SDK Diag Shell

RTL8690x

NOT FOR PUBLIC RELEASE

User Guide

(CONFIDENTIAL: Development Partners Only)

Rev. 0.0.1
30 Nov 2012

Realtek Semiconductor Corp.

No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan

Tel.: +886-3-578-0211 Fax: +886-3-577-6047

www.realtek.com

COPYRIGHT

TRADEMARKS

Realtek is a trademark of Realtek Semiconductor Corporation. Other names mentioned in this document are trademarks/registered trademarks of their respective owners.

DISCLAIMER

Realtek provides this document “as is”, without warranty of any kind, neither expressed nor implied, including, but not limited to, the particular purpose. Realtek may make improvements and/or changes in this document or in the product described in this document at any time. This document could include technical inaccuracies or typographical errors.

USING THIS DOCUMENT

This document is intended for use by the system engineer when integrating with Realtek Switch Software SDK. Though every effort has been made to assure that this document is current and accurate, more information may have become available subsequent to the production of this guide. In that event, please contact your Realtek representative for additional information that may help in the development process.

REVISION HISTORY

Revision	Release Date	Summary
0.0.1	201211-30	Initial version for 8690x chip

Table of Contents

Chapter 1 Introduction

This document provides a summary of the commands supported on the REALTEK SDK diag shell Command Line Interface(CLI). The document is divided into two sections. Section 1 describes the convention, argument types, privileges and command modes, command help, and command line editing. Section 2 gives a brief usage and description of all commands included in the diag shell.

1.1 Conventions

The following conventions are used in the command syntax throughout this document:

- Vertical bars (|) separate alternative, mutually exclusive arguments.
- Square brackets ([ ]) indicate optional elements.
- Braces (< >) indicate a required choice.
- Braces within square brackets ([< >]) indicate required choices within optional elements.
- **Bold** indicates commands and keywords.
- *Italics* indicate user variables.

1.2 Argument Types

The following argument types are recognized by the CLI and are used in the command syntax throughout this document:

- *MACADDR* – MAC address.
  Example: 00:3B:51:A9:CC:07.
- *PORT_LIST* – Port list, separated by “,” or “-”.
  Example: 1-3,6,8.
- *MASK_LIST* – mask list, separated by “,” or “-”.
  Example: 1-3,6,8.
- *IPV4ADDR* – IPv4 address.
  Example: 192.168.1.100.
- *IPV6ADDR* – IPv6 address.
  Example:101:234:689:ACD:151:03B:1A9:C07.
- *UINT* – unsigned integer.
  Example: 123.

1.3 Command Help

You may enter **?** at any command mode, and the CLI will return possible commands at that point, along with some description of the keywords:

```
RTK.0> vlan ?
create - create VLAN
destroy - destroy VLAN
set - Set configuration
get - get configuration
```

You may use the <Tab> key to do keyword auto completion:

```
RTK.0> vlan set p<Tab>
pvid - config port based vid
protocol-vlan – protocol
```

You do not need to type in the entire commands; you only need to type in enough characters for the CLI to recognize the command as unique.

Chapter 2 Commands

2.1. VLAN commands

vlan init

Description:

This command is used to initialize & reset VLAN module. By using this command, the following configuration will be appilied.
All VLAN will be deleted.
VLAN 1 will be created and all ports are untagged member.
The PVID of all ports are 1.

Syntax:

vlan init

Parameter:

None

Examples:

This example shows how to initialize VLAN.

```
RTK.0> vlan init
```

vlan create

Description:

Create a new VLAN

Syntax:

vlan create vlan-table vid *vid*

Parameter:

vid – specify the numeric VLAN identifier

Examples:

This example shows how to create a new VLAN which ID is 100:

```
RTK.0> vlan create vlan-table vid 100
```

vlan destroy

Description:

Destroy a VLAN

Syntax:

vlan destroy vlan-table vid *vid*
vlan destroy vlan-table all [restore-default-vlan]
vlan destroy vlan-table all untag
vlan destroy entry all
vlan destroy entry *index*

Parameter:

vid	– specify the numeric VLAN identifier
vlan-table	– vlan table
all	- all vlan
restore-default-vlan	– destroy all VLAN identifier except for default VLAN
Untag	- all port to untag member
index	- vlan member configuration index
entry	- vlan member configuration

Examples:

This example shows how to destroy VLAN 100:

```
RTK.0> vlan destroy vlan-table vid 100
```

vlan set vlan-table

Description:

This command can configure the 4k vlan entry.

Syntax:

vlan set vlan-table vid *vid* **member <***PORT_LIST: ports* **| all | none>**

vlan set vlan-table vid *vid* **tag-member <***PORT_LIST: ports* **| all | none>**

vlan set vlan-table vid *vid* **untag-member <***PORT_LIST: ports* **| all | none>**

vlan set vlan-table vid *vid* **ext-member <***PORT_LIST: ext* **| all | none>**

vlan set vlan-table vid *vid* **fid-msti** *fid_msti*

vlan set vlan-table vid *vid* **hash-mode <ivl | svl>**

vlan set vlan-table vid *vid* **vlan-based-policing state <disable | enable>**

vlan set vlan-table vid *vid* **meter** *meter*

vlan set vlan-table vid *vid* **vlan-based-priority state <disable | enable>**

vlan set vlan-table vid *vid* **vlan-based-priority priority** *priority*

Parameter:

vid	– specify the numeric VLAN identifier
ports	- specified port list
all	- specify all ports
none	- specify no port
member	- member configuration
tag-member	- tagging member configuration
untag-member	- untag set configuration
ext	- extention member ports
fid_msti	- the filter-id or msti for this vlan
ivl	- specify the hash key for this vlan lookup is using VID
svl	- specify the hash key for this vlan lookup is using MSTI/FID
disable	- disable
enable	- enable

meter - specify the meter index
priority – specify the priority value for this vlan

Examples:

This example shows how to congigurate the VLAN which ID is 100:
Add port 0,1,2 to vlan 100
Assign port 0,1 to tag member
Enable vlan based priority and set priority to 6
Enable vlan based policing and set meter index to 10

```
RTK.0> vlan create vlan-table vid 100
RTK.0> vlan set vlan-table vid 100
RTK.0> vlan set vlan-table vid 100 member 0-2
RTK.0> vlan set vlan-table vid 100 tag-member 0-1
RTK.0> vlan set vlan-table vid 100 vlan-based-priority state
enable
RTK.0> vlan set vlan-table vid 100 vlan-based-priority priority
6
RTK.0> vlan set vlan-table vid 100 vlan-based-policing state
enable
RTK.0> vlan set vlan-table vid 100 meter 10
```

vlan get vlan-table

Description:

Show VLAN table configuration of a specific vid

Syntax:

vlan get vlan-table vid *vid*

Parameter:

vid – specify the numeric VLAN identifier

Examples:

This example shows how to get the VLAN table configuration of vid 100:

```
RTK.0> vlan get vlan-table vid 100
Vid:100
Hash Policing Meter VlanPriEn Priority FID
SVL  En       10    En        6        0
member port          : 0-2
tag member port      : 0-1
extention member port: none
```

vlan set entry

Description:

This command can configure the 32 vlan member configuration.

Syntax:

vlan set entry *index* **enhanced-vid** *vid*
vlan set entry *index* **member <***PORT_LIST: ports* **| all | none>**
vlan set entry *index* **ext-member <***PORT_LIST: ext* **| all | none>**
vlan set entry *index* **fid-msti** *fid_msti*
vlan set entry *index* **hash-mode <ivl | svl>**
vlan set entry *index* **vlan-based-policing state <disable | enable>**
vlan set entry *index* **meter** *meter*
vlan set ventry *index* **vlan-based-priority state <disable | enable>**
vlan set entry *index* **vlan-based-priority priority** *priority*

Parameter:

vid	– specify the numeric VLAN identifier
enhanced-vid	- enhanced VLAN id
index	- vlan member configuration index
ports	- the vlan member or tag member ports
all	- specify all ports
none	- specify no port
ext	- extention member ports
fid_msti	- the filter-id or msti for this vlan
ivl	- specify the hash key for this vlan lookup is using VID
svl	- specify the hash key for this vlan lookup is using MSTI/FID
disable	- disable
enable	- enable
meter	- specify the meter index for this vlan based policing
priority	– specify the priority value for this vlan

Examples:

This example shows how to congigurate VLAN member configuration index 0 which ID is 100:

```
Add port 0,1,2 to vlan 100
Enable vlan based priority and set priority to 6
Enable vlan based policing and set meter index to 10

RTK.0> vlan set entry 0 enhanced-vid 100
RTK.0> vlan set entry 0 member 0-2
RTK.0> vlan set entry 0 vlan-based-priority state enable
RTK.0> vlan set entry 0 vlan-based-priority priority 6
RTK.0> vlan set entry 0 vlan-based-policing state enable
RTK.0> vlan set entry 0 meter 10
```

vlan get entry

Description:

Show VLAN member configuration of a specific index

Syntax:

vlan get entry *index*

Parameter:

index - vlan member configuration index

Examples:

Show VLAN member configuration index 0:

```
RTK.0> vlan get entry 0
vlan id:0
Evid PlyEn MtrIdx PriEn pri Fid
 100   En    10   En   6   0
member port          : 0-2
extention member port: 0-5
```

vlan set accept-frame-type

Description:

Configure accepted frame type for ingress frame per port

Syntax:

vlan set accept-frame-type port <*PORT_LIST:ports* | all> <all | tag-only | untag-only | priority-tag-and-tag>

Parameter:

ports	– specify the port list for apply this setting
all	- specify all ports
all	- accept all type frame
tag-only	- accept tag only frame
untag-only	- accept untag only frame
priority-tag-and-tag	- accept priority tag and 1q-tag frame

Examples:

This example shows how to accept all untag frame only for port 1 to 3:

```
RTK.0> vlan set accept-frame-type port 1-3 untag-only
```

vlan get accept-frame-type

Description:

Show accepted frame type for each port

Syntax:

vlan get accept-frame-type port <*PORT_LIST:ports* **| all>**

Parameter:

ports	– specify the port list for apply this setting
all	- specify all ports

Examples:

This example shows how to get accept frame types for all ports

```
RTK.0> vlan get accept-frame-type port 0-6
Port Accept Frame Type
0    all
1    untagged-only
2    untagged-only
3    untagged-only
4    all
5    all
6    all
```

vlan set egress

Description:

Show per egress port setting for VLAN mode real keep reference by ingress port

Syntax:

vlan set egress port <*PORT_LIST:egr_ports* **| all> keep-tag ingress-port <***PORT_LIST:igr_ports* **| all> state <enable | disable>**

Parameter:

egr_ports	– specify egress port list for apply this setting
all	– specify all ports
igr_ports	– The ingress port list to apply egress-keep setting
keep-tag	- keeping tag
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to enable the egress keep for port 1 the ingress packet is from port 2-5:

```
RTK.0> vlan set egress port 1 keep-tag ingress-port 2-5 state enable
```

vlan get egress

Description:

Show per egress port setting for VLAN mode real keep reference by ingress port

Syntax:

vlan get egress port <*PORT_LIST:egr_ports* **| all> keep-tag ingress-port <***PORT_LIST:igr_ports* **| all> state**

Parameter:

egr_ports	– specify egress port list for apply this setting
all	– specify all ports
igr_ports	– The ingress port list to apply egress-keep setting
keep-tag	- keeping tag
state	- state configuration

Examples:

This example shows how to get the egress keep setting for port 1

```
RTK.0> vlan get egress port 1 keep-tag ingress-port 0-6 state
Port P0      P1      P2      P3      P4      P5      P6
1    Disable Disable Enable  Enable  Enable  Enable  Disable
```

vlan set ingress-filter

Description:

Configure per-port VLAN ingress check.

Syntax:

vlan set ingress-filter port <*PORT_LIST:igr_ports* **| all> state <enable | disable>**

Parameter:

ports	- specified port list
all	– specify all ports
enable	- enable the vlan ingress filter function
disable	- disable the vlan ingress filter function

Examples:

This example shows how to enable the ingress filter for port 1

```
RTK.0> vlan set ingress-filter port 1 state enable
```

vlan get ingress-filter

Description:

Show per port ingress filter setting

Syntax:

vlan get ingress-filter port <*PORT_LIST:ports* **| all> state**

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get the ingress filter setting for port 0-6:

```
RTK.0> vlan get ingress-filter port 0-6 state
Port Ingress-filter
0    Enable
1    Enable
2    Enable
3    Enable
4    Enable
5    Enable
6    Enable
```

vlan set tag-mode

Description:

Configure egress port vlan tag mode.

Syntax:

vlan set tag-mode port <*PORT_LIST: ports* **| all> <original | keep-format | priority-tag>**

Parameter:

ports	- specified port list
all	– specify all ports
original	- Original mode. Output frame will follow VLAN untag setting
keep-format	- Keep format mode. Output frame will keep VLAN original format.(If packet not modified asic will not re-caculate CRC)
priority-tag	- Priority tag mode. Output frame will be priority tag.

Examples:

This example shows how to set egress tag mode for port 1 set to priority tag mode. Set port 3 to keep-format mode.

```
RTK.0> vlan set tag-mode port 1 priority-tag
RTK.0> vlan set tag-mode port 3 keep-format
```

vlan get tag-mode

Description:

Get per port egress tag mode.

Syntax:

vlan get tag-mode port <*PORT_LIST: ports* **| all>**

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get the egress tag mode for port 1-5

```
RTK.0> vlan get tag-mode port 1-5
Port Mode
1    priority-tag
2    original
3    keep-format
4    original
5    original
```

vlan set vlan-treat vid

Description:

Configure ingress packet with special vlan VID(0 and 4095) that will be treat as tagging pacet or un-tagging packet.

Syntax:

vlan set vlan-treat vid <0 | 4095> type <tagging | un-tagging>

Parameter:

0	– specify the ingress packet with vid 0
4095	– specify the ingress packet with vid 4095
type	- vlan treat type
tagging	- packet will be treat as tagging frame
un-tagging	- packet will be treat as un-tagging frame

Examples:

This example shows how to set ingress packet with vid 0 will be treat as tagging packet and packet with vid 4095 will be treat as un-tagging packet.

```
RTK.0> vlan set vlan-treat vid 0 type tagging
RTK.0> vlan set vlan-treat vid 4095 type un-tagging
```

vlan get vlan-treat vid

Description:

Get the setting of ingress packet with special vlan VID(0 and 4095) that will be treat as tagging pacet or un-tagging packet.

Syntax:

vlan get vlan-treat vid <0 | 4095> type

Parameter:

0	– specify the ingress packet with vid 0
4095	– specify the ingress packet with vid 4095
type	- vlan treat type

Examples:

This example shows how to get ingress packet with vid 0 and 4095 will be treat as tagging packet or un-tagging packet.

```
RTK.0> vlan get vlan-treat vid 0
vlan 0 treat Tagging
RTK.0> vlan get vlan-treat vid 4095
vlan 4095 treat Untagging
```

vlan set cfi-keep

Description:

Configure egress CFI value will keep ingress packet CFI value or set CFI to 0.

Syntax:

vlan set cfi-keep <cfi-to-0 | keep-cfi>

Parameter:

cfi-to-0 – egress packet cfi always set to 0
keep-cfi – keep ingress tag cfi value to egress tag

Examples:

This example shows how to set egress packet CFI always be 0.

```
RTK.0> vlan set cfi-keep cfi-to-0
```

vlan get cfi-keep

Description:

Get egress CFI keep mode.

Syntax:

vlan get cfi-keep

Parameter:

None

Examples:

This example shows how to get egress CFI mode.

```
RTK.0> vlan get cfi-keep
cfi-keep :Disable
```

vlan set transparent

Description:

Set vlan transparent mode. When transparent enable it will discard the “vlan set egress-keep port” setting.

Syntax:

vlan set transparent state <enable | disable>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to enable vlan transparent.

```
RTK.0> vlan set transparent state enable
```

vlan get transparent

Description:

Get vlan transparent mode.

Syntax:

vlan get transparent state

Parameter:

state - state configuration

Examples:

This example shows how to get vlan transparent state.

```
RTK.0> vlan get transparent state
vlan transparent state: Enable
```

vlan set ingress-filter

Description:

Set per port vlan ingress filter state.

Syntax:

vlan set ingress-filter port <*PORT_LIST:ports* **| all> state <enable | disable>**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to enable vlan ingress-filter for port 1-3.

```
RTK.0> vlan set ingress-filter port 1-3 state enable
```

vlan get ingress-filter

Description:

Get per port vlan ingress filter state.

Syntax:

vlan get ingress-filter port <*PORT_LIST:ports* | all> state

Parameter:

ports	– The port list
all	– specify all ports
state	- state configuration

Examples:

This example shows how to get vlan ingress-filter state for port 0-6.

```
RTK.0> vlan get ingress-filter port 0-6 state
Port Ingress-filter
0    Enable
1    Enable
2    Enable
3    Enable
4    Enable
5    Enable
6    Enable
```

vlan set state

Description:

Set system vlan filter state.

Syntax:

vlan set state <enable | disable>

Parameter:

disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to get system vlan filter state.

```
RTK.0> vlan set state enable
```

vlan get state

Description:

Get system vlan filter state.

Syntax:

vlan get state

Parameter:

None

Examples:

This example shows how to get system vlan filter state.

```
RTK.0> vlan get state
vlan state: Enable
```

vlan set pvid

Description:

Set port based VID. The index is index to vlan member configuration table.

Syntax:

vlan set pvid port <*PORT_LIST:ports* **| all>** *vid*
vlan set pvid port <*PORT_LIST:ports* **| all> vlan-index** *index*

Parameter:

ports	- specified port list
all	– specify all ports
index	- vlan member configuration index
vlan-idex	- vlan member configuration
vid	– specify the numeric VLAN identifier

Examples:

This example shows how to set port 5 port-base vlan to vlan 100.

```
RTK.0> vlan set pvid port 5 100
```

vlan get pvid

Description:

Get port based VID. The index is index to vlan member configuration table.

Syntax:

vlan get pvid port <*PORT_LIST:ports* **| all>**
vlan get pvid port <*PORT_LIST:ports* **| all> vlan-index**

Parameter:

ports	- specified port list
all	– specify all ports
vlan-idex	- vlan member configuration

Examples:

This example shows how to get port 5 port-base vlan setting.

```
RTK.0> vlan get pvid port 5

Port 5 PVID: 100
```

vlan set ext-pvid

Description:

Set port based VID for extention ports. The index is index to vlan member configuration table.

Syntax:

vlan set ext-pvid port <*PORT_LIST:ports* **| all>** *vid*

vlan set ext-pvid port <*PORT_LIST:ports* **| all> vlan-index** *index*

Parameter:

ports	- specified port list
all	– specify all ports
index	- vlan member configuration index
vlan-idex	- vlan member configuration
vid	– specify the numeric VLAN identifier

Examples:

This example shows how to set extention port 3 port-base vlan to vlan 100.

```
RTK.0> vlan set ext-pvid port 3 100
```

vlan get ext-pvid

Description:

Get port based VID for extention ports. The index is index to vlan member configuration table.

Syntax:

vlan get ext-pvid port <*PORT_LIST:ports* **| all>**
vlan get ext-pvid port <*PORT_LIST:ports* **| all> vlan-index**

Parameter:

ports	- specified port list
all	– specify all ports
vlan-idex	- vlan member configuration

Examples:

This example shows how to get extension port 3 port-base vlan setting.

```
RTK.0> vlan get ext-pvid port 3

 EXT Port 3 PVID: 100
```

vlan set protocol-vlan

Description:

Set protocol base vlan, for protocol group.

Syntax:

vlan set protocol-vlan group *index* **frame-type <ethernet | snap | llc-other>** *frame_type*

vlan set protocol-vlan port <*PORT_LIST:ports* **| all> group** *index* **vid** *vid* **priority** *priority*

vlan set protocol-vlan port <*PORT_LIST:ports* **| all> group** *index* **state <enable | disable>**

Parameter:

index	– index to protocol vlan group
ports	- specified port list
all	– specify all ports
ethernet	– specify protocol of this entry is ethernet
snap	– specify protocol of this entry is snap
llc-other	– specify protocol of this entry is llc-other
frame_type	- frame type value (example 0x8899)
vid	– specify the numeric VLAN identifier
priority	- priority configuration

Examples:

This example shows how to set protocol vlan group 0 the protocol is ehternet and the ether type of 8899.

```
RTK.0> vlan set protocol-vlan group 0 frame-type ethernet 0x8899
```

vlan get protocol-vlan

Description:

Get protocol base vlan setting.

Syntax:

vlan get protocol-vlan group *index*

vlan get protocol-vlan port <*PORT_LIST:ports* **| all>**

Parameter:

index	– index to protocol vlan group
ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get protocol vlan for group 0.

```
RTK.0> vlan get protocol-vlan group 0
Group  FRAME-TYPE  ETHER-TYPE
0      Ethernet    0x8899
```

vlan set port-based-fid

Description:

This command set per port based filter-id.

Syntax:

vlan set port-based-fid port <*PORT_LIST:ports* **| all> filter-id** *fid*

vlan set port-based-fid port <*PORT_LIST:ports* **| all> state <enable | disable>**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
fid	- filter-id

Examples:

This example shows how to set port 1 filter-id to 6.

```
RTK.0> vlan set port-based-fid port 1 filter-id 6
RTK.0> vlan set port-based-fid port 1 state enable
```

vlan get port-based-fid

Description:

This command get per port based filter-id.

Syntax:

vlan get port-based-fid port <*PORT_LIST:ports* **| all>**

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get port 1 filter-id setting.

```
RTK.0> vlan get port-based-fid port 1
Port  FID  State
1     6    Enable
```

vlan set port-based-pri

Description:

This command set port based priority.

Syntax:

vlan set port-based-pri port <*PORT_LIST:ports* **| all> priority** *priority*

Parameter:

ports	- specified port list
all	– specify all ports
priority	- priority configuration

Examples:

This example shows how to set port 1 priority to 6.

```
RTK.0> vlan set port-based-pri port 1 priority 6
```

vlan get port-based-pri

Description:

This command canget port based priority.

Syntax:

vlan get port-based-pri port <*PORT_LIST:ports* **| all>**

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get port 1 priority setting.

```
RTK.0> vlan get port-based-pri port 1
Port 1 VLAN based priority: 6
```

vlan set leaky

Description:

This command set VLAN leaky function

Syntax:

vlan set leaky ip-mcast port <*PORT_LIST:ports* **| all> state <enable | disable>**

vlan set leaky <cdp | csstp> state <enable | disable>

vlan set leaky rma *rma_tail* **state <enable | disable>**

Parameter:

ip-mcast	– ip mulitcast
ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
cdp	– cisco Discovery Protocol
csstp	– cisco Shared Spanning Tree Protocol
rma	- reserved multicast address
rma_tail	- tail of RMA MAC address

Examples:

This example shows how to enable CSSTP VLAN leaky function.

```
RTK.0> vlan set leaky csstp state enable
RTK.0>
```

vlan get leaky

Description:

This command get VLAN leaky function.

Syntax:

vlan get leaky ip-mcast port <*PORT_LIST:ports* **| all> state**
vlan get leaky <cdp | csstp> state
vlan get leaky rma *rma_tail* **state**

Parameter:

ip-mcast	– ip mulitcast
ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
cdp	– Cisco Discovery Protocol
csstp	– Cisco Shared Spanning Tree Protocol
rma	- reserved multicast address
rma_tail	- tail of RMA MAC address

Examples:

This example shows how to get CSSTP VLAN leaky function state.

```
RTK.0> vlan get leaky csstp state

Leaky state of Shared Spanning Tree Protocol: Enable
RTK.0>
```

vlan set keep-format

Description:

This command set VLAN keep format function. If this feature is enabled on specified type of frame, these frame will keep it VLAN tag format regardless of VLAN untag setting.

Syntax:

vlan set keep-format <cdp | sstp> state <enable | disable>
vlan set keep-format rma *rma_tail* **state <enable | disable>**

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration
cdp	– cisco Discovery Protocol
sstp	– cisco Shared Spanning Tree Protocol
rma	- reserved multicast address
rma_tail	- tail of RMA MAC address

Examples:

This example shows how to set CDP & SSTP keep-format function to enabled.

```
RTK.0> vlan set keep-format cdp state enable
RTK.0> vlan set keep-format sstp state enable
```

vlan get kee-format

Description:

This command get VLAN keep format function.

Syntax:

vlan get keep-format <cdp | sstp> state
vlan get keep-format rma *rma_tail* **state**

Parameter:

state	- state configuration
cdp	– Cisco Discovery Protocol
sstp	– Cisco Shared Spanning Tree Protocol
rma	- reserved multicast address
rma_tail	- tail of RMA MAC address

Examples:

This example shows how to get CDP & SSTP keep-format function.

```
RTK.0> vlan get keep-format cdp

VLAN Keep format state of Cisco Discovery Protocol: Enable
RTK.0> vlan get keep-format sstp

VLAN Keep format state of Shared Spanning Tree Protocol: Enable
```

2.2. SVLAN commands

svlan init

Description:

This command can initial svlan module, set svlan related configurations to default, and clear all svlan tables. Only after executing this command, other svlan commands can be executed and will not return FAILED.

Syntax:

svlan init

Parameter:

None

Examples:

This example shows how to initial svlan funtions and enable to execute other related svlan commands.

```
RTK.0> svlan init
RTK.0> svlan get service-port
Server Ports: none
RTK.0>
```

svlan create

Description:

This command can create a new svlan. After svlan is created, svlan member and other properties can be set.

Syntax:

svlan create svlan-table svid *svid*

Parameter:

svid – specify the numeric SVLAN identifier

Examples:

This example shows how to create a new svlan which svid is 100.

```
RTK.0> svlan init
RTK.0> svlan create svlan-table svid 100
RTK.0>
```

svlan destroy

Description:

This command can destroy a svlan or all created svlan.

Syntax:

svlan destroy svlan-table svid *svid*

svlan destroy svlan-table all

Parameter:

svid	– specify the numeric SVLAN identifier
svlan-table	– svlan table
all	- All vlan

Examples:

This example shows how to destroy svid 100.

```
RTK.0> svlan create svlan-table svid 100
RTK.0> svlan destroy svlan-table svid 100
RTK.0>
```

svlan set entry

Description:

This command can configure the svlan properties of member configuration entry. It is used for debug mode and will be executed whether svlan module had been initialed or not. But only svid of entry is created then the entry properties can be set.

Syntax:

svlan set entry *index* **svid** *svid*
svlan set entry *index* **member <***PORT_LIST:ports* **| all | none>**
svlan set entry *index* **tag-member <***PORT_LIST:ports* **| all | none>**
svlan set entry *index* **untag-member <***PORT_LIST:ports* **| all | none>**
svlan set entry *index* **priority** *priority*
svlan set entry *index* **fid-msti state <disable | enable>**
svlan set entry *index* **fid-msti** *fid_msti*
svlan set entry *index* **enhanced-fid state <disable | enable>**
svlan set entry *index* **enhanced-fid** *efid*

Parameter:

index	– index of svlan member configuration
svid	- specify the numeric sVLAN identifier
member	- member configuration
tag-member	- tagging member configuration
untag-member	- untag set configuration
ports	- the vlan member or tag member ports
all	- specify all ports
none	- specify no port
priority	- specify the priority value
fid_msti	- the filter-id or msti for this vlan
efid	- specify the enhanced filter-id
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to congigurate the svlan which svid is 1000:
Add port 0,1,2,3,4,5,6 to vlan 1000

```
Assign port 3 to tag member
Set svlan priority to 7
Enable svlan based fid and efid to fid 0 and efid 7

RTK.0> svlan destroy svlan-table all
RTK.0> svlan create svlan-table svid 1000
RTK.0> svlan set entry 0 member 0-6
RTK.0> svlan set entry 0 tag-member 3
RTK.0> svlan set entry 0 priority 7
RTK.0> svlan set entry 0 fid-msti state enable
RTK.0> svlan set entry 0 fid-msti 0
RTK.0> svlan set entry 0 enhanced-fid state enable
RTK.0> svlan set entry 0 enhanced-fid 7
RTK.0> svlan get entry 0
Index SVID Member  TagSet  Spri FidEn   FID EfidEn  Efid
0     1000 0-6     3      7    Enable  0   Enable  7
```

svlan get entry

Description:

Show svlan configuration of a specific entry.

Syntax:

svlan get entry *index*

svlan get entry all

Parameter:

index – index of svlan member configuration

all - All entries of SVLAN confingrations

Examples:

This example shows how to get the svlan table configuration of entry 0 and all entries retriving

```
RTK.0> svlan get entry 0
Index SVID Member  TagSet  Spri FidEn   FID EfidEn  Efid
0      1000 0-6     3         7    Enable  0    Enable  7
RTK.0>
RTK.0> svlan get entry all
Index SVID Member  TagSet  Spri FidEn   FID EfidEn  Efid
0      1000 0-6      3         7    Enable   0   Enable  7
1      1001           0-6     0     Disable 0   Disable 0
RTK.0>
```

svlan set svlan-table

Description:

This command can configure svlan with specify svid and only if this svlan is created

Syntax:

svlan set svlan-table svid *svid* **member <***PORT_LIST:ports* **| all | none>**

svlan set svlan-table svid *svid* **tag-member <***PORT_LIST: ports* **| all | none>**

svlan set svlan-table svid *svid* **untag-member <***PORT_LIST:ports* **| all | none>**

svlan set svlan-table svid *svid* **priority** *priority*

svlan set svlan-table svid *svid* **fid-msti state <disable | enable>**

svlan set svlan-table svid *svid* **fid-msti** *fid_msti*

svlan set svlan-table svid *svid* **enhanced-fid state <disable | enable>**

svlan set svlan-table svid *svid* **enhanced-fid** *efid*

Parameter:

svid	- specify the numeric sVLAN identifier
member	- member configuration
tag-member	- tagging member configuration
untag-member	- untag set configuration
ports	- the vlan member or tag member ports
all	- specify all ports
none	- specify no port
priority	- specify the priority value
fid_msti	- the filter-id or msti for this vlan
efid	- specify the enhanced filter-id
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to congigurate the svlan which svid is 2000:

Add port 0-4 to vlan 2000

```
Assign port 3 to tag member

RTK.0> svlan create svlan-table svid 2000
RTK.0> svlan set svlan-table svid 2000 member 0-4
RTK.0> svlan set svlan-table svid 2000 tag-member 3
RTK.0> svlan get svlan-table svid 2000
SVID Member  UntagSet Spri FidEn   FID EfidEn  Efid
2000 0-4      0-2,4-6  0     Disable 0   Disable 0
RTK.0>
```

svlan get svlan-table

Description:

Show svlan configuration of a specify svid.

Syntax:

svlan get svlan-table svid *svid*

Parameter:

svid - specify the numeric SVLAN identifier

Examples:

This example shows how to get the svlan table configuration of specify svid 2000

```
RTK.0> svlan get svlan-table svid 2000
SVID Member  UntagSet Spri FidEn   FID EfidEn  Efid
2000 0-4     0-2,4-6  0     Disable 0   Disable 0
RTK.0>
```

svlan set port svid

Description:

This command can configure port based svlan with specify svid.

Syntax:

svlan set port <*PORT_LIST:ports* **| all> svid** *svid*

Parameter:

svid	- specify the numeric VLAN identifier
ports	- specified port list
all	– specify all ports

Examples:

This example shows how to congigurate the port 0 port-based svlan with svid 100 and port 1 port-based svlan with svid 200

```
RTK.0> svlan create svlan-table svid 100
RTK.0> svlan create svlan-table svid 200
RTK.0> svlan set port 0 svid 100
RTK.0> svlan set port 1 svid 200
RTK.0>
```

svlan get port svid

Description:

Show port based svlan with specify svid.

Syntax:

svlan get port <*PORT_LIST:ports* **| all> svid**

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get the port 0/1 port-based svlan.

```
RTK.0> svlan get port 0-1 svid
Port SVID
0    100
1    200
RTK.0>
```

svlan set service-port

Description:

This command can configure svlan tag aware ports or svlan service ports.

Syntax:

svlan set service-port <*PORT_LIST:ports* **| all | none>**

Parameter:

ports	- specified port list
all	– specify all ports
none	- specify no port

Examples:

This example shows how to configurate svlan servive ports 0,3

```
RTK.0> svlan set service-port 0,3
RTK.0>
```

svlan get service-port

Description:

Show svlan tag aware ports or svlan service ports

Syntax:

svlan get service-port

Parameter:

None

Examples:

This example shows how to get svlan servive ports or stag aware ports

```
RTK.0> svlan set service-port 0,3
RTK.0> svlan get service-port
Server Ports: 0,3
RTK.0>
```

svlan set lookup-type

Description:

This command can configure svlan lookup from svlan member configuration entry or vlan 4k table. This command is supported depond on chip design or not.

Syntax:

svlan set lookup-type <svlan-table | vlan-4k-table>

Parameter:

svlan-table - 64 svlan member configurations
vlan-4k-table - 4K vlan table

Examples:

This example shows how to configurate svlan lookup mode to lokkup svlan member configuration entries.

```
RTK.0> svlan set lookup-type svlan-table
RTK.0>
```

svlan get lookup-type

Description:

Show svlan lookup mode

Syntax:

svlan get lookup-type

Parameter:

None

Examples:

This example shows how to get svlan lookup mode

```
RTK.0> svlan set lookup-type svlan-table
RTK.0> svlan get lookup-type
Lookup type: SVLAN 64 entries
RTK.0>
```

svlan set tpid

Description:

This command can configure svlan tagging ethertype

Syntax:

svlan set tpid *tpid*

Parameter:

tpid - ethertype for stag

Examples:

This example shows how to configurate ethertype of svlan tag to 0x9100.

```
RTK.0> svlan set tpid 0x9100
RTK.0>
```

svlan get tpid

Description:

Show svlan tagging ethertype

Syntax:

svlan get tpid

Parameter:

None

Examples:

This example shows how to get ethertype of svlan tag

```
RTK.0> svlan set tpid 0x9100
RTK.0> svlan get tpid
TPID: 0x9100
RTK.0>
```

svlan set priority-source

Description:

This command can configure egress stag priority field inserting source. This configuration will effect asic egress staging priority assignment only.

Syntax:

svlan set priority-source <internal-priority | dot1q-priority | svlan-member-cofig | port-based-priority>

Parameter:

internal-priority	- switch core internal priority
dot1q-priority	- ingress cvlan tagged priority field
svlan-member-cofig	- priority of svlan member configuration
port-based-priority	- port based priority

Examples:

This example shows how to configurate svlan tagging priority source from cvlan ingress tagged priority field. That means egress stag priority is as the same as ctag priority.

```
RTK.0> svlan set priority-source dot1q-priority
RTK.0>
```

svlan get priority-source

Description:

Show egress stag priority field inserting source

Syntax:

svlan get priority-source

Parameter:

None

Examples:

This example shows how to get source decision of svlan tagging priority field

```
RTK.0> svlan set priority-source dot1q-priority
RTK.0> svlan get priority-source
Priority source: 1q tag priority
RTK.0>
```

svlan set unmatch

Description:

This command can configure asic ation for ingress stag which svid is unmatched any of svlan member configuration entry if svlan lookup is used.

Syntax:

svlan set unmatch <drop | trap-to-cpu>
svlan set unmatch assign-svlan svid *svid*

Parameter:

drop	- drop packet
trap-to-cpu	- trap packet to cpu port
assign-svlan	- assign svlan property
svid	– specify the numeric SVLAN identifier

Examples:

This example shows how to assign action for packets ingress svid unmatched all svlan member configuration to svlan with svid 1001.

```
RTK.0> svlan create svlan-table svid 1001
RTK.0> svlan set unmatch assign-svlan svid 1001
RTK.0>
```

svlan get unmatch

Description:

Show action for ingress stag packets from service port which svid is unmatched all svlan member configuration entries

Syntax:

svlan get unmatch

Parameter:

None

Examples:

This example shows how to get action for packets from service ports that ingress svid unmatched all svlan member configuration

```
RTK.0> svlan set unmatch assign-svlan svid 1001
RTK.0> svlan get unmatch
Action: Assign to SVLAN SVID: 1001
RTK.0>
```

svlan set untag

Description:

This command can configure action for packets without stag from svlan service port

Syntax:

svlan set untag < drop | trap-to-cpu >
svlan set untag assign-svlan svid *svid*

Parameter:

drop	- drop packet
trap-to-cpu	- trap packet to cpu port
assign-svlan	- assign svlan property
svid	– specify the numeric SVLAN identifier

Examples:

This example shows how to assign trap action for packets without ingress stag from svlan service port.

```
RTK.0> svlan set untag trap-to-cpu
RTK.0>
```

svlan get untag

Description:

Show action for ingress un-stag packets from svlan service port

Syntax:

svlan get untag

Parameter:

None

Examples:

This example shows how to get action for ingress un-stag packets from svlan service port

```
RTK.0> svlan set untag assign-svlan svid 1001
RTK.0> svlan get untag
Action: Assign to SVLAN SVID: 1001
RTK.0>
```

svlan set trap-priority

Description:

This command can configure priority for packets trapped to cpu for related svlan functions

Syntax:

svlan set trap-priority *priority*

Parameter:

priority - priority for trapping packets

Examples:

This example shows how to assin priority 7 packets which meet svlan related trapped behavior

```
RTK.0> svlan set trap-priority 7
RTK.0>
```

svlan get trap-priority

Description:

Show svlan related assigned trapping priority

Syntax:

svlan get trap-priority

Parameter:

None

Examples:

This example shows how to get svlan trapping priority

```
RTK.0> svlan set trap-priority 6
RTK.0> svlan get trap-priority
Trap priority: 6
RTK.0>
```

svlan set dei-keep state

Description:

This command can configure egress keepping dei field of ingress stag. The DEI of stag will be keep only if this state is set, otherwise it will always be 0 in egress svlan tag.

Syntax:

svlan set dei-keep state <disable | enable>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to configurate enable keeping ingress dei field of stag packets for egress stag.

```
RTK.0> svlan set dei-keep state enable
RTK.0>
```

svlan get dei-keep state

Description:

Show egress keep dei field state of ingress stag packets

Syntax:

svlan get dei-keep state

Parameter:

state - state configuration

Examples:

This example shows how to get keep ingress dei field state of stagged packets for egress stag

```
RTK.0> svlan set dei-keep state enable
RTK.0> svlan get dei-keep
Keep DEI state: Enable
RTK.0>
```

svlan set vlan-aggregation

Description:

This command can configure svlan vlan-aggregation state for downstream egress port. Asic can tagging vid by ingress learning in downstream egress port while vlan aggregation state is set. Not only downstream stag only packet, but also ctag packets will be treated as egress ctag packets by vid which auto vid leaning with mac-address if forced-state is set.

Syntax:

svlan set vlan-aggregation port <*PORT_LIST:ports* **| all > state <disable | enable>**

svlan set vlan-aggregation forced-state <disable | enable>

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
forced-state	- forced state configuration

Examples:

This example shows how to configurate the egress port 1 downstream packet vlan aggregation state enable.

```
RTK.0> svlan set vlan-aggregation port 1 state enable
RTK.0> svlan get vlan-aggregation port 1 state
Port Status
1    Enable
RTK.0>
```

svlan get vlan-aggregation

Description:

Show svlan vlan-aggregation state for downstream egress port.

Syntax:

svlan get vlan-aggregation port <*PORT_LIST:ports* **| all > state**
svlan get vlan-aggregation forced-state

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
forced-state	- forced state configuration

Examples:

This example shows how to get port 0-3 downstream packet vlan aggregation state

```
RTK.0> svlan get vlan-aggregation port 0-3 state
Port Status
0    Disable
1    Enable
2    Disable
3    Disable
RTK.0>
```

svlan add vlan-conversion

Description:

This command can add ingress vlan conversion configuration from ingress vlan vid or multicast address to svlan svid. Also, this command can add egress vlan conversion configuration for different svlan and egress port to different egress vlan vid.

Syntax:

svlan add vlan-conversion c2s vid *vid* **port <***PORT_LIST:ports* **| all> svid** *svid*

svlan add vlan-conversion mc2s ip *ip* **ip-mask** *ip_mask* **svid** *svid*

svlan add vlan-conversion mc2s mac-address *mac* **mac-mask** *mac_mask* **svid** *svid*

svlan add vlan-conversion sp2c svid *svid* **port** *port* **vid** *vid*

Parameter:

c2s	– cvlan to svlan conversion
mc2s	- multicast to svlan conversion
sp2c	- svlan with egress port to cvlan conversion
vid	– specify the numeric VLAN identifier
svid	– specify the numeric SVLAN identifier
ports	- specified port list
all	– specify all ports
ip	- IPv4 address
ip_mask	- IPv4 address mask
mac	- MAC address
mac_mask	- MAC address mask
port	- specified port

Examples:

This example shows how to add vlan conversion configuration:
Ingress vid 100 from port 0 convert to svlan svid 2000
Ingress vid 100 from port 1 convert to svlan svid 2001
Ingress vid 200 from port 1 convert to svlan svid 2000
IPv4 dip 239.0.0.0~239.0.0.255 to svlan 2002
Layer 2 multicast address 01:xx:11:22:00:00~01:xx:11:22:FF:FF to svlan 2003
SVLAN svid 2000 and egress port 0 to vlan vid 100
SVLAN svid 2001 and egress port 1 to vlan vid 100

SVLAN svid 2000 and egress port 1 to vlan vid 200

```
RTK.0> svlan create svlan-table svid 2000
RTK.0> svlan create svlan-table svid 2001
RTK.0> svlan create svlan-table svid 2002
RTK.0> svlan create svlan-table svid 2003
RTK.0> svlan add vlan-conversion c2s vid 100 port 0 svid 2000
RTK.0> svlan add vlan-conversion c2s vid 200 port 1 svid 2000
RTK.0> svlan add vlan-conversion c2s vid 100 port 1 svid 2001
RTK.0> svlan add vlan-conversion mc2s ip 239.0.0.0 ip-mask
255.255.255.0 svid 2002
RTK.0> svlan add vlan-conversion mc2s mac-address
01:00:11:22:00:00 mac-mask 00:00:FF:FF:00:00 svid 2003
RTK.0> svlan add vlan-conversion sp2c svid 2000 port 0 vid 100
RTK.0> svlan add vlan-conversion sp2c svid 2000 port 1 vid 200
RTK.0> svlan add vlan-conversion sp2c svid 2001 port 1 vid 100
RTK.0>
```

svlan del vlan-conversion

Description:

This command can delete vlan conversion configuration with ingress vid to svid or svid to egress vid.

Syntax:

svlan del vlan-conversion c2s vid *vid* **port <***PORT_LIST:ports* **| all> svid** *svid*

svlan del vlan-conversion mc2s ip *ip* **ip-mask** *ip_mask*

svlan del vlan-conversion mc2s mac-address *mac* **mac-mask** *mac_mask*

svlan del vlan-conversion sp2c svid *svid* **port** *port*

Parameter:

c2s	– cvlan to svlan conversion
mc2s	- multicast to svlan conversion
sp2c	- svlan with egress port to cvlan conversion
vid	– specify the numeric VLAN identifier
svid	– specify the numeric SVLAN identifier
ports	- specified port list
all	– specify all ports
ip	- IPv4 address
ip_mask	- IPv4 address mask
mac	- MAC address
mac_mask	- MAC address mask
port	- specified port

Examples:

This example shows how to delete ipv4 dip 239.0.0.0~239.0.0.255 to svid 2002 and svid 2001 with egress port 1

```
RTK.0> svlan del vlan-conversion mc2s ip 239.0.0.0 ip-mask
255.255.255.0
RTK.0> svlan del vlan-conversion sp2c svid 2001 port 1
RTK.0>
```

svlan get vlan-conversion

Description:

Show vlan conversion configuration with ingress cvlan, multicast or egress svlan and egress port.

Syntax:

svlan get vlan-conversion c2s vid *vid* **port <PORT_LIST:***ports* **| all>**

svlan get vlan-conversion mc2s ip *ip* **ip-mask** *ip_mask*

svlan get vlan-conversion mc2s mac-address *mac* **mac-mask** *mac_mask*

svlan get vlan-conversion sp2c svid *svid* **port** *port*

Parameter:

c2s	– cvlan to svlan conversion
mc2s	- multicast to svlan conversion
sp2c	- svlan with egress port to cvlan conversion
vid	– specify the numeric VLAN identifier
svid	– specify the numeric SVLAN identifier
ports	- specified port list
all	– specify all ports
ip	- IPv4 address
ip_mask	- IPv4 address mask
mac	- MAC address
mac_mask	- MAC address mask
port	- specified port

Examples:

This example shows how to get the svlan vlan conversion configuration with ingress vid 100/200 and egress svid 2000 with egress port 0/1.

```
RTK.0> svlan get vlan-conversion c2s vid 100 port all
Vid Port Svid
100 0    2000
100 1    2001
RTK.0> svlan get vlan-conversion c2s vid 200 port all
Vid Port Svid
200 1    2000
```

```
RTK.0> svlan get vlan-conversion sp2c svid 2000 port 0
Svid Port Cvid
2000 0    100
RTK.0> svlan get vlan-conversion sp2c svid 2000 port 1
Svid Port Cvid
2000 1    200
RTK.0>
```

svlan set vlan-conversion sp2c unmatch-action

Description:

This command can configuration packet egress mode while svlan downstream packets which unmatch any entry of svid with egress port to egress vid conversion setting.

Syntax:

svlan set vlan-conversion sp2c unmatch-action <untag | ctag>

Parameter:

untag – egress cvlan untagging format
ctag - egress cvlan tagging format

Examples:

This example shows how to configurate always untag egress packets witch unmatch any entry of svid with egress port vlan conversion.

```
RTK.0> svlan set vlan-conversion sp2c unmatch-action untag
RTK.0>
```

svlan get vlan-conversion sp2c unmatch-action

Description:

Show configuration setting for svlan downstream packets which unmatch any entry of svid with destination port to egress vid conversion setting.

Syntax:

svlan get vlan-conversion sp2c unmatch-action

Parameter:

None

Examples:

This example shows how to get unmatched svid with egress port to egress vid conversion configuration

```
RTK.0> svlan get vlan-conversion sp2c unmatch-action
SP2C unmatching C-tag format: Untagging
RTK.0>
```

2.3. QoS commands

qos init

Description:

This command will initializae qos module to default setting.

Syntax:

qos init

Parameter:

None

Examples:

This example shows how to initialized QoS module.

```
RTK.0> qos init
RTK.0>
```

qos set priority-to-queue

Description:

This command can configure the priority to queue table, this table have 4 entries. Each entry can map 8 priority to specifiy queue-id.

This table be referenced by “qos set priority-to-queue-mapping”

Syntax:

qos set priority-to-queue table *index* **priority** *MASK_LIST:priority* **queue-id** *qid*

qos set priority-to-queue port <*PORT_LIST:ports* **| all> table** *index*

Parameter:

index	– index of priority to queue table
table	- priority to queue mapping table
priority	- priority configuration
qid	- specify the numeric queue identifier
ports	- specified port list
all	– specify all ports

Examples:

This example shows how to congigurate the prority to queue table entry 0 which mappimg is:

priority 0 to 7
priority 1 to 6
priority 2 to 5
priority 3 to 4
priority 4 to 3
priority 5 to 2
priority 6 to 1
priority 7 to 0
Mapping port 0 ~ 6 to prority to queue table entry 0

```
RTK.0> qos set priority-to-queue table 0 priority 0 queue-id 7
RTK.0> qos set priority-to-queue table 0 priority 1 queue-id 6
RTK.0> qos set priority-to-queue table 0 priority 2 queue-id 5
RTK.0> qos set priority-to-queue table 0 priority 3 queue-id 4
RTK.0> qos set priority-to-queue table 0 priority 4 queue-id 3
```

```
RTK.0> qos set priority-to-queue table 0 priority 5 queue-id 2
RTK.0> qos set priority-to-queue table 0 priority 6 queue-id 1
RTK.0> qos set priority-to-queue table 0 priority 7 queue-id 0
RTK.0> qos  set priority-to-queue port 0-6 table 0
```

qos get priority-to-queue

Description:

Show priority to queue mapping table

Syntax:

qos get priority-to-queue table *index*

qos get priority-to-queue port <*PORT_LIST:ports* **| all>**

Parameter:

index	– index of priority to queue table
table	- priority to queue mapping table
ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get prority to queue table.

```
Get prority to queue table entry 0
RTK.0> qos get priority-to-queue table 0
Index pri_0 pri_1 pri_2 pri_3 pri_4 pri_5 pri_6 pri_7
0          7     6     5     4     3     2     1     0

Get prority to queue table entry 0 seting for port 1
RTK.0> qos get priority-to-queue port 1
Port  Index
1     0
```

qos set remapping

Description:

Per system set priority remapping for dot1p, DSCP, port-ased priority and forward to CPU priority.

Syntax:

qos set remapping dot1p dot1p-priority *dot1p_priority* **internal-priority** *internal_priority*

qos set remapping dscp dscp *MASK_LIST:dscp* **internal-priority** *priority*

qos set remapping forward-to-cpu internal-priority *internal_priority* **remapping-priority** *remapping_priority*

qos set remapping port <*PORT_LIST:ports* **| all> internal-priority** *priority*

Parameter:

dot1p	- dot1p remapping configuration
dscp	- dscp remapping configuration
forward-to-cpu	- to cpu port remapping configuration
port	- port-based priority
dot1p-priority	- dot1p priority
internal-priority	- switch internal priority
remapping-priority	- priority for remapped configuration
ports	- specified port list
all	– specify all ports

Examples:

This example shows how to remappimg internal priority from packet dot1p priorty , the remapping setting as:
dot1p-priority 0 remapping to priority 7
dot1p-priority 1 remapping to priority 6
dot1p-priority 2 remapping to priority 5
dot1p-priority 3 remapping to priority 4

```
RTK.0> qos set remapping dot1p dot1p-priority 0
internal-priority 7
RTK.0> qos set remapping dot1p dot1p-priority 1
internal-priority 6
RTK.0> qos set remapping dot1p dot1p-priority 2
```

```
internal-priority 5
RTK.0> qos set remapping dot1p dot1p-priority 3
internal-priority 4
```

This example shows how to remappimg internal priority from DSCP priority, the remapping setting as:
DSCP priority 0~15 remapping to 0
DSCP priority 16~31 remapping to 1
DSCP priority 32~47 remapping to 2
DSCP priority 48~63 remapping to 3

```
RTK.0> qos set remapping dscp dscp 0-15 internal-priority 0
RTK.0> qos set remapping dscp dscp 16-31 internal-priority 1
RTK.0> qos set remapping dscp dscp 32-47 internal-priority 2
RTK.0> qos set remapping dscp dscp 48-63 internal-priority 3
```

This example shows how to remappimg forward to CPU packet priority as:
Internal priority 0 remapping to 0
Internal priority 1 remapping to 0
Internal priority 2 remapping to 3
Internal priority 3 remapping to 3
Internal priority 4 remapping to 5

```
RTK.0> qos set remapping forward-to-cpu internal-priority 0
remapping-priority 0
RTK.0> qos set remapping forward-to-cpu internal-priority 1
remapping-priority 0
RTK.0> qos set remapping forward-to-cpu internal-priority 2
remapping-priority 3
RTK.0> qos set remapping forward-to-cpu internal-priority 3
remapping-priority 3
RTK.0> qos set remapping forward-to-cpu internal-priority 4
remapping-priority 5
```

This example shows how to set port based priority as:
Port 0 remapping to priority 0
Port 1 remapping to priority 1
Port 2 remapping to priority 2

```
Port 3 remapping to priority 3
RTK.0> qos set remapping port 0 internal-priority 0
RTK.0> qos set remapping port 1 internal-priority 1
RTK.0> qos set remapping port 2 internal-priority 2
RTK.0> qos set remapping port 3 internal-priority 3
```

qos get remapping

Description:

Get remapping setting.

Syntax:

qos get remapping dot1p
qos get remapping dscp
qos get remapping forward-to-cpu
qos get remapping port <*PORT_LIST:ports* **| all>**

Parameter:

dot1p	- dot1p remapping configuration
dscp	- dscp remapping configuration
forward-to-cpu	- to cpu port remapping configuration
port	- port-based priority
ports	- specified port list
all	– specify all ports

Examples:

This example shows how get remapping setting:

```
RTK.0> qos get remapping dot1p
1p Priority    internal priority
0                7
1                1
2                2
3                3
4                4
5                5
6                6
7                7

RTK.0> qos get remapping dscp
DSCP  Priority
0     0
1     0
2     0
3     0
4     0
```

```
5     0
6     0
7     0
8     0
9     0
10    0
11    0
12    0
13    0
14    0
15    0
16    0
17    0
18    0
19    0

--More—

RTK.0> qos get remapping forward-to-cpu
Priority: 0 Remapping prority: 0
Priority: 1 Remapping prority: 0
Priority: 2 Remapping prority: 0
Priority: 3 Remapping prority: 0
Priority: 4 Remapping prority: 0
Priority: 5 Remapping prority: 0
Priority: 6 Remapping prority: 0
Priority: 7 Remapping prority: 0

RTK.0> qos get remapping port  0-6
Port prioirty
0    0
1    0
2    0
3    0
4    0
5    0
6    0
```

qos set priority-selector

Description:

There are 9 priority source in system. This command can assign weight to wach priority source. The source with highest weight will be select to internal priority.

There are 2 prioirty selector group. Per port can select which selector group this port will be used.

Syntax:

qos set priority-selector group-id *index* **<port| dot1q | dscp | acl | smac | svlan | vlan | l4 | lookup-table>** *weight*

qos set priority-selector port <*PORT_LIST:ports* **| all> group-id** *index*

Parameter:

index	- priority selector group index
port	- port based priority source
dot1q	- dot1q priority source
dscp	- dscp priority source
acl	- acl priority source
lut	- lookup table priority source
smac	- source mac priority source
svlan	- svlan priority source
vlan	- vlan priority source
l4	- L4 priority source
weight	- the weight for this priority source
ports	- specified port list
all	– specify all ports

Examples:

This example shows how to set acl priority weight to 15 and port-based priorty weight to 10 for selector group 0.
Set port 0-5 apply selector group 0.

```
RTK.0> qos set priority-selector group-id 0  acl 15
RTK.0> qos set priority-selector group-id 0 port 10


RTK.0> qos set priority-selector port 0-5 group-id 0
```

qos get priority-selector

Description:

Show system priority selector weight for each priority source.

Syntax:

qos get priority-selector group-id *index*

qos get priority-selector port <*PORT_LIST:ports* **| all> group-id**

Parameter:

index - priority selector group index

ports - specified port list

all – specify all ports

Examples:

This example shows how to get priority selector weight for each priority source.

```
RTK.0> qos get priority-selector group-id 0
TYPE          Weight
port          10
dot1q         0
dscp          0
acl           15
vlan          0
lookup-table  0
smac          0
svlan         0
l4            0
RTK.0> qos get priority-selector port  0-5  group-id
Port Group
0    0
1    0
2    0
3    0
4    0
5    0
```

qos set remarking dot1p

Description:

Per egress port enable/disable dot1p priority remarking and set per system internal priority to remarking dot1p mapping.

Syntax:

qos set remarking dot1p port <*PORT_LIST:ports* **| all> state <disable | enable>**

qos set remarking dot1p user-priority *priority* **dot1p-priority** *dot1p_priority*

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
dot1p-priority	- dot1p priority
user-priority	- switch user priority
priority	- priority configuration

Examples:

This example shows how to congigurate internal priority remarking to packet dot1p priority as:

The example shows how to enable dot1p priority remarking for port 1,2,3

```
internal priority 0 remarking dot1p priority 7
internal priority 1 remarking dot1p priority 6
internal priority 2 remarking dot1p priority 5
internal priority 3 remarking dot1p priority 4
internal priority 4 remarking dot1p priority 3
internal priority 5 remarking dot1p priority 2
internal priority 6 remarking dot1p priority 1
internal priority 7 remarking dot1p priority 0
```

```
RTK.0> qos set remarking dot1p user-priority 0 dot1p-priority
7
```

```
RTK.0> qos set remarking dot1p user-priority 1 dot1p-priority
6
RTK.0> qos set remarking dot1p user-priority 2 dot1p-priority
5
RTK.0> qos set remarking dot1p user-priority 3 dot1p-priority
4
RTK.0> qos set remarking dot1p user-priority 4 dot1p-priority
3
RTK.0> qos set remarking dot1p user-priority 5 dot1p-priority
2
RTK.0> qos set remarking dot1p user-priority 6 dot1p-priority
1
RTK.0> qos set remarking dot1p user-priority 7 dot1p-priority
0

RTK.0> qos set remarking dot1p port 1,2,3 state enable
```

qos get remarking dot1p

Description:

Per system set forward to CPU priority remapping.

Syntax:

qos get remarking dot1p port <*PORT_LIST:ports* **| all> state**

qos get remarking dot1p

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
source	- remarking source

qos set remarking dscp

Description:

Set dscp remarking function. Per port would enable/disable DSCP remarking. The remarking source can be selected for each port. The remarking source would be internal-prioirty, user-prioirty and packet original dscp value.

Syntax:

qos set remarking dscp port <*PORT_LIST:ports* **| all> state <disable | enable>**

qos set remarking port <*PORT_LIST:ports* **| all> dscp source <internal-priority | user-priority | dscp>**

qos set remarking dscp inter-priority *priority* **remarking-dscp** *dscp*

qos set remarking dscp user-priority *priority* **remarking-dscp** *dscp*

qos set remarking dscp original-dscp *MASK_LIST:dscp* **remarking-dscp** *remarking_dscp*

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
source	- remarking source
inter-priority	- switch internal/user priority
internal-priority	- switch internal priority
user-priority	- switch user priority
original-dscp	- ingress original dscp
remarking-dscp	- egress remarked dscp
dscp	- dscp configuration
priority	- priority configuration

Examples:

This example shows how to enable dscp remarking on port 4. The remarking source is based on packet original dscp value. The dscp remarking value will be set as

```
Packet original DSCP valu 0-31 remapping to 0
Packet original DSCP valu 32-63 remapping to 32

RTK.0> qos set remarking dscp port 4 state enable
RTK.0> qos set remarking port 4 dscp source dscp
RTK.0> qos set remarking dscp original-dscp 0-31 remarking-dscp
0
RTK.0> qos set remarking dscp original-dscp 31-63
remarking-dscp 31
RTK.0>
```

qos get remarking dscp

Description:

Get dscp remarking setting.

Syntax:

qos get remarking dscp port <*PORT_LIST:ports* | all> state
qos get remarking port <*PORT_LIST:ports* | all> dscp source
qos get remarking dscp <inter-priority | original-dscp>

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
source	- remarking source
inter-priority	- switch internal/user priority
original-dscp	- ingress original dscp

qos set scheduling algorithm

Description:

This command can set per port per queue scheduling algorithm.

Syntax:

qos set scheduling algorithm port <*PORT_LIST:ports* **| all> queue-id** *qid* **<strict | wfq>**

Parameter:

ports	- specified port list
all	– specify all ports
qid	- specify the numeric queue identifier
strict	- scheduling using strict queue type
wfq	- scheduling using WFQ type

Examples:

This example shows how to congigurate port 1 queue 0-3 using strict, queue 4-7 using WFQ:

```
RTK.0> qos set scheduling algorithm port 1 queue-id 0 strict
RTK.0> qos set scheduling algorithm port 1 queue-id 1 strict
RTK.0> qos set scheduling algorithm port 1 queue-id 2 strict
RTK.0> qos set scheduling algorithm port 1 queue-id 3 strict
RTK.0> qos set scheduling algorithm port 1 queue-id 4 wfq
RTK.0> qos set scheduling algorithm port 1 queue-id 5 wfq
RTK.0> qos set scheduling algorithm port 1 queue-id 6 wfq
RTK.0> qos set scheduling algorithm port 1 queue-id 7 wfq
```

qos get scheduling algorithm

Description:

Show per port per queue scheduling algorithm.

Syntax:

qos get scheduling algorithm port <*PORT_LIST:ports* **| all> queue-id** *qid*

Parameter:

ports	- specified port list
all	– specify all ports
qid	- specify the numeric queue identifier

Examples:

This example shows how to get queue scheduling algorithm for port 1 queue 0 and queue 1.

```
RTK.0> qos get scheduling algorithm port 1 queue-id 0
port: 1 queue: 0 type:strict
RTK.0> qos get scheduling algorithm port 1 queue-id 1
port: 1 queue: 1 type:strict
RTK.0>
```

qos set scheduling queue-weight

Description:

This command can set per port per queue WFQ scheduling weight .

Syntax:

qos set scheduling queue-weight port <*PORT_LIST:ports* **| all > queue-id** *qid* **weight** *weight*

Parameter:

ports	- specified port list
all	– specify all ports
qid	- specify the numeric queue identifier
weight	- the queue scheduling weight

Examples:

This example shows how to congigurate port 1 queue 0 weight 65535, queue 4 weight 15:

```
RTK.0> qos set scheduling queue-weight port 1 queue-id 0 weight
65535
RTK.0> qos set scheduling queue-weight port 1 queue-id 4 weight
15
RTK.0>
```

qos get scheduling queue-weight

Description:

Show per port per queue scheduling weight.

Syntax:

qos get scheduling queue-weight port <*PORT_LIST:ports* **| all> queue-id** *qid*

Parameter:

ports	- specified port list
all	– specify all ports
qid	- specify the numeric queue identifier

Examples:

This example shows how to get queue scheduling weight for port 1 queue 0-1.

```
RTK.0> qos get scheduling queue-weight port 1 queue-id 0
port: 1 queue: 0 weight:65535
RTK.0> qos get scheduling queue-weight port 1 queue-id 1
port: 1 queue: 1 weight:0
RTK.0>
```

qos set avb remapping

Description:

Per system set av bridge priority remapping.

Syntax:

qos set avb remapping internal-priority *priority* **user-priority** *user_priority*

qos set avb remapping port <*PORT_LIST:ports* **| all> state <disable | enable>**

Parameter:

internal-priority	- switch internal priority
user-priority	- switch user priority
ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to enable av bridge priority remaping on port 1 and the remapping table will as follow:

```
Internal priority 0 remapping to 0
Internal priority 1 remapping to 0
Internal priority 2 remapping to 3
Internal priority 3 remapping to 3
Internal priority 4 remapping to 3
Internal priority 5 remapping to 7
Internal priority 6 remapping to 7
Internal priority 7 remapping to 7

RTK.0> qos set avb remapping port 1 state enable
RTK.0> qos set avb remapping internal-priority 0 user-priority 0
RTK.0> qos set avb remapping internal-priority 1 user-priority 0
RTK.0> qos set avb remapping internal-priority 2 user-priority 3
RTK.0> qos set avb remapping internal-priority 3 user-priority
```

```
3
RTK.0> qos set avb remapping internal-priority 4 user-priority
3
RTK.0> qos set avb remapping internal-priority 5 user-priority
7
RTK.0> qos set avb remapping internal-priority 6 user-priority
7
RTK.0> qos set avb remapping internal-priority 7 user-priority
7
```

qos get avb remapping

Description:

Get AV bridage priority remapping setting.

Syntax:

qos get avb remapping internal-priority *priority*

qos get avb remapping port <*PORT_LIST:ports* **| all> state**

Parameter:

internal-priority	- switch internal priority
ports	- specified port list
all	– specify all ports
state	- state configuration

Examples:

```
RTK.0> qos get avb remapping port all
Port State
0    Disable
1    Enable
2    Disable
3    Disable
4    Disable
5    Disable

RTK.0> qos get avb remapping internal-priority 0
Priority  Remap-Pri
0           0
RTK.0> qos get avb remapping internal-priority 1
Priority  Remap-Pri
1           0
```

2.4. l2-table commands

l2-table init

Description:

This command can initialize & reset L2 module.

Syntax:

l2-table init

Parameter:

None

Examples:

This example shows how to initialize L2 module.

```
RTK.0> l2-table init
```

l2-table set aging-time

Description:

Thie command set aging time. The unit is 0.1 second.

Syntax:

l2-table set aging-time *time*

Parameter:

time - aging time, unit 0.1 second

Examples:

This example shows how to set aging time to 300 seconds.

```
RTK.0> l2-table set aging-time 3000
```

l2-table get aging-time

Description:

This command can get aging time.

Syntax:

l2-table get aging-time

Parameter:

None

Examples:

This example shows how to get aging time.

```
RTK.0> l2-table get aging-time

 Age Time: 3000
```

l2-table set limit-learning

Description:

This command can set per port or system limit learning number. The number specified in this command is only count dynamic L2 entries. For those packets which is not learned due to the current entry number is equal to limit learning number, an action can also be set for these packet.

Syntax:

l2-table set limit-learning port <*PORT_LIST:ports* **| all> count** *count*

l2-table set limit-learning port <*PORT_LIST:ports* **| all> count unlimited**

l2-table set limit-learning port <*PORT_LIST:ports* **| all> action <drop | forward | copy-to-cpu | trap-to-cpu>**

l2-table set limit-learning action <drop | forward | copy-to-cpu | trap-to-cpu>

l2-table set limit-learning count *count*

l2-table set limit-learning count unlimited

Parameter:

ports	– specify a port list
all	- specify all ports
drop	- drop packet
trap-to-cpu	- trap packet to cpu port
forward	- forward packet
copy-to-cpu	- copy packet to cpu port
count	- limited learning count
unlimited	- unlimited configuration
action	- over learning behavior

Examples:

This example shows how to set the following configuration.
Set system limit learning to 1000.
Set port 0-1 limit learning to 20.
Set system limit learning action as “Drop”
Set port 0-1 limit learning action as “Forward”

```
RTK.0> l2-table set limit-learning count 1000
```

```
RTK.0> l2-table set limit-learning action drop
RTK.0> l2-table set limit-learning port 0-1 count 20
RTK.0> l2-table set limit-learning port 0-1 action forward
RTK.0>
```

l2-table get limit-learning

Description:

This command can get the configuration of limit learning

Syntax:

l2-table get limit-learning port <*PORT_LIST:ports* **| all> count**
l2-table get limit-learning port <*PORT_LIST:ports* **| all> action**
l2-table get limit-learning action
l2-table get limit-learning count

Parameter:

ports	– specify a port list
all	- specify all ports
count	- limited learning count
action	- over learning behavior

Examples:

This example shows how to get system and port limit learning configuration.

```
RTK.0> l2-table get limit-learning count

 System Learning Limit: 1000
RTK.0> l2-table get limit-learning action

 System learning Over Action: Drop
RTK.0> l2-table get limit-learning port 0-6 count

 Port 0 learning limit: 20
 Port 1 learning limit: 20
 Port 5 learning limit: 2112
RTK.0> l2-table get limit-learning port 0-1 action

Port 0 learning limit over action: Forward
Port 1 learning limit over action: Forward
RTK.0>
```

l2-table get learning-exceed

Description:

This command get the status of limit learning exceed. If the current learning number is equal to limit learning, the status will becomes YES.

Syntax:

l2-table get learning-exceed

l2-table get learning-exceed port <*PORT_LIST:ports* **| all>**

Parameter:

ports – specify a port list

all - specify all ports

Examples:

This example shows how to get system and port learning-exceed.

```
RTK.0> l2-table get learning-exceed
Learning-exceed = No
RTK.0> l2-table get learning-exceed port 0-1
Port Id 0, learning-exceed = No
Port Id 1, learning-exceed = No
RTK.0>
```

l2-table set src-port-egress-filter

Description:

This command set source port egress filtering function.

Syntax:

l2-table set src-port-egress-filter port <*PORT_LIST:ports* | all> state <disable | enable>

l2-table set src-port-egress-filter ext <*PORT_LIST:ports*> state <disable | enable>

Parameter:

ports	– specify a port list
all	- specify all ports
ext	– specify a extention port list
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to disable source port filtering at port 0-2 and extension port 0-1

```
RTK.0> l2-table set src-port-egress-filter port 0-2 state disable
RTK.0> l2-table set src-port-egress-filter ext 0-1 state disable
RTK.0>
```

l2-table get src-port-egress-filter

Description:

This command can get source port filtering configuration.

Syntax:

l2-table get src-port-egress-filter port <PORT_LIST:ports | all> state

l2-table get src-port-egress-filter ext <PORT_LIST:ports> state

Parameter:

ports	– specify a port list
all	- specify all ports
ext	– specify a extention port list
state	- state configuration

Examples:

This example shows how to get source port filtering for all ports and all extension ports.

```
RTK.0> l2-table get src-port-egress-filter port 0-6

 SRC Port 0 egress filter state: Disable
 SRC Port 1 egress filter state: Disable
 SRC Port 2 egress filter state: Disable
 SRC Port 3 egress filter state: Enable
 SRC Port 4 egress filter state: Enable
 SRC Port 5 egress filter state: Enable
 SRC Port 6 egress filter state: Enable
RTK.0> l2-table get src-port-egress-filter ext 0-5

 EXT Port 0 egress filter state: Disable
 EXT Port 1 egress filter state: Disable
 EXT Port 2 egress filter state: Enable
 EXT Port 3 egress filter state: Enable
 EXT Port 4 egress filter state: Enable
 EXT Port 5 egress filter state: Enable
RTK.0>
```

l2-table clear learning-exceed

Description:

This command can clear the learning-exceed status.

Syntax:

l2-table clear learning-exceed

l2-table clear learning-exceed port <PORT_LIST:ports | all>

Parameter:

ports – specify a port list

all - specify all ports

Examples:

This example shows how to clear the learning-exceed status.

```
RTK.0> l2-table clear learning-exceed
Learning-exceed clear
RTK.0> l2-table clear learning-exceed port 0-6
Port Id 0, learning-exceed clear
Port Id 1, learning-exceed clear
Port Id 2, learning-exceed clear
Port Id 3, learning-exceed clear
Port Id 4, learning-exceed clear
Port Id 5, learning-exceed clear
Port Id 6, learning-exceed clear
RTK.0>
```

l2-table set aging-out

Description:

This command set the state of aging-out.

Syntax:

l2-table set aging-out port < *PORT_LIST:ports* | all > state <disable | enable>

Parameter:

ports	– specify a port list
all	- specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to disable aging-out at port 0-2.

```
RTK.0> l2-table set aging-out port 0-2 state disable
RTK.0>
```

l2-table get aging-out

Description:

This command get the state of aging-out.

Syntax:

l2-table get aging-out port <*PORT_LIST:ports* | all > state

Parameter:

ports	– specify a port list
all	- specify all ports
state	- state configuration

Examples:

This example shows how to get the state of aging-out.

```
RTK.0> l2-table get aging-out port 0-6

 Port 0 Age state: Disable
 Port 1 Age state: Disable
 Port 2 Age state: Disable
 Port 3 Age state: Enable
 Port 4 Age state: Enable
 Port 5 Age state: Enable
 Port 6 Age state: Enable
RTK.0>
```

l2-table add ip-mcast dip

Description:

This command used to add an IP multicast DIP only entry

Syntax:

l2-table add ip-mcast dip *dip* **port <***PORT_LIST:ports* **| all | none>**
l2-table add ip-mcast dip *dip* **ext <***PORT_LIST:ext* **| all | none>**
l2-table add ip-mcast dip *dip* **priority** *priority*
l2-table add ip-mcast dip *dip* **port <***PORT_LIST:ports* **| all> l3-interface** *index*
l2-table add ip-mcast dip *dip* **<priority | l3routing | forcedl3routing> state <disable | enable>**

Parameter:

dip	- destination ip address
ports	- specified port list
all	– specify all ports
none	- specify no port
state	- state configuration
disable	- disable configuration
enable	- enable configuration
priority	- priority configuration
ext	– specify a extention port list
l3-interface	- specify layer 3 routing interface
l3routing	- layer 3 routing configuration
forcedl3routing	- forced layer 3 routing configuration

Examples:

This example shows how to add an IP multicast DIP only entry.
DIP = 224.1.1.1
Portmask = Port 0-2
Enable Priority assignment and the priority is 6.

```
RTK.0> l2-table add ip-mcast dip 224.1.1.1 port 0-2

LUT address: 0x0000 (2K LUT)
RTK.0> l2-table add ip-mcast dip 224.1.1.1 priority state enable

LUT address: 0x0000 (2K LUT)
```

```
RTK.0> l2-table add ip-mcast dip 224.1.1.1 priority 6

LUT address: 0x0000 (2K LUT)
RTK.0>
```

l2-table get ip-mcast dip

Description:

This command can get an IP multicast DIP only entry

Syntax:

l2-table get ip-mcast dip *dip*

Parameter:

dip - destination ip address

Examples:

This example shows how to get an IP multicast DIP only entry which DIP = 224.1.1.1.

```
RTK.0> l2-table get ip-mcast dip 224.1.1.1

LUT address: 0x0000 (2K LUT)
DestinationIP   Member Fwd Pri State  Ext    DipOnly ForceExt
L3Route L3Idx
--------------- ------ --- --- ------ ------ ------- --------
------- -----
224.1.1.1       0-2    En  6   Auto         En      Dis     Dis
0

RTK.0>
```

l2-table del ip-mcast dip

Description:

This command can delete a IP multicast DIP only entry.

Syntax:

l2-table del ip-mcast dip *dip*

Parameter:

dip - destination ip address

Examples:

This example shows how to delete an IP multicast DIP only entry which DIP = 224.1.1.1.

```
RTK.0> l2-table del ip-mcast dip 224.1.1.1
RTK.0>
```

l2-table add ip-mcast sip dip

Description:

This command add an IP multicast entry with both DIP and SIP.

Syntax:

l2-table add ip-mcast sip *sip* **dip** *dip* **port <***PORT_LIST:ports* **| all | none>**

l2-table add ip-mcast sip *sip* **dip** *dip* **ext <***PORT_LIST:ext* **| all | none>**

l2-table add ip-mcast sip *sip* **dip** *dip* **priority** *priority*

l2-table add ip-mcast sip *sip* **dip** *dip* **priority state <disable | enable>**

Parameter:

sip	- source ip address
dip	- destination ip address
ports	- specified port list
all	– specify all ports
none	- specify no port
state	- state configuration
disable	- disable configuration
enable	- enable configuration
priority	- priority configuration
ext	– specify a extention port list

Examples:

This example shows how to add an IP multicast entry.
DIP = 224.1.1.1
SIP = 10.1.1.1
Portmask = Port 0-2
Enable Priority assignment and the priority is 6.

```
RTK.0> l2-table add ip-mcast sip 10.1.1.1 dip 224.1.1.1 port
0-2

LUT address: 0x0000 (2K LUT)
RTK.0> l2-table add ip-mcast sip 10.1.1.1 dip 224.1.1.1
priority state enable
```

```
LUT address: 0x0000 (2K LUT)
RTK.0> l2-table add ip-mcast sip 10.1.1.1 dip 224.1.1.1
priority 6

LUT address: 0x0000 (2K LUT)
RTK.0>
```

l2-table get ip-mcast sip dip

Description:

This command get an IP multicast entry with both DIP and SIP.

Syntax:

l2-table get ip-mcast sip *sip* **dip** *dip*

Parameter:

sip - source ip address
dip - destination ip address

Examples:

This example shows how to get an IP multicast entry.
DIP = 224.1.1.1
SIP = 10.1.1.1

```
RTK.0> l2-table get ip-mcast sip 10.1.1.1 dip 224.1.1.1

LUT address: 0x0000 (2K LUT)
DestinationIP  SourceIP     Member Fwd Pri State  Ext   DipOnly
-------------- ------------ ------ --- --- ------ ------
-------
224.1.1.1      10.1.1.1             Dis 0   Auto         Dis

RTK.0>
```

l2-table del ip-mcast sip dip

Description:

The command can delete an IP multicast entry with DIP and SIP

Syntax:

l2-table del ip-mcast sip *sip* **dip** *dip*

Parameter:

sip	- source ip address
dip	- destination ip address

Examples:

This example shows how to delete an IP multicast entry.

DIP = 224.1.1.1

SIP = 10.1.1.1

```
RTK.0> l2-table del ip-mcast sip 10.1.1.1 dip 224.1.1.1
RTK.0>
```

l2-table add ip-mcast vid dip

Description:

This command can add an IP multicast entry with VID & DIP

Syntax:

l2-table add ip-mcast vid *vid* **dip** *dip* **port <***PORT_LIST:ports* **| all | none>**

l2-table add ip-mcast vid *vid* **dip** *dip* **ext <***PORT_LIST:ext* **| all | none>**

l2-table add ip-mcast vid *vid* **dip** *dip* **priority** *priority*

l2-table add ip-mcast vid *vid* **dip** *dip* **priority state <disable | enable>**

Parameter:

vid	– specify the numeric VLAN identifier
dip	- destination ip address
ports	- specified port list
all	– specify all ports
none	- specify no port
state	- state configuration
disable	- disable configuration
enable	- enable configuration
priority	- priority configuration
ext	– specify a extention port list

Examples:

This example shows how to add an IP multicast entry.

DIP = 224.1.1.1

VID = 10

Portmask = Port 0-2

Enable Priority assignment and the priority is 6.

```
RTK.0> l2-table add ip-mcast vid 10 dip 224.1.1.1 port 0-2

LUT address: 0x0000 (2K LUT)
RTK.0> l2-table add ip-mcast vid 10 dip 224.1.1.1 priority
state enable

LUT address: 0x0000 (2K LUT)
```

```
RTK.0> l2-table add ip-mcast vid 10 dip 224.1.1.1 priority 6

LUT address: 0x0000 (2K LUT)
RTK.0>
```

l2-table get ip-mcast vid dip

Description:

This command can get an IP multicast entry with VID & DIP

Syntax:

l2-table get ip-mcast vid *vid* **dip** *dip*

Parameter:

vid – specify the numeric VLAN identifier
dip - destination ip address

Examples:

This example shows how to get an IP multicast entry.
DIP = 224.1.1.1
VID = 10

```
LUT address: 0x0000 (2K LUT)
DestinationIP  SourceIP     Member Fwd Pri State  Ext   DipOnly
-------------- -------------- ------ --- --- ------ ------
-------
224.1.1.1      10.1.1.1       0-2    En  6   Auto         Dis

RTK.0>
```

l2-table del ip-mcast vid dip

Description:

This command can delete an IP multicast entry with VID & DIP.

Syntax:

l2-table del ip-mcast vid *vid* **dip** *dip*

Parameter:

vid – specify the numeric VLAN identifier
dip - destination ip address

Examples:

This example shows how to delete an IP multicast entry.

```
DIP = 224.1.1.1
VID = 10

RTK.0> l2-table del ip-mcast vid 10 dip 224.1.1.1
RTK.0>
```

l2-table add mac-mcast filter-id

Description:

This command can add a L2 SVL mulitcast entry

Syntax:

l2-table add mac-mcast filter-id *fid* **mac-address** *mac* **port <***PORT_LIST:ports* **| all | none>**

l2-table add mac-mcast filter-id *fid* **mac-address** *mac* **ext <***PORT_LIST:ext* **| all | none>**

l2-table add mac-mcast filter-id *fid* **mac-address** ***mac*** **priority** *priority*

l2-table add mac-mcast filter-id *fid* **mac-address** *mac* **priority state <disable | enable>**

Parameter:

fid	- filter-id
mac	- mac address
ports	- specified port list
all	– specify all ports
none	- specify no port
state	- state configuration
disable	- disable configuration
enable	- enable configuration
priority	- priority configuration
ext	– specify a extention port list

Examples:

This example shows how to add a L2 SVL multicast entry.
MAC = 01:00:5E:01:02:03
FID = 2
Portmask = Port 0-2

```
RTK.0> l2-table add mac-mcast filter-id 2 mac-address
01:00:5E:01:02:03 port 0-2

LUT address: 0x0000 (2K LUT)
RTK.0>
```

l2-table get mac-mcast filter-id

Description:

This command can get a L2 SVL multicast entry.

Syntax:

l2-table get mac-mcast filter-id *fid* **mac-address** *mac*

Parameter:

fid - filter-id
mac - mac address

Examples:

This example shows how to get a L2 SVL multicast entry.
MAC = 01:00:5E:01:02:03
FID = 2

```
RTK.0> l2-table get mac-mcast filter-id 2 mac-address
01:00:5E:01:02:03

LUT address: 0x0000 (2K LUT)
MACAddress         Member FID  FwdPriEn Pri Ext
------------------ ------ ---- -------- --- ------
01:00:5E:01:02:03         2    Dis      0

RTK.0>
```

l2-table del mac-mcast filter-id

Description:

This command can delete a L2 SVL multicast entry.

Syntax:

l2-table del mac-mcast filter-id *fid* **mac-address** *mac*

Parameter:

fid	- filter-id
mac	- mac address

Examples:

This example shows how to delete a L2 SVL multicast entry.

```
MAC = 01:00:5E:01:02:03
FID = 2

RTK.0> l2-table del mac-mcast filter-id 2 mac-address
01:00:5E:01:02:03
RTK.0>
```

l2-table add mac-mcast vid

Description:

This command can add a L2 IVL mulitcast entry

Syntax:

l2-table add mac-mcast vid *vid* **mac-address** *mac* **port** **<***PORT_LIST:ports* **| all | none>**

l2-table add mac-mcast vid *vid* **mac-address** *mac* **ext** **<***PORT_LIST:ext* **| all | none>**

l2-table add mac-mcast vid *vid* **mac-address** *mac* **priority** *priority*

l2-table add mac-mcast vid *vid* **mac-address** *mac* **priority state** **<disable | enable>**

Parameter:

vid	– specify the numeric VLAN identifier
mac	- mac address
ports	- specified port list
all	– specify all ports
none	- specify no port
state	- state configuration
disable	- disable configuration
enable	- enable configuration
priority	- priority configuration
ext	– specify a extention port list

Examples:

This example shows how to add a L2 IVL multicast entry.

```
MAC = 01:00:5E:01:02:03
VID = 10
Portmask = Port 0-2

RTK.0> l2-table add mac-mcast vid 10 mac-address
01:00:5E:01:02:03 port 0-2

LUT address: 0x0000 (2K LUT)
RTK.0>
```

l2-table get mac-mcast vid

Description:

This command can get a L2 IVL mulitcast entry

Syntax:

l2-table get mac-mcast vid *vid* **mac-address** *mac*

Parameter:

vid – specify the numeric VLAN identifier
mac - mac address

Examples:

This example shows how to get a L2 IVL multicast entry.
MAC = 01:00:5E:01:02:03
VID = 10

```
RTK.0> l2-table get mac-mcast vid 10 mac-address
01:00:5E:01:02:03

LUT address: 0x0524 (2K LUT)
MACAddress        Member VID  FwdPriEn Pri Ext
----------------- ------ ---- -------- --- ------
01:00:5E:01:02:03  0-2    10   Dis      0

RTK.0>
```

l2-table del mac-mcast vid

Description:

This command can delete a L2 IVL mulitcast entry

Syntax:

l2-table del mac-mcast vid *vid* **mac-address** *mac*

Parameter:

vid – specify the numeric VLAN identifier
mac - mac address

Examples:

This example shows how to delete a L2 IVL multicast entry.
MAC = 01:00:5E:01:02:03
VID = 10

```
RTK.0> l2-table del mac-mcast vid 10 mac-address
01:00:5E:01:02:03
RTK.0>
```

l2-table add mac-ucast vid

Description:

This command can add a L2 IVL unicast entry

Syntax:

l2-table add mac-ucast vid *vid* **mac-address** *mac* **spn** *port*
l2-table add mac-mcast vid *vid* **mac-address** *mac* **ext-spn** *port*
l2-table add mac-ucast vid *vid* **mac-address** *mac* **filter-id** *fid*
l2-table add mac-ucast vid *vid* **mac-address** *mac* **age** *age*
l2-table add mac-ucast vid ***vid*** **mac-address** *mac* **priority** *priority*
l2-table add mac-ucast vid ***vid*** **mac-address** *mac* **<priority | sa-priority | arp-usage | auth | da-block | sa-block | static> state <disable | enable>**

Parameter:

vid	– specify the numeric VLAN identifier
mac	- mac address
spn	- specify learning source port
ext-spn	– specify learing extention port
fid	- filter-id
age	- aging time
priority	- priority configuration
state	- state configuration
disable	- disable configuration
enable	- enable configuration
priority	– lookup priority
sa-priority	- source address lookup priority
arp-usage	- entry for arp record
auth	- 1x authed
da-block	- destination address block
sa-block	- source address block
static	- static entry

Examples:

This example shows how to add a L2 IVL unicast entry.
MAC = 00:01:02:03:04:05
VID = 10
Source port = Port 2
Static = ENABLED

```
RTK.0> l2-table add mac-ucast vid 10 mac-address
00:01:02:03:04:05 static state enable

LUT address: 0x0000 (2K LUT)
RTK.0> l2-table add mac-ucast vid 10 mac-address
00:01:02:03:04:05 spn 2

LUT address: 0x0000 (2K LUT)
RTK.0>
```

l2-table get mac-ucast vid

Description:

This command can get a L2 IVL unicast entry

Syntax:

l2-table get mac-ucast vid *vid* **mac-address** *mac*

Parameter:

vid – specify the numeric VLAN identifier

mac - mac address

Examples:

This example shows how to get a L2 IVL unicast entry.

MAC = 00:01:02:03:04:05

VID = 10

```
RTK.0> l2-table get mac-ucast vid 10 mac-address
00:01:02:03:04:05

LUT address: 0x0000 (2K LUT)
                       Spa/                           Priority
Block
MACAddress        Ext Fid Efid Age Vid  State  LRN Sa  Fwd Pri
Auth Da  Sa  Arp
----------------- --- --- ---- --- ---- ------ --- --- --- ---
---- --- --- ---
00:01:02:03:04:05 0/0 0   0    0   10   Auto   IVL Dis Dis 0   Dis
Dis Dis Dis

RTK.0>
```

l2-table del mac-ucast vid

Description:

This command can delete a L2 IVL unicast entry

Syntax:

l2-table del mac-ucast vid *vid* **mac-address** *mac*

Parameter:

vid – specify the numeric VLAN identifier
mac - mac address

Examples:

This example shows how to delete a L2 IVL unicast entry.
MAC = 00:01:02:03:04:05
VID = 10

```
RTK.0> l2-table del mac-ucast vid 10 mac-address
00:01:02:03:04:05
RTK.0>
```

l2-table add mac-ucast filter-id

Description:

This command can add a L2 SVL unicast entry

Syntax:

l2-table add mac-ucast filter-id *fid* **mac-address** *mac* **spn** *port*

l2-table add mac-mcast filter-id *fid* **mac-address** *mac* **ext-spn** *port*

l2-table add mac-ucast filter-id *fid* **mac-address** *mac* **filter-id** *fid*

l2-table add mac-ucast filter-id *fid* **mac-address** *mac* **age** *age*

l2-table add mac-ucast filter-id *fid* **mac-address** *mac* **priority** *priority*

l2-table add mac-ucast filter-id *fid* **mac-address** *mac* **<priority | sa-priority | arp-usage | auth | da-block | sa-block | static> state <disable | enable>**

Parameter:

fid	- filter-id
mac	- mac address
spn	- specify learning source port
ext-spn	– specify learing extention port
fid	- filter-id
age	- aging time
priority	- priority configuration
state	- state configuration
disable	- disable configuration
enable	- enable configuration
priority	– lookup priority
sa-priority	- source address lookup priority
arp-usage	- entry for arp record
auth	- 1x authed
da-block	- destination address block
sa-block	- source address block
static	- static entry

Examples:

This example shows how to add a L2 SVL unicast entry.

MAC = 00:01:02:03:04:05

```
Filter-ID = 5
Source port = Port 2
Static = ENABLED

RTK.0> l2-table add mac-ucast filter-id 5 mac-address
00:01:02:03:04:05 static state enable

LUT address: 0x0000 (2K LUT)
RTK.0> l2-table add mac-ucast filter-id 5 mac-address
00:01:02:03:04:05 spn 2

LUT address: 0x0000 (2K LUT)
RTK.0>
```

l2-table get mac-ucast filter-id

Description:

This command can get a L2 SVL unicast entry

Syntax:

l2-table get mac-ucast filter-id *fid* **mac-address** *mac*

Parameter:

fid - filter-id
mac - mac address

Examples:

This example shows how to get a L2 SVL unicast entry.
MAC = 00:01:02:03:04:05
Filter-ID = 5

```
RTK.0> l2-table get mac-ucast filter-id 5 mac-address
00:01:02:03:04:05

LUT address: 0x0000 (2K LUT)
                    Spa/                          Priority      Block
MACAddress        Ext Fid Efid Age Vid  State  LRN Sa  Fwd Pri
Auth Da  Sa  Arp
------------------ --- --- ---- --- ---- ------ --- --- --- ---
---- --- --- ---
00:01:02:03:04:05 0/0 5   0    0   0    Auto   SVL Dis Dis 0   Dis
Dis Dis Dis

RTK.0>
```

l2-table del mac-ucast filter-id

Description:

This command can delete a L2 SVL unicast entry

Syntax:

l2-table del mac-ucast filter-id *fid* **mac-address** *mac*

Parameter:

fid	- filter-id
mac	- mac address

Examples:

This example shows how to delete a L2 SVL unicast entry.

```
MAC = 00:01:02:03:04:05
Filter-ID = 5

RTK.0> l2-table del mac-ucast filter-id 5 mac-address
00:01:02:03:04:05

RTK.0>
```

l2-table set mac-ucast enhanced-filter-id

Description:

This command can set enhanced filter ID. Enhanced filter ID is on hash key for unicast entry. When users add an unicast entry, enhance filter ID set by this command will be used.

Syntax:

l2-table set mac-ucast enhanced-filter-id *efid*

Parameter:

efid - enhanced filter-id

Examples:

This example shows how to set enhanced fileter ID to 3.

```
RTK.0> l2-table set mac-ucast enhanced-filter-id 3
RTK.0>
```

l2-table get entry

Description:

This command can get an entry by specifying a fixed index at Lookup table.

Syntax:

l2-table get entry address *address*

Parameter:

address – specify entry address

Examples:

This example shows how to get entry index 1000.

```
RTK.0> l2-table get entry address 1000

LUT address: 0x0000 (2K LUT)
L2 Unicast table:
MACAddress        Spa Fid Efid Age Vid  State  Hash
00:00:00:00:00:00 0   0   0    0   0    Auto   SVL
SaPriEn FwdPriEn Pri Auth DaBlock SaBlock Arp ExtDsl
Dis     Dis      0   Dis  Dis     Dis     Dis 0

RTK.0> l2-table set mac-ucast enhanced-filter-id 3
```

l2-table del all

Description:

This command can clear entire Lookup table.

Syntax:

l2-table del all [include-static]

Parameter:

include-static - include static entry

Examples:

This example shows how to clear entire Lookup table including static entry

```
RTK.0> l2-table del all include-static
RTK.0>
```

l2-table get next-entry

Description:

This command can get next entry after specified index. If specified index is a valid entry, this entry will be returned. If specified index is a non-valid entry, the next valid entry will be returned.

Syntax:

l2-table get next-entry address *address*
l2-table get next-entry mac-ucast address *address*
l2-table get next-entry mac-ucast address *address* **spn** *port*
l2-table get next-entry l2-mcast address *address*
l2-table get next-entry ip-mcast address *address*
l2-table get next-entry l2-ip-mcast address *address*

Parameter:

address	– specify entry address
mac-ucast	- unicast entry
spn	- specify learning source port
l2-mcast	- l2 multicast entry
ip-mcast	- ip multicast entry
l2-ip-mcast	- either l2 or ip multicast entry

Examples:

This example shows how to get next valid entry after index 1000.

```
RTK.0> l2-table get next-entry address 1000

LUT address: 0x0000 (2K LUT)
L2 Unicast table:
MACAddress        Spa Fid Efid Age Vid  State  Hash
00:00:00:00:00:00 0   0   0    0   0    Auto   SVL
SaPriEn FwdPriEn Pri Auth DaBlock SaBlock Arp ExtDsl
Dis     Dis      0   Dis  Dis     Dis     Dis 0

RTK.0>
```

l2-table set link-down-flush

Description:

This command can configure link-down-flush feature. By enabling this feature, all dynamic entry will be flushed when a port is like down.

Syntax:

l2-table set link-down-flush state <disable | enable>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to configure link-down-flush feature to disabled.

```
RTK.0> l2-table set link-down-flush state enable
RTK.0>
```

l2-table get link-down-flush

Description:

This command can get current state of link-down-flush feature.

Syntax:

l2-table get link-down-flush state

Parameter:

state - state configuration

Examples:

This example shows how to get link-down flush state.

```
RTK.0> l2-table get link-down-flush state

 Link down flush state: Enable
RTK.0>
```

l2-table set flush mac-ucast

Description:

This command can flush unicast entries in Lookup table.

Syntax:

l2-table set flush mac-ucat [include-static]

l2-table set flush mac-ucat static-only

l2-table set flush mac-ucat port <*PORT_LIST:ports* | all> [include-static]

l2-table set flush mac-ucat port <*PORT_LIST:ports* | all> static-only

l2-table set flush mac-ucat port <*PORT_LIST:ports* | all> filter-id *fid* **[include-static]**

l2-table set flush mac-ucat port <*PORT_LIST:ports* | all> filter-id *fid* **static-only**

l2-table set flush mac-ucat port <*PORT_LIST:ports* | all> vid *vid* **[include-static]**

l2-table set flush mac-ucat port <*PORT_LIST:ports* | all> vid *vid* **static-only**

Parameter:

ports	- specified port list
all	– specify all ports
include-static	- include static entry
Static-only	- static entry only
fid	- filter-id
vid	– specify the numeric VLAN identifier

Examples:

This example shows how to flush all dynamic entry at Port 2 only.

```
RTK.0> l2-table set flush mac-ucast port 2
RTK.0>
```

l2-table get learning-count

Description:

This command can get current dynamic entry learning counter for a specified port or system.

Syntax:

l2-table get learning-count

l2-table get learning-count port <*PORT_LIST:ports* **| all >**

Parameter:

ports – specify a port list

all - specify all ports

Examples:

This example shows how to get current learnig counter at port 2.

```
RTK.0> l2-table get learning-count port 2

 Port 2 learning counter: 0
RTK.0>
```

l2-table set ip-mcast-mode

Description:

This command can set hash mode for IPv4 multicast packet.

Syntax:

l2-table set ip-mcast-mode <dip-and-sip | dip-and-vid | vid-and-mac>

Parameter:

dip-and-sip	- lookup hashing with dip and sip
dip-and-vid	- lookup hashing with dip and vid
vid-and-mac	- lookup hashing with vid and mac address

Examples:

This example shows how to configure IPv4 multicast hash methed to DIP and SIP.

```
RTK.0> l2-table set ip-mcast-mode dip-and-sip
RTK.0>
```

l2-table get ip-mcast-mode

Description:

This command can get hash mode for IPv4 multicast packet.

Syntax:

l2-table get ip-mcast-mode

Parameter:

None

Examples:

This example shows how to get IPv4 multicast hash method

```
RTK.0> l2-table get ip-mcast-mode

 IPMC mode : DIP + SIP
RTK.0>
```

l2-table set lookup-miss action

Description:

This command can configure the per port lookup miss (IPv4 multicast/IPv6 multicast/L2 multicast/Unicat) packets action.

Syntax:

l2-table set lookup-miss port <*PORT_LIST:ports* | all > < ip-mcast | ip6-mcast> action <drop | flood-in-vlan | trap-to-cpu>

l2-table set lookup-miss port <*PORT_LIST:ports* | all > multicast action <drop | drop-exclude-rma | flood-in-vlan | trap-to-cpu>

l2-table set lookup-miss port <*PORT_LIST:ports* | all > unicast action <drop | flood-in-vlan | trap-to-cpu>

Parameter:

ports	– specify a port list
all	- specify all ports
ip-mcast	- ipv4 unknown multicast
ip6-mcast	- ipv6 unknown multicast
multicast	- l2 unknown multicast
unicast	- unknown unicast
drop	- drop packet
drop-exclude-rma	- drop packets but exclude rma
flood-in-vlan	- flood in vlan
trap-to-cpu	- trap packet to cpu port

Examples:

This example shows how to configure lookup miss packet action:
Port 0 unknown L2 multicast: Drop but exclude RMA
Port 1 unknown IPv4 multicast: Trap to CPU
Port 2 unknown IPv6 multicast: Flood in VLAN
Port 3 unknown unicast: Drop

```
RTK.0> l2-table set lookup-miss port 0 multicast action
drop-exclude-rma
RTK.0> l2-table set lookup-miss port 1 ip-mcast action
trap-to-cpu
RTK.0> l2-table set lookup-miss port 2 ip6-mcast action
flood-in-vlan
RTK.0> l2-table set lookup-miss port 3 unicast action drop
```

```
RTK.0>
```

l2-table get lookup-miss action

Description:

This command can get the per port lookup miss (IPv4 multicast/IPv6 multicast/L2 multicast/Unicat) packets action.

Syntax:

l2-table get lookup-miss port <*PORT_LIST:ports***> | all> <ip-mcast | ip6-mcast | multicast | unicast >**

Parameter:

ports	- specified port list
all	– specify all ports
ip-mcast	- ipv4 unknown multicast
ip6-mcast	- ipv6 unknown multicast
multicast	- l2 unknown multicast
unicast	- unknown unicast

Examples:

This example shows how to get L2 muliticast lookup miss packet action at port 0-3

```
RTK.0> l2-table get lookup-miss port 0-3 multicast

 Port 0 lookup-miss multicast Action: Drop exclude RMA
 Port 1 lookup-miss multicast Action: Forward
 Port 2 lookup-miss multicast Action: Forward
 Port 3 lookup-miss multicast Action: Forward
RTK.0>
```

l2-table set ip-mcast-data action

Description:

This command can set per port IP multicast packets receiving state. If user set this feature to drop, all ip multicast received will be dropped.

Syntax:

l2-table set ip-mcast-data port <*PORT_LIST:ports* **| all > action <drop | forward>**

Parameter:

ports	– specify a port list
all	- specify all ports
drop	- drop packet
forward	- forward packet

Examples:

This example shows how to configure the action of IP multicast packets at port 0 to “drop”

```
RTK.0> l2-table set ip-mcast-data port 0 action drop
RTK.0>
```

l2-table get ip-mcast-data action

Description:

This command can get per port IP multicast packets receiving state.

Syntax:

l2-table get ip-mcast-data port <*PORT_LIST:ports* **| all > action**

Parameter:

ports	– specify a port list
all	- specify all ports

Examples:

This example shows how to get the action of IP multicast packets at port 0-6

```
RTK.0> l2-table get ip-mcast-data port 0 action

 Port 0 IPMC action: Drop
RTK.0>
```

l2-table set port-move action

Description:

This command can configure source MAC address movement action. Only when the action is configured to “forward”, the MAC address will be moved to new source port.

Syntax:

l2-table set port-move action <copy-to-cpu | drop | forward | trap-to-cpu>

Parameter:

drop	- drop packet
trap-to-cpu	- trap packet to cpu port
forward	- forward packet
copy-to-cpu	- copy packet to cpu port

Examples:

This example shows how to configure port-move action to “drop”.

```
RTK.0> l2-table set port-move port 0-6 action drop
RTK.0>
```

l2-table get port-move action

Description:

This command can get source MAC address movement action.

Syntax:

l2-table get port-move action

Parameter:

None

Examples:

This example shows how to get port-move action

```
RTK.0> l2-table get port-move port 0-6

 Port 0 Port move Action: Drop
 Port 1 Port move Action: Drop
 Port 2 Port move Action: Drop
 Port 3 Port move Action: Drop
 Port 4 Port move Action: Drop
 Port 5 Port move Action: Drop
 Port 6 Port move Action: Drop
RTK.0>
```

l2-table set unknown-sa action

Description:

This command can set the action of unknown source MAC address.

Syntax:

l2-table set unknown-sa action <copy-to-cpu | drop | forward | trap-to-cpu>

Parameter:

drop	- drop packet
trap-to-cpu	- trap packet to cpu port
forward	- forward packet
copy-to-cpu	- copy packet to cpu port

Examples:

This example shows how to configure the action of unknown source MAC address to “trap to CPU”.

```
RTK.0> l2-table set unknown-sa port 0-6 action trap-to-cpu
RTK.0>
```

l2-table get unknown-sa action

Description:

This command can get the action of unknown source MAC address.

Syntax:

l2-table get unknown-sa action

Parameter:

None

Examples:

This example shows how to get the action of unknown source MAC address

```
RTK.0> l2-table get unknown-sa port 0-6 action

 Port 0 unknown SA Action: Trap to CPU
 Port 1 unknown SA Action: Trap to CPU
 Port 2 unknown SA Action: Trap to CPU
 Port 3 unknown SA Action: Trap to CPU
 Port 4 unknown SA Action: Trap to CPU
 Port 5 unknown SA Action: Trap to CPU
 Port 6 unknown SA Action: Trap to CPU
RTK.0>
```

l2-table set lookup-miss flood-ports

Description:

This command can configure the flooding portmask of lookup miss packets.

Syntax:

l2-table set lookup-miss <broadcast | unicast | multicast> flood-ports <*PORT_LIST:ports* | all | none>

Parameter:

broadcast	- broadcast
unicast	- unicast
multicast	- multicast
ports	- specified port list
all	– specify all ports
none	- specify no port

Examples:

This example shows how to configure the flooding portmask of broadcast / unknown unicast / unknown multicast packets:
Broadcast: flood to all ports
Unknown unicast: flood to port 1
Unknown multicast: flood to port 0-2

```
RTK.0> l2-table set lookup-miss broadcast flood-ports 0-6
RTK.0> l2-table set lookup-miss unicast flood-ports 1
RTK.0> l2-table set lookup-miss multicast flood-ports 0-2
RTK.0>
```

l2-table get lookup-miss flood-ports

Description:

This command can get the flooding portmask of lookup miss packets.

Syntax:

l2-table get lookup-miss <broadcast | unicast | multicast>

Parameter:

broadcast	- broadcast
unicast	- unicast
multicast	- multicast

Examples:

This example shows how to get the flooding portmask of broadcast / unknown unicast / unknown multicast packets:

```
RTK.0> l2-table get lookup-miss broadcast flood-ports

 Lookup-miss Broadcast Lookup miss flood portmask: 0-6
RTK.0> l2-table get lookup-miss unicast flood-ports

 Lookup-miss Unicast Lookup miss flood portmask: 1
RTK.0> l2-table get lookup-miss multicast flood-ports

 Lookup-miss Multicast Lookup miss flood portmask: 0-2
RTK.0>
```

l2-table set lookup-miss multicast trap-priority

Description:

This command can configure the priority for trapped unknown multicast packets.

Syntax:

l2-table set lookup-miss multicast trap-priority *priority*

Parameter:

priority - priority configuration

Examples:

This example shows how to configure the priority for trapped unknown multicast packets as 2:

```
RTK.0> l2-table set lookup-miss multicast trap-priority 2
RTK.0>
```

l2-table get lookup-miss multicast trap-priority

Description:

This command can get the priority for trapped unknown multicast packets.

Syntax:

l2-table get lookup-miss multicast trap-priority

Parameter:

None

Examples:

This example shows how to get the priority for trapped unknown multicast packets.

```
RTK.0> l2-table get lookup-miss multicast trap-priority

 Lookup-miss multicast trap-priority: 2
RTK.0>
```

l2-table set lookup-miss multicast reserved-flooding

Description:

This command can configure the action of reserved unknown IP multicast address packet. The reserved IP multicast is 224.0.0.0 ~ 224.0.0.255 for IPv4 and FFXX::00XX for IPv6.

Syntax:

l2-table set lookup-miss multicast reserved-flooding <disable | enable>

Parameter:

disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to configure the action of unknown reserved IP multicast packets to “flood”.

```
RTK.0> l2-table set lookup-miss multicast reserved-flooding
enable
RTK.0>
```

l2-table get lookup-miss reserved-flooding

Description:

This command can get the action of reserved unknown IP multicast address packet.

Syntax:

l2-table get lookup-miss reserved-flooding

Parameter:

None

Examples:

This example shows how to get the action of unknown reserved IP multicast packets

```
RTK.0> l2-table get lookup-miss multicast reserved-flooding

 Unknown reserved IP multicast Flooding: Enable
RTK.0>
```

2.5. trap commands

trap init

Description:

This command can initialize trap related functions.

Syntax:

trap init

Parameter:

None

Examples:

This example shows how to initial trap reltated functions.

```
RTK.0> trap init
RTK.0>
```

trap set cdp

Description:

This command can configure the action of CDP packets

Syntax:

trap set cdp action <drop | forward | forward-exclude-cpu | trap-to-cpu>

Parameter:

drop	- drop packet
trap-to-cpu	- trap packet to cpu port
forward	- forward packet
forward-exclude-cpu	- forward packet exclude to cpu port

Examples:

This example shows how to configure the action of CDP packet to “Drop”

```
RTK.0> trap set cdp action drop
RTK.0>
```

trap get cdp

Description:

This command can get the action of CDP packets

Syntax:

trap get cdp action

Parameter:

None

Examples:

This example shows how to get the action of CDP packets

```
RTK.0> trap get cdp action
CDP Drop
RTK.0>
```

trap set csstp

Description:

This command can configure the action CSSTP packets

Syntax:

trap set csstp action <drop | forward | forward-exclude-cpu | trap-to-cpu>

Parameter:

drop	- drop packet
trap-to-cpu	- trap packet to cpu port
forward	- forward packet
forward-exclude-cpu	- forward packet exclude to cpu port

Examples:

This example shows how to configure the action of CSSTP packet to Drop.

```
RTK.0> trap set csstp action drop
RTK.0>
```

trap get csstp

Description:

Thie command can get the action of CSSTP packets

Syntax:

trap get csstp action

Parameter:

None

Examples:

This example shows how to get the action of CSSTP packet

```
RTK.0> trap get csstp action
CSSTP Drop
RTK.0>
```

2.6. rma commands

rma set action

Description:

This command can configure the action of RMA (Reserved Multicast Address) packets. Make sure that ‘trap init’ command is executed before.

Syntax:

rma set address *rma_tail* **action <drop | forward | forward-exclude-cpu | trap-to-cpu>**

Parameter:

rma_tail	- last byte of 01:80:C2:00:00:xx address
drop	- drop packet
trap-to-cpu	- trap packet to cpu port
forward	- forward packet
forward-exclude-cpu	- forward packet exclude to cpu port

Examples:

This example shows how to configure the action of MAC_DA = 01:80:C2:00:00:01 to “trap to CPU”.

```
RTK.0> rma set address 01 action trap-to-cpu
RTK.0>
```

rma get action

Description:

This command can get the action of RMA (Reserved Multicast Address) packets. Make sure that ‘trap init’ command is executed before.

Syntax:

rma get address *rma_tail* **action**

Parameter:

rma_tail - last byte of 01:80:C2:00:00:xx address

Examples:

This example shows how to get the action of MAC_DA = 01:80:C2:00:00:01 packets.

```
RTK.0> rma get address 01 action
RMA 01-80-C2-00-00-01 action: Trap to CPU
RTK.0>
```

rma set priority

Description:

This command can configure the trapping priority when for RMA (Reserved Multicast Address) packets. Make sure that ‘trap init’ command is executed before.

Syntax:

rma set priority *priority*

Parameter:

priority - priority configuration

Examples:

This example shows how to configure the trapping priority of RMA packet to 4.

```
RTK.0> rma set priority 4
RTK.0>
```

rma get priority

Description:

This command can get the trapping priority when for RMA (Reserved Multicast Address) packets. Make sure that ‘trap init’ command is executed before.

Syntax:

rma get priority

Parameter:

None

Examples:

This example shows how to get the trapping priority of RMA packet.

```
RTK.0> rma get priority
The RMA trap priority = 4
RTK.0>
```

rma dump

Description:

This command can get all supported RMA configurations. Make sure that ‘trap init’ command is executed before.

Syntax:

rma dump

Parameter:

None

Examples:

This example shows how to get all RMA related configurations.

```
RTK.0> rma dump
RMA 01-80-C2-00-00-xx
trail action
00    Forward       01    Drop          02    Drop
03    Forward       04    Forward       05    Forward
06    Forward       07    Forward       08    Forward
09    Forward       0a    Forward       0b    Forward
0c    Forward       0d    Forward       0e    Forward
0f    Forward       10    Forward       11    Forward
12    Forward       13    Forward       14    Forward
15    Forward       16    Forward       17    Forward
18    Forward       19    Forward       1a    Forward
1b    Forward       1c    Forward       1d    Forward
1e    Forward       1f    Forward       20    Forward
21    Forward       22    Forward       23    Forward
24    Forward       25    Forward       26    Forward
27    Forward       28    Forward       29    Forward
2a    Forward       2b    Forward       2c    Forward
2d    Forward       2e    Forward       2f    Forward
RTK.0>
```

2.7. igmp commands

igmp set action

Description:

This command can configure action of IGMP/MLD packets. Make sure that ‘trap init’ command is executed before.

Syntax:

igmp set <igmpv1 | igmpv2 | igmpv3 | mldv1 | mldv2> port <*PORT_LIST:ports* | all> action <drop | forward | trap-to-cpu>

Parameter:

ports	– specify a port list
all	- specify all ports
igmpv1	- igmp version 1
igmpv2	- igmp version 2
igmpv3	- igmp version 3
mldv1	- mld version 1
mldv2	- mld version 2
drop	- drop packet
forward	- forward packet
trap-to-cpu	- trap packet to cpu port

Examples:

This example shows how to set IGMP/MLD packet action at Port 0 to “Trap to CPU”.

```
RTK.0> igmp set igmpv1 port 0 action trap-to-cpu
RTK.0> igmp set igmpv2 port 0 action trap-to-cpu
RTK.0> igmp set igmpv3 port 0 action trap-to-cpu
RTK.0> igmp set mldv1 port 0 action trap-to-cpu
RTK.0> igmp set mldv2 port 0 action trap-to-cpu
RTK.0>
```

igmp get action

Description:

This command can configure action of IGMP/MLD packets. Make sure that ‘trap init’ command is executed before.

Syntax:

trap get <igmpv1 | igmpv2 | igmpv3 | mldv1 | mldv2> port <*PORT_LIST:ports* **| all> action**

Parameter:

ports – specify a port list
all - specify all ports

Examples:

This example shows how to get IGMPv2 packet action at Port 0.

```
RTK.0> igmp get igmpv2 port 0 action

 Port: 0, IGMPv2 : Trap to CPU
RTK.0>
```

igmp set igmp-mld checksum-error

Description:

This command can configure action of IGMP/MLD checksum error packets. Make sure that ‘trap init’ command is executed before.

Syntax:

igmp set igmp-mld checksum-error action <drop | forward | trap-to-cpu>

Parameter:

drop	- drop packet
forward	- forward packet
trap-to-cpu	- trap packet to cpu port

Examples:

This example shows how to configure the action of IGMP/MLD checksum error packet as drop.

```
RTK.0> igmp set igmp-mld checksum-error action drop
RTK.0>
```

igmp get igmp-mld checksum-error

Description:

This command can get action of IGMP/MLD checksum error packets. Make sure that ‘trap init’ command is executed before.

Syntax:

igmp get igmp-mld checksum-error action

Parameter:

None

Examples:

This example shows how to get the action of IGMP/MLD checksum error packet

```
RTK.0> igmp get igmp-mld checksum-error action
IGMP/MLD Checksum error Action: Drop
RTK.0>
```

igmp set igmp-mld isolation-leaky

Description:

This command can configure the state of IGMP/MLD isolation leaky function. Make sure that ‘trap init’ command is executed before.

Syntax:

igmp set igmp-mld isolation-leaky state <disable | enable>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to configure the state of IGMP/MLD isolation leaky function as enabled.

```
RTK.0> igmp set igmp-mld isolation-leaky state enable
RTK.0>
```

igmp get igmp-mld isolation-leaky

Description:

This command can get the state of IGMP/MLD isolation leaky function. Make sure that ‘trap init’ command is executed before.

Syntax:

igmp get igmp-mld isolation-leaky state

Parameter:

None

Examples:

This example shows how to get the state of IGMP/MLD isolation leaky function.

```
RTK.0> igmp get igmp-mld isolation-leaky state
IGMP/MLD Isolation Leaky: Enable
RTK.0>
```

igmp set igmp-mld vlan-leaky

Description:

This command can configure the state of IGMP/MLD VLAN leaky function. Make sure that ‘trap init’ command is executed before.

Syntax:

igmp set igmp-mld vlan-leaky state <disable | enable>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to configure the state of IGMP/MLD VLAN leaky function as enabled.

```
RTK.0> igmp set igmp-mld vlan-leaky state enable
RTK.0>
```

igmp get igmp-mld vlan-leaky

Description:

This command can get the state of IGMP/MLD VLAN leaky function. Make sure that ‘trap init’ command is executed before.

Syntax:

igmp get igmp-mld vlan-leaky state

Parameter:

None

Examples:

This example shows how to get the state of IGMP/MLD VLAN leaky function.

```
RTK.0> igmp get igmp-mld vlan-leaky state
IGMP/MLD VLAN Leaky: Enable
RTK.0>
```

igmp set ip-mcast-lookup-mode

Description:

This command can configure IP multicast lookup mode. Make sure that ‘l2 init’ command is executed before.

Syntax:

igmp set ip-mcast-lookup-mode <dip-and-sip | dip-only>

Parameter:

dip-and-sip - lookup hashing with dip and sip
dip-only - lookup hashing with dip only

Examples:

This example shows how to configure IP multicast lookup mode to “DIP only”.

```
RTK.0> igmp set ip-mcast-lookup-mode dip-only
RTK.0>
```

igmp get ip-mcast-lookup-mode

Description:

This command can get IP multicast lookup mode. Make sure that ‘l2 init’ command is executed before.

Syntax:

igmp get ip-mcast-lookup-mode

Parameter:

None

Examples:

This example shows how to get IP multicast lookup mode.

```
RTK.0> igmp get ip-mcast-lookup-mode

 IPMC lookup mode: DIP only
RTK.0>
```

igmp set ip-mcast-table

Description:

This command can set an entry into IP multicast table.

Syntax:

igmp set ip-mcast-table index *index* **group-ip** *dip* **port** **<***PORT_LIST:ports* **| all | none>**

Parameter:

index	- entry index
dip	- destination ip address
ports	- specified port list
all	– specify all ports
none	- specify no port

Examples:

This example shows how to set an entry into index 0 of IP multicast table.

Group IP: 224.1.2.3

Port: 0-6

```
RTK.0> igmp set ip-mcast-table index 0 group-ip 224.1.2.3 port 0-6
RTK.0>
```

igmp get ip-mcast-table

Description:

This command shows how to get an entry from IP multicast table.

Syntax:

igmp get ip-mcast-table index *index*

Parameter:

index - entry index

Examples:

This example shows how to get index 0 from IP multicast table.

```
RTK.0> igmp get ip-mcast-table index 0

 Index    : 0
 DIP      : 224.1.2.3
 Portmask  : 0-6
RTK.0>
```

2.8. Storm Control commands

storm-control set broadcast state

Description:

This command can enable the storm finction. System only supports 4 storm type enable at same time. Before configuration storm control, user must enable the storm type first. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control set broadcast state <disable | enable>
storm-control set multicast state <disable | enable>
storm-control set unknown-multicast state <disable | enable>
storm-control set unknown-unicast state <disable | enable>
storm-control set arp-storm <disable | enable>
storm-control set dscp-storm state <disable | enable>
storm-control set igmp-mld-storm state <disable | enable>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to enable broadcast-storm, multicast-storm, dscp-storm and arp-storm.
System only supports 4 storm type enable at same time.

```
RTK.0> storm-control set broadcast state enable
RTK.0> storm-control set multicast state enable
RTK.0> storm-control set unknown-multicast state disable
RTK.0> storm-control set unknown-unicast state disable
RTK.0> storm-control set arp-storm state enable
RTK.0> storm-control set dhcp-storm state enable
RTK.0> storm-control set igmp-mld-storm state disable
RTK.0>
```

storm-control set boradcast

Description:

This command can configure broadcast storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control set broadcast state <disable | enable>

storm-control set broadcast port <*PORT_LIST:ports* **| all> state <disable | enable>**

storm-control set broadcast port <*PORT_LIST:ports* **| all> meter** *index*

Parameter:

ports	– specify a port list
all	- specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
meter	- policing meter configuration

Examples:

This example shows how to configure broadcast storm enable at port 0 and set the meter index as 0. The strom rate limit set to 8kbps.

```
RTK.0> meter set entry 0 rate 8
RTK.0> storm-control set broadcast state enable
RTK.0> storm-control set broadcast port 0 state enable
RTK.0> storm-control set broadcast port 0 meter 0
```

storm-control get boradcast

Description:

This command can get the broadcast storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control get broadcast

storm-control get broadcast port <*PORT_LIST:ports* **| all>**

Parameter:

ports – specify a port list

all - specify all ports

Examples:

This example shows how to get broadcast storm control function setting.

```
RTK.0> storm-control get broadcast
State: Enable
RTK.0> storm-control get broadcast port 0-6
Port State   Meter
0    Enable  0
1    Disable 0
2    Disable 0
3    Disable 0
4    Disable 0
5    Disable 0
6    Disable 0
RTK.0>
```

storm-control set multicast

Description:

This command can configure multicast storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control set multicast state <disable | enable>

storm-control set multicast port <*PORT_LIST:ports* **| all> state <disable | enable>**

storm-control set multicast port <*PORT_LIST:ports* **| all> meter** *index*

Parameter:

ports	– specify a port list
all	- specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
meter	- policing meter configuration

Examples:

This example shows how to configure multicast storm enable at port 0 and set the meter index as 0. The strom rate limit set to 8 kbps.

```
RTK.0> meter set entry 0 rate 8
RTK.0> storm-control set multicast state enable
RTK.0> storm-control set multicast port 0 state enable
RTK.0> storm-control set multicast port 0 meter 0
```

storm-control get multicast

Description:

This command can get the multicast storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control get multicast

storm-control get multicast port <*PORT_LIST:ports* | all>

Parameter:

ports – specify a port list

all - specify all ports

Examples:

This example shows how to get multicast storm control function setting.

```
RTK.0> storm-control get multicast
State: Enable
RTK.0> storm-control get multicast port 0-6
Port State   Meter
0    Enable  0
1    Disable 0
2    Disable 0
3    Disable 0
4    Disable 0
5    Disable 0
6    Disable 0
RTK.0>
```

storm-control set unknown-multicast

Description:

This command can configure unknown-multicast storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control set unknown-multicast state <disable | enable>

storm-control set unknown-multicast port <*PORT_LIST:ports* **| all> state <disable | enable>**

storm-control set unknown-multicast port <*PORT_LIST:ports* **| all> meter** *index*

Parameter:

ports	– specify a port list
all	- specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
meter	- policing meter configuration

Examples:

This example shows how to configure unknown-multicast storm enable at port 0 and set the meter index as 0. The unknown-multicast strom rate limit set to 8 kbps.

```
RTK.0> meter set entry 0 rate 8
RTK.0> storm-control set unknown-multicast state enable
RTK.0> storm-control set unknown-multicast port 0 state enable
RTK.0> storm-control set unknown-multicast port 0 meter 0
```

storm-control get unknown-multicast

Description:

This command can get the unknown-multicast storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control get unknown-multicast

storm-control get unknown-multicast port <*PORT_LIST:ports* **| all>**

Parameter:

ports	– specify a port list
all	- specify all ports
alternated	- alternated function

Examples:

This example shows how to get unknown-multicast storm control function setting.

```
RTK.0> storm-control get unknown-multicast
State: Enable
RTK.0> storm-control get unknown-multicast port 0-6
Port State  Meter
0    Enable  0
1    Disable 0
2    Disable 0
3    Disable 0
4    Disable 0
5    Disable 0
6    Disable 0
RTK.0>
```

storm-control set unknown-unicast

Description:

This command can configure unknown-unicast storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control set unknown-unicast state <disable | enable>

storm-control set unknown-unicast port <*PORT_LIST:ports* **| all> state <disable | enable>**

storm-control set unknown-unicast port <*PORT_LIST:ports* **| all> meter** *index*

Parameter:

ports	– specify a port list
all	- specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
meter	- policing meter configuration

Examples:

This example shows how to configure unknown-unicast storm enable at port 0 and set the meter index as 0. The unknown-unicast strom rate limit set to 8 kbps.

```
RTK.0> meter set entry 0 rate 8
RTK.0> storm-control set unknown-unicast state enable
RTK.0> storm-control set unknown-unicast port 0 state enable
RTK.0> storm-control set unknown-unicast port 0 meter 0
```

storm-control get unknown-unicast

Description:

This command can get the unknown-unicast storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control get unknown-unicast

storm-control get unknown-unicast port <*PORT_LIST:ports* **| all>**

Parameter:

ports	– specify a port list
all	- specify all ports
alternated	- alternated function

Examples:

This example shows how to get unknown-unicast storm control function setting.

```
RTK.0> storm-control get unknown-unicast
State: Enable
RTK.0> storm-control get unknown-unicast port 0-6
Port State   Meter
0    Enable  0
1    Disable 0
2    Disable 0
3    Disable 0
4    Disable 0
5    Disable 0
6    Disable 0
RTK.0>
```

storm-control set arp-storm

Description:

This command can configure arp storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control set arp-storm state <disable | enable>

storm-control set arp-storm port <*PORT_LIST:ports* **| all > state <disable | enable>**

storm-control set arp-storm port <*PORT_LIST:ports* **| all > meter** *index*

Parameter:

ports	– specify a port list
all	- specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
meter	- policing meter configuration

Examples:

This example shows how to configure arp-storm enable at port 0 and set the meter index as 0. The arp-storm rate limit set to 8 kbps.

```
RTK.0> meter set entry 0 rate 8
RTK.0> storm-control set arp-storm state enable
RTK.0> storm-control set arp-storm port 0 state enable
RTK.0> storm-control set arp-storm port 0 meter 0
```

storm-control get arp-storm

Description:

This command can get the arp storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control get arp-storm

storm-control get arp-storm port <*PORT_LIST:ports* **| all>**

Parameter:

ports – specify a port list

all - specify all ports

Examples:

This example shows how to get arp-storm control function setting.

```
RTK.0> storm-control get arp-storm
State: Enable
RTK.0> storm-control get arp-storm port 0-6
Port State   Meter
0    Enable  0
1    Disable 0
2    Disable 0
3    Disable 0
4    Disable 0
5    Disable 0
6    Disable 0
RTK.0>
```

storm-control set dhcp-storm

Description:

This command can configure dhcp storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control set dhcp-storm state <disable | enable>

storm-control set dhcp-storm port <*PORT_LIST:ports***> | all> state <disable | enable>**

storm-control set dhcp-storm port <*PORT_LIST:ports***> | all> meter** *index*

Parameter:

ports	– specify a port list
all	- specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
meter	- policing meter configuration

Examples:

This example shows how to configure dhcp-storm enable at port 0 and set the meter index as 0. The dhcp-storm rate limit set to 8 kbps.

```
RTK.0> meter set entry 0 rate 8
RTK.0> storm-control set dhcp-storm state enable
RTK.0> storm-control set dhcp-storm port 0 state enable
RTK.0> storm-control set dhcp-storm port 0 meter 0
```

storm-control get dhcp-storm

Description:

This command can get the dhcp storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control get dhcp-storm

storm-control get dhcp-storm port <*PORT_LIST:ports* **| all**

Parameter:

ports – specify a port list

all - specify all ports

Examples:

This example shows how to get dhcp-storm control function setting.

```
RTK.0> storm-control get dhcp-storm
State: Enable
RTK.0> storm-control get dhcp-storm port 0-6
Port State   Meter
0    Enable  0
1    Disable 0
2    Disable 0
3    Disable 0
4    Disable 0
5    Disable 0
6    Disable 0
RTK.0>
```

storm-control set igmp-mld-storm

Description:

This command can configure igmp-mld storm relative setting. Make sure that 'rate init' command is executed before.

Syntax:

storm-control set igmp-mld-storm state <disable | enable>

storm-control set igmp-mld-storm port <*PORT_LIST:ports* | all > state <disable | enable>

storm-control set igmp-mld-storm port <*PORT_LIST:ports* | all > meter *index*

Parameter:

ports	– specify a port list
all	- specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
meter	- policing meter configuration

Examples:

This example shows how to configure igmp-mld-storm enable at port 0 and set the meter index as 0. The igmp-mld-storm rate limit set to 8 kbps.

```
RTK.0> meter set entry 0 rate 8
RTK.0> storm-control set igmp-mld-storm state enable
RTK.0> storm-control set igmp-mld-storm port 0 state enable
RTK.0> storm-control set igmp-mld-storm port 0 meter 0
```

storm-control get igmp-mld-storm

Description:

This command can get the igmp-mld storm relative setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control get igmp-mld-storm

storm-control get igmp-mld-storm port <*PORT_LIST:ports* **| all**

Parameter:

ports	– specify a port list
all	- specify all ports

Examples:

This example shows how to get igmp-mld-storm control function setting.

```
RTK.0> storm-control get igmp-mld-storm
State: Enable
RTK.0> storm-control get igmp-mld-storm port 0-6
Port State   Meter
0    Enable  0
1    Disable 0
2    Disable 0
3    Disable 0
4    Disable 0
5    Disable 0
6    Disable 0
RTK.0>
```

storm-control set bypass-packet

Description:

This command can set the bypass strom packet type. Make sure that 'rate init' command is executed before.

Syntax:

storm-control set bypass-packet <igmp | cdp | csstp> state <disable | enable>

storm-control set bypass-packet rma *rma_tail* **state <disable | enable>**

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration
igmp	- igmp protocol
cdp	- cisco discovery protocol
csstp	- cisco shared spanning tree protocol
rma_tail	- last byte of 01:80:C2:00:00:xx address

Examples:

This example shows how to bypass cisco discovery protocol packet, cisco shared spanning tree protocol and BPDU packet.

```
RTK.0> storm-control set bypass-packet cdp state enable
RTK.0> storm-control set bypass-packet csstp state enable
RTK.0> storm-control  set bypass-packet rma 0x00 state enable
```

storm-control get bypass-packet

Description:

Get storm bypass-packet setting. Make sure that ‘rate init’ command is executed before.

Syntax:

storm-control get bypass-packet <igmp | cdp | csstp> state
storm-control get bypass-packet rma *rma_tail* **state**

Parameter:

state	- state configuration
igmp	- igmp protocol
cdp	- cisco discovery protocol
csstp	- cisco shared spanning tree protocol
rma_tail	- last byte of 01:80:C2:00:00:xx address

Examples:

This example show how to get storm-control bypass packet setting.

```
RTK.0> storm-control get bypass-packet cdp state
CDP bypass state: Enable
RTK.0> storm-control get bypass-packet csstp state
CSSTP bypass state: Enable
RTK.0> storm-control get bypass-packet igmp state
IGMP bypass state: Disable
RTK.0> storm-control get bypass-packet rma 0x0 state
RMA 01-80-C2-00-00-00 bypass Storm-Control state: Enable
RTK.0>
```

2.9. stp commands

stp init

Description:

This command can initialize spanning tree protocol (STP) related functions.

Syntax:

stp init

Parameter:

None

Examples:

This example shows how to initial STP reltated functions.

```
RTK.0> stp init
RTK.0>
```

stp set state

Description:

Per port and per STP instance set STP port state.

Syntax:

stp set stp-table instance *instance* **port <***PORT_LIST:ports* **| all> state <disable | blocking | forwarding | learning>**

Parameter:

ports	– specify a port list
all	- specify all ports
state	- state configuration
disable	- disable state
blocking	- blocking state
forwarding	- forwarding state
learning	- learning state

Examples:

This command shows how to set STP port state to “blocking” for port 2 STP instance 0.

```
RTK.0> stp set stp-table instance 0 port 2 state blocking
```

stp get state

Description:

Get STP port state

Syntax:

stp get stp-table instance *instance* **port <***PORT_LIST:ports* **| all> state**

Parameter:

Ports	– specify a port list
All	- specify all ports
State	- state configuration

Examples:

This command shows how to get STP port state for port 2 STP instance 0.

```
RTK.0> stp get stp-table instance 0 port 2 state
MSTI 0 Status:
        Port  2: BLOCKING
```

2.10. Mirror commands

mirror init

Description:

This command can reset & initialize Mirror module.

Syntax:

mirror init

Parameter:

None

Examples:

This example shows how to reset and initialize mirror module.

```
RTK.0> mirror init
RTK.0>
```

mirror set mirroring

Description:

This command can configure the mirror function.

Syntax:

mirror set mirroring-port *port* **mirrored-port port** *<PORT_LIST:ports* | **none>** **[rx-mirror] [tx-mirror]**

Parameter:

mirroring-port	- specify the mirroring port
mirrored-port	- specify the mirrored port
ports	- specified port list
none	- specify no port
rx-mirror	- mirror rx packets
tx-mirror	- mirror rX packets

Examples:

This example shows how to mirror both TX & RX packets from port 0 to Port 1.

```
RTK.0> mirror set mirroring-port 1 mirrored-port 0
RTK.0>
```

mirror dump

Description:

This command can get the mirror function

Syntax:

mirror dump

Parameter:

None

Examples:

This example shows get current mirror configuration.

```
RTK.0> mirror dump

 Monitor port: 1
 Mirroring TX portmask: none
 Mirroring RX portmask: none

 Mirror Egress Mode: All packets

RTK.0>
```

mirror set egress-mode

Description:

This command can configure egress-mode of mirror function

Syntax:

mirror set egress-mode <all-pkt | mirrored-only>

Parameter:

all-pkt	- all packets
mirrored-only	- mirrored packets

Examples:

This example shows how to set the egress-mode to “mirrored-only”.

```
RTK.0> mirror set egress-mode mirrored-only
RTK.0>
```

mirror get egress-mode

Description:

This command can get egress-mode of mirror function

Syntax:

mirror get egress-mode

Parameter:

None

Examples:

This example shows how to get the egress-mode.

```
RTK.0> mirror get egress-mode

 Mirror Egress Mode: Mirrored packets only
RTK.0>
```

2.11. meter commands

meter init

Description:

This command can reset & initialize Meter module.

Syntax:

meter init

Parameter:

None

Examples:

This example shows how to initialize Meter module.

```
RTK.0> meter init
RTK.0>
```

meter set entry

Description:

These command sets can set meter relative parameter for each meter entry.

The meter parameter would be:

- Burst size
- Packet rate
- Rate counting include IFG or not

Syntax:

meter set entry *index* **burst-size** *size*
meter set entry *index* **rate** *rate*
meter set entry *index* **ifg <include | exclude>**

Parameter:

index	- index of meter entry
burst-size	- burst size configuration
rate	- packet rate, unit kpbs
ifg	- inter-frame gap configuration
include	- include configuration
exclude	- exclude configuration

Examples:

This example shows how to set the packet rate to 1024 kbps and rate counting is including IFG for meter entry 0.

```
RTK.0> meter set entry 0 rate 1024
RTK.0> meter set entry 0 ifg include
RTK.0>
```

meter get entry

Description:

Get setting for each meter entry.

Syntax:

meter get entry *index*
meter get entry *index* **burst-size**
meter get entry *index* **rate**
meter get entry *index* **ifg**

Parameter:

index	- index of meter entry
burst-size	- burst size configuration
rate	- packet rate, unit kpbs
ifg	- inter-frame gap configuration

Examples:

This example shows how get meter setting for meter entry 0-4.

```
RTK.0> meter get entry 0-4
Meter idx = 0 , meter rate = 1024 , Kbps include IFG = Enable  ,
burst size = 0
Meter idx = 1 , meter rate = 0    , Kbps include IFG = Disable ,
burst size = 0
Meter idx = 2 , meter rate = 0    , Kbps include IFG = Disable ,
burst size = 0
Meter idx = 3 , meter rate = 0    , Kbps include IFG = Disable ,
burst size = 0
Meter idx = 4 , meter rate = 0    , Kbps include IFG = Disable ,
burst size = 0
```

meter get meter-exceed

Description:

Chen the meter staus is exceed or not.

Syntax:

meter get entry *index* **meter-exceed**

Parameter:

index - index of meter entry

Examples:

This example shows how to get meter exceed status for meter mentry 0-4.

```
RTK.0> meter get entry 0-4 meter-exceed
Meter idx = 0, meter exceed = No
Meter idx = 1, meter exceed = No
Meter idx = 2, meter exceed = No
Meter idx = 3, meter exceed = No
Meter idx = 4, meter exceed = No
```

meter reset meter-exceed

Description:

Reset meter exceed status.

Syntax:

meter reset entry *index* **meter-exceed**

Parameter:

index - index of meter entry

Examples:

This example shows how to clear meter exceed status for meter entry 0.

```
RTK.0> meter reset entry 0 meter-exceed
RTK.0>
```

meter set tick-token

Description:

This command can set the tick-token for meter global configuration.

Syntax:

meter set tick-token tick-period *period* token *token*

Parameter:

tick-period - timing tick, unit 1/system clock frequency
Token - toke size, unit byte

Examples:

This example shows how to set tick-period to 48 and token size set to 66 byte.

```
RTK.0> meter set tick-token tick-period 48 token 66
RTK.0>
```

meter get tick-token

Description:

Get meter tick token setting.

Syntax:

meter get tick-token

Parameter:

None

Examples:

This example shows how get meter tick-token setting.

```
RTK.0> meter get tick-token
tick period:48, token:66
RTK.0>
```

meter set pon-tick-token

Description:

This command can set the tick-token setting for pon port.

Syntax:

meter set pon-tick-token tick-period *period* token *token*

Parameter:

tick-period	- timing tick, unit 1/system clock frequency
Token	- toke size, unit byte

Examples:

This example shows how to set tick-period to 48 and token size set to 66 byte for pon port.

```
RTK.0> meter set pon-tick-token tick-period 48 token 66
RTK.0>
```

meter get pon-tick-token

Description:

This command can get the tick-token setting for pon port.

Syntax:

meter get pon-tick-token

Parameter:

None

Examples:

This example shows how to get meter tick-token setting for pon port.

```
RTK.0> meter get pon-tick-token
tick period:48, token:66
RTK.0>
```

2.12. bandwidth commands

bandwidth init

Description:

This command can reset & initialize bandwidth control module.

Syntax:

bandwidth init

Parameter:

None

Examples:

This example shows how to initial bandwidth control related functions.

```
RTK.0> bandwidth init
RTK.0>
```

bandwidth set egress ifg

Description:

This command can configure egress bandwidth control, which is included ifg calcuation or not.

Syntax:

bandwidth set egress ifg <include | exclude>

bandwidth set egress ifg port <*PORT_LIST:ports* **| all> <include | exclude>**

Parameter:

Include	- include configuration
Exclude	- exclude configuration
Ports	- specified port list
All	– specify all ports

Examples:

This example shows how to configure port 2 and port 3 egress bandwidth control which is included ifg calculation.

```
RTK.0> bandwidth set egress ifg port 2-3 include
RTK.0>
```

bandwidth get egress ifg

Description:

This command can get configuration of egress bandwidth control, which is included ifg calcuation or not.

Syntax:

bandwidth get egress ifg

bandwidth get egress ifg port <*PORT_LIST:ports* **| all>**

Parameter:

ports - specified port list

all – specify all ports

Examples:

This example shows how to get port 1 egress bandwidth control included ifg or not.

```
RTK.0> bandwidth get egress ifg port 1
port:1 Egress Rate counting ifg: exclude
RTK.0>
```

bandwidth set egress port

Description:

This command can configure per port or per queue egress rate. After configuring the egree queue APR rate, the configed rate meter needs to configure by meter commands and different port using different meter index range mapping to APR index. For 86900 chip, meter index and arp index mapping is as below.

Port	Meter-Range
0,4	0~7
1,5	8~15
2	16~23
3	24~31

Syntax:

bandwidth set egress port <*PORT_LIST:ports* **| all> rate** *rate*

bandwidth set egress port <*PORT_LIST:ports* **| all> qid** *qid* **apr-index** *index*

bandwidth set egress port <*PORT_LIST:ports* **| all> qid** *qid* **shared-bandwidth state <disable | enable>**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
rate	- packet rate, unit kpbs
qid	- specify the numeric queue identifier
arp-index	- meter index for arp
shared-bandwidth	- average shared bandwidth configuration

Examples:

This example shows how to configure port 2 egress rate to 100Mbps and egress queue 0/1 apr rate to related meter index 0/1.

```
RTK.0> bandwidth set egress port 2 rate 100000
RTK.0> bandwidth set egress port 2 queue-id 0 apr-index 0
RTK.0> bandwidth set egress port 2 queue-id 0 share-bandwidth state enable
```

```
RTK.0> bandwidth set egress port 2 queue-id 1 apr-index 1
RTK.0> bandwidth set egress port 2 queue-id 1share-bandwidth
state enable
RTK.0> meter set entry 16 rate 10240
RTK.0> meter set entry 17 rate 10240
RTK.0>
```

bandwidth get egress port

Description:

This command can get egress port rate and queue ARP index usage. It needs to use meter commands for retrieving related ARP rate of queue-id belong to desired egress port.

Syntax:

bandwidth get egress port <*PORT_LIST:ports* **| all>**

bandwidth get egress port <*PORT_LIST:ports* **| all> qid** *MASK_LIST:qid*

Parameter:

ports	- specified port list
all	– specify all ports
qid	- specify the numeric queue identifier

Examples:

This example shows how to egress bandwidth control of egress port 1.

```
RTK.0> bandwidth get egress port 1
port: 1  rate:1048568
          queue: 0  apr-index: 0
          queue: 1  apr-index: 0
          queue: 2  apr-index: 0
          queue: 3  apr-index: 0
          queue: 4  apr-index: 0
          queue: 5  apr-index: 0
          queue: 6  apr-index: 0
          queue: 7  apr-index: 0
RTK.0> meter get entry 8
Meter idx = 8 , meter rate = 0    , Kbps include IFG = Disable ,
burst size = 0
RTK.0>
```

bandwidth set ingress port

Description:

This command can configure per port ingress rate.

Syntax:

bandwidth set ingress port <*PORT_LIST:ports* **| all> rate** *rate*

Parameter:

ports	- specified port list
all	– specify all ports
rate	- packet rate, unit kpbs

Examples:

This example shows how to enable port 1 ingress rate with 10Mpbs configuration.

```
RTK.0> bandwidth set ingress port 1 rate 10240
RTK.0>
```

bandwidth get ingress port

Description:

This command can get per port ingress rate configuration.

Syntax:

bandwidth get ingress port <*PORT_LIST:ports* **| all>**

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get port 0-3 ingress bandwidth rate.

```
RTK.0> bandwidth get ingress port 0-3 rate
port: 0 rate:1024
port: 1 rate:10240
port: 2 rate:1048568
port: 3 rate:1048568
RTK.0>
```

bandwidth set ingress ifg

Description:

This command can configure ingress bandwidth control, which is included ifg calcuation or not.

Syntax:

bandwidth set ingress ifg port <*PORT_LIST:ports* **| all> <include | exclude>**

Parameter:

include	- include configuration
exclude	- exclude configuration
ports	- specified port list
all	– specify all ports

Examples:

This example shows how to configure port 2 and port 3 ingress bandwidth control which is included ifg calculation.

```
RTK.0> bandwidth set ingress ifg port 2-3 include
RTK.0>
```

bandwidth get ingress ifg

Description:

This command can get configuration of ingress bandwidth control, which is included ifg calcuation or not.

Syntax:

bandwidth get ingress ifg port <*PORT_LIST:ports* | all>

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get port 1 egress bandwidth control included ifg or not.

```
RTK.0> bandwidth get ingress ifg port 0-3
port: 0 ifg: exclude
port: 1 ifg: exclude
port: 2 ifg: include
port: 3 ifg: include
RTK.0>
```

bandwidth set ingress bypass-packet

Description:

This command can configure ingress rate control, which is bypassed some kinds of packet or not. Differet chip supports different kinds of packet to bypass ingress rate control.

Syntax:

bandwidth set ingress bypass-packet state <disable | enable>

Parameter:

State	- state configuration
Disable	- disable configuration
Enable	- enable configuration

Examples:

This example shows how to enable bypassing RMA, IGMP/MLD control packet and Realtek proprietary control frames for ingress bandwidth contol.

```
RTK.0> bandwidth set ingress bypass-packet state enable
RTK.0>
```

bandwidth get ingress bypass-packet

Description:

This command can get configuration of ingress rate control, which is bypassed some kinds of packet or not.

Syntax:

bandwidth get ingress bypass-packet state

Parameter:

State - state configuration

Examples:

This example shows how get packets bypass state for ingress bandwidth control.

```
RTK.0> bandwidth get ingress bypass-packet state
Ingress Rate byapss:Enable
byapss packet format:
    -DMAC=01-80-C2-00-00-xx
    -IGMP/MLD control packet
    -8899 frames
RTK.0>
```

bandwidth set ingress flow-control

Description:

This command can configure ingress bandwidth without UTP flow-control ability.

Syntax:

bandwidth set ingress flow-control port <*PORT_LIST:ports* **| all> state <disable | enable>**

Parameter:

Ports	- specified port list
All	– specify all ports
State	- state configuration
Disable	- disable configuration
Enable	- enable configuration

Examples:

This example shows how to set port 1-2 to keep flow-control ability while those ports are under rate control of ingress bandwidth.

```
RTK.0> bandwidth set ingress flow-control port 1-2 state enable
RTK.0>
```

bandwidth get ingress flow-control

Description:

This command used to get ingress bandwidth with UTP flow-control ability or not.

Syntax:

bandwidth get ingress flow-control port <*PORT_LIST:ports* **| all> state**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration

Examples:

This example shows how to get flow-control ability of port 0-3 ingress rate control.

```
RTK.0> bandwidth get ingress flow-control port 0-3 state
Port     Flow-Control
0           Disable
1           Enable
2           Enable
3           Disable
RTK.0>
```

2.13. mib commands

mib init

Description:

This command can initialize & reset mib module.

Syntax:

mib init

Parameter:

None

Examples:

This example shows how to initialize mib module.

```
RTK.0> mib init
```

mib dump counter

Description:

This command can display current MIB counter information.

Syntax:

mib dump counter dot1dTpLearnedEntryDiscards
mib dump counter port <*PORT_LIST:ports* | all>
mib dump counter port <*PORT_LIST:ports* | all> <
dot1dTpPortInDiscards | dot3ControlInUnknownOpcodes |
dot3InPauseFrames | dot3OutPauseFrames |
dot3StatsDeferredTransmissions | dot3StatsExcessiveCollisions |
dot3StatsLateCollisions | dot3StatsMultipleCollisionFrames |
dot3StatsSingleCollisionFrames | dot3StatsSymbolErrors |
etherStatsCRCAlignErrors | etherStatsCollisions |
etherStatsDropEvents | etherStatsFragments | etherStatsJabbers |
etherStatsUndersizeDropPkts | etherStatsPkts64Octets |
etherStatsPkts65to127Octets | etherStatsPkts128to255Octets |
etherStatsPkts256to511Octets | etherStatsPkts512to1023Octets |
etherStatsPkts1024to1518Octets | etherStatsPkts1519toMaxOctets |
etherStatsOversizePkts | etherStatsUndersizePkts |
etherStatsTxBroadcastPkts | etherStatsTxMulticastPkts |
inOamPduPkts | outOamPduPkts | ifInOctets | ifInUcastPkts |
ifInMulticastPkts | ifInBroadcastPkts | ifOutOctets |
ifOutUcastPkts | ifOutMulticastPkts | ifOutBroadcastPkts |
ifOutDiscards>

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to display dot1dTpLearnedEntryDiscards counter information.

```
RTK.0> mib dump counter dot1dTpLearnedEntryDiscards
dot1dTpLearnedEntryDiscards: 0
```

This example shows how to display port 0 MIB counter information:

```
RTK.0> mib dump counter port 0
Port: 0
ifInOctets                              :                    0
ifInUcastPkts                           :                    0
ifInMulticastPkts                       :                    0
ifInBroadcastPkts                       :                    0
ifOutOctets                             :                    0
ifOutDiscards                           :                    0
ifOutUcastPkts                          :                    0
ifOutMulticastPkts                      :                    0
ifOutBroadcastPkts                      :                    0
dot1dTpPortInDiscards                   :                    0
dot3InPauseFrames                       :                    0
dot3OutPauseFrames                      :                    0
dot3StatsSingleCollisionFrames          :                    0
dot3StatsMultipleCollisionFrames        :                    0
dot3StatsDeferredTransmissions          :                    0
dot3StatsLateCollisions                 :                    0
dot3StatsExcessiveCollisions            :                    0
dot3StatsSymbolErrors                   :                    0
dot3ControlInUnknownOpcodes             :                    0
etherStatsDropEvents                    :                    0
etherStatsFragments                     :                    0
etherStatsJabbers                       :                    0
etherStatsCollisions                    :                    0
etherStatsCRCAlignErrors                :                    0
etherStatsTxUndersizePkts               :                    0
etherStatsTxOversizePkts                :                    0
etherStatsTxPkts64Octets                :                    0
etherStatsTxPkts65to127Octets           :                    0
etherStatsTxPkts128to255Octets          :                    0
etherStatsTxPkts256to511Octets          :                    0
etherStatsTxPkts512to1023Octets         :                    0
etherStatsTxPkts1024to1518Octets        :                    0
etherStatsTxPkts1519toMaxOctets         :                    0
etherStatsTxBroadcastPkts               :                    0
etherStatsTxMulticastPkts               :                    0
etherStatsRxUndersizePkts               :                    0
etherStatsRxUndersizeDropPkts           :                    0
```

```
etherStatsRxOversizePkts          :                    0
etherStatsRxPkts64Octets          :                    0
etherStatsRxPkts65to127Octets     :                    0
etherStatsRxPkts128to255Octets    :                    0
etherStatsRxPkts256to511Octets    :                    0
etherStatsRxPkts512to1023Octets   :                    0
etherStatsRxPkts1024to1518Octets  :                    0
etherStatsRxPkts1519toMaxOctets   :                    0
inOamPduPkts                      :                    0
outOamPkdPkts                     :                    0
```

mib reset counter

Description:

This command reset MIB counter on a chip.

Syntax:

mib reset counter global

mib reset counter port <*PORT_LIST:ports* **| all>**

Parameter:

ports - specified port list

all – specify all ports

Examples:

This example shows how to reset port 1 MIB counter.

```
RTK.0> mib reset counter port 1
RTK.0>
```

This example shows how to reset global MIB counter.

```
mib reset counter global
RTK.0>
```

mib dump statistic

Description:

This command get ACL statistic counter on a chip.

Syntax:

mib dump statistic index *index*

Parameter:

index - specified statistic counter index

Examples:

This example shows how to get ACL statistic counter on index 0.

```
RTK.0> mib dump statistic index 0
Index 0: 0
```

mib reset statistic

Description:

This command reset ACL statistic counter on a chip.

Syntax:

mib reset statistic index *index*

Parameter:

index - specified statistic counter index

Examples:

This example shows how to reset ACL statistic counter on index 0.

```
RTK.0> mib reset statistic index 0
RTK.0>
```

mib set statistic mode

Description:

This command set ACL statistic counter m(0,2,4,…30) mode on a chip.

Syntax:

mib set statistic mode index *index* **<bits-32 | bits-64>**

Parameter:

index	- specified statistic counter index
bits-32	- 32 bits counter mode
bits-64	- 64 bits counter mode

Examples:

This example shows how to set ACL statistic counter mode of index 0 to bit 32.

```
RTK.0> mib set statistic mode index 0 bits-32
RTK.0>
```

mib get statistic mode

Description:

This command get ACL statistic counter m(0,2,4,…30) mode on a chip.

Syntax:

mib get statistic mode index *index*

Parameter:

index - specified statistic counter index

Examples:

This example shows how to get ACL statistic counter mode on index 0.

```
RTK.0> mib get statistic mode index 0
0: 32-bits
```

mib set statistic type

Description:

This command set ACL statistic counter type(packet/byte) count on a chip.

Syntax:

mib set statistic type index *index* **<byte-count | packet-count>**

Parameter:

index	- specified statistic counter index
byte-count	- byte counting type
packet-count	- packet counting type

Examples:

This example shows how to set ACL statistic counter type to byte count.

```
RTK.0> mib set statistic type index  0 byte-count
RTK.0>
```

mib get statistic type

Description:

This command get ACL statistic counter type(packet/byte) count on a chip.

Syntax:

mib get statistic type index *index*

Parameter:

index	- specified statistic counter index
byte-count	- byte counting type
packet-count	- packet counting type

Examples:

This example shows how to get ACL statistic counter type on index 0.

```
RTK.0> mib get statistic type index 0
0: Byte count
```

mib get packet-debug-reason

Description:

This command get per-port newest packet trap/drop reason and code number on a chip.

Syntax:

mib get packet-debug-reason port <*PORT_LIST:ports* | all>

Parameter:

ports - specified port list
all – specify all ports

Examples:

This example shows how to get port 0 newest packet trap/drop reason.

```
RTK.0> mib get packet-debug-reason port 0
 Port  code  Drop  Trap  Reason
 0      0                  normal
```

mib set sync-mode

Description:

This command set MIB data update mode (freerun/stop sync).

Syntax:

mib set syn-mode <freerun | stop>

Parameter:

freerun	- free running configuration
stop	- stopping configuration

Examples:

This example set MIB data update mode to freerun.

```
RTK.0> mib set sync-mode freerun
RTK.0>
```

mib get sync-mode

Description:

This command get MIB data update mode(freerun/stop sync).

Syntax:

mib get syn-mode

Parameter:

None

Examples:

This example shows how to get MIB update mode.

```
RTK.0> mib get sync-mode
MIB sync mode: normal free run sync
```

mib set reset-value

Description:

This command set mib reset value.

Syntax:

mib set reset-value <0 | 1>

Parameter:

0	- reset to 0
1	- reset to 1

Examples:

This example shows how to set MIB reset value to 0.

```
RTK.0> mib set reset-value 0
RTK.0>
```

mib get reset-value

Description:

This command get MIB reset value.

Syntax:

mib get reset-value

Parameter:

None

Examples:

This example shows how to get MIB reset value.

```
RTK.0> mib get reset-value
Reset MIB counter to: all 0
```

mib set ctag-length

Description:

This command can set RX/TX counters including or excluding C-tag when calculating packet length.

Syntax:

mib set ctag-length <rx-counter | tx-counter> <include | exclude>

Parameter:

rx-counter	- receiving counter
tx-counter	- transmitting counter
include	- include configuration
exclude	- exclude configuration

Examples:

This example shows how to set TX MIB counter to exclude C-tag length.

```
RTK.0> mib set ctag-length tx-counter exclude
RTK.0>
```

mib get ctag-length

Description:

This command can get RX/TX counters including or excluding C-tag when calculating packet length.

Syntax:

mib get ctag-length <rx-counter | tx-counter>

Parameter:

rx-counter	- receiving counter
tx-counter	- transmitting counter

Examples:

This example shows how to get if packet length in TX MIB counter include or exclude C-tag length.

```
RTK.0> mib get ctag-length tx-counter
tx-counter tag length: exclude
```

mib set count-mode

Description:

This command set MIB update mode.

Syntax:

mib set conut-mode freerun

mib set conut-mode by-timer latch-time *timer*

Parameter:

freerun	- free running configuration
by-timer	- running depond on timer setting
timer	- time setting, unit second

Examples:

This example shows how to set MIB data update mode to latch time to 10 seconds.

```
RTK.0> mib set count-mode by-timer latch-time 10
RTK.0>
```

mib get count-mode

Description:

This command get configuration of MIB update mode.

Syntax:

mib get conut-mode

Parameter:

None

Examples:

This example shows how to get MIB data update mode.

```
RTK.0> mib get count-mode
MIB count mode: control by MIB timer
Timer          : 10
```

2.14. ACL commands

acl init

Description:

This command can initial acl module, set acl related config to default, and clear acl entries. Only after executing this command, other acl commands can be executed and will not return FAILED.

Syntax:

acl init

Parameter:

None

Examples:

This example shows how to initial acl functions.

```
RTK.0> acl init
RTK.0> acl get port 0-6 state
Port State
0    Disable
1    Disable
2    Disable
3    Disable
4    Disable
5    Disable
6    Disable
RTK.0>
```

acl add entry

Description:

This command can add acl rule after rule contents are configured. In SVLAN and VLAN action setting, some configurations need to be making sure that related setting is done before adding rule. Otherwise, the adding rule can not be set to hardware or there will be unexpected behaviors.

Syntax:

acl add entry *index*

Parameter:

index - acl rule index

Examples:

This example shows how to add a acl rule to entry index 0 that all ingress packets from port 1 will be assigned priority to 7.

```
RTK.0> acl clear
RTK.0> acl set rule template entry 0
RTK.0> acl set rule state valid
RTK.0> acl set rule port 1
RTK.0> acl set action priority assign-priority 7
RTK.0> acl add entry 0
RTK.0>
```

acl del entry

Description:

This command can delete acl rule entry by desired index or delete all of rules. After rule entry being deleted, the content of entry will be cleared to default.

Syntax:

acl del entry *index*
acl del entry all

Parameter:

index - acl rule index
all - all entries

Examples:

This example shows how to delete acl rule index 1.

```
RTK.0> acl del entry 1
RTK.0> acl get entry 1
Valid: Invalid
Not: Disable
Cvlan action: Ingress Vlan vid: 0
Svlan action: Ingress Vlan svid: 0
Forward: Copy Ports:
Policing: Policing meter: 0
Pri-Remark: ACL priority acl-pri: 0
Interrupt: Disable
Classification: None
ACL index latch: Disable
Template: 0 [0] DMAC0
            [1] DMAC1
            [2] DMAC2
            [3] STAG
            [4] SMAC0
            [5] SMAC1
            [6] SMAC2
            [7] ETHERTYPE
Active Port: none
Tags: Mask:
```

```
field[0] data: 0x0000 mask: 0x0000
field[1] data: 0x0000 mask: 0x0000
field[2] data: 0x0000 mask: 0x0000
field[3] data: 0x0000 mask: 0x0000
field[4] data: 0x0000 mask: 0x0000
field[5] data: 0x0000 mask: 0x0000
field[6] data: 0x0000 mask: 0x0000
field[7] data: 0x0000 mask: 0x0000
RTK.0>
```

acl get entry

Description:

Show acl rule contents and action

Syntax:

acl get entry *index* **[action]**

Parameter:

index - acl rule index
action - rule action data configuration

Examples:

This example shows how to get content and action of acl rule entry 0 and action contents only of entry 1.

```
RTK.0> acl get entry 0
Valid: Valid
Not: Disable
Cvlan action: Disable
Svlan action: Disable
Forward: Redirect Ports: 2
Policing: Disable
Pri-Remark: Disable
Interrupt or Classification: Disable
Template: 0 [0] DMAC0
            [1] DMAC1
            [2] DMAC2
            [3] STAG
            [4] SMAC0
            [5] SMAC1
            [6] SMAC2
            [7] ETHERTYPE
Active Port: 0-6
Tags: Mask:
field[0] data: 0x0000 mask: 0xF0F0
field[1] data: 0x0000 mask: 0xF0F0
field[2] data: 0x0000 mask: 0xF0F0
field[3] data: 0x0000 mask: 0x0000
field[4] data: 0x0000 mask: 0x0000
```

```
field[5] data: 0x0000 mask: 0x0000
field[6] data: 0x0000 mask: 0x0000
field[7] data: 0x0000 mask: 0x0000
RTK.0> acl get entry 1 action
Valid: Valid
Not: Disable
Cvlan action: Disable
Svlan action: Disable
Forward: Disable
Policing: Policing meter: 1
Pri-Remark: ACL priority acl-pri: 7
Interrupt or Classification: Disable
RTK.0>
```

acl show

Description:

Show user configured acl contents and actions of rule be added to entry.

Syntax:

acl show

Parameter:

None

Examples:

This example shows how to show contents and actions of rule be configed before adding to entry.

```
RTK.0> acl clear
RTK.0> acl set rule template entry 0
RTK.0> acl set rule state valid
RTK.0> acl set rule port 0
RTK.0> acl set rule dmac data 0F:0F:0F:0F:0F:0F mask
F0:F0:F0:F0:F0:F0
RTK.0> acl set action priority assign-priority 7
RTK.0> acl show
Valid: Valid
Not: Disable
Cvlan action: Disable
Svlan action: Disable
Forward: Disable
Policing: Disable
Pri-Remark: ACL priority acl-pri: 7
Interrupt or Classification: Disable
Template: 0 [0] DMAC0
            [1] DMAC1
            [2] DMAC2
            [3] STAG
            [4] SMAC0
            [5] SMAC1
            [6] SMAC2
            [7] ETHERTYPE
```

```
Active Port: 0
Tags: Mask:
dmac data: 0F:0F:0F:0F:0F:0F
     mask: F0:F0:F0:F0:F0:F0
RTK.0>
```

acl clear

Description:

This command can clear contents of configured acl rule. It is necessary to use this command before editing and adding new rule to make sure action and rule contents are wanted.

Syntax:

acl clear

Parameter:

None

Examples:

This example shows how to clear pre-configured contents of acl rule which need to add.

```
RTK.0> acl show action
Valid: Valid
Not: Disable
Cvlan action: Disable
Svlan action: Disable
Forward: Redirect Ports: 1
Policing: Acl MIB counter: 1
Pri-Remark: ACL priority acl-pri: 7
Interrupt or Classification: Disable
RTK.0> acl clear
RTK.0> acl show action
Valid: Invalid
Not: Disable
Cvlan action: Disable
Svlan action: Disable
Forward: Disable
Policing: Disable
Pri-Remark: Disable
Interrupt or Classification: Disable
RTK.0>
```

acl set rule

Description:

This command can edit wanted rule contents before being added to device. Different chip version will support different contents to be configured. The rule content is depended on which template is used. So, configuring used template is more important before setting rule.

Syntax:

acl set rule <dmac | smac> data *mac* **[mask** *mac_mask***]**

acl set rule <sip | dip> data *ip* **[mask** *ip_mask***]**

acl set rule <sip6 | dip6> data *ip6* **[mask** *ip6_mask***]**

acl set rule <ethertype | gemport-llid | next-header | ext> data *data* **[mask** *mask***]**

acl set rule <range-l4port | range-vid | range-ip | range-length> care-range *MASK_LIST:list*

acl set rule ctag vid *vid* **priority** *priority* **cfi** *cfi* **[mask vid_mask** *vid_mask* **priority_mask** *priority_mask* **cfi_mask** *cfi_mask***]**

acl set rule stag vid *vid* **priority** *priority* **dei** *dei* **[mask vid_mask** *vid_mask* **priority_mask** *priority_mask* **dei_mask** *dei_mask***]**

acl set rule user-field *index* **data** *data* **[mask** *mask***]**

acl set rule pattern field-index *index* **data** *data* **[mask** *mask***]**

acl set rule field-valid care-field *MASK_LIST:list* **[mask_field** *MASK_LIST:mask***]**

acl set rule care-tags [ctag] [ip] [ipv6] [pppoe] [stag] [tcp] [ucp]

acl set rule port <*PORT_LIST: ports* **| all | none>**

acl set rule template entry *index*

acl set rule operation reverse-state <disable | enable>

acl set rule state <valid | invalid>

Parameter:

data	- rule matching data
mask	- rule matching mask
dmac	- destination mac address
smac	- source mac address
dip	- destination ipv4 address
sip	- source ipv4 address
dip6	- destination ipv6 address

sip6 - source ipv6 address
ethertype - ethernet type or payload lengthlength
gemport-llid - gemport id or llid
next-header - ipv6 next header
range-l4port - layer4 port number range checking list
range-vid - vid range checking list
range-ip - ip range checking list
range-length - packet length range checking list
stag - stag rule matching
priority - priority configuration
vid – specify the numeric VLAN identifier
dei - dei field of stag
ctag - ctag rule matching
cfi - cfi field of ctag
user-field - user defined field
field-valid - user defined field valid list
ext - extention source ports
pattern - rule raw data matching
field-index - rule field index
care-tags - packet types checking
port - rule actived port configuration
ports - specified port list
all – specify all ports
none - specify no port
template - rule matching template usage
state - rule valid state configuration
reverse-state - reverse configuration

Examples:

This example shows how to configure acl rule with specified destination mac, source mac and source ip address for ingress policing action.

```
RTK.0> acl clear
RTK.0> acl set rule template entry 0
RTK.0> acl set rule state valid
RTK.0> acl set rule port 0-6
RTK.0> acl set rule dmac data 01:80:C2:00:00:40
RTK.0> acl set rule smac data 00:01:02:03:04:05
```

```
RTK.0> acl set rule sip data 172.1.2.3
RTK.0> acl set action meter 3
RTK.0> acl show
Valid: Valid
Not: Disable
Cvlan action: Disable
Svlan action: Disable
Forward: Disable
Policing: Policing meter: 3
Pri-Remark: Disable
Interrupt or Classification: Disable
Template: 0 [0] DMAC0
           [1] DMAC1
           [2] DMAC2
           [3] SMAC0
           [4] SMAC1
           [5] SMAC2
           [6] IP4SIP0
           [7] IP4SIP1
Active Port: 0-6
Tags: Mask:
dmac data: 01:80:C2:00:00:40
     mask: FF:FF:FF:FF:FF:FF
smac data: 00:01:02:03:04:05
     mask: FF:FF:FF:FF:FF:FF
sip data: 172.1.2.3
    mask: 255.255.255.255
RTK.0>
```

acl show rule

Description:

Show user configured acl contents of rule which will be added to entry.

Syntax:

acl show rule

Parameter:

None

Examples:

This example shows how to show contents of rule have been configed before adding to entry.

```
RTK.0> acl clear
RTK.0> acl set rule template entry 0
RTK.0> acl set rule state valid
RTK.0> acl set rule port 0
RTK.0> acl set rule dmac data 0F:0F:0F:0F:0F:0F mask
F0:F0:F0:F0:F0:F0
RTK.0> acl show rule
Template: 0 [0] DMAC0
            [1] DMAC1
            [2] DMAC2
            [3] STAG
            [4] SMAC0
            [5] SMAC1
            [6] SMAC2
            [7] ETHERTYPE
Active Port: 0
Tags: Mask:
dmac data: 0F:0F:0F:0F:0F:0F
     mask: F0:F0:F0:F0:F0:F0
RTK.0>
```

acl clear action

Description:

This command can clear configured actions of editing rule. It should be execute before adding new configed rule. Otherwise, unexpected behavior may happen.

Syntax:

acl clear action

Parameter:

None

Examples:

This example shows how to clear configured actions of acl rule before adding.

```
RTK.0> acl show
Valid: Valid
Not: Disable
Cvlan action: Disable
Svlan action: Disable
Forward: Redirect Ports: 3
Policing: Policing meter: 31
Pri-Remark: ACL priority acl-pri: 7
Interrupt: Enable
Classification: None
ACL index latch: Disable
Template: 0 [0] DMAC0
            [1] DMAC1
            [2] DMAC2
            [3] STAG
            [4] SMAC0
            [5] SMAC1
            [6] SMAC2
            [7] ETHERTYPE
Active Port: none
Tags: Mask:
stag data: priority: 3 dei: 0 vid: 2000
     mask: priority: 7 dei: 1 vid: 4095
```

```
RTK.0>
RTK.0> acl clear action
RTK.0> acl show action
Valid: Valid
Not: Disable
Cvlan action: Disable
Svlan action: Disable
Forward: Disable
Policing: Disable
Pri-Remark: Disable
Interrupt or Classification: Disable
RTK.0>
```

acl set action

Description:

This command can edit wanted actions of rule will be added to device. Different chip version will support different actions to be configed.

Syntax:

acl set action cvlan inress vid *vid*
acl set action cvlan egress vid *vid*
acl set action cvlan using-svid
acl set action cvlan meter *index*
acl set action cvlan statistic *index*
acl set action cvlan remarking dscp *dscp*
acl set action cvlan remarking dot1p *priority*
acl set action svlan inress svid *svid*
acl set action svlan egress svid *svid*
acl set action svlan using-cvid
acl set action svlan meter *index*
acl set action svlan statistic *index*
acl set action svlan remarking dscp *dscp*
acl set action svlan remarking dot1p *priority*
acl set action priority remarking dscp *dscp*
acl set action priority remarking dot1p *priority*
acl set action priority meter *index*
acl set action priority statistic *index*
acl set action meter *index*
acl set action statistic *index*
acl set action trap-to-cpu
acl set action drop
acl set action copy port <*PORT_LIST:ext* **| all | none>**
acl set action redirect port <*PORT_LIST:ext* **| all | none>**
acl set action mirror port <*PORT_LIST:ext* **| all>**
acl set action interrrupt
acl set action latch-index
acl set action classf none
acl set action classf sid *sid*
acl set action classf llid *llid*
acl set action classf ext-member <*PORT_LIST:ext* **| all>**
acl set action none

Parameter:

cvlan	- cvlan related action configuration
svlan	- svlan related action configuration
priority	- priority related action configuration
ingress	- ingress vlan assign
egress	- egress vlan assign
using-cvid	- assign vid depend on VLAN identifier
using-svid	- assign vid depend on SVLAN identifier
remarking	- enable remarking
dot1p	- dot1p configuration
dscp	- dscp configuration
meter	- policing meter configuration
statistic	- statistic counter configuration
trap-to-cpu	- trap packet to cpu port
copy	- copy packet to assigned ports
redirect	- redirect packet to assigned ports
mirror	- mirror packet to assigned ports
interrupt	- interrupt action configuration
latch-index	- latch hit rule index
classf	- classification related action
none	- nop
sid	- stream id configuration
llid	- llid configuration
ext	- specify a extention port list

Examples:

This example shows how configure acl rule to redirect ingress port 0 packets to port 1 with assigned priority 7.

```
RTK.0> acl clear
RTK.0> acl set rule template entry 0
RTK.0> acl set rule state valid
RTK.0> acl set rule port 0
RTK.0> acl set action redirect port 1
RTK.0> acl set action priority assign-priority 7
RTK.0> acl show
Valid: Valid
Not: Disable
```

```
Cvlan action: Disable
Svlan action: Disable
Forward: Redirect Ports: 1
Policing: Disable
Pri-Remark: ACL priority acl-pri: 7
Interrupt or Classification: Disable
Template: 0 [0] DMAC0
            [1] DMAC1
            [2] DMAC2
            [3] STAG
            [4] SMAC0
            [5] SMAC1
            [6] SMAC2
            [7] ETHERTYPE
Active Port: 0
Tags: Mask:
RTK.0>
```

acl show action

Description:

This command can show configured actions of editing rule.

Syntax:

acl show action

Parameter:

None

Examples:

This example shows how to show configured actions of acl rule before adding.

```
RTK.0> acl show action
Valid: Valid
Not: Disable
Cvlan action: Disable
Svlan action: Disable
Forward: Redirect Ports: 3
Policing: Policing meter: 31
Pri-Remark: ACL priority acl-pri: 7
Interrupt: Enable
Classification: None
ACL index latch: Disable
RTK.0>
```

acl show template

Description:

This command can show configured content of editing template.

Syntax:

acl show template

Parameter:

None

Examples:

This example shows how the editing template is.

```
RTK.0> acl clear template
RTK.0> acl set template dmac
RTK.0> acl set template smac
RTK.0> acl set template sip
RTK.0> acl show template
Template: 0 [0] DMAC0
            [1] DMAC1
            [2] DMAC2
            [3] SMAC0
            [4] SMAC1
            [5] SMAC2
            [6] IP4SIP0
            [7] IP4SIP1
RTK.0>
```

acl clear template

Description:

This command can clear configured content of editing template.

Syntax:

acl clear template

Parameter:

None

Examples:

This example shows how to clear editing template

```
RTK.0> acl show template
Template: 0 [0] DMAC0
            [1] DMAC1
            [2] DMAC2
            [3] IP4DIP0
            [4] IP4DIP1
            [5] STAG
            [6] CTAG
            [7] unknown
RTK.0> acl clear template
RTK.0> acl show template
Template: 0 [0] unknown
            [1] unknown
            [2] unknown
            [3] unknown
            [4] unknown
            [5] unknown
            [6] unknown
            [7] unknown
RTK.0>
```

acl set template

Description:

This command can use to configure content of editing template. Each template contains limited packet pattern bits for acl rule matching. The ordering of editing template pattern bits is depended on editing order.

Syntax:

acl set template user-field <stag | ctag | ethertype | gem-llid | ipv6-next-header | dip | sip | smac | dmac | sip6 | dip6 | range-vid | range-ip | range-length | range-l4port | field-valid | ext-pmask | unknown>

acl set template user-field *index*

Parameter:

dmac	- destination mac address
smac	- source mac address
dip	- destination ipv4 address
sip	- source ipv4 address
dip6	- destination ipv6 address
sip6	- source ipv6 address
ethertype	- ethernet type or payload lengthlength
gemport-llid	- gemport id or llid
next-header	- ipv6 next header
range-l4port	- layer4 port number range checking list
range-vid	- vid range checking list
range-ip	- ip range checking list
range-length	- packet length range checking list
stag	- stag rule matching
ctag	- ctag rule matching
user-field	- user defined field
field-valid	- user defined field valid list
ext-pmask	- extention port mask
unknown	- unknown configuration

Examples:

This example shows how to configure acl template with dmac and smac and IPv4 dip.

```
RTK.0> acl clear template
RTK.0> acl set template dip
RTK.0> acl set template smac
RTK.0> acl set template dmac
RTK.0> acl show template
Template: 0 [0] IP4DIP0
            [1] IP4DIP1
            [2] SMAC0
            [3] SMAC1
            [4] SMAC2
            [5] DMAC0
            [6] DMAC1
            [7] DMAC2
RTK.0>
```

acl add template

Description:

This command can use to add configured template to device. The template number is supportted by different devices.

Syntax:

acl add template entry *index*

Parameter:

index - specified entry index

Examples:

This example shows how to add template with stag, ctag, dip, sip, user-defined field 0 and 1 to template entry 2.

```
RTK.0> acl clear template
RTK.0> acl set template stag
RTK.0> acl set template ctag
RTK.0> acl set template dip
RTK.0> acl set template sip
RTK.0> acl set template user-field 0
RTK.0> acl set template user-field 1
RTK.0> acl show template
Template: 0 [0] STAG
            [1] CTAG
            [2] IP4DIP0
            [3] IP4DIP1
            [4] IP4SIP0
            [5] IP4SIP1
            [6] Field_Sel0
            [7] Field_Sel1
RTK.0> acl add template entry 2
RTK.0> acl get template entry 2
Template: 2 [0] STAG
            [1] CTAG
            [2] IP4DIP0
            [3] IP4DIP1
            [4] IP4SIP0
            [5] IP4SIP1
```

```
                [6] Field_Sel0
                [7] Field_Sel1
    RTK.0>
```

acl get template

Description:

This command can be used to get templates which be configured to device.

Syntax:

acl get template entry *index*

Parameter:

Index - specified entry index

Examples:

This example shows how to get template index 0 and 1.

```
RTK.0> acl get template entry 0
Template: 0 [0] DMAC0
            [1] DMAC1
            [2] DMAC2
            [3] STAG
            [4] SMAC0
            [5] SMAC1
            [6] SMAC2
            [7] ETHERTYPE
RTK.0> acl get template entry 1
Template: 1 [0] CTAG
            [1] IP4SIP0
            [2] IP4SIP1
            [3] VIDRANGE
            [4] IPRANGE
            [5] PORTRANGE
            [6] IP4DIP0
            [7] IP4DIP1
RTK.0>
```

acl del template

Description:

This command can be used to delete templates which be configured to device. After deleting template, any packet will not hit acl rules with deleted template entry index.

Syntax:

acl del template entry *index*

Parameter:

index - specified entry index

Examples:

This example shows how to delete template entry 2.

```
RTK.0> acl get template entry 2
Template: 2 [0] STAG
            [1] CTAG
            [2] IP4DIP0
            [3] IP4DIP1
            [4] IP4SIP0
            [5] IP4SIP1
            [6] Field_Sel0
            [7] Field_Sel1
RTK.0> acl del template entry 2
RTK.0> acl get template entry 2
Template: 2 [0] unknown
            [1] unknown
            [2] unknown
            [3] unknown
            [4] unknown
            [5] unknown
            [6] unknown
            [7] unknown
RTK.0>
```

acl set range-vid

Description:

This command can configure vid, either svlan or cvlan, range check entry. While acl rule was configed to match desired vid range, valid entries will be compare with acl rule. If packets are without vlan tagging, then vid and svid will be assign as 0 to compared all vid range checking entries. The entry setting is be disabled while either state is configured to invalid or low-bound/up-bound is set as 0/4095.

Syntax:

acl set range-vid entry *index* **state <valid | invlaid>**
acl set range-vid entry *index* **type <svid | cvid>**
acl set range-vid entry *index* **<low-bound | up-bound>** *vid*

Parameter:

index	- specified entry index
state	- valid state configuration
type	- vid range type
low-bound	- low bound range
up-bound	- high bound range
vid	– specify the numeric VLAN identifier

Examples:

This example shows how to configured cvlan vid range for vid 100 to 200 and svlan vid range for svid 1000 to 2000.

```
RTK.0> acl set range-vid entry 0 state valid
RTK.0> acl set range-vid entry 0 type cvid
RTK.0> acl set range-vid entry 0 low-bound 100
RTK.0> acl set range-vid entry 0 up-bound 200
RTK.0> acl set range-vid entry 1 state valid
RTK.0> acl set range-vid entry 1 type svid
RTK.0> acl set range-vid entry 1 low-bound 1000
RTK.0> acl set range-vid entry 1 up-bound 2000
```

acl get range-vid

Description:

Show vid range checking entry contents.

Syntax:

acl get range-vid entry *index*

Parameter:

index - specified entry index

Examples:

This example shows configuration of vid range checking entry index 1.

```
RTK.0> acl get range-vid entry 1
Range check of VID
Index: 1 Upper: 2000 Lower: 1000 Type: SVID
RTK.0>
```

acl set range-ip

Description:

This command can configure IP address, either destination or source, range check entry. While acl rule be configed to match desired IP range, only valid entries will be compare with acl rule. The entry setting is be disabled, while either state is configed to invalid or low-bound/up-bound is set as 0.0.0.0/ 255.255.255.255. Different device supports different address bits checking for IPv6 IP range checking. At least IPv6 IP range checking is supporting with LSB 32-bits.

Syntax:

acl set range-ip entry *index* **state <valid | invlaid>**
acl set range-ip entry *index* **type <sip | dip | sip6 | dip6>**
acl set range-ip entry *index* **low-bound** *low_bound_ip* **up-bound** *up_bound_ip*

Parameter:

index	- specified entry index
state	- valid state configuration
type	- ip range type
dip	- destination ipv4 address
sip	- source ipv4 address
dip6	- destination ipv6 address
sip6	- source ipv6 address
low-bound	- low bound range
up-bound	- high bound range

Examples:

This example shows how to configure IP range checking entry 0 for IPv4 dip range within 172.1.1.1/16.

```
RTK.0> acl set range-ip entry 0 state valid
RTK.0> acl set range-ip entry 0 type dip
RTK.0> acl set range-ip entry 0 low-bound 172.1.1.1 up-bound
172.1.1.16
RTK.0> acl get range-ip entry 0
Range check of IP address
Index: 0 Upper: 172.1.1.16 Lower: 172.1.1.1 Type: IPv4 Dip
```

```
RTK.0>
```

acl get range-ip

Description:

Show IP range checking entry contents.

Syntax:

acl get range-ip entry *index*

Parameter:

index - specified entry index

Examples:

This example shows configuration of IP range checking entry index 1.

```
RTK.0> acl get range-ip entry 1
Range check of IP address
Index: 1 Upper: 192.168.1.255 Lower: 192.168.1.1 Type: Ipv4 Sip
RTK.0>
```

acl set range-l4port

Description:

This command can configure layer-4 tcp or udp port, either destination or source, range check entry. While acl rule was configed to match desired tcp or udp port range, only valid entries will be compare with acl rule. The entry setting is be disabled while either state is configured to invalid or low-bound/up-bound is set as 0/65535.

Syntax:

acl set range-l4port entry *index* **state <valid | invlaid>**
acl set range-l4port entry *index* **type <src-port | dst-port>**
acl set range-l4port entry *index* **<low-bound | up-bound>** *l4port*

Parameter:

index	- specified entry index
state	- valid state configuration
type	- ip range type
src-port	- source port number
dst-port	- destination port number
low-bound	- low bound range
up-bound	- high bound range
l4port	– specify the tcp/udp layer4 port number

Examples:

This example shows how to configure layer-4 port range checking entry index 0 to tcp source port range checking from 1024~2000.

```
RTK.0> acl set range-l4port entry 0 type src-port
RTK.0> acl set range-l4port entry 0 low-bound 1024
RTK.0> acl set range-l4port entry 0 up-bound 2000
RTK.0> acl set range-l4port entry 0 state valid
RTK.0> acl get range-l4port entry 0
Range check of L4 port
Index: 0 Upper: 2000 Lower: 1024 Type: Source Port
RTK.0>
```

acl get range-l4port

Description:

Show layer-4 tcp/udp port range checking entry contents.

Syntax:

acl get range-l4port entry *index*

Parameter:

index - specified entry index

Examples:

This example shows configuration of layer-4 tcp/udp port range checking entry index 0-1.

```
RTK.0> acl get range-l4port entry 0
Range check of L4 port
Index: 0 Upper: 2000 Lower: 1024 Type: Source Port
RTK.0> acl get range-l4port entry 1
Range check of L4 port
Index: 1 Upper: 1023 Lower: 0 Type: Destination Port
RTK.0>
```

acl set range-length

Description:

This command can configure length-range checking entry of ingress packet. While acl rule be configed to match packet length range, only valid entries will be compare with acl rule. The entry setting is be disabled while either state is configured to invalid or low-bound/up-bound is set as 0/16383. If reverse-state is set, it means packets which length is not within low-bound and up-bound setting are matched configuration.

Syntax:

acl set range-length entry *index* **reverse-state <disable | enable>**

acl set range-length entry *index* **<low-bound | up-bound>** *length*

Parameter:

index	- specified entry index
reverse-state	- reverse configuration
disable	- disable configuration
enable	- enable configuration
low-bound	- low bound range
up-bound	- high bound range
length	– specify length

Examples:

This example shows how to configure packet length range checking entry index 0 to lenth 64~1518.

```
RTK.0> acl set range-length entry 0 low-bound 64
RTK.0> acl set range-length entry 0 up-bound 1518
RTK.0> acl set range-length entry 0 reverse-state disable
RTK.0>
```

acl get range-length

Description:

Show packet length range checking entry contents.

Syntax:

acl get range-length entry *index*

Parameter:

index - specified entry index

Examples:

This example shows configuration of packet length range checking entry index 0 and index 1.

```
RTK.0> acl get range-length entry 0
Range check of packet length
Index: 0 Upper: 2000 Lower: 64 Type: Not Revise
RTK.0> acl get range-length entry 1
Range check of packet length
Index: 1 Upper: 1522 Lower: 68 Type: Not Revise
RTK.0>
```

acl get reason

Description:

Shows acl hit rule entry for each action. Each action will be executed by the first matched rule entry.

Syntax:

acl get reason <all | cvlan | svlan | priority | policing | forward | extend>

Parameter:

all	- all actions hit rule index
cvlan	- cvlan related action hit rule index
svlan	- svlan related action hit rule index
priority	- priority related action hit rule index
policing	- priority related action hit rule index
extend	- extend related action hit rule index

Examples:

This example shows how to get acl action hit rule entry number.

```
RTK.0> acl get reason all
Action    Hit Index
cvlan     yes 0
svlan     no  0
priority  no  0
policing  no  0
forward   no  0
extend    no  0
RTK.0>
```

acl set mode

Description:

This command can configure acl mode for different number of rules usage.

Syntax:

acl set mode <64-entries | 128-entries>

Parameter:

64-entries	- 64 rule entries usage
128-entries	- 128 rule entries usage

Examples:

This example shows how to configure acl mode to 128 rules.

```
RTK.0> acl init
RTK.0> acl set mode 128-entries
RTK.0>
```

acl get mode

Description:

Show acl number of rules usage.

Syntax:

acl get mode

Parameter:

None

Examples:

This example shows how to get valid number of acl rules.

```
RTK.0> acl init
RTK.0> acl get mode
ACL mode: 64 Entries
RTK.0>
```

acl set state

Description:

This command can configure per-port acl enable state. If acl ingress state is enabled, all packets from the acl enable port will be matching acl rules and switch will follow the 1st matched acl action to packets which hit configured acl rules.

Syntax:

acl set port <*PORT_LIST:ports* **| all> state <disable | enable>**

Parameter:

ports	- specified port list
All	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to configure acl state of ports from port 2 to port 4.

```
RTK.0> acl set port 2-4 state enable
RTK.0> acl get port 0-6 state
Port State
0    Disable
1    Disable
2    Enable
3    Enable
4    Enable
5    Disable
6    Disable
RTK.0>
```

acl get state

Description:

Show per-port acl enable state.

Syntax:

acl get port <*PORT_LIST:ports* **| all> state**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration

Examples:

This example shows how to get acl state of port 1,3 and 5.

```
RTK.0> acl get port 1,3,5 state
Port State
1    Disable
3    Enable
5    Disable
RTK.0>
```

acl set permit

Description:

This command can configure switch action for packets which unmatched ingress acl rules. If permit is not set, switch will drop packets, which are unmatched any acl rule.

Syntax:

acl set port <*PORT_LIST:ports* **| all> permit <disable | enable>**

Parameter:

ports	- specified port list
all	– specify all ports
permit	- permit configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to configure port 0-3 acl permit enable setting.

```
RTK.0> acl set port 0-3 permit enable
RTK.0> acl get port 0-6 permit
Port State
0    Enable
1    Enable
2    Enable
3    Enable
4    Disable
5    Disable
6    Disable
RTK.0>
```

acl get permit

Description:

Show per port acl permit state.

Syntax:

acl get port <*PORT_LIST:ports* **| all> permit**

Parameter:

ports	- specified port list
all	– specify all ports
permit	- permit configuration

Examples:

This example shows how to get the ports 1,3,5 acl permit state.

```
RTK.0> acl get port 1,3,5 permit
Port State
1    Enable
3    Enable
5    Disable
RTK.0>
```

field-selector set

Description:

This command can configure acl user defined field. Each field can set 16-bits content of packet which user wants to filter for acl uage. From pure raw packet to layer-4 content as tcp or udp, the field can be set for parsing content of packet inside the first 256 bytes.

Syntax:

field-selector set index *index* **format <default | raw | llc | arp | ipv4-header | ipv6-header | ip-payload | l4-payload> offset** *offset*

Parameter:

index	- specify index
format	- parsing format
default	- un-assigned format
raw	- parsing as l2 raw packet
llc	- parsing as llc packet
arp	- parsing as arp packet
ipv4-header	- parsing from ipv4 header
ipv6-header	- parsing from ipv6 header
ip-payload	- parsing from ipv4 payload
offset	- specify parsing byte offset

Examples:

This example show how to configure using acl to trap total length field of ipv4 header which greate than 128 by setting field selector 8 and udp ssdp “M-SEARCH” packets by setting field selector 9/10/11/12.

```
RTK.0> field-selector set index 8 format ipv4-header offset 2
RTK.0> field-selector set index 9 format ip-payload offset 8
RTK.0> field-selector set index 10 format ip-payload offset 10
RTK.0> field-selector set index 11 format ip-payload offset 12
RTK.0> field-selector set index 12 format ip-payload offset 14
```

field-selector get

Description:

Show user defined field selector configuration.

Syntax:

field-selector get index *index*

Parameter:

index - specify index

Examples:

This example shows how to get field selectors 8-10.

```
RTK.0> field-selector get index 8
Index Offset Mode
8     2      IPv4 header
RTK.0> field-selector get index 9
Index Offset Mode
9     29     IP payload
RTK.0> field-selector get index 10
Index Offset Mode
10    31     IP payload
RTK.0>
```

2.15. PON Mac commands

pon init

Description:

Initial PON mac module. For PON port the queue management and schedule must using PON command to configurate it.

Syntax:

pon init

Parameter:

None

Examples:

This example shows how to initialize pon module.

```
RTK.0> pon init
RTK.0>
```

pon set t-cont

Description:

The PON port queue management is based on T-CONT per tcont have 32 queues. If the PON port is used as UTP port, please use tcont 0 queue 0~7 to confgirate this port. This command set is buffer mode, user muset call “pon add t-cont” to apply these seeting to chip. For GPON mode please use “pon set t-cont” command

Syntax:

pon set t-cont *t-cont* **queue-id** *queue-id* **<cir | pir> rate** *rate*
pon set t-cont *t-cont* **queue-id** *queue-id* **scheduling type < strict | wfq >**
pon set t-cont *t-cont* **queue-id** *queue-id* **egress-drop state <disable | enable>**

Parameter:

t-cont	- GPON T-CONT id
queue-id	- Queue id
cir	- Committed Information Rate,
Pir	- Peak Information rate
Rate	- The PIR and CIR rate unit 64Kbps
Strict	- The traffic schedule is based on strick priority
Wfq	- The traffic schedule is based on wfq
Enable	- Enable per queue egress drop
Disable	- Disable per queue egress drop

Examples:

This example shows how to set t-cont 0 queue 0~1 as follow configurations.

Queue-id	schedule-type	PIR/CIR	egress-drop-state
0	Strict	100/10	disable
1	WFQ	100/10	disable

```
RTK.0> pon clear
RTK.0> pon set t-cont 0 queue-id 0 scheduling type strict
RTK.0> pon set t-cont 0 queue-id 0 pir rate 100
RTK.0> pon set t-cont 0 queue-id 0 cir rate 10
RTK.0> pon set t-cont 0 queue-id 0 egress-drop state disable
```

```
RTK.0> pon add t-cont 0 queue-id 0
RTK.0> pon clear
RTK.0> pon set t-cont 0 queue-id 1 scheduling type wfq
RTK.0> pon set t-cont 0 queue-id 1 pir rate 100
RTK.0> pon set t-cont 0 queue-id 1 cir rate 10
RTK.0> pon set t-cont 0 queue-id 1 egress-drop state disable
RTK.0> pon add t-cont 0 queue-id 1
```

pon get t-cont

Description:

This command can get queue setting. The “pon get t-cont queue-list” can get which queue is scheduled by this t-cont.

Syntax:

pon get t-cont *t-cont* **queue-id** *queue-id*
pon get t-cont *t-cont* **queue-list**

Parameter:

t-cont	- GPON T-CONT id
queue-id	- Queue id
queue-list	- Queue list for this t-cont

Examples:

This example shows how to get queue setting for t-cont 0 queue 0.

```
RTK.0> pon get t-cont 0 queue-id 0
CIR:10
PIR:100
queue Type:strict
WFQ weight:1
Egress Drop:Disable
```

This example shows how to get which queue is scheduled by t-cont 0.

```
RTK.0> pon get t-cont 0 queue-list
queue 0  :O
queue 1  :O
queue 2  :O
queue 3  :O
queue 4  :O
queue 5  :O
queue 6  :O
queue 7  :O
queue 8  :X
queue 9  :X
```

```
queue 10 :X
queue 11 :X
queue 12 :X
queue 13 :X
queue 14 :X
queue 15 :X
queue 16 :X
queue 17 :X
queue 18 :X
queue 19 :X
queue 20 :X
queue 21 :X
queue 22 :X
queue 23 :X
queue 24 :X
queue 25 :X
queue 26 :X
queue 27 :X
queue 28 :X
queue 29 :X
queue 30 :X
queue 31 :X
```

pon add t-cont

Description:

Use this command to apply pon queue setting to asic.

Syntax:

pon add t-cont *t-cont* **queue-id** *queue-id*

Parameter:

t-cont - GPON T-CONT id
queue-id - Queue id

Examples:

This example shows how to set t-cont 0 queue 0~1 as follow configurations.

Queue-id	schedule-type	PIR/CIR	egress-drop-state
0	Strict	100/10	disable
1	WFQ	100/10	disable

```
RTK.0> pon clear
RTK.0> pon set t-cont 0 queue-id 0 scheduling type strict
RTK.0> pon set t-cont 0 queue-id 0 pir rate 100
RTK.0> pon set t-cont 0 queue-id 0 cir rate 10
RTK.0> pon set t-cont 0 queue-id 0 egress-drop state disable
RTK.0> pon add t-cont 0 queue-id 0
RTK.0> pon clear
RTK.0> pon set t-cont 0 queue-id 1 scheduling type wfq
RTK.0> pon set t-cont 0 queue-id 1 pir rate 100
RTK.0> pon set t-cont 0 queue-id 1 cir rate 10
RTK.0> pon set t-cont 0 queue-id 1 egress-drop state disable
RTK.0> pon add t-cont 0 queue-id 1
```

pon del t-cont

Description:

Use this command to remove queue from given t-cont scheduler.

Syntax:

pon del t-cont *t-cont* **queue-id** *queue-id*

Parameter:

t-cont	- GPON T-CONT id
queue-id	- Queue id

Examples:

This example shows how to delete queue 0 from t-cont 0.

```
RTK.0> pon del t-cont 0 queue-id 0
```

pon set llid

Description:

The PON port queue management is based on LLID, per llid have 8 queues. If the PON port is used as UTP port, please use llid 0 queue 0~7 to confgirate this port. This command set is buffer mode, user muset call “pon add llid” to apply these seeting to chip. For EPON mode please use “pon set llid” command

Syntax:

pon set llid *llid* **queue-id** *queue-id* **<cir | pir> rate** *rate*
pon set llid *llid* **queue-id** *queue-id* **scheduling type < strict | wfq >**
pon set llid *llid* **queue-id** *queue-id* **egress-drop state <disable | enable>**

Parameter:

t-cont	- EPON llid
queue-id	- Queue id
cir	- Committed Information Rate,
Pir	- Peak Information rate
Rate	- The PIR and CIR rate unit 64Kbps
Strict	- The traffic schedule is based on strick priority
Wfq	- The traffic schedule is based on wfq
Enable	- Enable per queue egress drop
Disable	- Disable per queue egress drop

Examples:

This example shows how to set llid 0 queue 0~1 as follow configurations.

Queue-id	schedule-type	PIR/CIR	egress-drop-state
0	Strict	100/10	disable
1	WFQ	100/10	disable

```
RTK.0> pon clear
RTK.0> pon set llid 0 queue-id 0 scheduling type strict
RTK.0> pon set llid 0 queue-id 0 pir rate 100
RTK.0> pon set llid 0 queue-id 0 cir rate 10
RTK.0> pon set llid 0 queue-id 0 egress-drop state disable
RTK.0> pon add llid 0 queue-id 0
```

```
RTK.0> pon set llid 0 queue-id 0 scheduling type strict
RTK.0> pon set llid 0 queue-id 0 pir rate 100
RTK.0> pon set llid 0 queue-id 0 cir rate 10
RTK.0> pon set llid 0 queue-id 0 egress-drop state disable
RTK.0> pon add llid 0 queue-id 0
RTK.0> pon clear
RTK.0> pon set llid 0 queue-id 1 scheduling type wfq
RTK.0> pon set llid 0 queue-id 1 pir rate 100
RTK.0> pon set llid 0 queue-id 1 cir rate 10
RTK.0> pon set llid 0 queue-id 1 egress-drop state disable
RTK.0> pon add llid 0 queue-id 1
```

pon get llid

Description:

This command can get queue setting. The “pon get llid queue-list” can get which queue is scheduled by this llid.

Syntax:

pon get llid *llid* **queue-id** *queue-id*
pon get llid *llid* **queue-list**

Parameter:

llid	- EPON llid
queue-id	- Queue id
queue-list	- Queue list for this t-cont

Examples:

This example shows how to get queue setting for llid queue 0.

```
RTK.0> pon get llid 0 queue-id 0
CIR:10
PIR:100
queue Type:strict
WFQ weight:1
Egress Drop:Disable
```

This example shows how to get which queue is scheduled by llid 0.

```
RTK.0> pon get llid 0 queue-list
queue 0  :O
queue 1  :O
queue 2  :O
queue 3  :O
queue 4  :O
queue 5  :O
queue 6  :O
queue 7  :O
queue 8  :X
queue 9  :X
queue 10 :X
```

```
queue 11 :X
queue 12 :X
queue 13 :X
queue 14 :X
queue 15 :X
queue 16 :X
queue 17 :X
queue 18 :X
queue 19 :X
queue 20 :X
queue 21 :X
queue 22 :X
queue 23 :X
queue 24 :X
queue 25 :X
queue 26 :X
queue 27 :X
queue 28 :X
queue 29 :X
queue 30 :X
queue 31 :X
```

pon add llid

Description:

Use this command to apply pon queue setting to asic.

Syntax:

pon add llid *llid* **queue-id** *queue-id*

Parameter:

llid	- EPON llid
queue-id	- Queue id

Examples:

This example shows how to set llid 0 queue 0~1 as follow configurations.

Queue-id	schedule-type	PIR/CIR	egress-drop-state
0	Strict	100/10	disable
1	WFQ	100/10	disable

```
RTK.0> pon clear
RTK.0> pon set llid 0 queue-id 0 scheduling type strict
RTK.0> pon set llid 0 queue-id 0 pir rate 100
RTK.0> pon set llid 0 queue-id 0 cir rate 10
RTK.0> pon set llid 0 queue-id 0 egress-drop state disable
RTK.0> pon add llid 0 queue-id 0
RTK.0> pon clear
RTK.0> pon set llid 0 queue-id 1 scheduling type wfq
RTK.0> pon set llid 0 queue-id 1 pir rate 100
RTK.0> pon set llid 0 queue-id 1 cir rate 10
RTK.0> pon set llid 0 queue-id 1 egress-drop state disable
RTK.0> pon add llid 0 queue-id 1
```

pon del llid

Description:

Use this command to remove queue from given llid scheduler.

Syntax:

pon del llid *llid* **queue-id** *queue-id*

Parameter:

llid	-EPON llid
queue-id	-Queue id

Examples:

This example shows how to delete queue 0 from LLID 0.

```
RTK.0> pon del llid 0 queue-id 0
```

pon clear

Description:

This command can clear contents of configured pon queue setting. It is necessary to use this command before editing and adding new queue to make sure the setting are wanted.

Syntax:

pon clear

Parameter:

none

Examples:

This example shows how to set llid 0 queue 0~1 as follow configurations.

```
Queue-id  schedule-type  PIR/CIR  egress-drop-state
0         Strict         100/10   disable
1         WFQ            100/10   disable
RTK.0> pon clear
RTK.0> pon set llid 0 queue-id 0 scheduling type strict
RTK.0> pon set llid 0 queue-id 0 pir rate 100
RTK.0> pon set llid 0 queue-id 0 cir rate 10
RTK.0> pon set llid 0 queue-id 0 egress-drop state disable
RTK.0> pon add llid 0 queue-id 0
RTK.0> pon clear
RTK.0> pon set llid 0 queue-id 1 scheduling type wfq
RTK.0> pon set llid 0 queue-id 1 pir rate 100
RTK.0> pon set llid 0 queue-id 1 cir rate 10
RTK.0> pon set llid 0 queue-id 1 egress-drop state disable
RTK.0> pon add llid 0 queue-id 1
```

pon reset

Description:

This command can reset all pon mac setting to default as “pon init” was called. All queue will be remove from all t-cont or llid.

Syntax:

pon reset

Parameter:

None

Examples:

This example shows how to reset pon mac setting.

```
RTK.0> pon reset
```

2.16. Classfication commands

classf init

Description:

Classificatoin module initialization. It will delete all classification entries.

Syntax:

classf init

Parameter:

None

Examples:

This example shows how to initialize classification module.

```
RTK.0> classf init
RTK.0>
```

classf clear

Description:

Clear classification edit buffer. The edit buffer is used to store settings of classification rule and action, and will be applied by “classf add entry” command.

Syntax:

classf clear

Parameter:

None

Examples:

This example shows how to clear classification edit buffer. It must be used before a new classification entry is added.

```
RTK.0> classf clear
RTK.0>
```

classf show

Description:

Show the edit buffer of classification rule

Syntax:

classf show

Parameter:

None

Examples:

This example shows how to display edit buffer of classification rule:

```
RTK.0> classf show
Not: Disable
direction: Upstream
Rule:
tag vid data: 1000
        mask: 0xfff
Upstream action:
Stag action: No operation
Ctag action: No operation
SID action: ASSIGN to SID
Assign ID: 10
RTK.0>
```

classf set rule

Description:

Configure the edit buffer of classification rule

Syntax:

classf set rule direction <upstream | downstream>

classf set rule <tos-sid | tag-vid | tag-priority | internal-priority | svlan-bit | cvlan-bit | uni | ether-type | range-l4port | range-ip | hit-acl | wan-if | ipmc-bit | ip6mc-bit | igmp-bit | mld-bit | dei-cfi> data *data* **mask** *mask*

Parameter:

direction	– specify the rule is applied to upstream or downstream direction
upstream	– upstream configuration
downstream	– downstream configuration
tos-sid	- for upstream rule, this field specify to tos, for downstream rule this field specify to steam id
tag-vid	- specify to outer tag vid field
tag-priority	- specify to outer tag priority field
internal-priority	- specify to internal priority field
svlan-bit	- specify to stag exist field
cvlan -bit	- specify to ctag exist field
uni	- specify to ingress uni for downstream and specify to egress uni for upstream
ether-type	- specify to Ethernet type field
range-l4port	- layer4 port number range checking list
range-ip	- ip range checking list
hit-acl	- acl matched index
wan-if	- wan interface
ipmc-bit	- care field of ipv4 multicast packet
ip6mc-bit	- care field of ipv6 multicast packe
igmp-bit	- care field of igmp packet
mld-bit	- care field of mld packet
dei-cfi	- dei or cfi field of tag
data	- the data value of each field
mask	- the data mask of each field

Examples:

This example shows how to edit a classification downstream rule for variant fields.

```
RTK.0> classf clear
RTK.0> classf set rule direction downstream
RTK.0> classf set rule ether-type data 0x0800 mask 0xffff
RTK.0> classf set rule cvlan-bit data 1 mask 1
RTK.0> classf set rule svlan-bit data 1 mask 1
RTK.0> classf set rule tag-vid data 1000 mask 0xfff
RTK.0> classf set rule tag-priority data 5 mask 7
RTK.0> classf set rule uni data 1 mask 7
RTK.0> classf show
Not: Disable
direction: Downstream
Rule:
ether type data: 0x0800
           mask: 0xffff
Rule:
s-bit data: 1
     mask: 0x1
Rule:
c-bit data: 1
     mask: 0x1
Rule:
tag vid data: 1000
       mask: 0xfff
Rule:
tag priority data: 5
            mask: 0x7
Rule:
UNI data: 1
    mask: 0x7
Downstream action:
Stag action: No operation
Ctag action: No operation
Classf PRI action: Follow switch core
UNI action: No operation
UNI ports: none
```

```
RTK.0>
```

classf set downstream-action

Description:

Configure the edit buffer of classification downstream action

Syntax:

classf set downstream-action svlan-act <nop | vs-tpid | c-tpid | del | transparent | sp2c>

classf set downstream-action svlan-id-act assign *vid*

classf set downstream-action svlan-id-act <copy-outer | copy-inner> [*vid***]**

classf set downstream-action svlan-priority-act assign *priority*

classf set downstream-action svlan-priority-act <copy-outer | copy-inner | internal-priority> [*priority***]**

classf set downstream-action cvlan-act <nop | c-tag | sp2c | del | transparent>

classf set downstream-action cvlan-id-act assign *vid*

classf set downstream-action cvlan-id-act <follow-swcore | copy-outer | copy-inner | lookup-table> [*vid***]**

classf set downstream-action cvlan-priority-act assign *priority*

classf set downstream-action cvlan-priority-act <copy-outer | copy-inner | internal-priority> [*priority***]**

classf set downstream-action priority-act assign *priority*

classf set downstream-action priority-act follow-swcore

classf set downstream-action remark-dscp <disable | enable>

Parameter:

svlan-act	– specify the stag action
nop	- no operation
vs-tpid	- s-tagging with tpid in svlan configuration
c-tpid	- s-tagging with 802.1Q tpid 0x8100
del	- de-tagging vlan
transparent	- vlan transparent configuration
sp2c	- specify from the sp2c table
svlan-id-act	– specify the stag vid action
svlan-priority-act	– specify the stag priority action
assign	– specify assigned value

copy-outer - specify from outer tag
copy-inner - specify from inner tag
vid – specify the numeric VLAN identifier
priority - priority configuration
internal-priority - spefify from switch internal priority
cvlan-id-act – specify the ctag vid action
cvlan-priority-act – specify the ctag priority action
ctag - c-tagging
priority-act - classification priority assign
follow-swcore - follow switch normal decision
lookup-table - specify from lookup table
remark-dscp - dscp remarking
uni-forward-act - egress forward configuration
flood - flooding within assigned ports
forced - forced forward to assigned ports
ports - specified port list
all - specify all ports
none - specify no port

Examples:

This example shows how to edit a classification downstream action. Remove Stag. Assign Ctag with VID=100, priority=2. Assign internal priority=5. And force forward to UNI 0.

```
RTK.0> classf clear
RTK.0> classf set rule direction downstream
RTK.0> classf set rule ether-type data 0x0800 mask 0xffff
RTK.0> classf set rule cvlan-bit data 1 mask 1
RTK.0> classf set rule svlan-bit data 1 mask 1
RTK.0> classf set rule tag-vid data 1000 mask 0xfff
RTK.0> classf set rule tag-priority data 5 mask 7
RTK.0> classf set rule uni data 1 mask 7
RTK.0> classf set downstream-action svlan-act del
RTK.0> classf set downstream-action cvlan-act c-tag
RTK.0> classf set downstream-action cvlan-id-act assign 100
RTK.0> classf set downstream-action cvlan-priority-act assign
2
RTK.0> classf set downstream-action priority-act assign 5
```

```
RTK.0> classf set downstream-action uni-forward-act forced port
0
RTK.0> classf show
Not: Disable
direction: Downstream
Rule:
ether type data: 0x0800
           mask: 0xffff
Rule:
s-bit data: 1
      mask: 0x1
Rule:
c-bit data: 1
      mask: 0x1
Rule:
tag vid data: 1000
       mask: 0xfff
Rule:
tag priority data: 5
             mask: 0x7
Rule:
UNI data: 1
    mask: 0x7
Downstream action:
Stag action: Delete
Ctag action: Tagging
Ctag VID action: Assign
Ctag VID: 100
Ctag PRI action: Assign
Ctag PRI: 2
Classf PRI action: Assign
CF PRI: 5
UNI action: Force forward
UNI ports: 0
RTK.0>
```

classf set upstream-action

Description:

Configure the edit buffer of classification upstream action

Syntax:

classf set upstream-action svlan-act <nop | vs-tpid | c-tpid | del | transparent>

classf set upstream-action svlan-id-act assign *vid*

classf set upstream-action svlan-id-act <copy-outer | copy-inner> [*vid***]**

classf set upstream-action svlan-priority-act assign *priority*

classf set upstream-action svlan-priority-act <copy-outer | copy-inner | internal-priority> [*priority***]**

classf set upstream-action cvlan-act <nop | c-tag | c2s | del | transparent>

classf set upstream-action cvlan-id-act assign *vid*

classf set upstream-action cvlan-id-act <copy-outer | copy-inner> [*vid***]**

classf set upstream-action cvlan-priority -act assign *priority*

classf set upstream-action cvlan-priority-act <copy-outer | copy-inner | internal-priority> [*priority***]**

classf set upstream-action priority-act assign *priority*

classf set upstream-action priority-act follow-swcore

classf set upstream-action sid-act <sid | qid> *id*

classf set upstream-action sid-act nop

classf set upstream-action statistic *index*

classf set upstream-action remark-dscp <disable | enable>

classf set upstream-action drop <disable | enable>

Parameter:

svlan-act	– specify the stag action
nop	- no operation
vs-tpid	- s-tagging with tpid in svlan configuration
c-tpid	- s-tagging with 802.1Q tpid 0x8100
del	- de-tagging vlan
transparent	- vlan transparent configuration
svlan-id-act	– specify the stag vid action
svlan-priority-act	– specify the stag priority action

assign – specify assigned value
copy-outer - specify from outer tag
copy-inner - specify from inner tag
vid – specify the numeric VLAN identifier
priority - priority configuration
internal-priority - spefify from switch internal priority
cvlan-id-act – specify the ctag vid action
cvlan-priority-act – specify the ctag priority action
ctag - c-tagging
c2s - specify from the c2s table
priority-act - classification priority assign
follow-swcore - follow switch normal decision
sid-act – specify the stream id action
sid - stream id configuration
qid – specify the numeric queue identifier
statistic - statistic counter configuration
remark-dscp - dscp remarking
drop - drop packet
disable - disable configuration
enable - enable configuration

Examples:

This example shows how to edit a classification upstream action. Add Stag by VID 1000, priority copy from outer tag. Ctag transparent. And assign stream-ID to 1.

```
RTK.0> classf clear
RTK.0> classf set rule direction upstream
RTK.0> classf set rule svlan-bit data 0 mask 1
RTK.0> classf set rule cvlan-bit data 1 mask 1
RTK.0> classf set rule tag-vid data 100 mask 0xfff
RTK.0> classf set upstream-action svlan-act vs-tpid
RTK.0> classf set upstream-action svlan-id-act assign 1000
RTK.0> classf set upstream-action svlan-priority-act
copy-outer
RTK.0> classf set upstream-action cvlan-act transparent
RTK.0> classf set upstream-action sid-act sid 1
RTK.0> classf show
```

```
Not: Disable
direction: Upstream
Rule:
c-bit data: 0
      mask: 0x1
Rule:
s-bit data: 1
      mask: 0x1
Rule:
tag vid data: 100
        mask: 0xfff
Upstream action:
Stag action: VS_TPID
Stag VID action: Assign
Stag VID: 1000
Stag PRI action: Copy from 1st tag
Stag PRI: 0
Ctag action: Transparent
SID action: ASSIGN to SID
Assign ID: 1
RTK.0>
```

classf add entry

Description:

Add the classification entry according to the edit buffer of rule and upsteam/downstream action

Syntax:

classf add entry *index*

Parameter:

index – specify the entry index

Examples:

This example shows how to add a classification entry in detail. Including edit buffer of rule and downstream action.

```
RTK.0> classf clear
RTK.0> classf set rule direction downstream
RTK.0> classf set rule ether-type data 0x0800 mask 0xffff
RTK.0> classf set rule cvlan-bit data 1 mask 1
RTK.0> classf set rule svlan-bit data 1 mask 1
RTK.0> classf set rule tag-vid data 1000 mask 0xfff
RTK.0> classf set rule tag-priority data 5 mask 7
RTK.0> classf set rule uni data 1 mask 7
RTK.0> classf set downstream-action svlan-act del
RTK.0> classf set downstream-action cvlan-act c-tag
RTK.0> classf set downstream-action cvlan-id-act assign 100
RTK.0> classf set downstream-action cvlan-priority-act assign
2
RTK.0> classf set downstream-action priority-act assign 5
RTK.0> classf set downstream-action uni-forward-act forced port
0
RTK.0> classf add entry 0
RTK.0>
RTK.0> classf get entry 0
Valid: Valid
Not: Disable
direction: Downstream
databit: 0x4519-801f-0800
carebit: 0xff1f-807f-ffff
```

```
Downstream action:
Stag action: Delete
Ctag action: Tagging
Ctag VID action: Assign
Ctag VID: 100
Ctag PRI action: Assign
Ctag PRI: 2
Classf PRI action: Assign
CF PRI: 5
UNI action: Force forward
UNI ports: 0
RTK.0>
```

classf get entry

Description:

Show the raw data of classification entry

Syntax:

classf get entry *index*

Parameter:

index – specify the entry index

Examples:

This example shows how to display classification entry:

```
RTK.0> classf get entry 0
Valid: Valid
Not: Disable
direction: Downstream
databit: 0x4519-801f-0800
carebit: 0xff1f-807f-ffff
Downstream action:
Stag action: Delete
Ctag action: Tagging
Ctag VID action: Assign
Ctag VID: 100
Ctag PRI action: Assign
Ctag PRI: 2
Classf PRI action: Assign
CF PRI: 5
UNI action: Force forward
UNI ports: 0
RTK.0>
```

classf del entry

Description:

Delete the classification entry

Syntax:

classf del entry *index*

classf del entry all

Parameter:

index – specify the entry index

all - all entries

Examples:

This example shows how to del a classification entry

```
RTK.0> classf del entry 0
```

classf set operation

Description:

Configuration the operation mode as hit or not hit for upstream or downstream rule

Syntax:

classf set operation entry *index* **<upstream | downstream> < hit | not >**

Parameter:

index	– specify the entry index
upstream	– upstream configuration
downstream	– downstream configuration
hit	– hit operation
not	– reverse operation

Examples:

This example shows how to configure a downstream classification entry as hit operation

```
RTK.0> classf set operation entry 0 downstream hit
```

classf get operation

Description:

Get the operation mode for upstream or downstream rule

Syntax:

classf get entry *index*

Parameter:

index – specify the entry index

Examples:

This example shows how to display a classification entry operation

```
RTK.0> classf get entry 0
Valid: Valid
Not: Disable
direction: Downstream
databit: 0x4519-801f-0800
carebit: 0xff1f-807f-ffff
Downstream action:
Stag action: Delete
Ctag action: Tagging
Ctag VID action: Assign
Ctag VID: 100
Ctag PRI action: Assign
Ctag PRI: 2
Classf PRI action: Assign
CF PRI: 5
UNI action: Force forward
UNI ports: 0
```

classf set cf-sel-port

Description:

Configuration PON port or RGMII port as the classification uplink port

Syntax:

classf set cf-sel-port <pon | rg> <disable | enable>

Parameter:

pon	– pon port configuration
rg	- rg port configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to enable PON port as classification uplink port

```
RTK.0> classf set cf-sel-port pon enable
```

classf get cf-sel-port

Description:

Get the state of classification uplink port selection.

Syntax:

classf get cf-sel-port

Parameter:

None

Examples:

This example shows how to display a classification uplink port selection

```
RTK.0> classf get cf-sel-port
Cf select port: PON enabled
Cf select port: RGMII disabled
```

classf set upstream-unmatch-act

Description:

Configuration classification upstream unmatch action

Syntax:

classf set upstream-unmatch-act <drop | permit-without-pon | permit>

Parameter:

drop	– drop packet
permit	– forward packet
permit-without-pon	- forward packet exclude pon port

Examples:

This example shows how to configure upstream unmatch action as drop

```
RTK.0> classf set upstream-unmatch-act drop
```

classf get upstream-unmatch-act

Description:

Get the classification upstream unmatch action

Syntax:

classf get upstream-unmatch-act

Parameter:

None

Examples:

This example shows how to display a classification upstream unmatch action

```
RTK.0> classf get upstream-unmatch-act
Upstream un-match action: Drop
```

classf set permit-sel-range

Description:

Set classification entry range selection for upstream unmatch action. There are two options, one is for entry range 0-511, and the other is for entry range 64-511.

Syntax:

classf set permit-sel-range < 0to511 | 64to511>

Parameter:

0to511	– checking from rule 0 to rule 511
64to511	– checking from rule 64 to rule 511

Examples:

This example shows how to configure entry rang selection to 64-511.

```
RTK.0> classf set permit-sel-range 64to511
```

classf get permit-sel-range

Description:

This command can use to get classification entry range selection for upstream unmatch action. There are two options, one is for entry range 0-511, and the other is for entry range 64-511.

Syntax:

classf get us-unmatch-act

Parameter:

None

Examples:

This example shows how to display the classification entry rang selection

```
RTK.0> classf get permit-sel-range
Permit range from 64 to 511
```

classf set remarking dscp

Description:

Set DSCP remarking for classification DSCP_remarking action used.

Syntax:

classf set remarking dscp priority *priority* **dscp** *dscp*

Parameter:

priority	- priority configuration
dscp	- dscp configuration

Examples:

This example shows how to configure classification DSCP remarking as priority 0 mapping to dscp 7, priority 1 mapping to dscp 15, priority 2 mapping to dscp 23, priority 3 mapping to dscp 31, priority 4 mapping to dscp 39, priority 5 mapping to dscp 47, priority 6 mapping to dscp 55, priority 7 mapping to dscp 63

```
RTK.0> classf set remarking dscp priority 0 dscp 7
RTK.0> classf set remarking dscp priority 1 dscp 15
RTK.0> classf set remarking dscp priority 2 dscp 23
RTK.0> classf set remarking dscp priority 3 dscp 31
RTK.0> classf set remarking dscp priority 4 dscp 39
RTK.0> classf set remarking dscp priority 5 dscp 47
RTK.0> classf set remarking dscp priority 6 dscp 55
RTK.0> classf set remarking dscp priority 7 dscp 63
```

classf get remarking dscp

Description:

Get classification DSCP remarking.

Syntax:

classf get remarking dscp

Parameter:

None

Examples:

This example shows how to display a classification DSCP remarking table

```
RTK.0> classf get remarking dscp
CF_priority  DSCP
0            7
1            15
2            23
3            31
4            39
5            47
6            55
7            63
```

classf set range-ip

Description:

Set classification IP range for classification range-ip rule

Syntax:

classf set range-ip entry *index* **type <sip | dip >**

classf set range-ip entry *index* **low-bound** *low_bound_ip* **up-bound** *up_bound_ip*

Parameter:

index	- specified entry index
type	- ip range type
dip	- destination ipv4 address
sip	- source ipv4 address
low-bound	- low bound range
up-bound	- high bound range

Examples:

This example shows how to configure a upstream rule for source IP range 10.1.1.1 to 10.1.2.2

```
RTK.0> classf clear
RTK.0> classf set rule direction upstream
RTK.0> classf set rule range-ip data 0 mask 0x7
RTK.0> classf set upstream-action sid-act sid 0
RTK.0> classf add entry 0
RTK.0> classf set range-ip entry 0 type sip
RTK.0> classf set range-ip entry 0 low-bound 10.1.1.1 up-bound
10.1.2.2
```

classf get range-ip

Description:

Get classification IP range setting by specific entry index

Syntax:

classf get range-ip entry *index*

Parameter:

index - specified entry index

Examples:

This example shows how to display classification IP range setting

```
RTK.0> classf get range-ip entry 0
Range check of IP address
Index: 0 Upper: 10.1.2.2 Lower: 10.1.1.1 Type: Ipv4 Sip
```

classf set range-l4port

Description:

Set classification L4 port range for classification range-l4port rule

Syntax:

classf set range-l4port entry *index* **type <src-port | dst-port>**

classf set range-l4port entry *index* **low-bound** *l4lport* **up-bound** *l4uport*

Parameter:

index	- specified entry index
type	- ip range type
src-port	- source port number
dst-port	- destination port number
low-bound	- low bound range
up-bound	- high bound range

Examples:

This example shows how to configure a upstream rule for L4 source port range from 2000 to 3000

```
RTK.0> classf clear
RTK.0> classf set rule direction upstream
RTK.0> classf set rule range-l4port data 0 mask 0x7
RTK.0> classf set upstream-action sid-act sid 0
RTK.0> classf add entry 0
RTK.0> classf set range-l4port entry 0 type src-port
RTK.0> classf set range-l4port entry 0 low-bound 2000 up-bound 3000
```

classf get range-l4port

Description:

Get classification L4 port range setting by specific entry index

Syntax:

classf get range-l4port entry *index*

Parameter:

index - specified entry index

Examples:

This example shows how to display classification L4 port range setting

```
RTK.0> classf get range-l4port entry 0
Range check of L4 port
Index: 0 Upper: 3000 Lower: 2000 Type: Source Port
```

classf set us-1p-remark-prior

Description:

For upstream 802.1P remarking, it decide whether ACL action or classification action is high priority.

Syntax:

classf set us-1p-remark-prior < cf | acl >

Parameter:

cf	- classification is high priority
acl	- ACL is high priority

Examples:

This example shows how to configure a upstream 1P remark as ACL take high priority than classification

```
RTK.0> classf set us-1p-remark-prior acl
```

classf get us-1p-remark-prior

Description:

For upstream 802.1P remarking, it decide whether ACL action or classification action is high priority.

Syntax:

classf get range-l4port entry *index*

Parameter:

None

Examples:

This example shows how to display 802.1P remark decision for ACL and classification.

```
RTK.0> classf get us-1p-remark-prior
ACL prior than CF
```

2.17. GPON commands

gpon init

Description:

Initialize GPON module. It must be called before any GPON function.

Syntax:

gpon init

Parameter:

None

Examples:

This example shows how to initialize GPON module.

```
RTK.0> gpon init
RTK.0>
```

gpon deinitial

Description:

Deitialize GPON module.

Syntax:

gpon deinitial

Parameter:

None

Examples:

This example shows how to deinitialize GPON module.

```
RTK.0> gpon deinitial
RTK.0>
```

gpon set serial-number

Description:

Configure the serial number. It contain 4 bytes verdor ID, and 4 bytes serial number. It must be called before GPON activate command.

Syntax:

gpon set serial-number *vendor_id serial_number*

Parameter:

vendor_id	– specify the verdor ID, it is string format.
serial_number	- specify serial number, it is hex format.

Examples:

This example shows how to configure serial number.

```
RTK.0> gpon set serial-number real 0x98765432
RTK.0>
```

gpon get serial-number

Description:

Get the serial number.

Syntax:

gpon get serial-number

Parameter:

None

Examples:

This example shows how to get serial number.

```
RTK.0> gpon get serial-number
serial number: real 0x98765432
RTK.0>
```

gpon set password

Description:

Configure the serial number. It contain 10 bytes password. It must be called before GPON activate command.

Syntax:

gpon set password *password*

Parameter:

password – specify the password, it is string format.

Examples:

This example shows how to configure password.

```
RTK.0> gpon set password aaaaaaaaaa
RTK.0>
```

gpon get password

Description:

Get the password.

Syntax:

gpon get password

Parameter:

None

Examples:

This example shows how to get password.

```
RTK.0> gpon get password
password: aaaaaaaaaa
RTK.0>
```

gpon activate

Description:

Activate GPON module. After gpon activate, the GPON module will do ranging process.

Syntax:

gpon activate init-state < o1 | o7 >

Parameter:

o1	– specify the intial state to o1 state for ranging process.
o7	– specify the intial state to o7 state for ranging process.

Examples:

This example shows how to activate GPON.

```
RTK.0> gpon activate init-state o1
RTK.0>
```

gpon deactivate

Description:

Deactivate GPON module. After gpon deactivate, it will issue dying gasp messages.

Syntax:

gpon deactivate

Parameter:

None

Examples:

This example shows how to deactivate GPON.

```
RTK.0> gpon deactivate
RTK.0>
```

gpon get onu-state

Description:

Get the ONU status of ranging process.

Syntax:

gpon get onu-state

Parameter:

None

Examples:

This example shows how to get ONU status.

```
RTK.0> gpon get onu-state
ONU state: Initial State(O1)
RTK.0>
```

gpon get alarm-status

Description:

Get the ONU alarm status. It includes LOS alarm, LOF alarm and LOM alarm.

Syntax:

gpon get alarm-status

Parameter:

None

Examples:

This example shows how to get ONU alarm status. Clear means no alarm, and occur means alarm issued.

```
RTK.0> gpon get alarm-status
Alarm LOS, status: clear
Alarm LOF, status: occur
Alarm LOM, status: occur
RTK.0>
```

gpon set active-timer

Description:

Configure the timers used in raning process. It includes to1 imter and to2 timer. It must be called before GPON activate command.

Syntax:

gpon set active-timer to1 *to1_timer to2_timer*

Parameter:

to1_timer - specify the value of to1 timer
to2_timer - specify the value of to2 timer

Examples:

This example shows how to configure to1 timer and to2 timer.

```
RTK.0> gpon set active-timer to1 10000 to2 200
RTK.0>
```

gpon get active-timer

Description:

Disaply the active timer including to1 and to2.

Syntax:

gpon get active-timer

Parameter:

None

Examples:

This example shows how to display to1 and to2 timer:

```
RTK.0> gpon get active-timer
to1 timer: 10000 ms
to2 timer: 200 ms
RTK.0>
```

gpon set ds-laser

Description:

Configure downstream optical parameters. It must be called before GPON activate command.

Syntax:

gpon set ds-laser < opt_los | cdr_los > state < enable | disable > polarity < high | low >

gpon set ds-laser los_holdover < enable | disable >

Parameter:

opt_los	– specify the optical LOS input.
cdr_los	– specify the CDR LOS input.
enable	- specify to enable LOS input
disable	- specify to disable LOS input
high	– specify the polarity of LOS input is high active
low	– specify the polarity of LOS input is low active
los_holdover	– LOS will only be raised after being stable for more than 1ms.
enable	– specify to enable los_holdover function
disable	– specify to disable los_holdover function

Examples:

This example shows how to configure the downstream optical parameters including OPT LOS state, polarity, and LOS holdover function.

```
RTK.0> gpon set ds-laser opt_los state enable polarity low
RTK.0> gpon set ds-laser los_holdover disable
RTK.0>
```

gpon set ds-phy

Description:

Configure downstream physical parameters. It must be called before GPON activate command.

Syntax:

gpon set ds-phy descramble < enable | disable >

gpon set ds-phy fec-state < enable | disable > fec-threshold *threshold*

Parameter:

descramble	– specify the descramble function.
enable	– enable descramble function.
disable	– disable descramble function.
fec-state	- specify FEC function
enable	– enable downstream FEC correction function.
disable	– disable downstream FEC correction even when DS FEC encoding is enabled. The encoded parity bytes are ignored and the data will be passed to following processing modules.
fec-threshold	– D/S FEC status detection threshold, number of GPON frames. By default, it should be set to 1.
threshold	– specify the value of FEC threshold

Examples:

This example shows how to configure the downstream physical parameters including descramble, and downstream FEC functions.

```
RTK.0> gpon set ds-phy descramble enable
RTK.0> gpon set ds-phy fec-state enable fec-threshold 1
RTK.0>
```

gpon set ds-ploam

Description:

Configure downstream PLOAM parameters. It must be called before GPON activate command.

Syntax:

gpon set ds-ploam < drop-crc-error | filter-onuid | accept-broadcast > < enable | disable >

Parameter:

drop-crc-error	- specify to dropping or accept received PLOAMd messages with CRC error. (counters will be increased)
filter-onuid	– ONU_ID filtering for downstream PLOAM
accept-broadcast	– Accept broadcast PLOAMd messages
enable	– enable downstream PLOAM function
disable	– disable downstream PLOAM function

Examples:

This example shows how to configure the downstream PLOAM parameters including drop CRC error packet, filter ONU-ID, and accept broadcast functions.

```
RTK.0> gpon set ds-ploam drop-crc-error enable
RTK.0> gpon set ds-ploam filter-onuid enable
RTK.0> gpon set ds-ploam accept-broadcast enable
RTK.0>
```

gpon set ds-bwmap

Description:

Configure downstream BWMAP parameters. It must be called before GPON activate command.

Syntax:

gpon set ds-bwmap < drop-crc-error | filter-onuid | strict-plen > < enable | disable >

Parameter:

drop-crc-error	- specify to dropping or accept received BWMAP with CRC error
filter-onuid	– accept BWMap items matching provisioned T-CONTs
strict-plen	– process received PLENd in strict mode or standard mode. If set to enable, only 2 usable matching PLENd structures are accepted. If set to disable, processing in standard mode.
enable	– enable related downstream BWMAP function
disable	– disable related downstream BWMAP function

Examples:

This example shows how to configure the downstream PLOAM parameters including drop CRC error packet, filter ONU-ID, and strict PLEN functions.

```
RTK.0> gpon set ds-bwmap drop-crc-error enable
RTK.0> gpon set ds-bwmap filter-onuid enable
RTK.0> gpon set ds-bwmap strict-plen disable
RTK.0>
```

gpon set ds-gem

Description:

Configure downstream GEM parameters. It must be called before GPON activate command.

Syntax:

gpon set ds-gem assembly-threshold *threshold*

Parameter:

assembly-threshold	- specify number of GPON frames for assembly timeout threshold. The unit is 125us.
threshold	– specify the threshold value

Examples:

This example shows how to configure the downstream GEM assembly threshold parameters.

```
RTK.0> gpon set ds-gem assembly-threshold 16
RTK.0>
```

gpon set ds-eth

Description:

Configure downstream Ethernet parameters. It must be called before GPON activate command.

Syntax:

gpon set ds-eth drop-crc-error < enable | disable >
gpon set ds-eth pti-pettern *pettern* **pti-mask** *mask*

Parameter:

drop-crc-error	- specify dropping received Ethernet packet with CRC error
enable	– enable drop received Ethernet packet with CRC error
disable	– disable drop received Ethernet packet with CRC error
pettern	- PTI pattern for Ethernet fragments.
mask	- PTI mask for Ethernet fragments.

Examples:

This example shows how to configure the downstream Ethernet parameter, including drop CRC error frame, and PTI setting.

```
RTK.0> gpon set ds-eth drop-crc-error enable
RTK.0> gpon set ds-eth pti-pettern 0x1 pti-mask 0x1
RTK.0>
```

gpon set ds-omci

Description:

Configure downstream OMCI parameters. It must be called before GPON activate command.

Syntax:

gpon set ds-omci pti-pettern *pettern* **pti-mask** *mask*

Parameter:

pettern	- PTI pattern for OMCI fragments.
mask	- PTI mask for OMCI fragments.

Examples:

This example shows how to configure the downstream OMCI PTI parameter.

```
RTK.0> gpon set ds-omci pti-pettern 0x5 pti-mask 0x5
RTK.0>
```

gpon get ds parameters

Description:

Get downstream parameters.

Syntax:

gpon get < ds-laser | ds-phy | ds-ploam | ds-bwmap | ds-gem | ds-eth | ds-omci >

Parameter:

ds-laser	- get downstream optical parameter.
ds-phy	- get downstream physical parameter.
ds-ploam	- get downstream PLOAM parameter.
ds-bwmap	- get downstream BWMAP parameter.
ds-gem	- get downstream GEM parameter.
ds-eth	- get downstream ethernet parameter.
ds-omci	- get downstream OMCI parameter.

Examples:

This example shows how to display the parameters of downstream optical, physical, PLOAM, BWMAP, GEM, Ethernet, and OMCI.

```
RTK.0> gpon get ds-laser
OPT LOS state: enable
OPT LOS polarity: high
CDR LOS state: disable
CDR LOS polarity: low
LOS holdover: disable
RTK.0>
RTK.0> gpon get ds-phy
Descramble: enable
FEC state: enable
FEC threshold: 0
RTK.0>
RTK.0> gpon get ds-ploam
Accept broadcast: enable
Drop CRC error: enable
Filter ONU-ID: enable
RTK.0>
```

```
RTK.0> gpon get ds-bwmap
Drop CRC error: enable
Filter ONU-ID: enable
Strict PLENd: disable
RTK.0>
RTK.0> gpon get ds-gem
Assembly threshold: 16
RTK.0>
RTK.0> gpon get ds-eth
Drop CRC error: enable
PTI pattern: 0x1
PTI mask: 0x1
RTK.0>
RTK.0> gpon get ds-omci
PTI pattern: 0x1
PTI mask: 0x1
RTK.0>
```

gpon set us-laser

Description:

Configure upstream optical parameters. It must be called before GPON activate command.

Syntax:

gpon set us-laser on-offset *on_offset* **off-offset** *off_offset*

Parameter:

on_offset – specify the laser on offset
off_offset – specify the laser off offset

Examples:

This example shows how to configure the upstream optical parameters including laser-on and laser-off offset.

```
RTK.0> gpon set us-laser on-offset 40 off-offset 30
RTK.0>
```

gpon set us-phy

Description:

Configure upstream physical parameters. It must be called before GPON activate command.

Syntax:

gpon set us-phy scramble < enable | disable >
gpon set us-phy burst-polarity < high | low >
gpon set us-phy auto-sstart < enable | disable >
gpon set us-phy suppress-laser < enable | disable >

Parameter:

scramble	– specify the upstream scramble function.
enable	– enable scramble function.
disable	– disable scramble function.
burst-polarity	- specify burst enable polarity
high	– set burst enable polarity to high.
low	– set burst enable polarity to low.
auto-sstart	– Process Small SSTART (< BOH LEN) automatically
enable	– enable auto-sstart function.
disable	– disable auto-sstart function.
suppress-laser	– suppressing laser when ONT is outside of state 3, 4 and 5
enable	– enable suppress-laser function.
disable	– disable suppress-laser function.

Examples:

This example shows how to configure the upstream physical parameters including scramble, burst-enable polarity, auto and downstream FEC functions.

```
RTK.0> gpon set us-phy scramble enable
RTK.0> gpon set us-phy burst_polarity low
RTK.0> gpon set us-phy suppress_laser enable
RTK.0> gpon set us-phy auto_sstart enable
RTK.0>
```

gpon set us-ploam

Description:

Configure upstream PLOAM parameters. It must be called before GPON activate command.

Syntax:

gpon set us-ploam state < enable | disable >

Parameter:

enable	- sending normal upstream PLOAM
disable	– always sending no_msg PLOAM

Examples:

This example shows how to configure the upstream PLOAM function to send normal PLOAM.

```
RTK.0> gpon set us-ploam state enable
RTK.0>
```

gpon set us-dbr

Description:

Configure upstream DBA report function. It must be called before GPON activate command.

Syntax:

gpon set us-dbr state < enable | disable >

Parameter:

enable	– enable upstream DBA report function
disable	– disable upstream DBA report function If DBRu is requested by BWMAP, all ‘0’ will be sent.

Examples:

This example shows how to enable the upstream DBA report function.

```
RTK.0> gpon set us-dbr state enable
RTK.0>
```

gpon get us parameters

Description:

Get upstream parameters.

Syntax:

gpon get < us-laser | us-phy | us-ploam | us-dbr >

Parameter:

us-laser	- get upstream optical parameter.
us-phy	- get upstream physical parameter.
us-ploam	- get upstream PLOAM parameter.
ds-dbr	- get upstream DBR parameter.

Examples:

This example shows how to display the parameters of upstream optical, physical, PLOAM, and DBR.

```
RTK.0> gpon get us-laser
Laser-on offset: 40
Laser-off offset: 30
RTK.0>
RTK.0> gpon get us-phy
Scramble: enable
burst-enable polarity: low
Auto-sstart: enable
Suppressing laser: enable
RTK.0>
RTK.0> gpon get us-ploam
PLOAMu state: enable
RTK.0>
RTK.0> gpon get us-dbr
DBRu state: enable
RTK.0>
```

gpon add tcont alloc-id

Description:

Add a tcont entry by alloc-ID.

Syntax:

gpon add tcont alloc-id *id*

Parameter:

id – the alloc-ID value, from 0 to 4095

Examples:

This example shows how to add a tcont entry for alloc-ID 1000.

```
RTK.0> gpon add tcont alloc-id 1000
Add alloc_id 1000, return TCONT_id=0
RTK.0>
```

gpon del tcont alloc-id

Description:

Delete a tcont entry by alloc-ID.

Syntax:

gpon del tcont alloc-id *id*

Parameter:

id – the alloc-ID value, from 0 to 4095

Examples:

This example shows how to delete a tcont entry by alloc-ID 1000.

```
RTK.0> gpon del tcont alloc-id 1000
RTK.0>
```

gpon get tcont alloc-id

Description:

Get a tcont entry by alloc-ID.

Syntax:

gpon get tcont alloc-id *id*

Parameter:

id – the alloc-ID value, from 0 to 4095

Examples:

This example shows how to get a tcont entry by alloc-ID 1000.

```
RTK.0> gpon get tcont alloc-id 1000
alloc_id: 1000, TCONT_id 0:
RTK.0>
```

gpon show tcont

Description:

Display the added tcont entries.

Syntax:

gpon show tcont

Parameter:

None

Examples:

This example shows how to display whole tcont entries.

```
RTK.0> gpon show tcont
============================================================
     GPON ONU MAC TCONT Status
TCont ID | Alloc ID | Physical | Logical | Type
       0 |     1000 |        * |       * |    3
       1 |     2000 |        * |       * |    3
       2 |     3000 |        * |       * |    3
============================================================
RTK.0>
```

gpon add ds-flow flow-id

Description:

Add a downstream flow entry by flow-ID.

Syntax:

gpon add ds-flow flow-id *id* **gem-port** *gem* **< ether | OMCI > [multicast] [aes]**

Parameter:

id	– the flow-ID value, from 0 to 127
gem	– the GEM port value, from 0 to 4095
ether	– specify the flow is used for Ethernet packet
OMCI	– specify the flow is used for OMCI packet
multicast	– specify the flow is used for multicast packet
aes	– specify the flow is AES encryption

Examples:

This example shows how to add a downstream flow for flow-ID 0, GEM port 1000, Ethernet type, and no encryption.

```
RTK.0> gpon add ds-flow flow-id 0 gem-port 1000 ether
RTK.0>
```

gpon del ds-flow flow-id

Description:

Delete a downstream flow entry by alloc-ID.

Syntax:

gpon del ds-flow flow-id *id*

Parameter:

id – the flow-ID value, from 0 to 127

Examples:

This example shows how to delete a downstream flow by flow-ID 0.

```
RTK.0> gpon del ds-flow flow-id 0
RTK.0>
```

gpon get ds-flow flow-id

Description:

Display the downstream flow entry by flow-ID.

Syntax:

gpon get ds-flow flow-id *id*

Parameter:

id – the flow-ID value, from 0 to 127

Examples:

This example shows how to display the downstream flow entry by flow-ID 0.

```
RTK.0> gpon get ds-flow flow-id 0
flow_id  gem_port  type  multicast  AES
=======  ========  ====  =========  ===
      0      1000   ETH         no   no
RTK.0>
```

gpon show ds-flow

Description:

Display the added downstream flow entries.

Syntax:

gpon show ds-flow

Parameter:

None

Examples:

This example shows how to display whole downstream flow entries.

```
RTK.0> gpon show ds-flow
============================================================
      GPON ONU MAC D/S Flow Status
Flow ID | GEM Port | Type | Multicast | AES
       0 |     1000 |  ETH |           |
       1 |     1001 |  ETH |           |
       2 |     1002 |  ETH |           |
============================================================
RTK.0>
```

gpon add us-flow flow-id

Description:

Add a upstream flow entry by flow-ID.

Syntax:

gpon add us-flow flow-id *id* **gem-port** *gem* **< ether | OMCI >**

Parameter:

id	– the flow-ID value, from 0 to 127
gem	– the GEM port value, from 0 to 4095
ether	– specify the flow is used for Ethernet packet
OMCI	– specify the flow is used for OMCI packet

Examples:

This example shows how to add a upstream flow for flow-ID 0, GEM port 1000, Ethernet type.

```
RTK.0>  gpon add us-flow flow-id 0 gem-port 1000 ether
RTK.0>
```

gpon del us-flow flow-id

Description:

Delete a upstream flow entry by alloc-ID.

Syntax:

gpon del us-flow flow-id *id*

Parameter:

id – the flow-ID value, from 0 to 127

Examples:

This example shows how to delete a upstream flow by flow-ID 1.

```
RTK.0> gpon del us-flow flow-id 0
RTK.0>
```

gpon get us-flow flow-id

Description:

Display the upstream flow entry by flow-ID.

Syntax:

gpon get us-flow flow-id *id*

Parameter:

id – the flow-ID value, from 0 to 127

Examples:

This example shows how to display the upstream flow entry by flow-ID 1.

```
RTK.0> gpon get us-flow flow-id 0
flow_id  gem_port  type
=======  ========  ====
      0      1000   ETH
RTK.0>
```

gpon show us-flow

Description:

Display the added upstream flow entries.

Syntax:

gpon show us-flow

Parameter:

None

Examples:

This example shows how to display whole upstream flow entries.

```
RTK.0> gpon show us-flow
============================================================
    GPON ONU MAC U/S Flow Status
Flow ID | GEM Port | Type | TCont | Channel | OMCI
      0 |     1000 |  ETH |     0 |      16 |
      1 |     1001 |  ETH |     0 |      16 |
============================================================
RTK.0>
```

gpon add multicast-filter-entry

Description:

Add a multicast filter entry.

Syntax:

gpon add multicast-filter-entry mac-address *mac*

Parameter:

mac	– the multicast MAC address, the format is xx:xx:aa:bb:cc:dd. Only the LSB 32 bits aa:bb:cc:dd are used for filtering MAC DA.

Examples:

This example shows how to add a multicast filter entry for MAC address 01:00:5e:01:02:03.

```
RTK.0> gpon add multicast-filter-entry mac-address
01:00:5e:01:02:03
RTK.0>
```

gpon del multicast-filter-entry

Description:

Delete a multicast filter entry.

Syntax:

gpon del multicast-filter-entry mac-address *mac*

Parameter:

mac – the multicast MAC address, the format is xx:xx:aa:bb:cc:dd.

Examples:

This example shows how to delete a multicast filter entry by MAC address 01:00:5e:01:02:03.

```
RTK.0> gpon del multicast-filter-entry mac-address
01:00:5e:01:02:03
RTK.0>
```

gpon get multicast-filter-entry

Description:

Display the multicast filter entry by index.

Syntax:

gpon get multicast-filter-entry index *index*

Parameter:

index – the multicast filter table index, from 0 to 255

Examples:

This example shows how to display the multicast filter entry by index.

```
RTK.0> gpon get multicast-filter-entry index 0
Index | MAC Address
    0 | 01-00-5e-01-02-03
RTK.0>
```

gpon show multicast-filter-entry

Description:

Display the whole multicast filter table entries.

Syntax:

gpon show multicast-filter-entry

Parameter:

None

Examples:

This example shows how to display whole multicast filter table entries.

```
RTK.0> gpon show multicast-filter-entry
============================================================
    GPON ONU MAC Multicast MAC Filter Table
Index | MAC Address
    0 | 01-00-5e-01-02-03
============================================================
RTK.0>
```

gpon set multicast-filter

Description:

Configure multicast filter parameters.

Syntax:

gpon set multicast-filter < broadcast-pass | non-multicast-pass > < enable | disable >

gpon set multicast-filter < prefix-ipv4 | prefix-ipv6 > *prefix*

gpon set multicast-filter < force-ipv4 | force-ipv6 > < pass | drop | normal >

gpon set multicast-filter filter-mode < include | exclude >

Parameter:

broadcast-pass	– specify bypass broadcast packet
non-multicast-pass	– specify bypass non-multicast flow packet
enable	– specify to enable function
disable	– specify to disable function
prefix-ipv4	– specify multicast IPv4 prefix field
prefix-ipv6	– specify multicast IPv6 prefix field
prefix	– specify the prefix value. For IPv4, it is 3 bytes; for IPv6, it is 2 bytes.
force-ipv4	– specify multicast IPv4 force mode
force-ipv6	– specify multicast IPv6 force mode
pass	– specify to force bypass
drop	– specify to force drop
normal	– specify to non force mode
filter-mode	– specify multicast filter mode for multicast filter table entry
include	– specify to include mode, it means the DA MAC match to any entry in multicast filter table.
exclude	– specify to exclude mode, it means the DA MAC not match to any entry in multicast filter table.

Examples:

This example shows how to configure the parameters of multicast filter function. Set broadcast packet and non-multicast flow to

by pass check. Set multicast IPv4 prefix to 0x01005E, and IPv6 to 0x3333. Set non force mode for IPv4/6 multicast. And set filter mode to include mode.

```
RTK.0> gpon set multicast-filter broadcast-pass enable
RTK.0> gpon set multicast-filter non-multicast-pass enable
RTK.0> gpon set multicast-filter prefix-ipv4 0x01005e
RTK.0> gpon set multicast-filter prefix-ipv6 0x3333
RTK.0> gpon set multicast-filter force-ipv4 normal
RTK.0> gpon set multicast-filter force-ipv6 normal
RTK.0> gpon set multicast-filter filter-mode include
RTK.0>
```

gpon get multicast-filter

Description:

Display the multicast filter parameters.

Syntax:

gpon get multicast-filter

Parameter:

None

Examples:

This example shows how to get multicast filter parameters.

```
RTK.0> gpon get multicast-filter
Broadcast bypass enable
Non-multicast bypass enable
Ipv4 prefix 0x01005e
Ipv6 prefix 0x3333
Ipv4 force mode Normal
Ipv6 force mode Normal
Filter mode Including
RTK.0>
```

gpon set tx

Description:

Configure transmit laser force mode.

Syntax:

gpon set tx-laser < force-on | force-off | normal >
gpon set tx-force-idle < enable | disable >

Parameter:

tx-laser	– specify transmit laser force mode
force-on	– specify to force on
force-off	– specify to force on
normal	– specify to non force mode
tx-force-idle	– specify force transmit Idle pattern
enable	– enable force transmit Idle pattern
disable	– disable force transmit Idle pattern

Examples:

This example shows how to configure transmit laser to force send Idle pattern.

```
RTK.0> gpon set tx-laser force-on
RTK.0> gpon set tx-force-idle enable
RTK.0>
```

gpon get tx

Description:

Get transmit laser force mode.

Syntax:

gpon get tx

Parameter:

None

Examples:

This example shows how to display transmit laser mode.

```
RTK.0> gpon get tx
TX laser mode: Force on
TX force Idle: enable
RTK.0>
```

gpon show counter global

Description:

Show global counters.

Syntax:

gpon show counter global (active | ds-phy | ds-plm | ds-bw | ds-gem | ds-eth | ds-omci | us-phy | us-dbr | us-plm | us-gem | us-eth | us-omci)

Parameter:

active	– specify the activation counters, including SN_Req and Ranging_Req
ds-phy	– specify the downstream physical counters, including BIP, FEC, superframe los, and PLEN fail.
ds-plm	– specify the downstream PLOAM counters.
ds-bw	– specify the downstream BWMAP counters.
ds-gem	– specify the downstream GEM counters, including LOS, Idle, non_Idle, HEC, over_interleaving, length mismatch, and multiple flow match.
ds-eth	– specify the downstream ethernet counters, including unicast, multicast, forward multicast, leak multicast, and FCS error.
ds-omci	– specify the downstream OMCI counters, including total, CRC error, drop, and processed.
us-phy	– specify the upstream physical counters, including tx BOH.
us-dbr	– specify the upstream DBR counters, including tx DBRu.
us-plm	– specify the upstream PLOAM counters, including total, processed, urgent, processed urgent, normal, process normal, SN, and Nomsg.
us-gem	– specify the upstream GEM counters,

```
                          including block, and byte.
    us-eth                – specify the upstream ethernet
                           counters, including aborted packet.
    us-omci               – specify the upstream OMCI counters,
                           including processed OMCI.
```

Examples:

This example shows how to display global counters.

```
RTK.0> gpon show counter global active
============================================================
     GPON ONU MAC Device Counter: Activation
SN Req          : 0
Ranging Req     : 0
RTK.0>
RTK.0> gpon show counter global ds-phy
============================================================
     GPON ONU MAC Device Counter: DS PHY
BIP Error bits  : 0
BIP Error blocks: 0
FEC Correct bits: 0
FEC Correct bytes: 0
FEC Correct codewords: 0
FEC codewords Uncor: 65536
Superframe LOS  : 0
PLEN fail       : 0
RTK.0>
RTK.0> gpon show counter global ds-plm
============================================================
     GPON ONU MAC Device Counter: DS PLOAM
Total RX PLOAMd    : 1
CRC Err RX PLOAM   : 0
Corrected RX PLOAMd: 1
Proc RX PLOAMd     : 0
Overflow Rx PLOAM  : 0
============================================================
RTK.0>
```

gpon show counter tcont

Description:

Show Tcont counters, including send Idle bytes.

Syntax:

gpon show counter tcont *tcont_id*

Parameter:

tcont_id – specify the tcont-ID

Examples:

This example shows how to display Tcont counters.

```
RTK.0> gpon show counter tcont 0
============================================================
     GPON ONU MAC Device Counter: TCont 0 Gem
U/S sent GEM    : 0
============================================================
============================================================
     GPON ONU MAC Device Counter: TCont 0 Eth
U/S sent Eth Pkt: 0
============================================================
============================================================
     GPON ONU MAC Device Counter: TCont 0 Idle
U/S sent Idle Byte: 0
============================================================
RTK.0>
```

gpon show counter flow

Description:

Show flow counters, including downstream GEM packets, downstream GEM bytes, downstream RX Ethernet packets, downstream forward Ethernet packets, and upstream GEM bytes.

Syntax:

gpon show counter flow *flow_id*

Parameter:

flow_id – specify the flow-ID

Examples:

This example shows how to display flow counters.

```
RTK.0> gpon show counter flow 0
============================================================
     GPON ONU MAC Device Counter: D/S Flow 0 Gem
D/S GEM blocks  : 0
D/S GEM bytes   : 0
============================================================
============================================================
     GPON ONU MAC Device Counter: D/S Flow 0 Eth
RX Eth Pkt      : 0
Fwd Eth Pkt     : 0
============================================================
============================================================
     GPON ONU MAC Device Counter: U/S Flow 0 Gem
U/S GEM blocks  : 0
U/S GEM bytes   : 0
============================================================
RTK.0>
```

gpon omci_tx

Description:

Configure debug flag of GPON module. For debug used.

Syntax:

gpon omci_tx *data1 data2 data3 data4 data5 data6 data7 data8 data9 data10 data11*

Parameter:

data1 - daga11 – data block of OMCI packet. 4 bytes for each data block. total 44 byte.

Examples:

This example shows how to transmit a OMCI packet.

```
RTK.0> gpon omci_tx 0x11223344 0x55667788 0x99aabbcc 0x11223344
0x55667788 0x99aabbcc 0x11223344 0x55667788 0x99aabbcc
0x11223344 0x55667788
RTK.0>
```

gpon set auto-tcont

Description:

Configure to add tcont automatically when received assign ONU-ID PLOAM. Default is enable.

Syntax:

gpon set auto-tcont < enable | disable >

Parameter:

enable	– enable to add tcont automatically when received assign ONU-ID PLOAM.
disable	– disable to add tcont automatically when received assign ONU-ID PLOAM.

Examples:

This example shows how to disable adding tcont automatically when received assign ONU-ID PLOAM.

```
RTK.0> gpon set auto-tcont enable
RTK.0>
```

gpon get auto-tcont

Description:

Get to the state of add tcont automatically when received assign ONU-ID PLOAM.

Syntax:

gpon get auto-tcont

Parameter:

None

Examples:

This example shows how to get the state of add tcont automatically when received assign ONU-ID PLOAM.

```
RTK.0> gpon get auto-tcont
GPON auto TCONT state: Enable
RTK.0>
```

gpon set auto-boh

Description:

Configure to update BOH automatically when received burst overhead, extended overhead, and ranging time PLOAM. Default is enable.

Syntax:

gpon set auto- boh < enable | disable >

Parameter:

enable	– enable auto-boh.
disable	– disable auto-boh.

Examples:

This example shows how to enable auto-boh.

```
RTK.0> gpon set auto-boh enable
RTK.0>
```

gpon get auto-boh

Description:

Get to the state of auto-boh.

Syntax:

gpon get auto-boh

Parameter:

None

Examples:

This example shows how to get the state of auto-boh.

```
RTK.0> gpon get auto-boh
GPON auto BOH state: Enable
RTK.0>
```

gpon set eqd-offset

Description:

Configure to adjust eqd offset.

Syntax:

gpon set eqd-offset < plus | minus > *offset*

Parameter:

plus	– the eqd is plus offset.
minus	– the eqd is minus offset.
offset	– specify the offset value.

Examples:

This example shows how to adjust eqd by offset.

```
RTK.0> gpon set eqd-offset plus 100
RTK.0>
```

gpon get eqd-offset

Description:

Get to the eqd offset setting.

Syntax:

gpon get eqd-offset

Parameter:

None

Examples:

This example shows how to get the eqd offset.

```
RTK.0> gpon get eqd-offset
GPON EQD offset: 100
RTK.0>
```

gpon set debug

Description:

Configure debug flag of GPON module. For debug used.

Syntax:

gpon set debug < enable | disable >

Parameter:

enable	– specify to enable debug mode
disable	– specify to disable debug mode

Examples:

This example shows how to enable and disable GPON debug mode.

```
RTK.0> gpon set debug enable
RTK.0> gpon set debug disable
```

2.18. Switch commands

switch init

Description:

This command can reset & initialize switch control module.

Syntax:

switch init

Parameter:

None

Examples:

This example shows how to initialized switch module.

```
RTK.0> switch init
```

switch set 48-pass-1

Description:

This command can set switch 48-pass-1 state.

Syntax:

switch set 48-pass-1 state <disable | enable>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to enable switch 48-pass-1.

```
RTK.0> switch set 48-pass-1 state enable
```

switch get 48-pass-1

Description:

This command can get switch 48-pass-1 setting.

Syntax:

switch get 48-pass-1 state

Parameter:

None

Examples:

This example shows how to get switch 48-pass-1 setting.

```
RTK.0> switch get 48-pass-1 stat
48 Pass 1 function: Enable
RTK.0>
```

switch set ipg-compensation

Description:

This command can get switch igp compensation rate. The switch igp compensation can be enableot disable by this command.

Syntax:

switch set ipg-compensation state <disable | enable>

switch set ipg-compensation <65ppm | 90ppm>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration
65ppm	- 65ppm configuration
90ppm	- 90ppm configuration

Examples:

This example shows how to enable ipg compensation function and set rate to 90ppm.

```
RTK.0> switch get ipg-compensation
IPG compensation: 90PPM
RTK.0> switch set ipg-compensation state enable
```

switch get ipg-compensation

Description:

This command can get ipg compensation settiog.

Syntax:

switch get ipg-compensation state

switch get ipg-compensation

Parameter:

state - state configuration

Examples:

This example shows how to get ipg compensation setting.

```
RTK.0> switch get ipg-compensation state
Short IPG function: Enable
RTK.0> switch get ipg-compensation
IPG compensation: 90PPM
```

switch set bypass-tx-crc

Description:

This command can set switch tx CRC mode. Set state to disable will bypass tx CRC check, the CRC error packet will be send out.

Syntax:

switch set bypass-tx-crc state <disable | enable>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how set bypass TX CRC state to enable.

```
RTK.0> switch set bypass-tx-crc state enable
RTK.0>
```

switch get bypass-tx-crc

Description:

This command used to get bypass tx crc state.

Syntax:

switch get bypass-tx-crc state

Parameter:

None

Examples:

This example shows how to get bypass tx CRC state.

```
RTK.0> switch get bypass-tx-crc state
Bypass Tx CRC: Enable
RTK.0>
```

switch set rx-check-crc

Description:

This command can per port set rx CRC check state.

Syntax:

switch set rx-check-crc port <*PORT_LIST:ports* **| all> state <disable | enable>**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to set port 0 rx-check-crc status to disable. Port 2 can accept CRC error frames.

```
RTK.0> switch set rx-check-crc port 0 state disable
```

switch get rx-check-crc

Description:

This command can per port get rx-check-crc status.

Syntax:

switch get rx-check-crc port <*PORT_LIST:ports* **| all> state**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration

Examples:

This example shows how to get rx-check-crc status for port 0.

```
RTK.0> switch get rx-check-crc port 0 state
Port       Status
----------------------------
0          Disable
```

switch set mac-address

Description:

This command can set switch MAC address.

Syntax:

switch set mac-address *mac*

Parameter:

Mac - mac address

Examples:

This example shows how to set switch MAC address to 00:00:11:52:33:22.

```
RTK.0> switch set mac-address 00:00:11:52:33:22.
```

switch get mac-address

Description:

This command can get switch MAC address.

Syntax:

switch get mac-address

Parameter:

None

Examples:

This example shows how to get switch MAC address.

```
RTK.0> switch get mac-address
Switch MAC Address: 00:00:11:52:33:22
```

switch set max-pkt-len

Description:

This command can set maximum accept packet length for each port. Please set max-pkt-len first, and then assign each port to max-pkt-len index.

Syntax:

switch set max-pkt-len <fe | ge> port <*PORT_LIST:ports* **| all> index** *index*

switch set max-pkt-len index *index* **length** *lenth*

Parameter:

fe	- fast ethernet
ge	- giga ethernet
ports	- specified port list
all	– specify all ports
index	- index of length configuration
length	- packet length, unit byte

Examples:

This example shows how to set port 0 link in giga mode the maximum accept packet length to 1518 bytes.

```
RTK.0> switch set max-pkt-len index 0 length 1518
RTK.0> switch set max-pkt-len ge port 0 index 0
```

switch get max-pkt-len

Description:

This command can get the maximum accept packet length setting.

Syntax:

switch get max-pkt-len <fe | ge> port <*PORT_LIST:ports* **| all>**

switch get max-pkt-len index *index*

Parameter:

fe	- fast Ethernet
ge	- giga Ethernet
ports	- specified port list
all	– specify all ports
index	- index of length configuration

Examples:

This example shows how to get maximum accept packet length setting.

```
RTK.0> switch get max-pkt-len index 0
Max-Length Index 0 is Length 0 bytes.
RTK.0> switch get max-pkt-len ge port 0
Port       Speed      Config
----------------------------
0          1000M      0
```

switch set limit-pause

Description:

This command can set switch limit pause frame state.

Syntax:

switch set limit-pause state <disable | enable>

Parameter:

State	- state configuration
Disable	- disable configuration
Enable	- enable configuration

Examples:

This example shows how to set switch limit pause frame state to enable.

```
RTK.0> switch set limit-pause state enable
```

switch get limit-pause

Description:

This command can set switch limit pause frame state.

Syntax:

switch get limit-pause state

Parameter:

None

Examples:

This example shows how to get switch limit pause frame state.

```
RTK.0> switch get limit-pause state
Limit Pause Frame: Enable
```

switch set small-ipg-tag

Description:

This cmmand can set each port can accept small IPG tag or not.

Syntax:

switch set small-ipg-tag port <*PORT_LIST:ports* **| all> state <disable | enable>**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to accept small ipg for port 0.

```
RTK.0> switch set small-ipg-tag port 0 state enable
```

switch get small-ipg-tag

Description:

This command used to get accept small ipg tag setting for each port.

Syntax:

switch get small-ipg-tag port <*PORT_LIST:ports* **| all> state**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration

Examples:

This example shows how to get small ipg setting for port 0.

```
RTK.0> switch get small-ipg-tag port 0
Port       Status
----------------------------
0           Enable
RTK.0>
```

switch set small-pkt

Description:

Thos command can set each port can accept small packet or not.

Syntax:

switch set small-pkt port <*PORT_LIST:ports* **| all> state <disable | enable>**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to accept small packet for port 0.

```
RTK.0> switch set small-pkt port 0 state enable
```

switch get small-pkt

Description:

Syntax:

switch get small-pkt port <*PORT_LIST:ports* **| all> state**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration

Examples:

This example shows how to get small packet setting for port 0.

```
RTK.0> switch get small-ipg-tag port 0
Port       Status
----------------------------
0          Disable
```

switch set output-drop

Description:

When this function enabled packet will be droped in ingress port when traffic conjuest occurs. The packet type would be broadcast, unknown-unicast and multicast.

Syntax:

switch set output-drop port <*PORT_LIST:ports* | all> state <disable | enable>

switch set output-drop <broadcast | unknown-unicast | multicast> state <disable | enable>

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
broadcast	- broadcast
unknown-unicast	- unknown unicast
multicast	- multicast

Examples:

This example shows how to enable output-drop function on port 0 and set check packet type to broadcast, unknown-unicast and multicast.

```
RTK.0> switch set output-drop port 0 state enable
RTK.0> switch set output-drop broadcast state enable
RTK.0> switch set output-drop unknown-unicast state enable
RTK.0> switch set output-drop multicast state enable
```

switch get output-drop

Description:

This command can get output-drop configuration.

Syntax:

switch get output-drop port <*PORT_LIST:ports* **| all> state**

switch get output-drop <broadcast | unknown-unicast | multicast> state

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
broadcast	- broadcast
unknown-unicast	- unknown unicast
multicast	- multicast

Examples:

This example shows how to get output-drop setting for port 0.

```
RTK.0> switch get output-drop port 0 state
Port Status
0    Disable
```

switch set back-pressure

Description:

This cmmand can set switch back-pressure mode.

Syntax:

switch set back-pressure <jam | defer>

Parameter:

jam	- jam configuration
defer	- defer configuration

Examples:

This example shows how to set switch back-pressure to jam mode.

```
RTK.0> switch set back-pressure jam
```

switch get back-pressure

Description:

This command get get switch back-pressure mode.

Syntax:

switch get back-pressure

Parameter:

None

Examples:

This example shows how to get switch back-pressure mode.

```
RTK.0> switch get back-pressure
Back-pressure: jam mode
```

switch reset

Description:

This command used to reset switch.

Syntax:

switch reset <global | chip>

Parameter:

global	- global reset
chip	- whole chip reset

Examples:

This example shows how to reset the switch chip.

```
RTK.0> switch reset chip
```

2.19. Dot1x commands

dot1x init

Description:

This command can reset & initialize Dot1x module.

Syntax:

dot1x init

Parameter:

None

Examples:

This example shows how to initial Dot1x module.

```
RTK.0> dot1x init
RTK.0>
```

dot1x set port-based

Description:

This command can set Dot1x port-based state, authentication state and direction.

Syntax:

dot1x set port-based port <*PORT_LIST:ports* **| all> state <disable | enable>**

dot1x set port-based port <*PORT_LIST:ports* **| all> <auth | unauth>**

dot1x set port-based port <*PORT_LIST:ports* **| all> direction <in | both>**

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
auth	- authed configuration
unauth	- un-authed configuration
direction	- auth packet checking direction
in	- in direction
both	- both in and out direction

Examples:

This example shows how to enable port 0 port-based 1X. Set port 0 to Authed state. The auth packet checking direction set to both in and out direction.

```
RTK.0> dot1x set port-based port 0 state enable
RTK.0> dot1x set port-based port 0 auth
RTK.0> dot1x set port-based port 0 direction both
```

dot1x get port-based

Description:

This command can get port-based Dot1x setting.

Syntax:

dot1x get port-based port <*PORT_LIST:ports* **| all>**

Parameter:

ports - specified port list

all – specify all ports

Examples:

This example shows how to get port-based Dot1x status for port 0.

```
Port Direction State   AuthedStatus
0    BOTH      Enable Authorized
```

dot1x set mac-based

Description:

This command can set Dot1x mac based state, authentication state and direction.

Syntax:

dot1x set mac-based port <*PORT_LIST:ports* **| all> state <disable | enable>**

dot1x set mac-based direction <in | both>

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
disable	- disable configuration
enable	- enable configuration
direction	- auth packet checking direction
in	- in direction
both	- both in and out direction

Examples:

This example shows how to enable port 1 dot1x mac-based state to enable. The auth packet checking direction set to rx direction.

```
RTK.0> dot1x set mac-based port 1 state enable
RTK.0>
RTK.0> dot1x set mac-based direction in
RTK.0>
```

dot1x get mac-based

Description:

This command can get mac-based Dot1x setting.

Syntax:

dot1x get mac-based port <*PORT_LIST:ports* **| all> state**

dot1x get mac-based direction

Parameter:

ports	- specified port list
all	– specify all ports
state	- state configuration
direction	- authorized checking direction

Examples:

This example shows how to get mac-based Dot1x status for port 0.

```
RTK.0> dot1x get mac-based direction
Mac-based direction: BOTH
RTK.0> dot1x get mac-based port 1
Port State
1    Disable
RTK.0>
```

dot1x set trap-priority

Description:

Configure priority for packets trapped to CPU for related dot1x functions

Syntax:

dot1x set trap-priority *priority*

Parameter:

priority - priority for trapping packets

Examples:

This example shows how to configurate dot1x trap priority to 7.

```
RTK.0> dot1x set trap-priority 7
RTK.0>
```

dot1x get trap-priority

Description:

Retriving priority for packets trapped to cpu for related dot1x functions

Syntax:

dot1x get trap-priority

Parameter:

None

Examples:

This example shows how to get dot 1x trap priority.

```
RTK.0> dot1x get trap-priority
Trap priority:7
```

dot1x set unauth-packet

Description:

This command can set forwad action for unauth-packet.

Syntax:

dot1x set unauth-pakcet port <*PORT_LIST:ports* **| all> action <drop | guest-vlan | trap-to-cpu>**

Parameter:

ports	- specified port list
all	– specify all ports
drop	- drop packet
trap-to-cpu	- trap packet to cpu port
guest-vlan	- assign to guest vlan

Examples:

This example shows how to set the unauth-packet all trap-to-cpu for port 0.

```
RTK.0> dot1x set unauth-packet port 0 action trap-to-cpu
```

dot1x get unauth-packet

Description:

This command can get unauth-packet action.

Syntax:

dot1x get unauth-pakcet port <*PORT_LIST:ports* **| all> action**

Parameter:

ports	- specified port list
all	– specify all ports
action	- packet be treated configuration

Examples:

This example shows how to get unauth-packet action for port 0.

```
RTK.0> dot1x get unauth-packet port 0 action
Port Unauth Action
0    Drop
RTK.0>
```

dot1x set guest-vlan

Description:

This command use to assign vid for guest vlan and it can set the packet belong to guset vlan can forward to authed destination mac or not.

Syntax:

dot1x set guest-vlan vid *vid*

dot1x set guest-vlan to-auth-da <allow | disallow>

Parameter:

vid	– specify the numeric VLAN identifier
to-auth-da	– forward to authed destination mac
Allow	- allow configuration
Disallow	- disallow configuration

Examples:

This example can assign VID 200 to guest vlan and it set guest vlan packet can not forward to authed destination mac.

```
RTK.0> vlan create vlan-table vid 200
RTK.0> dot1x set guest-vlan vid 200
RTK.0> dot1x set guest-vlan to-auth-da disallow
```

dot1x get guest-vlan

Description:

This command can get dot1x guest vlan setting.

Syntax:

dot1x get guest-vland

Parameter:

None

Examples:

This example shows how to get dot1x guest vlan setting.

```
RTK.0> dot1x get guest-vlan
Guest vlan: 200
Allow guest vlan talk to auth. DA: Disable
RTK.0>
```

2.20. Trunk commands

trunk init

Description:

This command can reset & initialize port trunking module.

Syntax:

trunk init

Parameter:

None

Examples:

This example shows how to reset initial port trunking function.

```
RTK.0> trunk init
RTK.0> trunk get member-port

 Trunk portmask: none

RTK.0>
```

trunk set distribute-algorithm

Description:

This command can configure the distribution algorithm for trunk group.

Syntax:

trunk set distribute-algorithm [dst-ip] [dst-l4-port] [dst-mac] [src-ip] [src-l4-port] [src-mac] [src-port]

Parameter:

dst-ip	- Destination IP
dst-l4-port	- Destination Layer 4 port number
dst-mac	- Destination MAC address
src-ip	- Source IP
src-l4-port	- Source Layer 4 port number
src-mac	- Source MAC address
src-port	- Source port

Examples:

This example shows how to configure the distribution algorithm as “Destination IP” + “Source IP”.

```
RTK.0> trunk set distribute-algorithm dst-ip src-ip
RTK.0>
```

trunk get distribute-algorithm

Description:

This command can get the distribution algorithm for trunk group.

Syntax:

trunk get distribute-algorithm

Parameter:

None

Examples:

This example shows how to get the distribution algorithm.

```
RTK.0> trunk get distribute-algorithm

 Distribute Algorithm:
  Source IP
  Destination IP

RTK.0>
```

trunk set flood-mode

Description:

This command can configure the flood mode for trunk group.

Syntax:

trunk set flood-mode < normal | to-1st-logic-port >

Parameter:

normal	- Flooding packets will be forwarded according to distribution algorithm
to-1st-logic-port	- Flooding packets will be forwarded according to 1st logical port

Examples:

This example shows how to configure the flood mode to “to-1st-logical port”.

```
RTK.0> trunk set flood-mode to-1st-logic-port
RTK.0>
```

trunk get flood-mode

Description:

This command can get the flood mode for trunk group.

Syntax:

trunk get flood-mode

Parameter:

None

Examples:

This example shows how to get the flood mode.

```
RTK.0> trunk get flood-mode

 Flood Mode: Forward to first port

RTK.0>
```

trunk set flow-control

Description:

This command can configure the flow control ability for trunk group. When enabling this ability, Apollo will send pause frame to the ports which receive packet and forward the packets to trunk group. This ability is used when users want to change distribution algorithm of member ports

Syntax:

trunk set flow-control state < disable | enable >

Parameter:

disable	- Disable flow Control
enabled	- Enable flow control

Examples:

This example shows how to configure the flow control to enabled.

```
RTK.0> trunk set flow-control state enable
RTK.0>
```

trunk get flow-control

Description:

This command can get the flow control ability for trunk group.

Syntax:

trunk get flow-control

Parameter:

None

Examples:

This example shows how to get the flow control ability.

```
RTK.0> trunk get flow-control

Trunk Flow Control: Enable

RTK.0>
```

trunk set hash-mapping

Description:

This command can configure the mapping table of hash value and port ID in trunk group.

Syntax:

trunk set hash-mapping hash-value < *value* **| all > port** *port*

Parameter:

value	- Specify the hash value (0 ~ 15)
all	- Specify all hash value
port	- The port ID in trunk group

Examples:

This example shows how to configure the 16 hash values to 4 trunk ports.

```
RTK.0> trunk set hash-mapping hash-value 0 port 0
RTK.0> trunk set hash-mapping hash-value 1 port 0
RTK.0> trunk set hash-mapping hash-value 2 port 0
RTK.0> trunk set hash-mapping hash-value 3 port 0
RTK.0> trunk set hash-mapping hash-value 4 port 1
RTK.0> trunk set hash-mapping hash-value 5 port 1
RTK.0> trunk set hash-mapping hash-value 6 port 1
RTK.0> trunk set hash-mapping hash-value 7 port 1
RTK.0> trunk set hash-mapping hash-value 8 port 2
RTK.0> trunk set hash-mapping hash-value 9 port 2
RTK.0> trunk set hash-mapping hash-value 10 port 2
RTK.0> trunk set hash-mapping hash-value 11 port 2
RTK.0> trunk set hash-mapping hash-value 12 port 3
RTK.0> trunk set hash-mapping hash-value 13 port 3
RTK.0> trunk set hash-mapping hash-value 14 port 3
RTK.0> trunk set hash-mapping hash-value 15 port 3
RTK.0>
```

trunk get hash-mapping

Description:

This command can get the mapping table of hash value and port ID in trunk group.

Syntax:

trunk get hash-mapping hash-value all
trunk get hash-mapping hash-value *value*

Parameter:

all - Specify all hash value
value - specify the hash value (0 ~ 15)

Examples:

This example shows how to get all hash value in mapping table.

```
RTK.0> trunk get hash-mapping hash-value all

 Hash Value 0 to Port 0 in trunk group
 Hash Value 1 to Port 0 in trunk group
 Hash Value 2 to Port 0 in trunk group
 Hash Value 3 to Port 0 in trunk group
 Hash Value 4 to Port 1 in trunk group
 Hash Value 5 to Port 1 in trunk group
 Hash Value 6 to Port 1 in trunk group
 Hash Value 7 to Port 1 in trunk group
 Hash Value 8 to Port 2 in trunk group
 Hash Value 9 to Port 2 in trunk group
 Hash Value 10 to Port 2 in trunk group
 Hash Value 11 to Port 2 in trunk group
 Hash Value 12 to Port 3 in trunk group
 Hash Value 13 to Port 3 in trunk group
 Hash Value 14 to Port 3 in trunk group
 Hash Value 15 to Port 3 in trunk group

RTK.0>
```

trunk set member-port

Description:

This command can configure the member ports of trunk group.

Syntax:

trunk set member-port < *PORT_LIST:ports* | none >

Parameter:

ports	- Trunk port mask (0-1, 4-5)
none	- Specify no ports in trunk

Examples:

This example shows how to configure a trunk group with member ports 0-1.

```
RTK.0> trunk set member-port 0-1
RTK.0>
```

trunk get member-port

Description:

This command can get the member ports of trunk group.

Syntax:

trunk get member-port

Parameter:

None

Examples:

This example shows how to get a trunk group member ports

```
RTK.0> trunk get member-port

 Trunk portmask: 0-1

RTK.0>
```

trunk set mode

Description:

This command can configure the mode of trunk group.

Dumb mode: The destination port choosen is based on system hash value mapping table.

Normal mode: The destination port choosen is based on customer hash value mapping table.

Syntax:

trunk set mode < dumb | normal >

Parameter:

dumb	- Dumb mode
normal	- Normal mode

Examples:

This example shows how to configure a trunk mode as “normal mode”.

```
RTK.0> trunk set mode normal
RTK.0>
```

trunk get mode

Description:

This command can get the mode of trunk group.

Syntax:

trunk get mode

Parameter:

None

Examples:

This example shows how to get trunk mode.

```
RTK.0> trunk get mode
Trunk Mode: Normal mode

RTK.0>
```

trunk get queue-empty

Description:

This command can get the port mask wich all queues are empty.

Syntax:

trunk get queue-empty

Parameter:

None

Examples:

This example shows how to get queue empty mask.

```
RTK.0> trunk get queue-empty

 Queue Empty Port Mask: none

RTK.0>
```

2.21. Auto Fallback commands

auto-fallback set port

Description:

This command can configure the state of auto fallback per port basis.

Syntax:

auto-fallback set port *PORT_LIST:ports* **state < disable | enable >**

Parameter:

ports	- Specify a port list
disable	- Disable Auto fallback
enable	- Enable Auto fallback

Examples:

This example shows how to enable auto fallback at port 0 & 1.

```
RTK.0> auto-fallback set port 0-1 state enable
RTK.0>
```

auto-fallback get port

Description:

This command can get the state / status / counter of auto fallback per port basis .

Syntax:

auto-fallback get port *PORT_LIST:ports* **state**

auto-fallback get port *PORT_LIST:ports* **error-counter**

auto-fallback get port *PORT_LIST:ports* **monitor-counter**

auto-fallback get port *PORT_LIST:ports* **restore**

auto-fallback get port *PORT_LIST:ports* **valid-flow**

Parameter:

ports	- Specify a port list
state	- state of auto fallback
error-counter	- Error packet counter
monitor-counter	- Monitor packet counter
restore	- the state of restore power level
valid-flow	- the state of valid flow

Examples:

This example shows how to get auto fallback information at port 0 & 1.

```
RTK.0> auto-fallback get port 0-1 state

 Port 0 Auto Fallback State: Enable
 Port 1 Auto Fallback State: Enable

RTK.0> auto-fallback get port 0-1 error-counter

 Port 0 Error counter: 0
 Port 1 Error counter: 0
```

```
RTK.0> auto-fallback get port 0-1 monitor-counter

 Port 0 Monitort counter: 0
 Port 1 Monitort counter: 0

RTK.0> auto-fallback get port 0-1 restore

 Port 0 Restore Power Level: Not Restore Power Level
 Port 1 Restore Power Level: Not Restore Power Level

RTK.0> auto-fallback get port 0-1 valid-flow

 Port 0 Valid Flow state: None Valid Flow
 Port 1 Valid Flow state: None Valid Flow

RTK.0>
```

auto-fallback set error-count

Description:

This command can configure the error count threshold of auto fallback..

Syntax:

auto-fallback set error-count < 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 >

Parameter:

1	- Specify the error count threshold to 1
2	- Specify the error count threshold to 2
4	- Specify the error count threshold to 4
8	- Specify the error count threshold to 8
16	- Specify the error count threshold to 16
32	- Specify the error count threshold to 32
64	- Specify the error count threshold to 64
128	- Specify the error count threshold to 128

Examples:

This example shows how to configure the error count threshold as 8

```
RTK.0> auto-fallback set error-count 8
RTK.0>
```

auto-fallback get error-count

Description:

This command can get the error count threshold of auto fallback..

Syntax:

auto-fallback set error-count

Parameter:

None.

Examples:

This example shows how to get the error count threshold.

```
RTK.0> auto-fallback get error-count

 Max Error Count: 8 packets

RTK.0>
```

auto-fallback set monitor-count

Description:

This command can configure the monitor count threshold of auto fallback.

Syntax:

auto-fallback set monitor-count < 8K | 16K | 32K | 64K | 128K | 256K | 512K | 1M >

Parameter:

8K	- Specify the monitor count threshold to 8K
16K	- Specify the monitor count threshold to 16K
32K	- Specify the monitor count threshold to 32K
64K	- Specify the monitor count threshold to 64K
128K	- Specify the monitor count threshold to 128K
256K	- Specify the monitor count threshold to 256K
512K	- Specify the monitor count threshold to 512K
1M	- Specify the monitor count threshold to 1M

Examples:

This example shows how to configure the monitor count threshold as 8K

```
RTK.0> auto-fallback set monitor-count 8K
RTK.0>
```

auto-fallback get monitor-count

Description:

This command can get the monitor count threshold of auto fallback.

Syntax:

auto-fallback set monitor-count < 8K | 16K | 32K | 64K | 128K | 256K | 512K | 1M >

Parameter:

None.

Examples:

This example shows how to get the monitor count threshold.

```
RTK.0> auto-fallback get monitor-count

 Max Monitor Count: 8K packets

RTK.0>
```

auto-fallback set ignore-timeout

Description:

This command can configure the ignore timeout ability of auto fallback. When enabling this ability, TCP timeout event will not be treated as an error.

Syntax:

auto-fallback set ignore-timeout < disable | enable >

Parameter:

disable - enable ignore timeout ability
enable - disable ignore timeout ability

Examples:

This example shows how to configure the ignore timeout ability as enabled

```
RTK.0> auto-fallback set ignore-timeout enable
RTK.0>
```

auto-fallback get ignore-timeout

Description:

This command can get the ignore timeout ability of auto fallback.

Syntax:

auto-fallback get ignore-timeout

Parameter:

None.

Examples:

This example shows how to get the ignore timeout ability as enabled

```
RTK.0> auto-fallback get ignore-timeout

 Ignore Timeout State: Enable

RTK.0>
```

auto-fallback set reduce-power-level

Description:

This command can configure the ability of reducing power level when the error event is great than error count threshold.

Syntax:

auto-fallback set reduce-power-level < disable | enable >

Parameter:

disable - enable reduce power level ability
enable - disable reduce power level ability

Examples:

This example shows how to configure the reduce power level ability as enabled

```
RTK.0> auto-fallback set reduce-power-level enable
RTK.0>
```

auto-fallback get reduce-power-level

Description:

This command can get the ability of reducing power level.

Syntax:

auto-fallback get reduce-power-level

Parameter:

None

Examples:

This example shows how to get the reduce power level ability as enabled

```
RTK.0> auto-fallback get reduce-power-level

 Reduce Power Level State: Enable

RTK.0>
```

auto-fallback set timer

Description:

This command can configure the state of timer and its value.

Syntax:

auto-fallback set timer state < disable | enable >

auto-fallback set timer *timer*

Parameter:

disable	- disable timer
enable	- enable timer
timer	- the value of timer

Examples:

This example shows how to configure the timer as enabled and its value as 1000

```
RTK.0> auto-fallback set timer 1000
RTK.0> auto-fallback set timer state en
RTK.0>
```

auto-fallback get timer

Description:

This command can get the state of timer and its value.

Syntax:

auto-fallback get timer state

auto-fallback get timer

Parameter:

None

Examples:

This example shows how to get the state of timer its value.

```
RTK.0> auto-fallback get timer state

 Timer State: Enable

RTK.0> auto-fallback get timer

 Timeout threshold: 4ms

RTK.0>
```

2.22. OAM commands

oam init

Description:

This command can reset & initialize OAM module. The oam module used to control received oam packet behavior.

Syntax:

oam init

Parameter:

None

Examples:

This example shows how to initialize OAM module.

```
RTK.0> oam init
RTK.0>
```

oam set state

Description:

This command can enable or disable oam function. Enable oam function will trap oam packet to CPU.

Syntax:

oam set state <disable | enable>

Parameter:

disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to enable oam function.

```
RTK.0> oam set state enable
RTK.0>
```

oam get state

Description:

This command can get the oam status.

Syntax:

oam get state

Parameter:

None

Examples:

This example shows how to get oam state.

```
RTK.0> oam get state
OAM State is Enable
```

oam set multiplexer

Description:

This command used to set oam transmitting multiplexing function.

Syntax:

oam set multiplexer port <*PORT_LIST:ports* **| all> action <forwad | discard | from-cpu-only>**

Parameter:

ports	- specified port list
all	– specify all ports
forward	- forward non-oam packet
discard	- discard non-oam packet
from-cpu-only	- accept packet from cpu port only

Examples:

This example shows how to set port 0 oam multiplexer action to forward.

```
RTK.0> oam  set multiplexer port 0 action forward
0           Forward
RTK.0>
```

oam get multiplexer

Description:

This command can get per port oam multiplexer action.

Syntax:

oam get multiplexer port <*PORT_LIST:ports* **| all>**

Parameter:

ports - specified port list

all – specify all ports

Examples:

This example shows how to get port 0 oam multiplexer action.

```
RTK.0> oam  get multiplexer port 0
Port       Multiplexer Action
0           Forward
RTK.0>
```

oam set parser

Description:

This command can set oam parser action.

Syntax:

oam set parser port <*PORT_LIST:ports* **| all> action <forwad | loopback | discard>**

Parameter:

ports	- specified port list
all	– specify all ports
forward	- forward non-OAMPDUs
lookback	- loopback non-OAMPDUs
	- drop CRC and receiving FAILED packets
	- trap OAMPDUs to CPU
discard	- discard non-OAMPDUs

Examples:

This example shows how to set port 0 oam parser acxion to lookback.

```
RTK.0> oam set parser port 0 action loopback
Set OAM Parser
Port    Parser Action
-----------------------------
0           Loop Back
RTK.0>
```

oam get parser

Description:

This command can per port get oam parser action.

Syntax:

oam get parser port <*PORT_LIST:ports* **| all>**

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get oam parser action for port 0.

```
RTK.0> oam get parser port  0
Get OAM Parser
Port       Parser Action
----------------------------
0           Forward
RTK.0>
```

oam set trap-priority

Description:

Configure priority for packets trapped to cpu for related oam functions

Syntax:

oam set trap-priority *priority*

Parameter:

priority - priority for trapping packets

Examples:

This example shows how to configurate oam trap priority to 7.

```
RTK.0> oam set trap-priority 7
```

oam get trap-priority

Description:

Retriving priority for packets trapped to cpu for related oam function

Syntax:

oam get trap-priority

Parameter:

None

Examples:

This example shows how to get oam trap priority.

```
RTK.0> oam get trap-priority
OAM Trap Priority is 7
RTK.0>
```

2.23. Loop commands

rldp init

Description:

This command can reset & initialize RLDP module.

Syntax:

rldp init

Parameter:

None

Examples:

This example shows how to initialize RLDP module.

```
RTK.0> rldp init
RTK.0>
```

rldp set state

Description:

These commands configure the enable/disable RLDP function of the entire chip and each port. Chip level state configuration has higher precedence then per port state control.

Syntax:

rldp set state <disable | enable>

rldp set port <*PORT_LIST:ports* | all> state <disable | enable>

Parameter:

disable	- disable configuration
enable	- enable configuration
ports	- specified port list
all	– specify all ports

Examples:

This example shows how to enable the chip and port 1-3’s RLDP function.

```
RTK.0> rldp set state enable
RLDP: Disable -> Enable
RTK.0> rldp set port 1-3 state enable
Set RLDP port state
Port state
---- ------------------
1    Disable -> Enable
2    Disable -> Enable
3    Disable -> Enable
RTK.0>
```

rldp get state

Description:

These commands get the chip and port RLDP enable/disable configuration.

Syntax:

rldp get state

rldp get port <*PORT_LIST:ports* **| all> state**

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get chip level and port 0-3’s RLDP configuraitons.

```
RTK.0> rldp get state
RLDP Enable
RTK.0> rldp get port 0-3 state
Get RLDP port state
Port state
---- --------
0    Disable
1    Enable
2    Enable
3    Enable
```

rldp get status

Description:

This command gets the currently loop status of each port. It display if the port is currently in looping status and if each port ever entering/leaving looping status.

Syntax:

rldp get port <*PORT_LIST:ports* | all> status

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to display port 1 and port 3s' RLDP status.

```
RTK.0> rldp get port 1,3 status
Get RLDP port looped state
Port looping enter leave
---- ------- ----- -----
1
3
RTK.0>
```

rldp clear status

Description:

This command clears the entering/leaving status of specified port.

Syntax:

rldp clear port <*PORT_LIST:ports* | all> status <entering | leaving>

Parameter:

ports	- specified port list
all	– specify all ports
entering	- enter looping status
leaving	- leave looping status

Examples:

This example shows how to clear port 0, 3s’ entering status and port 1, 2s’ leaving status.

```
RTK.0> rldp clear port 0,3 status entering
Clear RLDP port looped state
Port enter leave
---- ----- -----
0      V
3      V
RTK.0>
RTK.0> rldp clear port 1,2 status leaving
Clear RLDP port looped state
Port enter leave
---- ----- -----
1            V
2            V
RTK.0>
```

rldp set magic

Description:

This command sets the magic of outgoing RLDP frames.

Syntax:

rldp set magic *mac*

Parameter:

mac - mac address

Examples:

This example shows how to configure the magic of RLDP.

```
RTK.0> rldp set magic 01:02:03:04:05:06
RLDP Magic: 00:00:00:00:00:00 -> 01:02:03:04:05:06
RTK.0>
```

rldp get magic

Description:

This command gets the magic used by RLDP.

Syntax:

rldp get magic

Parameter:

None

Examples:

This example shows how to get magic of RLDP.

```
RTK.0> rldp get magic
RLDP Magic: 01:02:03:04:05:06
RTK.0>
```

rldp get identifier

Description:

This command gets the current identifier used by RLDP which is generated by random number generator.

Syntax:

rldp get identifier

Parameter:

None

Examples:

This example shows how to get identifier of RLDP.

```
RTK.0> rldp get identifier
RLDP Identifier: 5A:E7:BD:3C:81:96
RTK.0>
```

rldp set compare-typ

Description:

This command configures the RLDP compare type to either compare both identifier and magic or magic only.

Syntax:

rldp set compare-typ <magic-and-identifier | magic-only>

Parameter:

magic-and-identifier	- checking both magic and identifier
magic-only	- checking magic only

Examples:

This example shows how to configure RLDP to compare magic only.

```
RTK.0> rldp set compare-type magic-only
RLDP Compare Type: Magic Number + Identifier -> Magic Number
RTK.0>
```

rldp get compare-typ

Description:

This command gets the current compare type used by RLDP.

Syntax:

rldp get compare-typ

Parameter:

None

Examples:

This example shows how to get compare type of RLDP.

```
TK.0> rldp get compare-type
RLDP Compare Type: Magic Number
RTK.0>
```

rldp set period

Description:

This command set the check/loop state period of RLDP.

Syntax:

rldp set <check | loop> period *time*

Parameter:

check	- checking state
loop	- looping state
time	- time to send packets, unit 1 ms

Examples:

This example shows how to configure RLDP’s check and loop state period.

```
RTK.0> rldp set check period 5000
RLDP check period: 2000 ms -> 5000 ms
RTK.0>
RTK.0> rldp set loop period 5000
RLDP loop period: 2000 ms -> 5000 ms
RTK.0>
```

rldp get period

Description:

This command gets the check/loop state period of RLDP.

Syntax:

rldp get <check | loop> period

Parameter:

check	- checking state
loop	- looping state

Examples:

This example shows how to get RLDP’s check/loop state period.

```
RTK.0> rldp get check period
RLDP check period: 5000 ms
RTK.0> rldp get loop period
RLDP loop period: 5000 ms
RTK.0>
```

rldp set number

Description:

This command sets the check/loop state of RLDP frame send count.

Syntax:

rldp set <check | loop> number *count*

Parameter:

check	- checking state
loop	- looping state
count	- number of sending packets

Examples:

This example shows how to configure RLDP’s check and loop state frame send count.

```
RTK.0> rldp set check number 5
RLDP check number count: 4 -> 5
RTK.0> rldp set loop number 3
RLDP loop number count: 4 -> 3
RTK.0>
```

rldp get number

Description:

This command gets the check/loop state of RLDP send count.

Syntax:

rldp get <check | loop> number

Parameter:

check	- checking state
loop	- looping state

Examples:

This example shows how to get send count of RLDP check and loop state.

```
RTK.0> rldp get check number
RLDP check number count: 5
RTK.0> rldp get loop number
RLDP loop number count: 3
RTK.0>
```

rldp set control-state

Description:

This command set the force looping state of each port to indicate that the ports has entering loop state.

Syntax:

rldp set port <*PORT_LIST:ports* **| all> control-state <looping | non-looping>**

Parameter:

ports	- specified port list
all	– specify all ports
looping	- looping state
non-looping	- non-looping state

Examples:

This example shows how to configure port 0-2 enter loop state and leave.

```
RTK.0> rldp set port 0-2 control-state looping
RTK.0>
RTK.0> rldp get port 0-3 status
Get RLDP port looped state
Port looping enter leave
---- ------- ----- -----
0       V           V
1       V           V
2       V           V
3
RTK.0> rldp set port 1,2 control-state none-looping
RTK.0>
RTK.0> rldp get port 0-3 status
Get RLDP port looped state
Port looping enter leave
---- ------- ----- -----
0       V           V
1                   V
2                   V
3
RTK.0>
```

rldp get control-state

Description:

This command gets the force looping state of RLDP.

Syntax:

rldp get port <*PORT_LIST:ports* **| all> control-state**

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how get port 0-4s’ the force looping state of RLDP.

```
RTK.0> rldp get port 0-4 control-state
RLDP port control-state
Port state
---- ------------
   0: Looping
   1: None-looping
   2: None-looping
   3: None-looping
   4: None-looping
```

rldp get looped-port-id

Description:

This command get the port ID detected by RLDP which is looped with specified port.

Syntax:

rldp get port <*PORT_LIST:ports* **| all> looped-port-id**

Parameter:

ports	- specified port list
all	– specify all ports

Examples:

This example shows how to get the looped port id of the specified port.

```
RTK.0> rldp get port 0-4 looped-port-id
Get RLDP port looped port id
Port looped port
---- -----------
0              0
1              0
2              0
3              0
4              0
```

rldp set re-generate-identifier

Description:

This command regenerates the identifier used by RLDP.

Syntax:

rldp set re-generate-identifier

Parameter:

None

Examples:

This example shows how to regenerate the identifier of RLDP.

```
RTK.0> rldp get identifier
RLDP Identifier: AD:86:E0:14:AE:5B
RTK.0> rldp set re-generate-identifier
RTK.0> rldp get identifier
RLDP Identifier: 72:74:33:30:4B:0D
RTK.0>
```

rldp set handle

Description:

This command configures that RLDP frame should processed by chip or software.

Syntax:

rldp set handle <hardware | software>

Parameter:

hardware	- detect by hardware
software	- detect by software

Examples:

This example shows how to configure the RLDP should processed by chip hardware.

```
RTK.0> rldp set handle hardware
RTK.0>
```

rldp get handle

Description:

This command gets the current RDLP handler.

Syntax:

rldp get handle

Parameter:

None

Examples:

This example shows how get the current RLDP handler.

```
RTK.0> rldp set handle hardware
RTK.0>
```

rldp set mode

Description:

This command configures the RLDP detection mode to either sa-moving or periodic mode.

Syntax:

rldp set mode <sa-moving | periodic>

Parameter:

sa-movin	- detect by host port moving
periodic	- detect periodicly

Examples:

This example shows how to configure the RLDP to periodic mode.

```
RTK.0> rldp set mode periodic
RTK.0>
```

rldp get mode

Description:

This command gets the current RLDP detection mode.

Syntax:

rldp get mode

Parameter:

None

Examples:

This example shows how to get the current RLDP detection mode.

```
RTK.0> rldp get mode
RLDP Mode: Peroidical
```

rldp set bypass-flow-control

Description:

This command configures if RLDP affects by flow control mechanism.

Syntax:

rldp set bypass-flow-control state <disable | enable>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how to configure RLDP to bypass flow control mechanism.

```
RTK.0> rldp set bypass-flow-control state enable
RLDP Bypass: Disable ->Enable
RTK.0>
```

rldp get bypass-flow-control

Description:

This command gets the current state of RLDP bypass flowcontrol.

Syntax:

rldp get bypass-flow-control state

Parameter:

state - state configuration

Examples:

This example shows how to get the current bypass flowcontrol state of RLDP.

```
RTK.0> rldp get bypass-flow-control
RLDP Bypass: Enable
RTK.0>
```

rlpp set trap

Description:

This command configure if the RLPP packet is trapped to CPU or not.

Syntax:

rldp set trap state <disable | enable>

Parameter:

state	- state configuration
disable	- disable configuration
enable	- enable configuration

Examples:

This example shows how set RLPP packet to no trapped to CPU.

```
RTK.0> rlpp set trap state disable
RTK.0>
```

rlpp get trap

Description:

This command gets current RLPP trap state.

Syntax:

rlpp get trap state

Parameter:

state - state configuration

Examples:

This example shows how to get the trap state of RLPP.

```
RTK.0> rlpp get trap
RLPP Trap to CPU: Disable
RTK.0>
```

Interrupt commands

interrupt init

Description:

This command can reset interrupt mask and clear interrupt status.

Syntax:

interrupt init

Parameter:

None

Examples:

This example shows how intitalize interrupt.

```
RTK.0> interrupt init
```

interrupt set mask

Description:

This command can turn on/off the interrupt mask.

Syntax:

interrupt set mask < all | link-change | meter-exceed | learn-over | speed-change | special-congestion | loop-detection | cable-diag | acl | gphy | serdes | gpon | epon | ptp | dying-gasp | thermal | adc > state < disable | enable >

Parameter:

all	- select all interrupt mask bit
link-change	- mask for link change interrupt
meter-exceed	- mask for meter exceed interrupt
learn-over	- mask for L2 table learn over interrupt
speed-change	- mask for port speed change interrupt
special-congestion	- mask for special congestion interrupt
loop-detection	- mask for loop dection interrupt
cable-diag	- mask for cable diagnostic interrupt
acl	- mask for acl interrupt
gphy	- mask for GPHY interrupt
serdes	- mask for Serdes interrupt
gpon	- mask for GPON interrupt
epon	- mask for EPON interrupt
ptp	- mask for PTP interrupt
dying-gasp	- mask for DyingGasp interrupt
thermal	- mask for Thermal exceed interrupt
adc	- mask for ADC interrupt
disable	- disable the interrupt mask
enable	- enable the interrupt mask

Examples:

This example show how to turn on GPON interrupt mask.

```
RTK.0> interrupt set mask gpon state enable
```

interrupt get mask

Description:

This command can turn on/off the interrupt mask.

Syntax:

interrupt get mask < all | link-change | meter-exceed | learn-over | speed-change | special-congestion | loop-detection | cable-diag | acl | gphy | serdes | gpon | epon | ptp | dying-gasp | thermal | adc > state

Parameter:

all	- select all interrupt mask bit
link-change	- mask for link change interrupt
meter-exceed	- mask for meter exceed interrupt
learn-over	- mask for L2 table learn over interrupt
speed-change	- mask for port speed change interrupt
special-congestion	- mask for special congestion interrupt
loop-detection	- mask for loop dection interrupt
cable-diag	- mask for cable diagnostic interrupt
acl	- mask for acl interrupt
gphy	- mask for GPHY interrupt
serdes	- mask for Serdes interrupt
gpon	- mask for GPON interrupt
epon	- mask for EPON interrupt
ptp	- mask for PTP interrupt
dying-gasp	- mask for DyingGasp interrupt
thermal	- mask for Thermal exceed interrupt
adc	- mask for ADC interrupt

Examples:

This example show how to display GPON interrupt mask.

```
RTK.0> interrupt get mask gpon state
IMR: Enable
```

interrupt get status

Description:

This command get the interrupt status.

Syntax:

interrupt get status < all | link-change | meter-exceed | learn-over | speed-change | special-congestion | loop-detection | cable-diag | acl | gphy | serdes | gpon | epon | ptp | dying-gasp | thermal | adc >

Parameter:

all	- select all interrupt mask bit
link-change	- status for link change interrupt
meter-exceed	- status for meter exceed interrupt
learn-over	- status for L2 table learn over interrupt
speed-change	- status for port speed change interrupt
special-congestion	- status for special congestion interrupt
loop-detection	- status for loop dection interrupt
cable-diag	- status for cable diagnostic interrupt
acl	- status for acl interrupt
gphy	- status for GPHY interrupt
serdes	- status for Serdes interrupt
gpon	- status for GPON interrupt
epon	- status for EPON interrupt
ptp	- status for PTP interrupt
dying-gasp	- status for DyingGasp interrupt
thermal	- status for Thermal exceed interrupt
adc	- status for ADC interrupt

Examples:

This example show how to display GPON interrupt status.

```
RTK.0> interrupt get status gpon
IMS: Disable
```

interrupt clear status

Description:

This command clear the interrupt status.

Syntax:

interrupt clear status < all | link-change | meter-exceed | learn-over | speed-change | special-congestion | loop-detection | cable-diag | acl | gphy | serdes | gpon | epon | ptp | dying-gasp | thermal | adc >

Parameter:

all	- select all interrupt mask bit
link-change	- status for link change interrupt
meter-exceed	- status for meter exceed interrupt
learn-over	- status for L2 table learn over interrupt
speed-change	- status for port speed change interrupt
special-congestion	- status for special congestion interrupt
loop-detection	- status for loop dection interrupt
cable-diag	- status for cable diagnostic interrupt
acl	- status for acl interrupt
gphy	- status for GPHY interrupt
serdes	- status for Serdes interrupt
gpon	- status for GPON interrupt
epon	- status for EPON interrupt
ptp	- status for PTP interrupt
dying-gasp	- status for DyingGasp interrupt
thermal	- status for Thermal exceed interrupt
adc	- status for ADC interrupt

Examples:

This example show how to clear GPON interrupt status.

```
RTK.0> inter clear status gpon
```

interrupt get status detail

Description:

This command get the specific interrupt status for ports or PHYs.

Syntax:

interrupt get status < speed-change | link-up | link-down | gphy > detail

Parameter:

speed-change	- status for port speed change interrupt
link-up	- status for link up interrupt
link-down	- status for link down interrupt
gphy	- status for GPHY interrupt

Examples:

This example shows how to display link up interrupt detail status.

```
RTK.0> interrupt get status link-up detail
Status: 0x00000000
```

interrupt clear status detail

Description:

This command clear the specific interrupt status for ports or PHYs.

Syntax:

interrupt clear status < speed-change | link-up | link-down | gphy > detail

Parameter:

speed-change	- status for port speed change interrupt
link-up	- status for link up interrupt
link-down	- status for link down interrupt
gphy	- status for GPHY interrupt

Examples:

This example show how to clear all ports link up interrupt status.

```
RTK.0> interrupt clear status link-up detail
```

interrupt set polarity

Description:

This command configure the interrupt polarity is high active or low active while trigged.

Syntax:

interrupt set polarity < high | low >

Parameter:

high	- high active
low	- low active

Examples:

This example show how to set the interrupt polarity as high active.

```
RTK.0> interrupt set polarity high
```

interrupt get polarity

Description:

This command get the interrupt polarity is high active or low active while trigged.

Syntax:

interrupt set polarity

Parameter:

None

Examples:

This example show how to display the interrupt polarity.

```
RTK.0> interrupt get polarity
Polarity: High
```

Realtek Semiconductor Corp.

Headquarters

No. 2, Innovation Road II

Hsinchu Science Park, Hsinchu 300, Taiwan

Tel.: +886-3-578-0211. Fax: +886-3-577-6047

www.realtek.com