
id	ID of policer registered in the OFS	Decimal number (1 to 4294967295). *"N/A" is displayed if **status** is "down."
vexternal	vExternal name related to policer registration	Up to 31 characters including one-byte alphanumeric characters and underscores
port	Physical port which the policer settings are mapped to	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
status	Setting status of policer registration	{active|down(detail)} (Lowercase) *The reason for "down" is displayed in detail.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.6.2 Map policing profile to vExternal

Processing request

Method

PUT

request URI

- XML format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/policing.xml`
- JSON format

  `/vtns/`**`vtn_name`**`/vexternals/`**`vex_name`**`/policing.json`

Table 8-124 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

- XML format

```
<policing>
  <profile prf_name="prf_name"/>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name"
    }
  }
}
```

Table 8-125 Description of Elements in profile

Element	Description	Valid Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Map policing profile **prf_name** to VTN **vtn_name** vExternal **vex_name** succeeded."

- On failure

```
"Map policing profile prf_name of vExternal vex_name of VTN vtn_name
failed. [Invalid argument(prf_name)](<Error information from pfcshel
l>)"
```

Remark

None

8.6.3 Unmap policing profile from vExternal

Processing request

Method

DELETE

request URI

- XML format

  `/vtns/vtn_name/vexternals/vex_name/policing.xml`

- JSON format

  `/vtns/vtn_name/vexternals/vex_name/policing.json`

Table 8-126 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores

Settings of request body

None

Processing result

Details of response body

None

HTTP status code for response

- On success

  200 (OK)

- On failure

  400,500

Miscellaneous

Log

- On success

  "Unmap policing profile from VTN **vtn_name** vExternal **vex_name** succeeded."

- On failure

  "Unmap policing profile from VTN **vtn_name** vExternal **vex_name** failed. (<Error information from pfcshell>)"

Remark

None

8.6.4 Show policing profile detail on vExternal

Processing request

Method

GET

request URI

- XML format

  /vtns/**vtn_name**/vexternals/**vex_name**/policing/detail.xml

- JSON format

  /vtns/**vtn_name**/vexternals/**vex_name**/policing/detail.json

- Parameter

  The following parameters are passed in the URI query character string (in the format of ?param1=***&param2=***).

  seqnum=**seqnum**

Table 8-127 Description of Parameter in URI

Element	Description	Valid Value
vtn_name	VTN name	Up to 31 characters including one-byte alphanumeric characters and underscores
vex_name	vExternal name	Up to 31 characters including one-byte alphanumeric characters and underscores
seqnum	Policy sequence number	Decimal number (1 to 255)

Settings of request body

None

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<policing>
  <profile prf_name="prf_name">
    <profileentries>
      [
      <profileentry seqnum="seqnum">
        <tworatethreecolor>
          <meter rateunit="rateunit"
            cir="cir"
            cbs="cbs"
            pir="pir"
            pbs="pbs"/>
          <greenaction green_action="green_action"
            ga_priority="ga_priority"
            ga_dscp="ga_dscp"
            ga_drop_precedence="ga_drop_precedence"/>
          <yellowaction yellow_action="yellow_action"
            ya_priority="ya_priority"
            ya_dscp="ya_dscp"
            ya_drop_precedence="ya_drop_precedence"/>
          <redaction red_action="red_action"
            ra_priority="ra_priority"
            ra_dscp="ra_dscp"
            ra_drop_precedence="ra_drop_precedence"/>
        </tworatethreecolor>
        <statistics total_packets="total_packets"
          total_bytes="total_bytes"
          green_yellow_packets="green_yellow_packets"
          green_yellow_bytes="green_yellow_bytes"
          red_packets="red_packets"
          red_bytes="red_bytes"/>
        <ofses>
          [
          <ofs dp_id="dp_id">
            <policer id="id"
              vexternal="vexternal"
              port="port"
              vlan_id="vlan_id"
              status="status"/>
            <statistics total_packets="total_packets"
              total_bytes="total_bytes"
              green_yellow_packets="green_yellow_packets"
              green_yellow_bytes="green_yellow_bytes"
              red_packets="red_packets"
              red_bytes="red_bytes"/>
          </ofs>
          ]
        </ofses>
      </profileentry>
      ]
    </profileentries>
  </profile>
</policing>
```

- JSON format

```
{
  "policing" : {
    "profile" : {
      "prf_name" : "prf_name",
      "profileentries" : [
        {
          "seqnum" : "seqnum",
          "tworatethreecolor" : {
```

```
            "meter" : {
              "rateunit" : "rateunit",
              "cir" : "cir",
              "cbs" : "cbs",
              "pir" : "pir",
              "pbs" : "pbs"
            },
            "greenaction" : {
              "green_action" : "green_action",
              "ga_priority" : "ga_priority",
              "ga_dscp" : "ga_dscp",
              "ga_drop_precedence" : "ga_drop_precedence"
            },
            "yellowaction" : {
              "yellow_action" : "yellow_action",
              "ya_priority" : "ya_priority",
              "ya_dscp" : "ya_dscp",
              "ya_drop_precedence" : "ya_drop_precedence"
            },
            "redaction" : {
              "red_action" : "red_action",
              "ra_priority" : "ra_priority",
              "ra_dscp" : "ra_dscp",
              "ra_drop_precedence" : "ra_drop_precedence"
            }
          },
          "statistics" : {
            "total_packets" : "total_packets",
            "total_bytes" : "total_bytes",
            "green_yellow_packets" : "green_yellow_packets",
            "green_yellow_bytes" : "green_yellow_bytes",
            "red_packets" : "red_packets",
            "red_bytes" : "red_bytes"
          },
          "ofses" : [
            {
              "dp_id" : "dp_id",
              "policer" : {
                "id" : "id",
                "vexternal" : "vexternal",
                "port" : "port",
                "vlan_id" : "vlan_id",
                "status" : "status"
              },
              "statistics" : {
                "total_packets" : "total_packets",
                "total_bytes" : "total_bytes",
                "green_yellow_packets" : "green_yellow_packets",
                "green_yellow_bytes" : "green_yellow_bytes",
                "red_packets" : "red_packets",
                "red_bytes" : "red_bytes"
              }
            }
          ]
        }
      ]
    }
  }
}
```

Table 8-128 Description of Elements in profile

Element	Description	Return Value
prf_name	Policing profile name	Up to 32 characters including one-byte alphanumeric characters and underscores
profileentries	Policing profile entry list	profileentries

Table 8-129 Description of Elements in profileentries

Element	Description	Return Value
profileentry	Policing profile entry	profileentry

Table 8-130 Description of Elements in profileentry

Element	Description	Return Value
seqnum	Policy sequence number	Decimal number (1 to 255)
tworatethreecolor	2rate3color policing	tworatethreecolor
statistics	Packet and byte information	statistics type
ofses	List of OFSs	ofses

Table 8-131 Description of Elements in tworatethreecolor

Element	Description	Return Value
meter	Rate and burst size meter information	meter
greenaction	Green action (for display)	greenaction
yellowaction	Yellow action (for display)	yellowaction
redaction	Red action (for display)	redaction

Table 8-132 Description of Elements in meter

Element	Description	Return Value
rateunit	Rate unit	{kbps|pps} (Lowercase)
cir	The lower rate of two rate types	Decimal number (1 to 4294967295)
cbs	Burst size for the lower rate of two rate types	Decimal number (1 to 4294967295)
pir	The higher rate of two rate types	Decimal number (1 to 4294967295)
pbs	Burst size for the higher rate of two rate types	Decimal number (1 to 4294967295)

Table 8-133 Description of Elements in greenaction

Element	Description	Return Value
green_action	Action	{pass|drop|penalty} (Lowercase)
ga_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ga_dscp	DSCP value	Decimal number (0 to 63)
ga_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-134 Description of Elements in yellowaction

Element	Description	Return Value
yellow_action	Action	{pass|drop|penalty} (Lowercase)
ya_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ya_dscp	DSCP value	Decimal number (0 to 63)
ya_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-135 Description of Elements in redaction

Element	Description	Return Value
red_action	Action	{pass|drop|penalty} (Lowercase)
ra_priority	Traffic priority within the OpenFlow network	Decimal number (0 to 7)
ra_dscp	DSCP value	Decimal number (0 to 63)
ra_drop_precedence	Discarding priority	Decimal number (1 to 3)

Table 8-136 Description of Elements in statistics type

Element	Description	Return Value
total_packets	Total number of packets	Counter
total_bytes	Total number of bytes	Counter
green_yellow_packets	Total number of packets which green or yellow actions were applied to	Counter
green_yellow_bytes	Total number of bytes which green or yellow actions were applied to	Counter
red_packets	Total number of packets which red actions were applied to	Counter
red_bytes	Total number of bytes which red actions were applied to	Counter

Table 8-137 Description of Elements in ofses

Element	Description	Return Value
ofs	OFS	ofs

Table 8-138 Description of Elements in ofs

Element	Description	Return Value
dp_id	DPID of the OFS registered as the policer	HHHH-HHHH-HHHH-HHHH format (H: Hexadecimal number). *It is not allowed to specify F's to all digits.
policer	Policer information	policer
statistics	Packet and byte information	statistics type

Table 8-139 Description of Elements in policer

Element	Description	Return Value
id	ID of policer registered in the OFS	Decimal number (1 to 4294967295). *"N/A" is displayed if **status** is "down."
vexternal	vExternal name related to policer registration	Up to 31 characters including one-byte alphanumeric characters and underscores
port	Physical port which the policer settings are mapped to	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
vlan_id	VLAN ID	Decimal number (1 to 4095)
status	Setting status of policer registration	{active|down(detail)} (Lowercase) *The reason for "down" is displayed in detail.

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

8.7 Operate OFS QoS Flow Entry Information

8.7.1 Show QoS flow entries on OFSs

Processing request

Method

GET

request URI

- XML format

  `/realnetwork/qosflowentries.xml`
- JSON format

  `/realnetwork/qosflowentries.json`
- Parameter

  The following parameters are passed in the `URI` query character string (in the format of `?param1=***&param2=***`).

  `dp_id=`**`dp_id`**

Table 8-140 Description of Parameter in URI

Element	Description	Valid Value
dp_id	DPID of OFS	HHHH-HHHH-HHHH-HHHH format (H: Hexadecimal number). *It is not allowed to specify F's to all digits.

Settings of request body

None.

Processing result

Details of response body

- XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<qosflowentry qosflowentry_count="qosflowentry_count">
  <ofses>
    [
    <ofs dp_id="dp_id" status="status">
      <match inport="inport"
        srcmac="srcmac"
        dstmac="dstmac"
        vlan_id="vlan_id"
        vlanpriority="vlanpriority"
        ethernettype="ethernettype"
        srcip="srcip"
        dstip="dstip"
        iptos="iptos"
        ipprotocol="ipprotocol"
        l4srcport_icmptype="l4srcport_icmptype"
        l4dstport_icmpcode="l4dstport_icmpcode"/>
      <action policerid="policerid"/>
    </ofs>
    ]
  </ofses>
</qosflowentry>
```

- JSON format

```
{
  "qosflowentry" : {
    "qosflowentry_count" : "qosflowentiry_count",
    "ofses" : [
      {
        "dp_id" : "dp_id",
        "status" : "status",
        "match" : {
          "inport" : "inport",
          "srcmac" : "srcmac",
          "dstmac" : "dstmac",
          "vlan_id" : "vlan_id",
          "vlanpriority" : "vlanpriority",
          "ethernettype" : "ethernettype",
          "srcip" : "srcip",
          "dstip" : "dstip",
          "iptos" : "iptos",
          "ipprotocol" : "ipprotocol",
          "l4srcport_icmptype" : "l4srcport_icmptype",
          "l4dstport_icmpcode" : "l4dstport_icmpcode"
        },
        "action" : {
          "policerid" : "policerid"
        }
      }
    ]
  }
}
```

Table 8-141 Description of Elements in qosflowentry

Element	Description	Return Value
qosflowentry_count	Total number of displayed flow entries	Decimal number (1 to 20000)
ofses	OFS information (multiple)	ofses

Table 8-142 Description of Elements in ofses

Element	Description	Return Value
ofs	OFS information	ofs

Table 8-143 Description of Elements in ofs

Element	Description	Return Value
dp_id	DPID of OFS	HHHH-HHHH-HHHH-HHHH format (H: Hexadecimal number). *It is not allowed to specify F's to all digits.
status	Setting status of flow entry for QoS control	{active|down(detail)} (Lowercase) *The reason for "down" is displayed in detail.
match	Flow entry matching condition information	match
action	Flow entry action information	action

Table 8-144 Description of Elements in match

Element	Description	Return Value
inport	Received OFS port	Up to 16 characters including ascii alphanumeric characters except for a question mark (?)
srcmac	Source MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
dstmac	Destination MAC address	hhhh.hhhh.hhhh format (h: hexadecimal number)
vlan_id	VLAN ID	Decimal number (1 to 4095)
vlanpriority	VLAN PRIORITY	Decimal number (0 to 7)
ethernettype	Ether type	Hexadecimal format including "0x"
srcip	Source IP address	IPAddress/Prefix IPAddress: IPv4 dot-separated format (Example: 192.168.1.1) Prefix: 1 to 32
dstip	Destination IP address	IPAddress/Prefix IPAddress: IPv4 dot-separated format (Example: 192.168.1.1) Prefix: 1 to 32
iptos	ToS value	Hexadecimal format including "0x"
ipprotocol	L3 protocol type	Decimal number (1 to 255)
l4srcport_icmptype	Source port number or ICMP type	Source port number or ICMP type (10x)(0x(16x) / Wildcard for source port number (10x)(0x(16x)
l4dstport_icmpcode	Destination port number or ICMP code	Destination port number or ICMP code(10x)(0x(16x) / wildcard (10x)(0x(16x) of destination port number

Table 8-145 Description of Elements in action

Element	Description	Valid Value
policerid	Policier ID that the flow entry referrs to	Decimal number (1 to 4294967295)

HTTP status code for response

- On success

  200 (OK)
- On failure

  400,500

Miscellaneous

Log

None

Remark

None

PF6800 Ver. 6.0
WebAPI User's Guide

PFC00EN0600-01

July, 2014 First Edition

NEC Corporation