

Font Size

[A](#) ([javascript:changeStyle1\(\);](#)) [A](#) ([javascript:changeStyle2\(\);](#)) [A](#) ([javascript:changeStyle3\(\);](#))

# How to find mapping between voq and voq connector id on DNX device.

**Doc ID:** HT2000 ([permalink \(https://support.broadcom.com/KMS/Main.aspx?id=HT2000\)](https://support.broadcom.com/KMS/Main.aspx?id=HT2000))

**Version:** 2.0

**Published date:** 08/10/2015

**Products:** [88370 \(Qumran-MX\)](#) ([index?page=content&channel=HOW\\_TO&cat=88370\\_QUMRANMX](#)), [88670 \(Jericho\)](#) ([index?page=content&channel=HOW\\_TO&cat=88670\\_JERICH0](#)), [88660 \(Arad+\)](#) ([index?page=content&channel=HOW\\_TO&cat=88660\\_ARAD](#)), [88650 \(Arad\)](#) ([index?page=content&channel=HOW\\_TO&cat=88650\\_ARAD](#))

**Domains:** [Queuing](#) ([index?page=content&channel=HOW\\_TO&cat=QUEING](#))

## Summary

This article shows how to find mapping between voq and voq-connector id (aka flow-id). For completeness, this article shows other diag commands germane to voq configurations and egq mapping which can provide useful information when debugging traffic related problems.

## Description

When debugging traffic problems affecting only certain flow(s), one wants to know the flow-id of the affected voq to investigate voq or voq flow related configuration problems. Prior to sdk-6.4.4, there is no user-friendly diag command which allows users to retrieve desired information. The sdk-6.4.4 has added new commands, “*cosq conn ing/egr*”, which shows mapping between voq and voq-connector. These diag commands can be used to identify the misconfiguration of voq and voq-connector such as mismatched voq-connector-id's between ingress and egress and missing voq-connectors. On Jericho, when the device is working in the symmetric mode, the voq configuration must be replicated on both cores such that the same voq id is allocated on both cores but a different flow-id must be allocated to each voq since it is required that flow-id's be unique in the egress FAP. When traffic to the same destination works only on one core but not on the other, one can dump the voq/voq-connector information via "cosq conn" and verify that two different flow-id's are allocated to the same voq-id's on core-0 and core-1.

# voq to voq-connector-id mapping on the ingress device  
**cosq conn ing**

# voq to voq-connector-id mapping on the egress device  
**cosq conn egr**

# modid/modport and voq mapping for a gport  
**gport**

# voq configuration at the device level  
**diag cosq voq**

# detailed configurations of a selected voq  
**diag cosq voq id=<voq-id> detail=1**

# port to EGQ mapping information  
**diag cosq qpair egq map**

# graphical representation of port to EGQ mapping

**diag cosq qpair egq ps=<port>**

Examples on Negev:

**# voq mapping on the ingress device**

BCM.0> cosq conn ing

Voq ID	NOF Voqs	Core	Remote Voq Connector	Remote Modid
32	8	-17	48	0
128	8	-17	240	0
136	8	-17	256	0
144	8	-17	272	0
152	8	-17	288	0
160	8	-17	304	1
320	8	-17	1136	1
1944	8	-17	9376	0
1952	8	-17	7456	0
1960	8	-17	9392	1
1968	8	-17	7472	1
2064	8	-17	32	0

**# voq mapping on the egress device**

BCM.0> cosq conn egr

Voq Connector ID	NOF Connectors	Core	Ingress Voq	Ingress Modid
32	8	0	2064	0
544	8	0	2064	1
48	8	0	32	0
560	8	0	32	1
240	8	0	128	0
752	8	0	128	1
256	8	0	136	0
768	8	0	136	1
272	8	0	144	0
784	8	0	144	1
288	8	0	152	0
800	8	0	152	1
7456	8	0	1952	0
7968	8	0	1952	1
9376	8	0	1944	0
9888	8	0	1944	1
304	8	1	160	0
816	8	1	160	1
1136	8	1	320	0
1648	8	1	320	1
7472	8	1	1968	0
7984	8	1	1968	1
9392	8	1	1960	0
9904	8	1	1960	1

**# modid/modport and voq mapping of a gport**

BCM.0> **gport**

gp:0x241c0020	cos:8	Unicast Queue Group	mod: 0	port: 1 voq: 32)
gp:0x241c0080	cos:8	Unicast Queue Group	mod: 0	port:13 voq: 128)
gp:0x241c0088	cos:8	Unicast Queue Group	mod: 0	port:14 voq: 136)
gp:0x241c0090	cos:8	Unicast Queue Group	mod: 0	port:15 voq: 144)
gp:0x241c0098	cos:8	Unicast Queue Group	mod: 0	port:16 voq: 152)
gp:0x241c00a0	cos:8	Unicast Queue Group	mod: 1	port:17 voq: 160)

gp:0x241c0140	cos:8	Unicast Queue Group	mod: 1	port:19 voq: 320)
gp:0x241c0798	cos:8	Unicast Queue Group	mod: 0	port:255 voq:1944)
gp:0x241c07a0	cos:8	Unicast Queue Group	mod: 0	port:240 voq:1952)
gp:0x241c07a8	cos:8	Unicast Queue Group	mod: 1	port:255 voq:1960)
gp:0x241c07b0	cos:8	Unicast Queue Group	mod: 1	port:240 voq:1968)
gp:0x241c0810	cos:8	Unicast Queue Group	mod: 0	port: 0 voq:2064)
gp:0x78200020	cos:8	VOQ Connector	flow:32)	
gp:0x78200030	cos:8	VOQ Connector	flow:48)	
gp:0x782000f0	cos:8	VOQ Connector	flow:240)	
gp:0x78200100	cos:8	VOQ Connector	flow:256)	
gp:0x78200110	cos:8	VOQ Connector	flow:272)	
gp:0x78200120	cos:8	VOQ Connector	flow:288)	
gp:0x78201d20	cos:8	VOQ Connector	flow:7456)	
gp:0x782024a0	cos:8	VOQ Connector	flow:9376)	
gp:0x78220130	cos:8	VOQ Connector	flow:304)	
gp:0x78220470	cos:8	VOQ Connector	flow:1136)	
gp:0x78221d30	cos:8	VOQ Connector	flow:7472)	
gp:0x782224b0	cos:8	VOQ Connector	flow:9392)	

### # voq configuration on a device

BCM.0> **diag cosq voq**

Core mode is symmetric!

Dram-mix threshold range is: Min: (26214) and Max: (26542).

Multicast Queue is enabled. The range is: Min: (0) and Max: (3).

Shaper Queue is enabled. The range is: Min: (98303) and Max: (98302).

Fabric Queue is not enabled. The range is: Min: (4) and Max: (98303).

Credit watchdog Queue is enabled. The range is: Min: (0) and Max: (98271).

### # detailed voq configuration of voq id=32

BCM.0> **diag cosq voq id=32 detail=1**

Basic info

Q type: voq

num cos: 8, cosq class: 0

Base queue id: 32, base queue gport: 0x24000020

Credit request type: BCM\_COSQ\_DELAY\_TOLERANCE\_10G\_SLOW\_ENABLED Adjusted for slow enabled 10Gb ports

Watchdog enable in common status message mode

Is queue in credit watchdog queue range: True

Credit watchdog message time: 1000

Delete queue time: 32768

Backoff enter queue credit balance threshold: 2147349664

Backoff exit queue credit balance threshold: 31029972

Backlog enter queue credit balance threshold: 2147349672

Backlog exit queue credit balance threshold: 2147349672

Empty queue satisfied credit balance threshold: 60001960

Max empty queue credit balance threshold: 2147350100

Exceed max empty queue credit balance threshold: 16782854

Off-To-Slow credit balance threshold: 60008204

Off-To-Normal credit balance threshold: 2147349648

Slow-To-Normal credit balance threshold: 56756864

Normal-To-Slow credit balance threshold: 16782854

Delay Tolerance is OCB only

Delay Tolerance is High Q Priority

Slow Level Thresh Down in slow level number: 0 is: 22

Slow Level Thresh Down in slow level number: 1 is: 1

Slow Level Thresh Down in slow level number: 2 is: 118

Slow Level Thresh Down in slow level number: 3 is: 6

Slow Level Thresh Down in slow level number: 4 is: 60008204

Slow Level Thresh Down in slow level number: 5 is: 2147349744

Slow Level Thresh Down in slow level number: 6 is: 2147349744

Slow Level Thresh Up in slow level number: 0 is: 1

Slow Level Thresh Up in slow level number: 1 is: 2147349696

Slow Level Thresh Up in slow level number: 2 is: 2147349704  
 Slow Level Thresh Up in slow level number: 3 is: 2147349704  
 Slow Level Thresh Up in slow level number: 4 is: 97560764  
 Slow Level Thresh Up in slow level number: 5 is: 4  
 Slow Level Thresh Up in slow level number: 6 is: 2147350100  
 Credit value(local): 1024, Credit value(remote): 2048  
 Credit discount value: 2  
 Rate class info  
   WRED info:  
     Green : enable(False) min\_thresh(0) max\_thresh(0) drop\_probability(0)  
     Yellow: enable(False) min\_thresh(0) max\_thresh(0) drop\_probability(0)  
     Red : enable(False) min\_thresh(0) max\_thresh(0) drop\_probability(0)  
     Black : enable(False) min\_thresh(0) max\_thresh(0) drop\_probability(0)  
   Tail drop info:  
     Green : max queue size in bytes(1048576), max queue size in BDs(1048576)  
     Yellow: max queue size in bytes(1048576), max queue size in BDs(1048576)  
     Red : max queue size in bytes(1048576), max queue size in BDs(1048576)  
     Black : max queue size in bytes(1048576), max queue size in BDs(1048576)  
   Guaranteed info:  
     Green : min queue size in bytes(0), min queue size in BDs(0)  
     Yellow: min queue size in bytes(0), min queue size in BDs(0)  
     Red : min queue size in bytes(0), min queue size in BDs(0)  
     Black : min queue size in bytes(0), min queue size in BDs(0)  
   Fair adaptive tail drop info:  
     Enable: False  
     Green : alpha(7)  
     Yellow: alpha(7)  
     Red : alpha(7)  
     Black : alpha(7)  
   ECN wred info: enable(False) min\_thresh(0) max\_thresh(0) drop\_probability(0)  
   ECN max queue size in bytes(524288), max queue size in BDs(524288)  
 VSQ-related  
   Category class: 2  
   Traffic class: 0  
   Connection class: 0  
   Green : Admission-test-templates(0)  
   Yellow: Admission-test-templates(0)  
   Red : Admission-test-templates(0)  
   Black : Admission-test-templates(0)  
   AdmissionTestProfileA[0]: 'category' 'category, traffic class' 'category, connection class' 'statistics tag'  
   AdmissionTestProfileB[0]: 'category' 'category, traffic class' 'category, connection class' 'statistics tag'  
 current queue size: 0 bytes, current queue bds size: 0  
 Attached VOQ connector: 0x00000030  
 Destination sys port: 0x00000001  
 OCB eligibility: False

**# port to EGQ mapping; PS refers to Port Scheduler.**

BCM.0> **diag cosq qpair egq map**

EGQ MAPPING:  
\*\*\*\*\*

Port #	Priorities	Base Q-Pair	PS #	Core
0	2	192	24	0
1	2	194	24	0
13	2	196	24	0
14	2	198	24	0
15	2	104	13	0
16	2	106	13	0
17	2	104	13	1
19	2	106	13	1
240	2	240	30	0

240		2		240		30		1	
248		2		248		31		0	
248		2		248		31		1	

For a graphic display of Port Scheduler <i> enter 'diag cosq qpair egq ps=<i>'.

# EGQ mapping of port=1

BCM.0> diag cosq qpair egq ps=1

Graphic representation of port 1 through port scheduler number 24:

\*\*\*\*\*

Port Index 001				Port Index 001			
Q-Pair 194(P0)				Q-Pair 195(P1)			
U   M		U   M					
-----		-----					
\ H L /		\ H L /					
\ -SP- /		\ -SP- /					
-----		-----					
N		N					
N		N					
(0101001094Kbs)				(0101001094Kbs)			
-----		-----					
\ H		L /					
\		/					
SP							
-----		-----					