Document Number	EDCS- 837266
Based on Template	EDCS-160995 Rev 11
Created By	Cisco Systems, Inc.

VQE-C
CLI Command Reference
Release 3.5.10

This document provides the CLI command set reference for the VQE-C (Visual Quality Experience Client).

Modification History

Revision	Date	Originator	Comments
0.1	01/11/10	Cisco Systems, Inc.	Changes over 3.4 version: 1. 'rcc with loss' counter added to "show counters" output 2. 'drop' command description revised to reflect new 'session' keyword option and changed behavior of 'interval' keyword.
0.2	02/05/10	Cisco Systems, Inc.	Added section for “show tech-support” command
0.3	02/05/10	Cisco Systems, Inc.	Updated the section for “help”
0.4	04/06/10	Cisco Systems, Inc.	Changes for 3.5.3: 1. Updated “show tuner” command output with source failover fields. 2. Updated “debug” and “show debug” for dp-failover option.
0.5	09/16/10	Cisco Systems, Inc.	Changes for 3.5.5: 1. Updated “show tuner” command output with new source failover timing fields.

Contents

1 Overview

This document is intended as a reference for the VQE-C CLI commands. It assumes the reader is familiar with terms and concepts such as dropping packets, tuners, and error repair as they relate to VQE-C. More information on these topics can be found in the high-level VQE reference documents and the VQE-C System Integration Guide.

The VQE-C CLI, based on the open source library libcli (http://sourceforge.net/projects/libcli), is designed primarily for testing and debugging the VQE-C software. It is not intended to be IOS-compatible, **nor is it intended for deployment**, although it will be accessible by vendors. The scope of the CLI is limited to the VQE-C software only; it will not be able to control or provide information on any other functions of a set-top box or any host system on which VQE-C is running.

The CLI in VQE-C is accessible by telnet. VQE-C listens for incoming telnet connections on the port configured in the VQE-C system configuration. More information about this and other VQE-C system configuration parameters can be found in the VQE-C System Configuration Reference document.

2 Conventions and Format

The convention used in this document for the command grammar is **keyword** *argument* [*optional-argument*] {*argument-choice-1* | *argument-choice-2*}.

Similar to the IOS CLI, the VQE-C CLI has different operating modes. Each mode allows a certain set of commands:

- EXEC mode
  - Allows commands that are not potentially destructive
- Privileged EXEC mode
  - Allows all EXEC mode commands (except 'enable') and commands that have more control over VQE-C's functionality and also have greater potential for destruction
- Configure mode
  - Allows commands to control and configure the basic functionality of VQE-C

The VQE-C CLI has some built-in features to promote ease-of-use. A list of these features to keep in mind while using the VQE-C CLI is as follows:

- type '**?**' at any time to display possible completions or usage information for a command.
- use the up-arrow and down-arrow to cycle backwards and forwards through command history.
- use "*command* **| include** *pattern*" to only display lines from the resulting output of *command* that contain the string *pattern*.

Whenever any change is made to the VQE-C configuration via the CLI, the new configuration is only guaranteed to take effect on the next channel change, after all tuners have been unbound from that particular channel. For example, enabling error repair will only become active on the next channel change (where the channel being bound to has no prior bindings to it). Also, it should be noted that some commands may mask the effects of others. In particular, the **drop interval** and **drop percentage** commands will have no noticeable effect until **drop enable** is given. It is also important to note that none of the effects of these commands are saved through a restart of VQE-C.

3 List of Commands

EXEC mode

help
history
quit
logout
exit
enable
show channel
show counters
show debug
show dp
show drop
show error-repair
show fec
show rcc
show nat
show pak-pool
show proxy-igmp
show stream-output
show system-config
show tech-support
show tuner
show update
show ipc

Privileged EXEC mode (excluding those also available in EXEC mode)

disable
configure terminal
clear counters
send-debugs-to-cli
debug
monitor

Configure mode

app-delay
channel tr-135
drop
error-repair
error-repair policer
fec
rcc
update
parse sdp
proxy-igmp-join
stream-output
tuner bind
tuner create
tuner destroy
tuner unbind

3.1 EXEC mode

help

To show a list of available commands in the current mode, use the **help** command.

help

Command Modes All modes

Examples The following example lists all commands that are available in user EXEC mode:

```
vqec# help

  help                Show available commands
  quit                Disconnect
  logout              Disconnect
  exit                Exit from current mode
  enable              Turn on privileged commands
  disable             Turn off privileged commands
  configure           Enter configuration mode
  clear               clear cmds
  send-debugs-to-cli  Set whether or not debug messages are to be printed
                       on the CLI output
  debug               Set debugging flags
  monitor             Performance monitoring tools
  show                Commands to display VQE-C state information
```

history

To show a list of previously entered commands, use the history command.

history

Command Modes All modes

Examples The following example lists all commands that were previously issued:

```
vqec# history
Command history:
  0. enable
  1. show debug
```

quit
logout

To disconnect the telnet session from the CLI, use the **quit** or **logout** command.

quit
logout

Command Modes All modes

Examples The following example disconnects the telnet session:

```
vqec> quit
Connection closed by foreign host.
```

exit

To exit from the current command level (except when trying to get back to non-privileged EXEC mode), use the **exit** command.

exit

Command Modes All modes

Examples The following example exits global configuration mode and returns to privileged EXEC mode, then exits the telnet session after the command is issued a second time:

```
vqec(config)# exit
vqec# exit
Connection closed by foreign host.
```

enable

To enter privileged EXEC mode, use the **enable** command.

enable

Command Modes EXEC

Examples The following example enters privileged EXEC mode from user EXEC mode:

```
vqec> enable
vqec#
```

show channel

To show a list of configured channels, or to show details for one a configured channel, use the **show channel** command.

show channel [{counters {all | url <channel_url>} [cumulative] } |
{config {all | url <channel-url>}}]

Syntax Description

counters {all | url <channel_url>} [cumulative]	(optional) Prints the counters associated with the specified channel(s). 'All' specifies all channels. If the optional keyword "cumulative" is supplied, the counters in the time-window from channel initialization to 'now' will be shown. If the cumulative keyword is absent, then counters in the time window from last 'clear counters/initialization' to 'now' would be shown.
config {all | url <channel_url>}	(optional) Prints the configuration associated with the specified channel(s). 'All' specifies all channels.

Command Modes

EXEC

Usage Guidelines

When no arguments are provided, this command will simply display a list of the configured channels by their URLs and names.

The **show channel counters url <channel_url>** [cumulative] command displays the counters for a single channel (since last reset/initialization to 'now'), and **show channel counters all** [cumulative] displays the counters for each channel one after the other. If the optional keyword 'cumulative' is supplied, the command will display the counters for each channel since its initialization to 'now', and if absent, the command will display counters since last reset/channel initialization (whichever occurred latest) to 'now'. See the **show counters command** for descriptions of fields common to both these commands. The following are counters that are unique to "show channel counters".

Field Name	Type	Description
primary rtp expected	uint64	Number of packets expected for this channel's primary session. From RFC 3550: The number of packets expected can be computed by the receiver as the difference between the highest sequence number received and the first sequence number received.
primary rtp lost	uint64	Number of packets lost for this channel's primary session From RFC 3550: The number of packets lost is defined to be the number of packets expected less the number of packets actually received.
TR-135 Packet Counters		
buffersize (usec)	uint64	Channel's current jitter buffer size, which is currently derived from the

		System Configuration at the time the channel was created.
underruns (events)	uint64	Underruns are counted at the time the jitter buffer is read, and no packets are available for reading.
overruns (events)	uint64	Overruns are counted at the time packets arrive, and an attempted write of a packet to the jitter buffer fails due to an overflow condition.
gmin	uint64	Minimum number of consecutive received packets after the end of an RTP loss event. A loss event is defined as a sequence of lost packets, possibly including islands of received packets. Each island consists of up to (Gmin – 1) received packets (a sequence of Gmin received packets terminates the loss event, and so is not an island).
severe loss minimum distance	uint64	The minimum distance required between error events before an RTP loss event is considered severe
packets expected after error correction	uint64	Total number of RTP packets expected for a given AV session as described in RFC 3550 after error correction
packets lost before error correction	uint64	Total number of RTP packets lost for a given session. These statistics are collected when no error correction is applied
packets lost after error correction	uint64	Packets lost after error correction is applied
loss events before error correction	uint64	A loss event is defined as a sequence of lost packets, possibly including islands of received packets. Each island consists of up to (Gmin – 1) received packets (a sequence of Gmin received packets terminates the loss event, and so is not an island).
loss events after Error correction	uint64	Loss events after EC is applied
severe loss index count	uint64	This is the total number of loss events closer than Severe Loss Min. Distance. These stats are collected after error correction is applied
minimum loss distance	uint64	The smallest number of RTP packets between two consecutive loss events, measured after error correction is applied.
maximum loss period	uint64	The maximum number of RTP packets of the longest loss event measured after error correction is applied.

The show **channel config url <channel_url>** command displays the configuration for a single channel, and **show channel config all** displays the configuration for each channel one after the other. See the **show tuner command** for descriptions of each output field.

Examples

The following example lists all the configured channels, and then displays more details on the second channel listed:

```
vqec> show channel
VQE-C channel cfg update:   not in progress
Last update received:       Jan 21 13:03:22
Channel cfg file version:   136a4169327228e2f980f1b03c95e022
Total number of channels:   10

 rtp://230.151.1.1:10000 (Channel 230.151.1.1)
 rtp://230.151.1.2:10004 (Channel 230.151.1.2)
 rtp://230.151.1.3:10008 (Channel 230.151.1.3)
 rtp://230.151.1.4:10012 (Channel 230.151.1.4)
 rtp://230.151.1.5:10016 (Channel 230.151.1.5)
 rtp://230.151.1.6:10020 (Channel 230.151.1.6)
 rtp://230.151.1.7:10024 (Channel 230.151.1.7)
 rtp://230.151.1.8:10028 (Channel 230.151.1.8)
 rtp://230.151.1.9:10032 (Channel 230.151.1.9)
 rtp://230.151.1.10:10036 (Channel 230.151.1.10)

vqec> show channel config url rtp://230.151.1.1:10000
Channel name: Channel 230.151.1.1
Channel sdp_handle: o=- 1209133068 1209133067 IN IP4 venus-iptv
Channel handle: 0x89000001
Channel session identifier: INIP4#-#1209133068#venus-iptv
Channel version: 1209133067
Configuration data: complete
Channel mode: lookaside
Original source multicast address: 230.151.1.1
Source address for original source stream: 5.8.37.2
Original source port: 10000
Original source RTCP port: 10001
Original source RTP payload type: 33
Original source RTCP sender bandwidth: 46875
Original source RTCP receiver bandwidth: 140625
Original source RTCP per receiver bandwidth: 37
Original source RTCP XR Loss RLE Report: Off
Original source RTCP XR Stat Summary Report: 0x0000
RTCP XR Post Repair Loss RLE Report: Off
Maximum bit rate: 3750000
Retransmission/FBT address: 8.36.1.1
Retransmission RTP port: 10002
Retransmission RTCP port: 10003
Retransmission associated payload type: 33
Repair stream RTCP sender bandwidth: 37
Repair stream RTCP receiver bandwidth: 37
Error repair: enabled
Rapid channel change: enabled

vqec> show channel counters url rtp://230.151.1.1:10000
primary rtp expected:    470767

primary rtp lost:        473


-- TR-135 Packet Counters --
```

```
buffersize (usec):              200000
underruns (events):             0
overruns (events):              1
gmin:                           1
severe loss minimum distance:   2
                            Before-EC     After-EC
packets expected:           -             15868
packets received:           -             15653
packets lost:               331           215
loss events:                266           174
severe loss index count:    -             2
minimum loss distance:      -             1
maximum loss period:        -             3

-- Stream Packet Counters --
                    Inputs        Outputs       Drops (Late)
primary rtp:        470767        -             473 (0)
primary rtcp:       0             436           -
repair/rcc rtp:     32823         -             1 (1)
repair/rcc rtcp:    147           -             -
fec:                0             -             0 (0)
repair rtp stun:    0             0             -
repair rtcp stun:   0             0             -
post-repair:        -             502240        -
tuner Q drops:      -             -             0
underruns:          0             -             -

-- Repair Packet Counters --
                    Pre-Repair    Post-Repair
stream loss:        473           215
 rcc loss:          -             0
                    Requested     Policed
error repair:       473           0
                    Recovered
fec:                0

-- Channel Change Counters --
requests:           0
 rcc requests:      0
  rcc with loss:    0
  rcc aborts:       0
    server    stun      response  response  burst     burst
    rejects   timeout   timeout   invalid   start     activity
                  other
    0         0         0         0         0         0         0
vqec> show channel counters url rtp://230.151.1.1:10000 cumulative
primary rtp expected:    488653

primary rtp lost:        2227


-- TR-135 Packet Counters --
buffersize (usec):              200000
underruns (events):             1
overruns (events):              1
gmin:                           1
severe loss minimum distance:   2
                            Before-EC     After-EC
packets expected:           -             33753
packets received:           -             32353
packets lost:               2075          1400
loss events:                1664          1084
severe loss index count:    -             22
```

```
minimum loss distance:      -            1
maximum loss period:        -            5

-- Stream Packet Counters --
                   Inputs        Outputs       Drops (Late)
primary rtp:       488653        -             2227 (0)
primary rtcp:      0             1691          -
repair/rcc rtp:    51341         -             1 (1)
repair/rcc rtcp:   154           -             -
fec:               0             -             0 (0)
repair rtp stun:   0             0             -
repair rtcp stun:  0             0             -
post-repair:       -             536371        -
tuner Q drops:     -             -             0
underruns:         0             -             -

-- Repair Packet Counters --
                   Pre-Repair    Post-Repair
stream loss:       2227          1400
 rcc loss:         -             0
                   Requested     Policed
error repair:      2226          0
                   Recovered
fec:               0

-- Channel Change Counters --
requests:          1
 rcc requests:     1
  rcc with loss:   0
  rcc aborts:      0
    server    stun      response  response  burst     burst
    rejects   timeout   timeout   invalid   start     activity
                     other
    0         0         0         0         0         0         0
```

show counters

To show counters, use the **show counters** command.

show counters [{name <tuner-name>} | {cumulative}]

Syntax Description

tuner-name Name of an active tuner.

Command Modes

EXEC

Usage Guidelines

To show historical/rolled-up counters for events which have occurred across all tuners/channels, including events which occurred on tuners/channels which may no longer exist (or whose binding has changed), use the “show counters" command. When the “show counters” command is invoked with the “cumulative” keyword, cumulative statistics for each channel (since it became active to “now”) are rolled up and then displayed.. In the absence of the “cumulative” keyword, “show counters” displays historical/rolled up statistics for each channel since the channel was last reset/initialized.

The fields shown in the output of this command are described in the following table:

Field Name	Type	Description
Stream Packet Counters		
primary rtp inputs	int	primary rtp packets received
primary rtp drops	int	primary rtp packets dropped, due to reasons such as: 1. rtp parse failure 2. packet arrived too early (before join) 3. packet arrived too late for playout 4. drop simulator tool dropped it
primary rtp drops (late)	int	primary rtp packets dropped due to arriving too late (after time needed by output scheduler)
primary rtcp inputs	int	primary rtcp packets received
primary rtcp outputs	int	primary rtcp packets sent
repair/rcc rtp inputs	int	repair/rcc rtp packets received
repair/rcc rtp drops	int	repair/rcc rtp packets dropped, due to reasons such as: 1. rtp parse failure 2. packet arrived too early (before join) 3. packet arrived too late for playout 4. repair preceded first sequence number from RCC APP. 5. drop simulator tool dropped it This counter EXCLUDES drops due to duplicate packets being received.
repair/rcc rtp drops (late)	int	repair/rcc rtp packets dropped due to arriving too late (after time needed by output scheduler)
repair/rcc rtcp inputs	int	repair/rcc rtcp packets received
fec inputs	int	fec packets received
fec drops	int	fec packets dropped, due to reasons such as: 1. invalid rtp header 2. invalid fec header 3. packet arrived too late 4. internal error while processing fec packet (e.g. memory allocation failure)
fec drops (late)	int	fec packets which arrived too late (a primary packet to which it refers has already been scheduled for output)
repair rtp stun inputs	int	STUN packets received on repair rtp port
repair rtp stun outputs	int	STUN packets sent on repair rtp port
repair rtcp stun inputs	int	STUN packets received on repair rtcp port
repair rtcp stun outputs	int	STUN packets sent on repair rtcp port
post repair outputs	int	post repair stream packets (common to all tuners which are tuned to the same channel)
tuner queue drops	int	drops during packet enqueue on

		tuner/sink (e.g. due to queue limit reached)
underruns	int	underruns upon inserting packets of the input streams
Repair Packet Counters		
pre-repair losses	int	number of packets not arriving within the stream. E.g. an arriving packet stream of sequence numbers 1,4,5,7,8 will bump this counter 3 times.
post-repair losses	int	number of packets which were missing (not repaired) upon output to the tuner
post-repair losses rcc	int	number of packets which were missing (not repaired) from within an RCC burst upon output to the tuner. Subset of post-repair losses rcc (above).
repairs requested	int	number of repair packets requested by VQE-C
repairs policed	int	number of repair requests not sent due to rate limiting
fec recovered packets	int	packets successfully regenerated/repaired by fec
Channel Change Counters		
channel change requests	int	number of channel change requests (tuner bind requests) attempted
rcc requests	int	number of channel change requests (tuner bind requests) attempted in which an RCC operation was initiated
rcc with loss	int	number of times an RCC occurred (did not abort) but experienced non-zero loss within the RCC burst of the post-repair stream. i.e., an RCC occurred for which the "post-repair losses rcc" counter incremented and the channel change was not aborted.
rcc aborts total	int	total number of rapid channel changes which aborted, for reasons itemized by counters below.
server reject	int	number of times RCC request was rejected by VQE-S
stun timeout	int	number of times STUN response not received from VQE-S
response timeout	int	number of times APP packet not received from VQE-S
response invalid	int	number of times APP packet received contains invalid contents
burst start	int	number of times burst failed to start (first repair packet not received in time)
burst activity	int	number of times burst activity timed out prior to completion

To show counters for a single tuner which accumulated since its channel last became active (was last bound to any tuner), use the "show counters <tuner>" form of the command.

The fields shown in the output of this command are described in the following table:

Field Name	Type	Description
tuner-name	string	Name of tuner being displayed.
Inputs	int	Total number of packets received.
Drops	int	Total number of packets lost (not received).
Primary	int	Total number of primary packets received.
Repair	int	Total number of repair packets received.
Rtcp	int	Total number of RTCP packets received.
Rtp	int	Total number of RTP packets received.
app timeouts	int	Total number of times that there was an APP packet expected that did not show up before the timeout.
null app	int	Total number of NULL (empty) APP packets received.
Outputs	int	Total number of packets that have been sent out from the output queue.
Output Q drops	int	Total number of packets that have been dropped from the output queue because of overflow.
Fec repairs	int	Total number of repair packets that were recovered by FEC.

To clear counters for either form of the "show counters" command, use the "clear counters" command.

Examples

The following example shows the cumulative counters across all tuners/channels:

```
vqec# sh cou

-- Stream Packet Counters --
                    Inputs        Outputs       Drops (Late)
primary rtp:        2244          -             0 (0)
primary rtcp:       0             1             -
repair/rcc rtp:     2326          -             0 (0)
repair/rcc rtcp:    1             -             -
fec:                0             -             0 (0)
repair rtp stun:    0             0             -
repair rtcp stun:   0             0             -
post-repair:        -             4497          -
tuner Q drops:      -             -             4497
underruns:          0             -             -

-- Repair Packet Counters --
                    Pre-Repair    Post-Repair
stream loss:        0             0
```

```
 rcc loss:          -              0
                    Requested      Policed
error repair:       0              0
                    Recovered
fec:                0

-- Channel Change Counters --
requests:           0
 rcc requests:      0
  rcc with loss:    0
  rcc aborts:       0
    server    stun      response  response  burst     burst
    rejects   timeout   timeout   invalid   start     activity
other
    0         0         0         0         0         0         0

vqec# sh cou cum

-- Stream Packet Counters --
                    Inputs         Outputs        Drops (Late)
primary rtp:        159271         -              0 (0)
primary rtcp:       0              371            -
repair/rcc rtp:     31354          -              0 (0)
repair/rcc rtcp:    27             -              -
fec:                0              -              0 (0)
repair rtp stun:    0              0              -
repair rtcp stun:   0              0              -
post-repair:        -              189617         -
tuner Q drops:      -              -              189217
underruns:          0              -              -

-- Repair Packet Counters --
                    Pre-Repair     Post-Repair
stream loss:        0              0
 rcc loss:          -              0
                    Requested      Policed
error repair:       0              0
                    Recovered
fec:                0

-- Channel Change Counters --
requests:           2
 rcc requests:      2
  rcc with loss:    0
  rcc aborts:       1
    server    stun      response  response  burst     burst
    rejects   timeout   timeout   invalid   start     activity
other
0         0         1         0         0         0         0
```

The following example shows the counters for a tuner named “tuner1”:

```
vqec> show counters tuner1
tuner-name:      tuner1
 inputs:         1466955
 drops:          30
 primary:        1463475
 repair:         3231
 rtcp:           248
 rtp:            1466706
 app timeouts:   0
 null app:       0
 outputs:        1466076
 output Q drops: 0
```

```
fec repairs:    0
```

show debug

To show the current debug flag settings, use the **show debug** command.

show debug

Command Modes EXEC

Usage Guidelines Each debug flag is responsible for a certain set of possible debug messages. When one of these flags is enabled, debug messages related to that component will begin to appear. Each flag can have a value of **enabled** or **disabled**.

Examples The following example lists all the current debug flags:

```
vqec> show debug
channel:           disabled
chan_cfg:          disabled
cpchan:            disabled
error-repair:      disabled
event:             disabled
rcc:               disabled
igmp:              disabled
input:             disabled
output:            disabled
pcm:               disabled
recv-socket:       disabled
rtcp:              disabled
nat :              disabled
timer:             disabled
tuner:             disabled
upcall:            disabled
updater:           disabled
dp-tlm:            disabled
dp-inputshim:      disabled
dp-outputshim:     disabled
dp-nll:            disabled
dp-nll-adjust:     disabled
dp-pcm:            disabled
dp-pcm-pak:        disabled
dp-error-repair:   disabled
dp-rcc:            disabled
dp-fec:            disabled
dp-failover:       disabled
```

show dp

To show current VQE-C dataplane information, use the **show dp** command.

show dp [counters]

Command Modes EXEC

Usage Guidelines Currently **counters** is the only option which will display any information.

Examples The following example shows some example output:

```
vqec# show dp counters
---Dataplane IPC---
 IRQ sent                  173
 IRQ dropped               0
 Ejected packets sent      1
 Ejected packets dropped   0
---Dataplane Channel---
 Creates                   0
 Destroys                  0
 Creation failures         0
---Dataplane RTP---
 Source creates            2
 Source destroys           0
 Source table full         0
 Source limit exceeded     0
 Source aged out           0
 RTP IS creates            2
 RTP IS deletes            0
 RTP IS limit exceeded     0
 RTP IS ejected paks       1
 XR stats malloc failures  0
 SSRC filter drops         0
 Repair stream filter drops 0
```

show drop

To show the current drop simulation settings, use the **show drop** command.

show drop

Command Modes EXEC

Usage Guidelines The fields shown in the output of this command are described in the following table:

Field Name	Type	Description
primary stream dropping *or* repair stream dropping	enabled (string) / disabled	Indicates current drop simulator status and dropped packet selection method for the given stream.
Dropping	int	Number of consecutive packets being dropped at the beginning of each drop interval
interval	int	Length of drop interval.
Percentage	percentage	If dropping randomly selected packets,

		this is the target rate at which to randomly select and drop packets.

Examples

The following example displays VQE-C's drop simulation configuration:

```
vqec> show drop
primary stream dropping:    enabled (using percentage)
 dropping:                  0
 interval:                  0
 percentage:                5%
repair stream dropping:     enabled (using percentage)
 dropping:                  0
 interval:                  0
 percentage                 3%
```

show error-repair

To show the current error repair feature status, use the **show error-repair** command.

show error-repair

Command Modes EXEC

Examples The following example shows the current status of error repair:

```
vqec> show error-repair
error-repair:               enabled
 repair policer:            enabled
  rate:                     5%
  burst:                    10000ms
  packet requests policed:  20
```

The default or configured values for the error-repair policer are shown, along with a `packet requests policed` counter which displays the total number of error repairs policed for all streams tuned by the VQE-C. This counter is reset via the "clear counters" command and during initialization of VQE-C.

show fec

To show the current FEC feature status, use the **show fec** command.

show fec

Command Modes EXEC

Examples The following example shows the current status of FEC:

```
vqec> show fec
fec:                  enabled
```

The default value for the FEC is shown here.

show rcc

To show the current RCC feature status, use the **show rcc** command.

show rcc

Command Modes EXEC

Examples The following example shows the current status of RCC:

```
vqec# show rcc
rcc:                  disabled
```

The default value for RCC is shown here.

show nat

To show the current NAT feature status, use the **show nat** command.

show nat

Command Modes EXEC

Examples The following example shows the current NAT status:

```
vqec# show nat
NAT protocol:               STUN
NAT bindings open:          2
NAT id:                     4043309057
 NAT status:                Not Behind NAT
 Internal address:          5.8.48.2:32797
 Public address:            5.8.48.2:32797
 Last request time:         1214316087525
 Last response time:        1214316087527
NAT id:                     2902458370
 NAT status:                Not Behind NAT
 Internal address:          5.8.48.2:32795
 Public address:            5.8.48.2:32795
 Last request time:         1214316087525
 Last response time:        0
```

show pak-pool

To show the current status of the packet memory pool used by VQE-C, use the **show pak-pool** command.

show pak-pool

Command Modes

EXEC

Usage Guidelines

If you specify an optional tuner name, you will see information for that tuner.

If you specify no optional arguments, you will see information for all active tuners.

The fields shown in the output of this command are described in the following table:

Field Name	Type	Description
max entries	int	Maximum number of packets that can be allocated from the packet memory pool at any one time.
Used entries	int	Number of packets currently allocated in the packet memory pool.
High water entries	int	Record high number of used entries at any given runtime.
Fail pak alloc drops	int	Number of packets dropped due to there not being enough free space in the packet pool.

Examples

The following example shows the current status of the packet memory pool:

```
vqec> show pak-pool
global input pak pool stats:
 max entries:               1000
 used entries:              274
 high water entries:        322
 fail pak alloc drops:      0
```

show proxy-igmp

To show the current status of proxy-igmp, use the **show proxy-igmp** command.

show proxy-igmp [*tuner-name*]

Syntax Description

tuner-name (Optional) Name of an active tuner.

Command Modes

EXEC

Usage Guidelines

If you specify an optional tuner name, you will see information for that tuner.

If you specify no optional arguments, you will see information for all active tuners.

The fields shown in the output of this command are described in the following table:

Field Name	Type	Description
Tuner name	string	Name of the current tuner.
IGMP Proxy State	string	Status of IGMP proxy for the current tuner
VQEC Interface	string	Physical interface to which the STB is connected
STB IP Address	IP	The IP address of the STB interface
Destination URL	protocol://IP:port	The URL to which the tuner streams output to
Packets sent	int	Number of packets that were successfully streamed out.
Packets dropped	int	Number of packets that were not successfully transmitted.

Examples

The following example shows the stream-output status of a tuner with name “0”:

```
vqec> show proxy-igmp
Tuner name:                 tuner1
 IGMP Proxy State:          Enabled
 VQEC Interface:            eth2
 STB IP Address:            192.168.1.130
 destination URL:           rtp://224.1.1.1:50000
 packets sent:              2720
 packets dropped:           0
```

show stream-output

To show the current status of output streaming, use the **show stream-output** command.

show stream-output [*tuner-name*]

Syntax Description

tuner-name (Optional) Name of an active tuner.

Command Modes

EXEC

Usage Guidelines

If you specify an optional tuner name, you will see information for that tuner.

If you specify no optional arguments, you will see information for all active tuners.

The fields shown in the output of this command are described in the following table:

Field Name	Type	Description
Tuner name	string	Name of the current tuner.
Packets sent	int	Number of packets that were successfully streamed out.
Packets dropped	int	Number of packets that were not successfully transmitted.

Examples

The following example shows the stream-output status of a tuner with name "0":

```
vqec# show stream-output 0
Tuner name:                0
 packets sent:             25600
 packets dropped:          0
```

show system-config

To show the current build information and VQE-C configuration settings, use the **show system-config** command.

show system-config [start-up | network | override | defaults]

Syntax Description

start-up	*(Optional)* Shows.contents of system configuration file supplied by integrator.
network	*(Optional)* Shows contents of cached network configuration file, if one exists.
override	*(Optional)* Shows contents of cached override configuration file, if one exists.
defaults	*(Optional)* Shows VQE-C software default values for all configuration parameters
<cr>	Shows the running system configuration currently in use by VQE-C. This configuration is derived from merging override configuration, network configuration, start-up configuration, and VQE-C software defaults, in that order.

Command Modes

EXEC

Usage Guidelines

The fields that are displayed by this command are explained in detail in the VQE-C System Configuration Reference document.

Examples

The following example shows the current VQE-C build information and running configuration:

```
vqec# show system-config
VQE-C 3.0.0 build 101 (development-build)
Built by: ansawchu
Workspace: /auto/wscisco/main/vam/eva
Timestamp: Jun 24 2008 09:58:18
max_tuners = 10;
channel_lineup = "./vqe_channels.cfg";
jitter_buff_size = 200;
repair_trigger_point = 30;
pakpool_size = 2000;
so_rcvbuf = 128000;
strip_rtp = true;
input_ifname = "eth1";
sig_mode = "nat";
nat_binding_refresh_interval = 30;
max_paksize = 1508;
cdi_enable = false;
domain_name_override = "";
libcli_telnet_port = 8183;
output_pakq_limit = 200;
update_window = 60;
app_paks_per_rcc = 5;
error_repair_enable = true;
error_repair_policer =
  {
     enable = false;
```

```
    rate = 5;
    burst = 10000;
  };
log_level = 4;
fec_enable = true;
fec_max_block_size = 100;
rcc_enable = true;
rcc_start_timeout = 120;
num_byes = 2;
bye_delay = 40;
reorder_delay = 20;
vcds_server_ip = "0.0.0.0";
vcds_server_port = 8554;
cli_ifname = "lo";
tuner_list = (
  {
    name = "t";
    url = "rtp://230.151.1.1:10000";
  }
);
rtcp_dscp_value = 24;
src_ip_filter_enable = false;
```

show tech-support

To facilitate an aggregate display of several CLI show commands that are useful for diagnostics, use the "show tech-support" command. This command, in turn, executes the following CLI commands one after the other, and displays the output of each to the CLI console.

- show system-config
- show tuner all detail
- show channel
- show counters
- show dp counters
- show update
- show drop
- show error-repair
- show fec
- show rcc
- show fast-fill
- show nat
- show pak-pool
- show stream-output
- show ipc
- show proxy-igmp

The command could be used by an integrator to get an aggregate view of the VQE-C state that could in turn be useful to verify or diagnose VQE-C integrations.

show tech-support

Command Modes EXEC

Usage Guidelines The fields that are displayed by this command are explained in detail elsewhere in this document.

Examples

The following example shows an edited output of the "show tech-support" command. The command output displays each sub-command in a header, followed by the output of that command.

```
vqec> show tech-support


# Command "show system-config" :
VQE-C 3.5.0 build 101 (development-build)
Built by: kanjoshi
Workspace: /auto/wskanjoshi/libcli/vam/eva
Timestamp: Feb  4 2010 10:44:18
Dataplane operating mode: USER
max_tuners = 3;.
.
.
.
# Command "show proxy-igmp" :
Tuner name:                 tuner1
 IGMP Proxy State:          Disabled
```

show tuner

To show statistics and current configuration for a tuner, use the **show tuner** command.

show tuner {join-delay |
{{all | name *tuner-name*} [brief] [pcm] [fec] [nat]
[counters] [channel] [log] [ipc] [rcc] [detail]}

Syntax Description

join-delay	Prints a histogram of join-delay times for all channel changes (measured across all tuners)
all | name *tuner-name*	Specify tuner name. 'All' specifies all tuners.
brief	(Optional) Show brief statistics for the specified tuner.
pcm	(Optional) Show the pcm statistics.
fec	(Optional) Show the fec statistics.
nat	(Optional) Show the nat statistics.
channel	(Optional) Show the channel statistics.
log	(Optional) Show stats for packet seq-no/timestamps and loss.
ipc	(Optional) Show stats for the ipc interface.
rcc	(Optional) Show stats relating to rapid channel change.
detail	(Optional) Show all statistics for the specified tuner.

Command Modes

EXEC

Usage Guidelines

Displays information associated with tuners.

If 'all' or 'name <*tuner-name*>' are specified, then information for all or the specified tuner(s) will be displayed. The displayed output may be qualified by one or more keyword options which restrict the output to a "brief" display (the default), a "detailed" display, or a display containing a subset of the output based (e.g. fec, log, etc.)..

The fields shown in the output of detail option of the show tuner command are described in the following table:

Field Name	Type	Description
Tuner name	String	Name of the current tuner.
Tuner ID	int	Control-plane ID of the tuner.
DP Tuner ID	int	Data-plane ID of the tuner.
Bound to Channel ID	int	ID of the channel the tuner is currently bound to.
Channel Information		
Original source multicast	address	Muticast address of the current channel
Original source port	port	Port of the current multicast channel.
Dataplane channel ID	int	ID of the channel in the data-plane.
Dataplane graph ID	int	ID of the dataplane graph structure representing the current channel.
Repair RTP ephemeral port	port	Port number of the repair receive port, chosen by the operating system at bind-time.
Primary RTP ID	int	Hex ID for the primary RTP stream.
Repair RTP ID	int	Hex ID for the repair RTP stream.
FEC0 RTP ID	int	Hex ID for the first FEC RTP stream.
FEC1 RTP ID	int	Hex ID for the second FEC RTP stream.
Current Channel Information		
name	string	Name of the current channel.
sdp_handle	string	SDP handle for the current channel.
handle	int	Internal handle for the current channel.
session identifier	string	
version	int	Version of the current channel's information.
configuration data	string	
mode	string	Lookaside/Source – the current operating mode of VQE for this channel.
original source multicast address	address	Address for the primary multicast stream.
Source address for original source stream	address	Address of the sender for the primary multicast stream.
Original source port	port	Port number from the primary multicast sender.

Original source RTCP port	port	Port number from the primary multicast RTCP stream.
Original source RTP payload type		RTP payload type of the primary stream.
Original source RTCP sender bandwidth		
Original source RTCP receiver bandwidth		
Original source RTCP per receiver bandwidth		
Original source RTCP XR Loss RLE Report	boolean	Indicates whether RTCP Extended Reporting is enabled or disabled.
Original source RTCP XR Stat Summary Report	int	Indicates the options for RTCP Extended Report summaries.
RTCP XR Post Repair Loss RLE Report	boolean	Indicates whether or not RTCP Extended Reporting is enabled for post-repair data or not.
Maximum bit rate	int	Highest bitrate of the current stream.
Retransmission/FBT address	address	Address of the VQE-S servicing this channel.
Retransmission RTP port	port	Port on the VQE-S to receive repair packets from.
Retransmission RTCP port	port	Port on the VQE-S to send feedback information to.
Retransmission associated payload type		RTP payload type of the retransmission stream.
Repair stream RTCP sender bandwidth		
Repair stream RTCP receiver bandwidth		
Error repair	boolean	Indicates whether or not error repair services are enabled.
Rapid channel change	boolean	Indicates whether or not fast channel change (RCC) services are enabled.
Active channel cfg updates	Int	Number of times an active channel's configuration was updated since the channel was tuned (e.g. the channel was configured with a new source due to failover)
Active channel's last cfg update	Int	Absolute time (in milliseconds) at which the active channel's configuration was last

		updated. If the active channel's configuration has not changed (since it became active), this value will be zero.
Error repair policer	boolean	Indicated whether or not the error repair policer is enabled.
Primary data received	boolean	Indicates if primary data has been received on this channel.
RTP Primary Session		
ssrc	int	RTP SSRC.
cname	string	RTP CNAME.
nmembers	int	Number of session members.
nsenders	int	Number of session senders.
RTCP compound packet stats		
sent	int	RTCP packets sent for the session.
send_errors	int	RTCP sends that failed.
rcvd	int	RTCP packets received for the session.
rcvd_errors	int	RTCP packets that failed receive processing.
badver	int	Number of primary RTCP packets dropped due to a bad RTP version number.
runts	int	Number of primary RTCP packets dropped due to the message at the end of a compound packet being either shorter than the minimum length of a message (4 bytes), or shorter than the length indicated in the header.
badlen	int	Number of primary RTCP packets dropped due to a length that is invalid for the given message type.
unexp	int	'Unexpected RTCP message type': RTCP message types that are valid but are not expected by the application.
avg_pkt_size	int	RTCP received average packet size.
avg_pkt_size_sent	int	RTCP transmitted average packet size.
Sender info for sender 1[1]		
ssrc	int	RTP SSRC

[1] From RFC 3550

cname	string	RTP CNAME
received	int	RTP packets received
cum_loss	int	RTP cumulative number of packets lost
ehsr	int	RTP extended highest sequence number
jitter	int	RTP interarrival jitter
Sender stats for sender 1		
max_seq	uint16	Highest sequence number seen
cycles	int	Shifted count of sequence number cycles
bad_seq	int	Last 'bad' sequence number + 1
base_seq	int	Base sequence number
transit	int	Relative transit time for previous packet
received	int	RTP packets received
last_arr_ts_media	int	Last packet arrival timestamp in RTP timestamp units
seqjumps	int	Packets received with 'large' sequence number jumps in comparison to max_seq.
initseq_count	int	The number of times a base sequence number has been established or 're-established' for the seding source.
out_of_order	int	Packets recevied with sequence numbers that were in the 'immediate' past of the max_seq.
Primary RTCP RRs sent	int	Number of RTCP receiver reports that have been sent for this primary session.
DP sources	int	Number of sources in the dataplane for this RTP session.
Source info for source 1		
ssrc, src ip, port	string	Identifies a DP source (see above).
source status	string	Indicates the status of a DP source. "active"/"inactive" indicates whether packets are currently being received from the source or not. "pktflow" indicates that the source's packets are being accepted into the PCM.
source seq num offset	int	Indicates the signed 16-bit

		offset by which this DP source's sequence numbers may be recovered from PCM sequence numbers (vqec_seq_num_t). E.g., a value of 0 implies that the source's RTP sequence number may be recovered by subtracting 0 from the PCM sequence number and taking the lower 16-bits the result.
Repair data received	boolean	Indicates whether or not repair packets have been received from the repair stream.
RTP Repair Session		
(same field and definitions as above, but for the repair RTP session instead of the primary RTP session)		
Dataplane channel stats		
total recvd paks	int	Total packets received by the RTP module.
total recvd primary paks	int	Total primary packets received by the RTP module.
total recvd repair paks	int	Total repair packets received by the RTP module.
total recvd rtp paks	int	Total RTP packets received by the RTP module.
total rtp drops	int	Total RTP packets that were dropped by the RTP module.
total sim drops	int	Total packets that were intentionally thrown away by the RTP module drop simulation.
total early drops	int	Total packets that were dropped due to early arrival.
failover source	string	Identifies an alternate source whose most recent packets are being queued for use should the primary (packetflow) source disappear. Only applicable only for unicast channels which are receiving traffic from an alternate source.
num paks queued	int	Number of packets queued from an alternate source (see "failover

		source" above).
queue head RTP seq num	int	RTP sequence number of packet at the head of the failover queue (see "failover source" above).
queue tail RTP seq num	int	RTP sequence number of packet at the tail of the failover queue (see "failover source" above).
prev src last rcv time	Int	Absolute time (in milliseconds) at which the last packet from the channel's previous source was received. (Only displayed if the channel has experienced a source change.) See next field.
curr src first rcv time	Int	Absolute time (in milliseconds) at which the first packet from the channel's current source was received. (Only displayed if the channel has experienced a source change.) When compared to the "prev src last rcv time" field above, the difference can indicate the amount of overlap or gap between previous and current sources during the last source failover event.
runts	int	Number of primary RTCP packets dropped due to the message at the end of a compound packet being either shorter than the minimum length of a message (4 bytes), or shorter than the length indicated in the header.
badver	int	Number of primary RTCP packets dropped due to a bad RTP version number.
badlen	int	Number of primary RTCP packets dropped due to a length that is invalid for the given message type.
badcreate	int	Number of primary packets dropped because lookup or create RTP member failed.
seqjumps	int	Packets received with

		'large' sequence number jumps in comparison to max_seq.
initseq_count	int	The number of times a base sequence number has been established or 're-established' for the seding source.
runts	int	Number of primary RTCP packets dropped due to the message at the end of a compound packet being either shorter than the minimum length of a message (4 bytes), or shorter than the length indicated in the header.
badver	int	Number of primary RTCP packets dropped due to a bad RTP version number.
badlen	int	Number of primary RTCP packets dropped due to a length that is invalid for the given message type.
badcreate	int	Number of primary packets dropped because lookup or create RTP member failed.
seqjumps	int	Packets received with 'large' sequence number jumps in comparison to max_seq.
initseq_count	int	The number of times a base sequence number has been established or 're-established' for the seding source.
total recvd rtcp paks	int	Total RTCP packets receieved by the RTP module.
generic nack counter	int	Number of generic NACK packets sent.
total repairs requested	int	Number of packets requested as repairs (total contents of generic NACKs)
total repairs policed	int	Number of packets not requested as repairs due to error-repair policing.
total repairs unrequested	int	Number of packets not requested as repairs because either: (1) the repair requests did not fit into a single RTCP packet (the internal limit

		defined for the number of Feedback Control Information bitmaps was exceeded), or (2) the VQE-C was unable to transmit the RTCP packet containing the constructed repair requests
failed to send rtcp pak	int	
failed to report gap	int	
PCM status		
Head	int	Sequence number of the packet at the front of the PCM.
Tail	int	Sequence number of the packet at the end of the PCM.
highest_er_seq_num	int	Sequence number of the newest packet that is eligible for retransmission error repair.
last_reqstd_er_seq_num	int	Sequence number most recently requested for retransmission error repair.
last_rx_seq_num	int	Last received sequence number.
num_paks_in_pak_seq	int	The current number of packets contained in the PCM.
Primary_received	boolean	Indicates whether or not primary stream packets have been received.
Repair_received	boolean	Indicates whether or not repair stream packets have been received.
Repair_trigger_time	int	The time interval (in ms) that specifies how often the gap reporter should be triggered to report loss in the PCM.
Reorder_delay	int	The time (in ms) that loss created by out-of-order arriving packets should be held before they are eligible for retransmission-based error-repair.
fec_delay	int	Time (in ms) that lost packets should be held in order for FEC to have an adequate opportunity to repair them.
Original jitter buff size	int	Duration (in ms) of the

		jitter buffer, as specified by the configuration.
Total delay including fec	int	Duration (in ms) of the total delay within the PCM, including jitter buffer, reorder_delay, and delays from FEC.
gap_hold_time	int	Total duration (in ms) that lost packets will be held in the PCM before allowing retransmission to try to repair them.
PCM Counters		
late packets	int	Number of packets that arrived late.
head ge last seq reqstd	int	Number of times that the sequence number of the PCM's head became greater than or equal to the sequence number that was most recently requested for retransmission error repair.
Primary packets	int	Number of primary packets received.
Repair packets	int	Number of repair packets received.
Input loss packets	int	Number of missing packets detected on the input (received) stream.
Input loss holes	int	Number of missing packet holes (one or more contiguous missing packets) on the input stream.
Output loss packets	int	Number of missing packets on the ouput stream.
Output loss holes	int	Number of missing packet holes on the output stream.
Pcm_insert drops	int	Number of packets that were dropped due to a PCM insertion failure.
Duplicate packets	int	Number of duplicate packets received in input stream.
Pak_seq_insert fail paks	int	Number of packets that were dropped due to a packet sequence module insertion failure.
Bad seq range	int	Number of packets that tried to be inserted into

		PCM, but whose seq_num was far outside of the range of the PCM head and tail.
Under runs	int	Number of times the PCM has run out of packets to send.
Output early packets	int	Packets that were scheduled to be played out earlier than they should have been.
Bad receive timestamp	int	Number of packets with invalid timestamps.
duplicate repair packets	int	Number of repair packets that were received as duplicates.
Last tx seq num	int	Sequence number of last packet transmitted from PCM to sink.
Last tx time	int	Time of last packet transmitted from PCM to sink.
Total tx packets	int	Total packets transmitted from PCM to sink.
Total tx bursts	int	Total bursts transmitted from PCM to sink. PCM determines one burst as a series of more than 10 packets that are transmitted in <1 ms.
NLL state		
mode	int	Mode of the NLL, non-tracking (0) or tracking (1).
pred base	uint64	Predicted absolute time for the last packet scheduled.
last act time	uint64	Actual receive time for the last packet scheduled.
primary offset	uint64	Time offset between the repair & primary streams for RCC.
pcr32 base	int	The RTP timestamp for the last packet scheduled.
exp disc	int	Explicit timestamp discontinuities input to the NLL.
imp disc	int	Implicit timestamp discontinuities declared by the NLL.
observations	int	Number of packets processed.
resets	int	Number of times the NLL has been reset.
past predicts	int	Number of times the NLL

		has predicted send times for a packet in the past.
reset_base w/o time	int	NLL resets using the current time as the receive time for the observation.
FEC status		
fec enabled in channel	boolean	Show if FEC is enabled. True if enable, false, otherwise
fec streams	int	Number of FEC streams available at current session. Show as %d_D
fec_column_stream_avail	boolean	True if column FEC stream is received from network; false, otherwise
fec_row_stream_avail	boolean	True if row FEC stream is received from network; false, otherwise
L value	uint8	Number of columns in one FEC encoding block
D value	uint8	Number of rows in one FEC encoding block
column head	vqec_seq_num_t	The smallest sequence number in column FEC buffer
column tail	vqec_seq_num_t	The largest sequence number in column FEC buffer
row head	vqec_seq_num_t	The smallest sequence number in row FEC buffer
row tail	vqec_seq_num_t	The largest sequence number in row FEC buffer
FEC counters		
late fec packets	uint64	The late FEC packet counter. A FEC packet is late if it protected primary video packet are sent to display
fec recovered packets	uint64	Number of packets recovered by FEC
no need to decode paks	uint64	Number of FEC packets that do not need to be decoded. If there is no loss of the primary packets protected by this FEC packet, then, this FEC packet is not decoded.
total fec packets	uint64	Total FEC packets received
duplicate fec packets		Number of duplicate FEC packets received. A packet is considered duplicate if it is received

		more than once.
rtp hdr invalid paks	uint64	Number of FEC packets with invalid RTP header.
fec hdr invalid paks	uint64	Number of FEC packets with invalid FEC header
fec paks, unrecoverable	uint64	Number of valid FEC packets that can not be used to recover lost packets. If there are more than one packet losses of the primary packets that are protected by one FEC packet, this FEC packet is called unrecoverable.
fec paks, other	uint64	Number of FEC packets being dropped due to internal errors, such as memory alloc failure, etc
fec gaps detected	uint64	Number of lost FEC packets according to RTP sequence number gap.
RCC status		
rcc enabled	boolean	true if enabled, otherwise false
rcc result	string	Rapid channel change results as defined below: “success”: rapid channel change succeeds, all the processes are finished in designed order. “on-going”: rapid channel change is in process. “failure”: rapid channel change is aborted.
cp failure reason	string	The reason the rapid channel change was aborted at control plane. NONE: no error, rapid channel change state machine completed in control plane. NAT_TIMEOUT: NAT server does not have any response within the rcc_start_timeout. (This

		only happens when sig_mod is in NAT mode. APP_TIMEOUT: APP packet was not received within the rcc_start_timeout.
dp failure reason	string	NULL_APP: null APP from VQE-S. The server could not process this rapid channel change. Check the server statistics for information regarding this NULL APP. INVALID_APP: the received APP packet is not valid, i.e one or more fields in the APP packet are not properly set from the VQE-S. RCC_DISABLED: rapid channel change is not enabled, either in VQE-C system configuration setup or in channel lineup. UNKNOWN: none of the above reason was detected, but the rapid channel change was aborted.
Buffer fill (ms)		
minimum buffer fill	rel_time_t	Minimum buffer fill requirement in ms, calculated based on FEC and ER requirement. This is sent from client to server.
maximum buffer fill	rel_time_t	Maximum buffer fill limit in ms, calculated based on the pak-pool size and max_rcc_backfill_scaler. This is sent from client to server.
buffer fill from APP	rel_time_t	Expected buffer fill from server. This is the value from APP packet.
APP expected relative times (ms)		
Join	rel_time_t	Earliest multicast join time
ER	rel_time_t	Expected ER turn on time
End-Of-Burst	rel_time_t	Expected burst end time

Actual relative times (ms)		
CC	rel_time_t	The time the channel change was issued, measured at the time the tuner bind channel. This is the starting point for the calculation of all these relative times below
Pli	rel_time_t	The time the PLI packet was sent to VQE-S
APP	rel_time_t	The time the APP packet was received.
Rep	rel_time_t	The time the first repair packet was received.
Join	rel_time_t	The time the multicast join was issued.
Prim	rel_time_t	The time the primary packet was received.
ER	rel_time_t	The time the error-repair was turned on.
Join-lat	rel_time_t	The IGMP join latency. This relative time is the time difference between the multicast join time and the first primary packet received time.
Pcm snapshots		
JOIN		List the PCM snapshot at multicast join, including Head: head of the PCM (smallest sequence number at PCM). Tail: tail of the PCM (largest sequence number at PCM). Paks: number of packets cached in PCM.
PRIM		List the PCM snapshots at the time when the first primary packet was received, including Head: head of the PCM (smallest sequence number at PCM) Tail: tail of the PCM (largest sequence number at PCM) Paks: number of packets cached in PCM.
EREN		List the PCM snapshots at the time when error-repair was enabled, including

		Head: head of the PCM (smallest sequence number at PCM) Tail: tail of the PCM (largest sequence number at PCM) Paks: number of packets cached in PCM.
Output statistics		
first primary sequence	vqec_seq_num_t	Sequence number of the first primary packet received.
rcc output loss packets	uint32_t	Total number of lost packets seen at output scheduler during rapid channel change period.
rcc output loss holes	uint32_t	Total number of holes seen at output scheduler during rapid channel change period.
rcc duplicate packets	uint32_t	Total number of duplicated packets received during rapid channel change period
repairs in 1st nack	uint32_t	Total number of repair packets requested in the first NACK after rapid channel change.
first packet output time	rel_time_t	The relative time the first packet of the RCC repair burst was scheduled out to output queue.
last packet output time	rel_time_t	The relative time the last RCC packet was scheduled out to output queue.
first packet decode time	rel_time_t	The relative time the first packet was decoded at STB when changing to a new channel.
NAT Status	string	Indicates whether or not the VQE-C is behind a NAT device (repair RTP port).
Internal address	IP:port	Internal address used for the RTP repairs
Public address	IP:port	External/Public address used for the RTP repairs
Last request time	int	Time of last NAT update request (RTP).
Last response time	int	Time of last NAT update response (RTP).
NAT Status	string	Indicates whether or not the VQE-C is behind a NAT device (repair RTCP

		port).
Internal address	IP:port	Internal address used for the RTCP repairs
Public address	IP:port	External/Public address used for the RTCP repairs
Last request time	int	Time of last NAT update request (RTCP).
Last response time	int	Time of last NAT update response (RTCP).
Logs		
last 10 repair seq	vqec_seq_num_t	List of the last 10 repair sequence numbers
first 10 primary seq	vqec_seq_num_t	List of first 10 primary packet sequence numbers
last 10 fec repair seq	vqec_seq_num_t	List last 10 FEC corrected primary packets
Input Loss (holes)	vqec_seq_num_t	List of the last 10 input loss (holes) sequence number intervals
Output loss (holes)	vqec_seq_num_t	List of the last 10 output loss (holes) sequence number intervals
primary stream dropping *or* repair stream dropping	string	Indicates whether or not the VQE-C drop simulation is enabled.
dropping	int	Number of contiguous packets that are dropped at the start of each interval.
interval	int	Length of the drop interval, when dropping contiguous packets.
percentage	int (%)	When in random drop mode, approximate randomized percentage of packets that are being dropped.
Information about output streams (from DP input shim; one for each output stream)		
Output Stream *(with hexadecimal ID specified)*		
Encaps type	string	Encapsulations supported by this output stream.
Capabilities	string	Capabilities supported by this output stream.
Filter (scheduling class *x*)	title/int	*x* represents the scheduling class to which the current filter is associated.
protocol	string	Protocol that the filter is bound to.
source IP	address	IP address to accept incoming packets from.
source port	int	Port number to accept incoming packets from.

dest IP	address	IP address that the filter is bound to.
dest port	int	Port number that the filter is bound to.
Connected Input Stream ID	int	ID of the input stream to which this output stream is connected.
Stats		
packets transmitted	int	Total number of packets that have been transmitted over the stream.
bytes transmitted	int	Total number of bytes that have been transmitted over the stream.
packets dropped	int	Total number of packets that failed to be transmitted.
Output shim status		
state	string	Current state of the output shim module.
is_creates	int	Number of input streams created.
is_destroys	int	Number of output streams created.
num_tuners	int	Number of DP tuners created.
Tuner status (one for each tuner)		
Cp_tid	int	CP ID of this tuner.
qid	int	ID of the output queue.
isid	int	ID of the associated input stream.
qinputs	int	Number of packets input into the output queue.
qdrops	int	Number of packets dropped in the output queue.
qdepth	int	Number of packets currently in the output queue.
qoutputs	int	Number of packets output from the output queue.
Output shim input streams (one for each input stream)		
stream id	int	ID of this stream.
capabilities	string	Stream transmission capabilities.
encapsulation	string	Encapsulation of this stream.
mapped TunIDs	multiple ints	IDs of all tuners mapped to this input stream.
connected os	int	ID of the connected output stream.
packets	int	Number of packets on the input stream.
bytes	int	Number of bytes on the input stream.

drops	int	Number of packet drops on the input stream.

If "join-delay" is specified, then a histogram will be displayed whose data points are the intervals (in milliseconds) between

a) the request of the channel (which, for a multicast channel, aligns with the join request), and
b) the arrival of the first primary stream packet

A single "join-delay" histogram is maintained across all defined tuners. Data points in the histogram are cleared only via the "clear counters" command.

Examples

The following example shows the basic statistics and information of a tuner with name "tuner1":

```
vqec# show tuner name tuner1
Tuner name:                 tuner1
Channel information for channel 0x56000001
 Original source multicast: 230.151.1.1
 Original source port:      10000
 Dataplane channel ID:      a5000001
 Dataplane graph ID:        ba000001
 Repair RTP ephemeral port: 32795
 Primary RTP ID:            a0000000
 Repair RTP ID:             a0000001
 FEC0 RTP ID:               0
 FEC1 RTP ID:               0
Channel name: Channel 230.151.1.1
Source multicast address: 230.151.1.1
Source port: 10000
Retransmission/FBT address: 8.36.1.1
Maximum bit rate: 3750000
PCM counters:
 input primary packets:     4677652
 input loss packets:        0
 input loss holes:          0
 repair received:           3206
 output loss packets:       0
 output loss holes:         0
 under runs:                0
 late packets:              1
FEC status:
 fec enabled in channel:    false
FEC counters:
 input fec packets:         0
 fec recovered packets:     0
 fec paks, unrecoverable:   0
 late fec packets:          0
 fec paks, other:           0
--- RCC status ---
 rcc enabled:               true
 rcc result:                success
 cp failure reason:         NONE
 dp failure reason:         NONE
```

The following example shows all statistics and configuration settings of a tuner with name "tuner1":

```
vqec> show tuner all detail
Tuner name:                 tuner1
  Tuner ID:                 0
```

```
  DP Tuner ID:              1
  Bound to Channel ID:      0x56000
Channel information for channel 0x56000001
 Original source multicast: 230.151.1.1
 Original source port:      10000
 Dataplane channel ID:      a5000001
 Dataplane graph ID:        ba000001
 Repair RTP ephemeral port: 34965
 Primary RTP ID:            a0000000
 Repair RTP ID:             a0000001
 FEC0 RTP ID:               0
 FEC1 RTP ID:               0
Current channel information:
Channel name: Channel 230.151.1.1
Channel sdp_handle: o=- 1209133068 1209133067 IN IP4 venus-iptv
Channel handle: 0x89000001
Channel session identifier: INIP4#-#1209133068#venus-iptv
Channel version: 1209133067
Configuration data: complete
Channel mode: lookaside
Original source multicast address: 230.151.1.1
Source address for original source stream: 5.8.37.2
Original source port: 10000
Original source RTCP port: 10001
Original source RTP payload type: 33
Original source RTCP sender bandwidth: 46875
Original source RTCP receiver bandwidth: 140625
Original source RTCP per receiver bandwidth: 37
Original source RTCP XR Loss RLE Report: Off
Original source RTCP XR Stat Summary Report: 0x0000
RTCP XR Post Repair Loss RLE Report: Off
Maximum bit rate: 3750000
Retransmission/FBT address: 8.36.1.1
Retransmission RTP port: 10002
Retransmission RTCP port: 10003
Retransmission associated payload type: 33
Repair stream RTCP sender bandwidth: 37
Repair stream RTCP receiver bandwidth: 37
Error repair: enabled
Fast channel change: enabled
Active channel cfg updates: 1
Active channel's last cfg update time: 1275677891921
Error repair policer:           disabled
Primary data received:     TRUE
RTP Primary session
  ssrc:                     1f33c921
  cname:                    00:0e:0c:c6:f3:20
  nmembers:                 2
  nsenders:                 1
 RTCP compound packet stats
  sent:                     3
  send_errors:              0
  rcvd:                     0
  rcvd_errors:              0
  badver:                   0
  runts:                    0
  badlen:                   0
  unexp:                    0
  avg_pkt_size:             103.223633
  avg_pkt_size_sent         103.223633
 Sender info for sender 1
  ssrc:                     e91c5214
  cname:
```

```
  received:                 6474
  cum_loss:                 0
  ehsr:                     44203
  jitter:                   2
 Sender stats for sender 1
  max_seq:                  44203
  cycles:                   0
  bad_seq:                  65537
  base_seq:                 37730
  transit:                  1661617379
  received:                 6474
  last_arr_ts_media:        1531668
  seqjumps:                 0
  initseq_count:            1
  out_of_order:             0
Primary RTCP RRs sent:      3
 DP sources:                1
  ssrc=e91c5214, src ip: 5.8.37.2, port:49152
     source status:         (active, pktflow)
     source seq num offset: 0
Repair  data received:     TRUE
RTP Repair  session
  ssrc:                     1f33c921
  cname:                    00:0e:0c:c6:f3:20
  nmembers:                 2
  nsenders:                 1
 RTCP compound packet stats
  sent:                     1
  send_errors:              0
  rcvd:                     1
  rcvd_errors:              0
  badver:                   0
  runts:                    0
  badlen:                   0
  unexp:                    0
  avg_pkt_size:             107.281250
  avg_pkt_size_sent         100.250000
 Sender info for sender 1
  ssrc:                     e91c5214
  cname:                    vqe-dev-71
  received:                 2206
  cum_loss:                 0
  ehsr:                     35473
  jitter:                   945
 Sender stats for sender 1
  max_seq:                  35473
  cycles:                   0
  bad_seq:                  65537
  base_seq:                 33268
  transit:                  1662133060
  received:                 2206
  last_arr_ts_media:        521131
  seqjumps:                 0
  initseq_count:            1
  out_of_order:             0
Repair  RTCP RRs sent:      1
 DP sources:                1
  ssrc=e91c5214, src ip:8.36.1.1, port:10002
     source status:         (active, pktflow)
     source seq num offset: 0

--- Dataplane channel stats ---
 total recvd paks:          12021
```

```
 total recvd primary paks:  9814
 total recvd repair paks:   2206
 total recvd rtp paks:      12020
 total rtp drops:           0
 total sim drops:           0
 total early drops:         0
 runts:                     0
 badver:                    0
 badlen:                    0
 badcreate:                 0
 seqjumps:                  0
 initseq_count:             1
 runts:                     0
 badver:                    0
 badlen:                    0
 badcreate:                 0
 seqjumps:                  0
 initseq_count:             1
 total recvd rtcp paks:     1
 generic nack counter:      0
 total repairs requested:   0
 total repairs policed:     0
 total repairs unrequested: 0
 failed to send rtcp pak:   0
 failed to report gap:      0

PCM status:
 head:                      4294948904
 tail:                      4294949303
 highest_er_seq_num:        4294949288
 last_reqstd_er_seq_num:    4294949280
 last_rx_seq_num:           4294949303
 num_paks_in_pak_seq:       400
 primary_received:          true
 repair_received:           true
 repair_trigger_time:       59
 reorder_delay:             40
 fec_delay:                 0
 original jitter buff size: 200
 total delay including fec: 0
 gap_hold_time:             40
PCM counters:
 late packets:              1
 head ge last seq reqstd:   1
 primary packets:           9814
 repair packets:            2180
 input loss packets:        0
 input loss holes:          0
 output loss packets:       0
 output loss holes:         0
 pcm_insert drops:          31
 duplicate packets:         30
 pak_seq_insert fail paks:  0
 bad seq range:             0
 under runs:                0
 output early packets:      0
 bad receive timestamp:     0
 duplicate repair packets:  0
 last tx seq num:           4294948903
 last tx time:              1213381889988602
 total tx packets:          11593
 total tx bursts:           0
```

```
NLL state:
 mode:                      1
 pred base                  1213381889988
 last act time              1213381889988
 primary offset             1054653
 pcr32 base                 2635578012
 exp disc                   2
 imp disc                   0
 observations               11589
 resets                     0
 past predicts              0
 reset_base w/o time        0

FEC status:
 fec enabled in channel:    false
 fec streams:               0_D
 fec_column_stream_avail:   false
 fec_row_stream_avail:      false
 L value:                   0
 D value:                   0
 column head:               0
 column tail:               0
 row head:                  0
 row tail:                  0

FEC counters:
 late fec packets:          0
 fec recovered packets:     0
 no need to decode paks:    0
 total fec packets:         0
 duplicate fec packets:     0
 rtp hdr invalid paks:      0
 fec hdr invalid paks:      0
 fec paks, unrecoverable:   0
 fec paks, other:           0
 fec gaps detected:         0

--- RCC status ---
 rcc enabled:               true
 rcc result:                success
 cp failure reason:         NONE
 dp failure reason:         NONE
--- Buffer Fill (ms) ---
 minimum buffer fill:       200
 maximum buffer fill:       5614
 buffer fill from APP:      1054
--- APP expected relative times (ms) ---
 Join      ER       End-Of-Burst
 4640      1154     5794
---Actual relative times (ms)---
 CC       Pli      APP      Rep      Join     Prim     ER
Join-lat
 0        1        2        20       4658     4701     5810     22
--- Pcm snapshots ---
       Head            Tail            Paks
 JOIN  4294939073      4294939424      352
 PRIM  4294939097      4294939433      337
 EREN  4294939516      4294939919      404
--- Output statistics ---
 first primary sequence:    4294939490
 rcc output loss packets:   0
 rcc output loss holes:     0
 rcc duplicate packets:     30
```

```
 repairs in 1st nack:       0
 first packet output time   41
 last packet output time    5738
 first packet decode time   0
 NAT status:                Not Behind NAT
 Internal address:          5.8.48.2:34965
 Public address:            5.8.48.2:34965
 Last request time:         1213381860532
 Last response time:        0
 NAT status:                Not Behind NAT
 Internal address:          5.8.48.2:34967
 Public address:            5.8.48.2:34967
 Last request time:         1213381860532
 Last response time:        1213381860533

last 10 repair seq:
 4294939480  4294939481  4294939482  4294939483  4294939484
4294939485  4294939486  4294939487  4294939488  4294939489
first 10 primary seq:
 4294939490  4294939491  4294939492  4294939493  4294939494
4294939495  4294939496  4294939497  4294939498  4294939499
last 10 fec repair seq:
 0  0  0  0  0  0  0  0  0  0
Input loss (holes):
 0 - 0 time 0,  0 - 0 time 0,  0 - 0 time 0,  0 - 0 time 0,  0 - 0
time 0,  0 - 0 time 0,  0 - 0 time 0,  0 - 0 time 0,  0 - 0 time
0,  0 - 0 time 0,
Output loss (holes):
 0 - 0 time 0,  0 - 0 time 0,  0 - 0 time 0,  0 - 0 time 0,  0 - 0
time 0,  0 - 0 time 0,  0 - 0 time 0,  0 - 0 time 0,  0 - 0 time
0,  0 - 0 time 0,

primary stream dropping:    disabled
 dropping:                  0
 interval:                  0
 percentage:                0%
repair stream dropping:     disabled
 dropping:                  0
 interval:                  0
 percentage:                0%
Information about 2 output streams:
Output Stream [ID '0x9a000002']
 Encaps type:                 RTP
 Capabilities:                PUSH PUSH_VECTORED PUSH_POLL
 Filter (scheduling class 0):
  protocol:                   UDP
  source IP:                  8.36.1.1
  source port:                10002
  dest IP:                    5.8.48.2
  dest port:                  34965
 Connected Input Stream ID:   2684354561
 Stats:
  packets transmitted:        2207
  bytes transmitted           2934032
  packets dropped:            0

Output Stream [ID '0x03000001']
 Encaps type:                 RTP
 Capabilities:                PUSH PUSH_VECTORED PUSH_POLL
 Filter (scheduling class 0):
  protocol:                   UDP
  source IP:                  <any>
  source port:                <any>
```

```
  dest IP:                    230.151.1.1
  dest port:                  10000
 Connected Input Stream ID:   2684354560
 Stats:
  packets transmitted:        9814
  bytes transmitted           13032992
  packets dropped:            0

Output shim status:
 state:                      OPERATIONAL
 is_creates:                 1
 is_destroys:                0
 num_tuners:                 1
Tuner status (tunerid = 1):
 cp_tid:         0
 qid:            0
 isid:           1
 qinputs:        11593
 qdrops:         0
 qdepth:         0
 qoutputs:       11593
Output shim input streams:
 stream id:      1
  capabilities:  PUSH PUSH_VECTORED
  encapsulation: UDP
  mapped TunIDs: 1
  connected os:  1
  packets:       0
  bytes:         0
  drops:         0
```

The following example shows the join-delay intervals experienced across all tuners. It indicates that there have been a total 6 channel changes since initialization or last issue of the "clear counters" command, each of which fall within the join-delay bucket ranges listed on the left:

```
vqec# show tuner join-delay
Histogram of Join to First Primary Pkt Delay (in ms):
          0 -          9    [          1  ]
         10 -         19    [          1  ]
         20 -         29    [          2  ]
```

show update

To show information about network and channel configuration updates that have been attempted in the past and scheduled for the future, use the **show update** command.

show update

Command Modes

EXEC

Usage Guidelines

The fields shown in the output of this command are described in the following table:

Field Name	Description
Updater state	State of the updater service.
identity	Unique identifier (CNAME) used by VQE-C when requesting configuration files. The VCDS may supply a customized configuration based on this identifier.
update window	Period (in seconds) over which VQE-C may defer a background update by a random amount, as defined by the **update_window** configuration parameter.
polling	TRUE if polling is enabled, FALSE otherwise
poll interval	Amount of time (in seconds) to wait between background updates, excluding the randomized "update window" delay component. Based on the **update_interval_max** configuration parameter.
Next Update Request:	Approximate time of next scheduled update request, if polling is enabled.
Last Update Request	Time of last update request
Servers attempted/eligible	Number of VQE-C Configuration Delivery Servers which VQE-C attempted to contact during its last configuration update attempt. ("attempted"). Number of VQE-C Configuration Delivery Servers which VQE-C learned from DNS as being eligible/configured in the network to handle update requests during its last configuration update attempt ("eligible").
VCDS Selected for request	IP address of VCDS used for the last configuration update attempt.
VCDS version	Version string within RTSP DESCRIBE response message supplied by the server.
Index file retrieval	Results of the VQE-C's last attempt to retrieve an index file containing version identifiers of the network and channel configuration files for this STB.
Network config update result	Result of trying to retrieve a network configuration file from the VCDS during the last update attempt.

Channel config update result	Result of trying to retrieve a channel configuration file from the VCDS during the last update attempt.
Last successful update transfer times:	
Network config update response time	Time elapsed between issue of a request for an updated network configuration file and its complete arrival.
Channel config update response time	Time elapsed between issue of a request for an updated channel configuration file and its complete arrival.
Updater counters:	
Index retrieval attempts (failures).	Total number of updates attempted by VQE-C (and number of update attempts which were unsuccessful due an index file being unavailable).
Network Config update attempts (failures)	Number of times VQE-C determined that its network configuration differed from that offered by a VCDS, and requested an update to the file (and number of such updates which failed).
Channel Config update attempts (failures)	Number of times VQE-C determined that its channel configuration differed from that offered by a VCDS, and requested an update to the file (and number of such updates which failed).

Note: **cdi_enable** must be configured as TRUE for the updates to be performed and this display to be available.

Examples

The following example shows that the VQE-C performed a successful update with VCDS whose IP address is 138.5.3.1 on January 21 at 16:02. Only the channel configuration was retrieved, as the network configuration file cached by VQE-C was current.

```
vqec# show up

Updater state:                                  running
 identity:                                      00-14-5e-80-6a-4a
 update window:                                 30
 polling:                                       enabled
 poll interval (s):                             3600
Next update request:                            Jan 21 17:03:10
Last update request:                            Jan 21 16:02:11
 Servers attempted/eligible:                    1/1
 VCDS selected for request:                     138.5.3.1:8554
 VCDS version:                                  vcds 3.2.1
 Index file retrieval:                          success
 Network Config update result:                  update not necessary
 Channel Config update result:                  success
Last successful update transfer times:
 Network Config update response time (s):       n/a
 Channel Config update response time (s):       0.151819
Updater counters:
 Index retrieval attempts (failures):           1 (0)
 Network Config update attempts (failures):     0 (0)
 Channel Config update attempts (failures):     1 (0)
```

show ipc

To show current VQE-C control/data plane IPC information, use the **show ipc** command.

show ipc

Command Modes EXEC

Examples The following example shows some example output:

```
vqec# show ipc
---Dataplane IPC---
 IRQ sent                   174
 IRQ dropped                0
 Ejected packets sent       1
 Ejected packets dropped    0
--- IRQ events ---
 Sock name:                 /tmp/.vqec_irqsk9847
 Total Events:              174
 Lost Events:               0
 Error Events:              0
 Ack Errors:                0

--- NAT events ---
 Sock name:                 /tmp/.vqec_paksk9847
 Total Events:              1
 Lost Events:               0
 Error Events:              0
 Ack Errors:                0
```

3.2 Privileged EXEC mode

disable

To leave privileged EXEC mode and return to user EXEC mode, use the **disable** command.

disable

Command Modes Privileged EXEC

Examples The following example returns to user EXEC mode from privileged EXEC mode:

```
vqec# disable
vqec>
```

configure terminal

To enter the configuration mode, use the **configure terminal** command.

configure terminal

Command Modes Privileged EXEC

Examples The following example enters the configuration mode from privileged EXEC mode:

```
vqec# configure terminal
vqec(config)#
```

clear counters

To clear all counters, use the **clear counters** command.

clear counters

Command Modes Privileged EXEC

Usage Guidelines This command clears all counters and statistics.

Examples The following example clears all counters:

```
vqec# clear counters
```

send-debugs-to-cli

Toggles whether or not the VQE-C debug messages are sent to only syslog, or syslog and the CLI as well.

send-debugs-to-cli *{enable | disable}*

Syntax Description

enable	Enable the debug messages to go to the CLI.
disable	Disable the debug messages from going to the CLI.

Defaults

By default, this is enabled, and the debug messages will go to the CLI.

Command Modes

Privileged EXEC

Examples

The following example disables debug messages from going to the CLI:

```
vqec# send-debugs-to-cli disable
```

debug

To enable or disable specific debug flags, use the **debug** command.

debug *debug-type* {*enable* | *disable*}

Syntax Description

debug-type	The debug flag to operate on.
enable	Enable this debug flag.
disable	Disable this debug flag.

Defaults

By default, all debug flags are disabled.

Command Modes

Privileged EXEC

Usage Guidelines

The *debug-type* must be one of the following:

all	Represents set of all other types
channel	Include debug messages from channel
cpchan	Include debug messages from CP channel
error-repair	Include debug messages from error repair
event	Include debug messages from event
rcc	Include debug messages from rcc
igmp	Include debug messages from igmp
input	Include debug messages from input
output	Include debug messages from output
pcm	Include debug messages from pcm
recv-socket	Include debug messages from recv_socket
rtcp	Include debug messages from rtcp
timer	Include debug messages from timer

tuner	Include debug messages from tuner
nat	Include debug messages from NAT
chan_cfg	Include debug messages from channel configuration
upcall	Include debug messages from upcalls
updater	Include debug messages from updater
dp-error-repair	Include debug messages from DP error repair
dp-nll	Include debug messages from NLL in DP
dp-nll-adjust	Include debug messages from NLL adjustments
dp-pcm	Include debug messages from PCM in DP
dp-pcm-pak	Include debug messages from PCM for each packet
dp-inputshim	Include debug messages from DP input shim
dp-outputshim	Include debug messages from DP output shim
dp-tlm	Include debug messages from DP toplevel manager
dp-rcc	Include debug messages from RCC feature in DP
dp-fec	Include debug messages from FEC feature in DP
dp-failover:	Include debug messages from source failover events in DP

Examples

The following example enables debug flags for *fcc* and *igmp*, and then disables the *fcc* flag:

```
vqec# debug rcc enable
vqec# debug igmp enable
vqec# debug rcc disable
vqec# show debug
channel:           disabled
chan_cfg:          disabled
cpchan:            disabled
error-repair:      disabled
event:             disabled
rcc:               disabled
igmp:              enabled
input:             disabled
output:            disabled
pcm:               disabled
recv-socket:       disabled
rtcp:              disabled
nat :              disabled
timer:             disabled
tuner:             disabled
upcall:            disabled
updater:           disabled
dp-tlm:            disabled
dp-inputshim:      disabled
dp-outputshim:     disabled
dp-nll:            disabled
dp-nll-adjust:     disabled
dp-pcm:            disabled
dp-pcm-pak:        disabled
dp-error-repair:   disabled
dp-rcc:            disabled
dp-fec:            disabled
dp-failover:       disabled
```

The following example enables debug flags for *channel*, *event*, *igmp*, and *pcm*, and then disables *all* debug flags:

```
vqec# debug channel enable
vqec# debug event enable
vqec# debug igmp enable
vqec# debug pcm enable
vqec# debug dp-pcm enable
vqec# show debug
channel:           enabled
chan_cfg:          disabled
cpchan:            disabled
error-repair:      disabled
event:             enabled
rcc:               disabled
igmp:              enabled
input:             disabled
output:            disabled
pcm:               enabled
recv-socket:       disabled
rtcp:              disabled
nat :              disabled
timer:             disabled
tuner:             disabled
upcall:            disabled
updater:           disabled
dp-tlm:            disabled
dp-inputshim:      disabled
dp-outputshim:     disabled
dp-nll:            disabled
dp-nll-adjust:     disabled
dp-pcm:            enabled
dp-pcm-pak:        disabled
dp-error-repair:   disabled
dp-rcc:            disabled
dp-fec:            disabled
dp-failover:       disabled

vqec# debug all disable
vqec# show debug
channel:           disabled
chan_cfg:          disabled
cpchan:            disabled
error-repair:      disabled
event:             disabled
rcc:               disabled
igmp:              disabled
input:             disabled
output:            disabled
pcm:               disabled
recv-socket:       disabled
rtcp:              disabled
nat :              disabled
timer:             disabled
tuner:             disabled
upcall:            disabled
updater:           disabled
dp-tlm:            disabled
dp-inputshim:      disabled
dp-outputshim:     disabled
dp-nll:            disabled
dp-nll-adjust:     disabled
dp-pcm:            disabled
dp-pcm-pak:        disabled
dp-error-repair:   disabled
```

```
dp-rcc:           disabled
dp-fec:           disabled
dp-failover:      disabled
```

monitor

To use the system monitoring tools, use the **monitor** command.

The **monitor output-sched** command

monitor output-sched *{show | on | off | reset}*

may be used to collect and observe the intervals (measured in milliseconds) between instances of output scheduling. The implementation uses a timer for the purposes of awaking and updating the packets available for reading from its tuners every VQEC_PCM_OUTPUT_SCHED_INTERVAL (20) milliseconds. The histogram keeps track of the actual intervals (in milliseconds) at which the implementation was awoken to update the tuners' output queues.

Syntax Description

show	Shows the output scheduling histogram.
on	Turns on measurement logging for the histogram.
off	Turns off measurement logging for the histogram.
reset	Clears the entries in the output scheduling histogram.

Defaults

By default, output scheduling measurement logging is **off**.

Command Modes

Privileged EXEC

Usage Guidelines

The **monitor output-sched** measurement logging may be turned on to troubleshoot reasons for drops on a tuner's output queue.

Examples

The following example shows an output scheduling histogram containing measurements from 2034 intervals, all within the range of 20-29 milliseconds.

```
vqec# monitor output-sched show

Histogram of Output Scheduling Intervals (in ms):
       20 -         29    [       2034  ]
```

3.3 Configure mode

app-delay

To adjust the playout timing of replicated APP packets after a successful RCC, use the **app-delay** command.

app-delay *delay*

Syntax Description

delay	Integer number of ms to delay each APP packet copy.

Defaults

By default, *delay* is set to 0.

Command Modes

Configure mode

Usage Guidelines

When VQE-C is replicating APP packets (i.e. more than one packet containing APP data is being sent on output at the beginning of a new channel), this command can be used to send out the replicated APP packets at a specific rate.

Examples

The following example enables the VQE-C to delay each packet containing APP data by 40 milliseconds. So, in other words, assuming that VQE-C is replicating APP packets to send a total of 3 of them at the beginning of a new channel's output stream, the first 3 packets on the stream would be sent at times (in ms) T, T+40, T+80, followed by the rest of the MPEG data packets:

```
vqec(config)# app-delay 40
```

channel tr-135

To change an active channel's TR-135 writable parameters, use the "channel tr-135" command.

channel tr-135 <channel-url> gmin <gmin> slmd <slmd>

Syntax Description

channel-url	URL of the active channel whose TR-135 parameters are to be changed.
gmin	Integer specifying value of gmin, a TR-135 writable parameter.
slmd	Integer specifying value of "Severe Loss Minimum Distance", a TR-135 writable parameter.

Defaults

By default, when a channel becomes active, gmin and slmd are set to 0, and TR-135 statistics, that are related to gmin and slmd are disabled..

Command Modes

Configure mode

Usage Guidelines

After a channel becomes active (when a tuner binds to that channel), the above command can be used to modify the channel's TR-135 writeable parameters: gmin and slmd. The address and protocol of the channel is specified in the form of channel-url, which has the form: output-type://address:port
where
output-type must be “udp" or "rtp".
address is a valid IPv4 address to be used as the destination address of the output stream
port is an integer in the range [1, 65535] to be used as the destination port of the output stream.

Examples

The following example sets gmin to 1 and slmd to 2 for an active channel identified by the URL rtp://224.1.1.1:50000

vqec(config)# ch tr-135 rtp://224.1.1.1:50000 gmin 1 slmd 2

drop

To configure VQE-C packet drop simulation, use the **drop** command.

drop [session { primary | repair }] { enable | disable |
interval *continuous-drop interval* | **percentage** *percentage* **}**

Syntax Description

continuous-drop	Integer in the range [0, *interval*].
interval	Integer greater than *continuous-drop*.
percentage	Integer in the range [0, 100].

Defaults

By default, *continuous-drop*, *interval,* and *percentage* are set to 0.

Command Modes

Configure mode

Usage Guidelines

Drop simulation makes VQE-C intentionally drop some packets and behave as if it never received packets that it intentionally drops. This is useful, for example, when attempting to verify error-repair operation of VQE-C despite the primary stream being received without losses from the network.

Drop characteristics of each session type may be configured independently via use of the **session** keyword. If the **session** keyword is omitted, then the command is interpreted as configuring drop characteristics of primary streams.

Use of the **enable** keyword enables drop the simulator for a particular stream type (primary or repair), while use of the **disable** keyword disables the drop simulator for the given stream type. When drops for a particular session type are enabled, configuration of the **interval** and **percentage** keywords controls the drop function.

When **interval** is configured, VQE-C intentionally drops the first *continuous-drop* sequential packets for every *interval* packets it receives on the specified session of a channel. The order of arrival of packets on that session (rather than the order of sender’s

transmission or sequence number) determines which *continuous-drop* packets are dropped .

When **percentage** is configured (and both arguments of **interval** are set to 0) then VQE-C intentionally drops the configured percentage of the packets it receives, as if it had never received them.

Examples

The following example enables the VQE-C drop simulation and sets VQE-C to drop 3 sequential packets for every 50 packets it receives on the primary session. In other words, if VQE-C starts receiving packets with sequence number 0 and receives them all sequentially, it will drop any whose sequence numbers fall in the pattern {0, 1, 2, 50, 51, 52, 100, 101, 102, …}:

```
vqec(config)# drop session primary enable
vqec(config)# drop session primary interval 3 50
```

The following example enables VQE-C drop simulation for repair streams and sets VQE-C to drop a random 10% of the packets it receives:

```
vqec(config)# drop session repair enable
vqec(config)# drop session repair percentage 10
```

The following example disables VQE-C drop simulation for both primary and repair streams :

```
vqec(config)# drop session primary disable
vqec(config)# drop session repair disable
```

error-repair

Toggles global 'error-repair' processing state. When in the 'error-repair disable' state, VQE-C will not perform any retransmission based error repair (even if the channel is described as having error repair enabled). When in the default 'error-repair enable' state VQE-C will perform retransmission repair only when a channel is described as having error-repair configured. Error-repair is configured on a per-channel basis via the channel lineup. Note that error repair enable will not turn on retransmission based error repair for a channel that does not have retransmission based error repair configured in the channel_lineup".

error-repair *{enable | disable | policer}*

Syntax Description

enable	Enable the error repair feature in VQE-C.
disable	Disable the error repair feature in VQE-C.
policer	Configure the error repair policer (see **error-repair policer** command description).

Defaults

By default, the VQE-C error repair feature is enabled.

Command Modes

Configure mode

Usage Guidelines

Use this command to disable retransmission based error repair to see how the video for a retransmission based error repair enabled channel will appear without retransmission based error repair. This command allows a quick way to toggle between viewing a retransmission based error repair corrected stream and a stream with no retransmission based correction. Note that use of retransmission based error repair is configured on a per-channel basis and this command allows overriding the configuration to force retransmission based error repair off. This command is used primarily for demonstrating the effects of retransmission based error repair.

.

Examples

The following example disables the VQE-C error repair feature on the next bind after all tuners have been unbound from the channel. Once this feature is disabled, any packets that are lost in the network will not be recovered by VQE:

```
vqec(config)# error-repair disable
```

error-repair policer

To configure the error repair policer in VQE-C, use the **error-repair policer** sub-command.

error-repair policer *{enable | disable | rate | burst}*

Syntax Description

enable	Enable the error repair policing feature in VQE-C.
disable	Disable the error repair policing feature in VQE-C.
rate	Set the allowed rate of the token bucket for policing error repair requests, expressed as a percentage of the primary stream rate ("b=AS") rate
burst	Set the capacity of the token bucket for policing error repair requests. The value is expressed as a duration of time (in milliseconds) at which the token bucket's capacity would be reached if the bucket were empty and filled at rate **rate**, with no tokens drained. A larger capacity indicates that more repair requests may be sent over the short term without being policed.

Defaults

The VQE-C error repair policer is disabled by default. Default values and supported ranges for the token bucket parameters are as follows:

rate	5% (default)
	1% (Minimum)
	100% (Maximum)
burst	*10000 ms (default)*
	1ms (Minimum)
	60000ms (Maximum)

Command Modes

Configure mode

Usage Guidelines

When the VQE-C experiences a high drop rate for a stream, a large number of error repair requests may be sent to the VQE-S. If it is desirable to limit the error repair requests sent into the network by the VQE-C (e.g. to avoid flooding the network due to a lossy stream), this feature may be enabled.

The VQE-C uses the relative policer values configured above, along with the primary stream's rate and an assumed packet size of 1356 bytes, to compute the token bucket parameters used for error repair policing. The default packet size is derived as follows:

Default packet size = 7 (MPEG TS pkts) * 188 (bytes/MPEG TS pkt) +
12 (bytes/RTP header) + 12 (bytes/UDP header) +
20 (bytes/IP header).
= 1356

The token bucket's absolute rate (rate', expressed in repair packets/s) and burst (burst', expressed in packets) parameters used for policing repair requests are then computed from configured values as follows:

rate' (packets/s) = <rate>/100 * <stream-rate (bps)> / (8 bits/byte * 1356 bytes/packet)
burst' (packets) = <rate' (packets/s)> * <burst (ms)> / (1000 ms per second)

Changes to the configured values take effect with the next channel change (streams currently being received are *not* affected).

Examples

The following example enables error repair policing at a rate of 5%, and with a burst value of 10000ms:

```
vqec(config)# error-repair policer enable
vqec(config)# error-repair policer rate 5
vqec(config)# error-repair policer burst 10000
```

Assuming a primary stream rate of 6Mbps, the rate' and burst' values used by the policer (as described in the "Usage Guidelines" section above) are 28 packets/s and 277 packets, respectively. These values are displayed in the `show tuner` output.

fec

Toggles global fec processing state. When in the 'fec disable' state, VQE-C will not perform any FEC repair (even if the channel is described as having FEC enabled). When in the default 'fec enable' state VQE-C will perform FEC based repair only when a channel is described as having FEC configured. FEC is configured on a per-channel basis via the channel lineup. Note that FEC enable will not turn on FEC for a channel that does not have FEC configured in the channel_lineup".

This is mainly a demo command, if we turn FEC on, we can see packets are corrected, if we turn FEC off, no packet is decoded by FEC module .

fec *{enable | disable}*

Syntax Description

enable	Enable the fec decoding feature in VQE-C.
disable	Disable the fec decoding feature in VQE-C.

Defaults

By default, the VQE-C fec decoding feature is enabled.

Command Modes

Configure mode

Usage Guidelines

.
Use this command to disable FEC to see how the video for a FEC enabled channel will appear without FEC. This command allows a quick way to toggle between viewing a FEC corrected stream and a stream with no FEC correction. Note that use of FEC is configured on a per-channel basis and this command allows overriding the configuration to force FEC off. This command is used primarily for demonstrating the effects of FEC.

Examples

The following example disables the VQE-C fec decoding feature. Once this feature is disabled, no received FEC packets are decoded:

```
vqec(config)# fec disable
```

rcc

Toggles global rcc processing state. When in the 'rcc disable' state, VQE-C will not perform any rapid channel change (even if the channel is described as having rapid channel change enabled). When in the default 'rcc enable' state VQE-C will perform rapid channel change only when a channel is described as having rapid channel change configured. Rapid channel change is configured on a per-channel basis via the channel lineup. Note that RCC enable will not turn on RCC for a channel that does not have rapid channel change configured in the channel_lineup".

This is mainly a demo command, if we turn RCC on, we can see rapid channel change is performed when changing to a new channel, if we turn RCC off, no rapid channel change is performed during a channel change.

rcc *{enable | disable}*

Syntax Description

enable	Enable the rapid channel change feature in VQE-C.
disable	Disable the rapid channel change feature in VQE-C.

Defaults

By default, the VQE-C rcc is enabled.

Command Modes

Configure mode

Usage Guidelines

.
Use this command to disable RCC to see how the channel change for a RCC enabled channel will appear without rapid channel change. This command allows a quick way to toggle between viewing a channel change with and without rapid channel change feature. Note that use of RCC is configured on a per-channel basis and this command allows overriding the configuration to force RCC off. This command is used primarily for demonstrating the effects of rapid channel change.

Examples

The following example disables the VQE-C rapid channel change feature. Once this feature is disabled, no rapid channel change is performed:

```
vqec(config)# rcc disable
```

update

To update the system and/or and channel configurations in VQE-C, use the **update** command.

update [file <*filename*> type {network|override-tags|channel}]

If no parameters are specified (i.e. "**update <cr>**"), then VQE-C will attempt to update its Channel and Network Configuration (if configured) via CDI.

If the file and type parameters are provided, then the update is assumed to come from a local file instead of being supplied by CDI.

Syntax Description

file <*filename*> Identifies the local file whose contents are used for updating a VQE-C configuration. Expected file syntax based on the file type is as follows:

- network – assumes same as VQE-C start-up file (see VQE-C Configuration Guide)
- override-tags – assumes one parameters and value per line, separated by whitespace
- channel – assumes SDP syntax

type {network|override-tags|channel}
Identifies the type of configuration to be updated.

<cr> Triggers a CDI-based update.

Defaults

By default, VQE-C will update necessary components via CDI when needed and as specified by the VQE-C system configuration.

Command Modes

Configure mode

Usage Guidelines

Use "update <cr>" to trigger a configuration update ahead of the next polled update scheduled by the updater, or if update polling is not configured.

Use the "update file <filename> type <type>" form of this command to force an update to VQE-C configuration using the contents of a local file. This command should not typically be needed and is provided for special situations only (e.g. as a means to test VQE-C behavior with different configuration files or to supply a configuration file when a VCDS is not available).

Note that if CDI is enabled, file-based configuration updates may be overwritten by CDI updates.

Examples

The following example will update the VQE-C configuration from the latest configuration provided by the VCDS referenced by the DNS server:

```
vqec(config)# update
```

parse sdp

To parse a file containing a single SDP channel description into a list of channel parameters, use the **parse sdp** command.

parse sdp { vod | linear } *input-file* [output *output-file*] [*params-list*]

Syntax Description

{ vod | linear}	SDP channel description type: linear for multicast, vod for unicast.
input-file	Path of file containing SDP channel description.
output-file	Path of file to which resulting parameters are written.
params-list	Optional parameters list used to override SDP parameters.

Command Modes

Configure mode

Usage Guidelines

The **parse sdp** command exposes the VQE-specific SDP parser and validator and allows external SDP files to be parsed into parameter lists. The validation type is determined by the first argument: "vod" for a video on demand SDP or "linear" for a multicast channel SDP. The input and output file paths should not include special shell symbols (such as "~" or ".."); they will not be expanded. The command output will be written to the output file, if specified, or to the CLI otherwise.

Since certain SDP files will not contain all the necessary parameters for binding to a channel (such as transport addresses for a VoD session), additional override parameters may be provided. The format of these input parameters is consistent with the parse command's output. See the examples section for more information.

Examples

The following example parses a file "mychannel.cfg" containing a multicast SDP channel description and outputs the results to the CLI. These resulting parameters may be copied as inputs to the **tuner bind** command. If an output file were specified, the same list would be written to the output file (which could then be provided to the tuner bind command as well).

```
vqec(config)# parse sdp linear mychannel.cfg

primary-dest-addr 229.1.1.8 primary-dest-port 53198 primary-dest-
rtcp-port 53199 primary-src-addr 9.3.13.2 primary-src-port 0
primary-src-rtcp-port 53199 primary-payload-type 96 primary-bit-rate
14910000 primary-rtcp-sndr-bw 53 primary-rtcp-rcvr-bw 530000
primary-rtcp-per-rcvr-bw 53 primary_rtcp_xr_loss_rle_enable
primary_rtcp_xr_per_loss_rle_enable primary-rtcp-xr-stat-flags
loss,dup,jitt fbt-addr 5.3.19.100 er_enable rcc_enable rtx-src-addr
5.3.19.100 rtx-src-port 50000 rtx-src-rtcp-port 50001 rtx-dest-addr
0.0.0.0 rtx-dest-port 0 rtx-dest-rtcp-port 0 rtx-payload-type 99
rtx-rtcp-sndr-bw 53 rtx-rtcp-rcvr-bw 53
```

proxy-igmp-join

To enable the VQE-C proxy mode, use the **proxy-igmp-join** command.

proxy-igmp-join *tuner-name stb-if-name stb-ip-addr*

Syntax Description

tuner-name	Name of an active tuner.
stb-if-name	Name of the ethernet interface the STB is connected to.
stb-ip-addr	IP address of the STB.

Defaults

By default, VQE-C proxy mode is disabled.

Command Modes

Configure mode

Usage Guidelines

When the VQE-C proxy mode is enabled for a tuner, that tuner will listen on *stb-if-name* interface for IGMP join requests coming from a set-top box with address *stb-ip-addr*. When the tuner receives an IGMP leave and join report from the set-top box, the tuner will then tune to the new channel being joined to by the STB. It will then begin to send a repaired output multicast stream on the *stb-if* interface for the set-top box to display the stream.

To disable the VQE-C proxy mode for a tuner, use the same command, but provide "0.0.0.0" as *stb-ip-addr*.

Usage Restrictions

Needs superuser privileges to successfully execute this command. Works only for IEEE 802.3 encapsulation, and will show unexpected behavior for Ethernet II, and 802.1Q (VLAN) encapsulations. Works only for a single STB behind the VQE-C device, and supports UDP-only stream output. Also has only host device support, and does not support operation on a router or bridge. A user cannot configure more than one IGMP proxy for the same STB (IP Address).

Examples

The following example enables the VQE-C proxy mode for a tuner named "0" and listens for IGMP messages on eth1, where eth1 is the interface on the VQE-C-machine, to which the STB is connected. Here, the address of the STB is 192.168.1.150. When VQE-C receives IGMP join report for some active channel, say, channel 224.1.1.*t,* from the STB, VQE-C's tuner will tune to this address and stream output to the 224.1.1.*t* address on the eth1 interface, as a result of which, the STB would then receive multicast data for 224.1.1.*t*.

```
vqec(config)# proxy-igmp-join 0 eth1 192.168.1.150
```

stream-output

To enable VQE-C output streaming, use the **stream-output** command.

stream-output *tuner-name if-name output-url*

Syntax Description

tuner-name	Name of an active tuner.
if-name	Name of an Ethernet interface.
output-url	Destination URL of the output stream.

Defaults By default, VQE-C output streaming mode is disabled.

Command Modes Configure mode

Usage Guidelines When the VQE-C output streaming mode is enabled for a tuner, that tuner will begin sending a repaired output stream on the *tuner-if* interface. The address and protocol of the repaired output multicast stream is determined from the *output-url*, which has the form:

output-type://*address*:*port*
where
output-type must be "udp"
address is a valid IPv4 address to be used as the destination address of the output stream
port is an integer in the range [1, 65535] to be used as the destination port of the output stream

Examples The following example enables the VQE-C output streaming mode for a tuner named "0" and begins to send a repaired UDP multicast stream from the interface eth1 to the address 192.168.1.128 on port 50000:

```
vqec(config)# stream-output 0 eth1 udp://192.168.1.128:50000
```

tuner bind

To bind an active tuner to a either valid channel configured in the channel lineup or a temporary channel described by a channel parameters list, use the **tuner bind** command.

tuner bind <tuner-name> {<url> | chan-params {file <filename> | list <param_list>}}
[no_rcc]
[fastfill]
[max-fastfill <max_fastfill>]
[rcc-bw <max_recv_bw_rcc>]
[er-bw <max_recv_bw_er>]
[tr-135 gmin <gmin> slmd <slmd>]

Syntax Description

tuner-name	Name of an active tuner.
channel-url	URL of the channel to be bound.
params-file	Path of a file containing a channel parameter list.
params-list	A list of key-value channel parameters.
no_rcc	(Optional) If this keyword is specified, the channel change will occur with RCC disabled.
tr-135	(Optional) If this keyword is specified, after a tuner binds to this channel, the channel's TR-135 writable parameters : gmin and severe loss minimum distance (slmd) are updated.

Command Modes Configure mode

Usage Guidelines

The *channel-url* must be of the form:

protocol://*address*:*port*
where
protocol is the protocol of the channel. Usually "igmp".
address is a valid IPv4 multicast address of the channel
port is an integer in the range [1, 65535] and should match the primary rtp port of the channel as configured in the channel lineup

The *params-list* or the contents of the *params-file* should take the following format:

primary-dest-addr *addr* primary-dest-port *port* [primary-dest-rtcp-port *port*] [primary-src-addr *addr*] [primary-src-port *port*] [primary-src-rtcp-port *port*] [primary-payload-type *payload-type*] primary-bit-rate *bit-rate* [primary-rtcp-sndr-bw *rtcp-bw*] [primary-rtcp-rcvr-bw *rtcp-bw*] [primary-rtcp-per-rcvr-bw *rtcp-bw*] [primary_rtcp_xr_loss_rle_enable] [primary_rtcp_xr_per_loss_rle_enable] [primary-rtcp-xr-stat-flags loss[,dup][,jitt]] [fbt-addr *addr*] [er_enable] [rcc_enable] [rtx-src-addr *addr*] [rtx-src-port *port*] [rtx-src-rtcp-port *port*] [rtx-dest-addr *addr*] [rtx-dest-port *port*] [rtx-dest-rtcp-port *port*] [rtx-payload-type *payload-type*] [rtx-rtcp-sndr-bw *rtcp-bw*] [rtx-rtcp-rcvr-bw *rtcp-bw*] [rtx_rtcp_xr_loss_rle_enable] [rtx-rtcp-xr-stat-flags loss[,dup][,jitt]] [fec_enable] [fec-mode {1D | 2D}] [fec1-mcast-addr *addr*] [fec1-mcast-port *port*][fec1-mcast-rtcp-port *port*] [fec1-src-addr *addr*] [fec1-payload-type *payload-type*] [fec1-rtcp-sndr-bw *rtcp-bw*] [fec1-rtcp-rcvr-bw *rtcp-bw*] [fec2-mcast-addr *addr*] [fec2-mcast-port *port*] [fec2-mcast-rtcp-port *port*] [fec2-src-addr *addr*] [fec2-payload-type *payload-type*] [fec2-rtcp-rcvr-bw *rtcp-bw*] [fec2-rtcp-sndr-bw *rtcp-bw*]
where
addr is a dotted decimal IP address
port is a valid UDP port number
payload-type is an RTP payload type in the dynamic range 96 – 128
bit-rate is the rate of the primary stream in bps
rtcp-bw is the maximum bandwidth used for RTCP in bps

The **tuner bind** command may use the parameters list or parameters file created as the output from the **parse sdp** command. Using the chan-params option in a bind call allows the user to create and listen to channels beyond those described in the channel lineup.

It should be noted that it is not necessary to use **tuner unbind** prior to using **tuner bind** multiple consecutive times.

Examples

The following example binds a tuner named "tuner1" to channel 230.151.1.1 at port 10000 using the RTP protocol with RCC enabled if available:

```
vqec(config)# tuner bind tuner1 rtp://230.151.1.1:10000
```

tuner create

To create a tuner, use the **tuner create** command.

tuner create *tuner-name*

Syntax Description

tuner-name	Name of the tuner to be created.

Command Modes

Configure mode

Examples

The following example creates a tuner named "newtuner":

```
vqec(config)# tuner create newtuner
```

tuner destroy

To destroy a tuner, use the **tuner destroy** command.

tuner destroy *tuner-name*

Syntax Description

tuner-name	Name of an active tuner to be destroyed.

Command Modes

Configure mode

Examples

The following example first creates a tuner named "newtuner", and then destroys it:

```
vqec(config)# tuner create newtuner
vqec(config)# tuner destroy newtuner
```

tuner unbind

To unbind an active tuner from its currently bound channel, use the **tuner unbind** command.

tuner unbind *tuner-name*

Syntax Description

tuner-name	Name of an active tuner currently bound to a channel.

Command Modes

Configure mode

Usage Guidelines

When a tuner is unbound from a channel, that tuner will no longer receive any packets from any channels until it is bound to another channel via the **tuner bind** command.

Examples

The following example binds a tuner named "0" to channel 224.1.1.1 at port 50000 using the IGMP protocol, and then unbinds that same tuner so it will stop receiving packets from that channel:

```
vqec(config)# tuner bind 0 igmp://224.1.1.1:50000
vqec(config)# tuner unbind 0
```

End of Document