Troubleshooting Commands

This module describes commands used for troubleshooting routers running Cisco IOS XR software.

The commands in this chapter with the cisco-support task ID are used in the *Cisco IOS XR Troubleshooting Guide for Cisco XR 12000 Series Router* as part of the troubleshooting process. For information about commands with the cisco-support task ID that are not documented in this chapter, please contact Cisco Technical Support.

Caution These Cisco support commands are normally reserved for use by Cisco Technical Support personnel only. There is some risk that they may cause performance or other issues that impact products without proper usage, and we highly recommend that you contact Cisco Technical Support prior to using any of these commands.

show arp trace

To display Address Resolution Protocol (ARP) entries in the buffer, use the **show arp trace** command in EXEC mode.

show arp trace

Syntax Description

This command has no keywords or arguments.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

Use the **show arp trace** command to display ARP entries in the buffer.

Task ID

Task ID	Operations
cisco-support	read

Examples

The following example shows the output of the **show arp trace** command:

```
RP/0/0/CPU0:router# show arp trace events

Tue Nov 10 04:13:22.766 PST

22 unique entries (4096 possible, 54 filtered)
Nov  5 19:48:27.624 ipv4_arp/slow 0/RP0/CPU0 1# t1  ARP-EVENT: Repopulating AIB
Nov  5 19:48:49.768 ipv4_arp/slow 0/RP0/CPU0 1# t1  ARP-DEV-EVENT: Unbinding frs
Nov  5 19:49:01.590 ipv4_arp/slow 0/RP0/CPU0 1# t1  ARP-EVENT: IM ORE received
Nov  5 19:54:12.448 ipv4_arp/slow 0/RP0/CPU0 5# t1  ARP-EVENT: Processing MAC c3
Nov  5 19:54:12.467 ipv4_arp/slow 0/RP0/CPU0 5# t1  ARP-EVENT: Interface attrib2
Nov  5 19:54:12.555 ipv4_arp/fast 0/RP0/CPU0 10# t1  ARP-EVENT: received interf3
Nov  5 19:54:12.595 ipv4_arp/fast 0/RP0/CPU0 5# t1  ARP-EVENT: Copying MAC addr3
Nov  5 19:54:12.614 ipv4_arp/fast 0/RP0/CPU0 6# t1  ARP-EVENT: Received VLAN ID)
Nov  5 19:54:12.614 ipv4_arp/fast 0/RP0/CPU0 3# t1  ARP-EVENT: Processing VLAN )
Nov  5 19:54:15.434 ipv4_arp/slow 0/RP0/CPU0 5# t1  ARP-EVENT: Interface not up0
Nov  5 19:54:15.437 ipv4_arp/slow 0/RP0/CPU0 1# t3  ARP-EVENT: IMP caps add suc0
Nov  5 19:54:15.581 ipv4_arp/fast 0/RP0/CPU0 5# t1  ARP-EVENT: Completing IDB i0
Nov  5 19:54:15.673 ipv4_arp/slow 0/RP0/CPU0 1# t1  ARP-EVENT: interface_entry 0
Nov  5 19:54:15.793 ipv4_arp/pkt 0/RP0/CPU0 1# t1  ARP-EVENT: Discarding arp pa2
Nov  5 19:57:22.531 ipv4_arp/fast 0/RP0/CPU0 4# t1  ARP-EVENT: received DPC for1
Nov  5 21:30:08.234 ipv4_arp/slow 0/RP0/CPU0 9# t1  ARP-EVENT: clearing ARP AIB1
```

```
Nov  5 21:46:04.169 ipv4_arp/slow 0/RP0/CPU0 18# t1  ARP-EVENT: updating arp-id)
Nov  5 21:46:04.169 ipv4_arp/slow 0/RP0/CPU0 9# t1  ARP-EVENT: adding ARP AIB e1
Nov  5 21:46:04.316 ipv4_arp/fast 0/RP0/CPU0 18# t1  ARP-EVENT: Interface Bundlp
Nov  6 17:03:53.443 ipv4_arp/pkt 0/RP0/CPU0 3# t1  PROBE: Timer expired on Mgmt1
Nov  6 17:04:23.052 ipv4_arp/pkt 0/RP0/CPU0 3# t1  PROBE: MgmtEth0/RP0/CPU0/0 eE
Nov  6 17:23:16.156 ipv4_arp/slow 0/RP0/CPU0 46# t1  ARP-EVENT: updated aib ent0
160 wrapping entries (4096 possible, 805 filtered, 965 total)
Nov  5 19:48:27.771 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Repopulating AIB
Nov  5 19:48:49.915 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-DEV-EVENT: Unbinding from s
Nov  5 19:49:01.737 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: IM ORE received
Nov  5 19:49:01.761 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Copying MAC address0
Nov  5 19:49:01.761 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Interface attribute2
Nov  5 19:49:01.761 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Processing MAC chan6
Nov  5 19:49:01.769 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received interface 0
Nov  5 19:49:01.769 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received interface 0
Nov  5 19:54:12.258 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received interface 8
Nov  5 19:54:12.258 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received interface 8
Nov  5 19:54:12.294 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Copying MAC address8
Nov  5 19:54:12.294 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Interface attribute2
Nov  5 19:54:12.294 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Processing MAC chan3
Nov  5 19:54:12.555 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received interface 1
Nov  5 19:54:12.555 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received interface 2
Nov  5 19:54:12.555 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received interface 3
Nov  5 19:54:12.555 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received interface 1
Nov  5 19:54:12.555 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received interface 2
Nov  5 19:54:12.555 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received interface 3
Nov  5 19:54:12.595 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Copying MAC address1
Nov  5 19:54:12.595 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Copying MAC address2
Nov  5 19:54:12.595 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Copying MAC address3
Nov  5 19:54:12.595 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Received VLAN ID no)
Nov  5 19:54:12.595 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Received VLAN ID no)
Nov  5 19:54:12.595 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Received VLAN ID no)
Nov  5 19:54:12.595 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Interface attribute2
Nov  5 19:54:12.595 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Processing MAC chan3
Nov  5 19:54:12.595 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Processing MAC chan3
Nov  5 19:54:12.595 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Processing MAC chan3
Nov  5 19:54:12.595 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Interface attribute2
Nov  5 19:54:12.614 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Received VLAN ID no)
Nov  5 19:54:12.614 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Processing VLAN ID )
Nov  5 19:54:12.614 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Received VLAN ID no)
Nov  5 19:54:12.614 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Processing VLAN ID )
Nov  5 19:54:12.614 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Received VLAN ID no)
Nov  5 19:54:12.614 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Processing VLAN ID )
Nov  5 19:54:12.614 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Interface attribute2
Nov  5 19:54:12.692 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etn
Nov  5 19:54:12.692 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etn
Nov  5 19:54:12.692 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etn
Nov  5 19:54:12.692 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etn
Nov  5 19:54:12.692 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 19:54:12.692 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: clearing ARP AIB en8
Nov  5 19:54:12.692 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 19:54:12.692 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: clearing ARP AIB en1
Nov  5 19:54:12.692 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 19:54:12.692 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: clearing ARP AIB en2
Nov  5 19:54:12.692 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 19:54:12.692 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: clearing ARP AIB en3
Nov  5 19:54:12.749 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Completing IDB ifh:8
Nov  5 19:54:12.749 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Completing IDB ifh:1
Nov  5 19:54:12.749 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Completing IDB ifh:2
Nov  5 19:54:12.749 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Completing IDB ifh:3
Nov  5 19:54:12.749 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Interface not up ca8
Nov  5 19:54:12.749 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Interface not up ca1
Nov  5 19:54:12.749 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Interface not up ca2
Nov  5 19:54:12.749 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Interface not up ca3
Nov  5 19:54:15.567 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface MgmtEth0/n
Nov  5 19:54:15.567 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 19:54:15.567 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: clearing ARP AIB en0
Nov  5 19:54:15.581 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Completing IDB ifh:0
Nov  5 19:54:15.581 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: Interface not up ca0
Nov  5 19:54:15.584 ipv4_arp/slow 0/RP0/CPU0 t3  ARP-EVENT: IMP caps add succee0
Nov  5 19:54:15.793 ipv4_arp/pkt 0/RP0/CPU0 t1  ARP-EVENT: Discarding arp packe2
Nov  5 19:54:15.819 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface MgmtEth0/p
Nov  5 19:54:15.819 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
```

```
Nov  5 19:54:15.819 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: adding ARP AIB entr0
Nov  5 19:54:15.820 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 19:54:15.820 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: interface_entry (170
Nov  5 19:57:21.623 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 19:57:22.463 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received DPC for if8
Nov  5 19:57:22.531 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received DPC for if3
Nov  5 19:57:22.531 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received DPC for if2
Nov  5 19:57:22.531 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: received DPC for if1
Nov  5 19:57:29.136 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:27:42.950 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:27:42.969 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:27:43.202 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:27:54.590 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:30:38.679 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:30:38.943 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:30:45.788 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:30:46.342 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:30:46.458 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:32:57.516 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:33:38.988 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etp
Nov  5 20:33:38.988 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 20:33:38.988 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: adding ARP AIB entr8
Nov  5 20:33:39.065 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etp
Nov  5 20:33:39.065 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etp
Nov  5 20:33:39.065 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etp
Nov  5 20:33:39.065 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 20:33:39.065 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: adding ARP AIB entr3
Nov  5 20:33:39.065 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 20:33:39.065 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: adding ARP AIB entr2
Nov  5 20:33:39.065 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 20:33:39.065 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: adding ARP AIB entr1
Nov  5 20:41:37.128 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 20:41:37.144 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 21:23:17.059 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 21:23:18.347 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 21:26:41.271 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 21:30:08.361 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etn
Nov  5 21:30:08.361 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 21:30:08.361 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: clearing ARP AIB en8
Nov  5 21:30:08.367 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etn
Nov  5 21:30:08.367 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 21:30:08.367 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: clearing ARP AIB en3
Nov  5 21:30:08.373 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etn
Nov  5 21:30:08.373 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 21:30:08.373 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: clearing ARP AIB en2
Nov  5 21:30:08.381 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etn
Nov  5 21:30:08.381 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 21:30:08.381 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: clearing ARP AIB en1
Nov  5 21:46:04.302 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etp
Nov  5 21:46:04.302 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 21:46:04.302 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: adding ARP AIB entr8
Nov  5 21:46:04.316 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etp
Nov  5 21:46:04.316 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etp
Nov  5 21:46:04.316 ipv4_arp/fast 0/RP0/CPU0 t1  ARP-EVENT: Interface Bundle-Etp
Nov  5 21:46:04.316 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 21:46:04.316 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: adding ARP AIB entr3
Nov  5 21:46:04.316 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 21:46:04.316 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: adding ARP AIB entr2
Nov  5 21:46:04.316 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updating arp-idb ip)
Nov  5 21:46:04.316 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: adding ARP AIB entr1
Nov  5 22:39:30.728 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:32:03.427 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:32:03.625 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:33:37.230 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:33:37.765 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:35:13.706 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:35:45.392 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:43:24.043 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:45:39.659 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:56:36.519 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:56:47.521 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:56:54.402 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:57:12.595 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
```

```
Nov  5 23:57:22.204 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  5 23:57:23.449 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 00:10:29.938 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 00:15:14.864 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 00:20:46.274 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 00:22:13.307 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 00:24:17.723 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 00:25:17.797 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 02:33:04.239 ipv4_arp/pkt 0/RP0/CPU0 t1  PROBE: Timer expired on MgmtEth1
Nov  6 02:33:30.807 ipv4_arp/pkt 0/RP0/CPU0 t1  PROBE: MgmtEth0/RP0/CPU0/0 exceE
Nov  6 12:23:26.295 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 13:16:12.876 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 13:16:13.026 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 13:17:37.082 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 13:17:37.130 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 14:54:55.415 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
Nov  6 16:12:07.269 ipv4_arp/pkt 0/RP0/CPU0 t1  PROBE: Timer expired on MgmtEth1
Nov  6 16:12:35.727 ipv4_arp/pkt 0/RP0/CPU0 t1  PROBE: MgmtEth0/RP0/CPU0/0 exceE
Nov  6 17:03:53.443 ipv4_arp/pkt 0/RP0/CPU0 t1  PROBE: Timer expired on MgmtEth1
Nov  6 17:04:23.052 ipv4_arp/pkt 0/RP0/CPU0 t1  PROBE: MgmtEth0/RP0/CPU0/0 exceE
Nov  6 17:23:16.303 ipv4_arp/slow 0/RP0/CPU0 t1  ARP-EVENT: updated aib entry (0
```

Related Commands

Command	Description
show arp	Displays the ARP.

show captured packets

To display information on packets that are switched and punted in the software, use the **show captured packets** command in EXEC mode.

show captured packets {**ingress**| **egress**} [**interface** *type interface-path-id*] **[hexdump]** [**last** *number*] **[single-line] location** *node-id*

Syntax Description

ingress	Specifies ingress dropped packets.
egress	Specifies egress dropped packets.
interface	(Optional) Specifies an interface.
type	Interface type. For more information, use the question mark (**?**) online help function.
interface-path-id	Physical interface or virtual interface. **Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (**?**) online help function.
hexdump	(Optional) Displays the packet contents in hex.
last *number*	(Optional) Specifies the last number of packets in the queue to display.
single-line	(Optional) Displays a one-line summary of the captured packets to facilitate the use of the include and exclude operators.
location *node-id*	Displays packet information for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

Use the **show captured packets** command to display information on packets that are switched and punted in the software.

The **capture software packets** command must be enabled at the interface level to use this command.

Task ID

Task ID	Operations
cisco-support	read

Examples

The following example shows the output of the **show captured packets** command:

```
RP/0/0/CPU0:router# show captured packets ingress interface tengige0/0/0/3 location
0/0/CPU0

-------------------------------------------------------
packets captured on interface in ingress direction buffer overflow pkt drops:0, current:
6, non wrapping: 0 maximum: 200
-------------------------------------------------------
           Wrapping entries
-------------------------------------------------------
[1] Mar 22 16:30:43.797, len: 114, hits: 1, i/p i/f: TenGigE0/0/0/3
    [punt reason: IFIB]
    [ether dst: 0015.fa99.590b src: 0010.a4e6.22fc type/len: 0x800]
    [IPV4: source 172.18.2.2, dest 172.18.2.1 ihl 5, ver 4, tos 0
     id 22556, len 100, prot 1, ttl 64, sum c655, offset 0]
    00008612 51010000 abcdabcd abcdabcd abcdabcd abcdabcd abcdabcd abcdabcd
    abcdabcd abcdabcd abcdabcd abcd
```

This table describes the significant fields shown in the display.

Table 1: show captured packets Field Descriptions

Field	Description
punt reason: IFIB	Packet was switched in the software due to the Internal Forwarding Information Base (IFIB) entry.
ether	Source, destination, and type or length values in the Ethernet header.
IPV4	Depending on the type of packet, the layer 3 packet header follows.

show cfgmgr trace

To display trace information for the configuration manager (CFGMGR), use the **show cfgmgr trace** command in EXEC mode.

show cfgmgr trace

Syntax Description

This command has no keywords or arguments.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

Use the **show cfgmgr trace** command to display cfgmgr trace information.

The following lines of the **show cfgmgr trace** command output indicate that the startup configuration has started and that it has completed on the active RP:

```
Feb  6 21:28:37.145 /ltrace/cfgmgr/common 0/RP0/CPU0 t5  Startup confi
g apply requested with option '0x1'
Feb  6 21:31:30.874 /ltrace/cfgmgr/common 0/RP0/CPU0 t7  Startup confi
g done (and infra band already ready)
```

Note These traces are not present if the original active RP has ever reloaded (for example, if there have been any RP switchover events since the system first booted).

Task ID

Task ID	Operations
cisco-support	read

Examples

The following example shows the output of the **show cfgmgr trace** command:

```
RP/0/0/CPU0:router#show cfgmgr trace

130 wrapping entries (2048 possible, 0 filtered, 130 total)
Apr 23 21:15:58.587 cfgmgr/common 0/RP0/CPU0 t5  Req '4': Save interface config]
```

```
Apr 23 21:15:58.707 cfgmgr/common 0/RP0/CPU0 t5  Req '4': Save node specific co]
Apr 23 21:15:59.000 cfgmgr/common 0/RP0/CPU0 t5  OIR announcement made for 'nod'
Apr 23 21:17:40.975 cfgmgr/common 0/RP0/CPU0 t5  The request queue IS NOT curred
Apr 23 21:17:40.975 cfgmgr/common 0/RP0/CPU0 t5  Process OIR save request.
Apr 23 21:17:41.040 cfgmgr/common 0/RP0/CPU0 t5  Validating 'LR' configuration ]
Apr 23 21:17:41.055 cfgmgr/common 0/RP0/CPU0 t5  Validating 'admin' configurati]
Apr 23 21:17:41.304 cfgmgr/common 0/RP0/CPU0 t5  Req '5': Save interface config]
Apr 23 21:17:41.349 cfgmgr/common 0/RP0/CPU0 t5  Req '5': Save interface config]
Apr 23 21:17:41.995 cfgmgr/common 0/RP0/CPU0 t5  Req '5': Save interface config]
Apr 23 21:17:42.041 cfgmgr/common 0/RP0/CPU0 t5  Req '5': Save interface config]
Apr 23 21:17:42.254 cfgmgr/common 0/RP0/CPU0 t5  Req '5': Save interface config]
Apr 23 21:17:42.356 cfgmgr/common 0/RP0/CPU0 t5  Req '5': Save node specific co]
Apr 23 21:17:42.580 cfgmgr/common 0/RP0/CPU0 t5  OIR announcement made for 'nod'
Apr 25 15:26:49.372 cfgmgr/common 0/RP0/CPU0 t1  Config media returned from dis.
Apr 25 18:15:06.142 cfgmgr/common 0/RP0/CPU0 t1  Config media returned from dis.
Apr 26 03:35:10.170 cfgmgr/common 0/RP0/CPU0 t1  Config media returned from dis.
Apr 26 05:54:37.528 cfgmgr/common 0/RP0/CPU0 t1  Config media returned from dis.
Apr 26 06:18:47.118 cfgmgr/common 0/RP0/CPU0 t1  Config media returned from dis.
Apr 26 09:07:01.662 cfgmgr/common 0/RP0/CPU0 t1  Config media returned from dis.
Apr 26 09:28:22.311 cfgmgr/common 0/RP0/CPU0 t1  Config media returned from dis.
Apr 26 11:56:55.677 cfgmgr/common 0/RP0/CPU0 t1  Config media returned from dis.
```

Related Commands

Command	Description
show cfgmgr commitdb	Displays the contents of the commit database for the configuration manager.

show im database

To display the information stored in the shared memory database of interface manager (IM), use the **show im database** command in EXEC mode.

show im database [**brief**| **detail**| **ifhandle**| **interface**| **summary**| **verbose**| **view**] *interface-type interface-instance* **location** *node-id*

Syntax Description

brief	(Optional) Displays brief information about IM database.
detail	(Optional) Displays detailed information about IM database.
ifhandle	(Optional) Select a specific interface by handle.
interface	(Optional) Select a specific interface by name.
summary	(Optional) Displays IM database summary information.
verbose	(Optional) Displays verbose information about IM database.
view	(Optional) Specify a database view to filter the information based on the view
interface-type	Interface type. For more information, use the question mark (?) online help function.
interface-instance	Either a physical interface instance or a virtual interface instance as follows: • Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation. ◦ *rack*: Chassis number of the rack. ◦ *slot*: Physical slot number of the modular services card or line card. ◦ *module*: Module number. A physical layer interface module (PLIM) is always 0. ◦ *port*: Physical port number of the interface. **Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RP0 or RP1) and the module is CPU0. Example: interface MgmtEth0/RP1/CPU0/0. • Virtual interface instance. Number range varies depending on interface type. For more information about the syntax for the router, use the question mark (?) online help function.
location *node-id*	Displays IM database information for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.8.0	This command was introduced.

Usage Guidelines

Task ID

Task ID	Operations
cisco-support	read
interface	read

Examples

The following example shows the output of the **show im database** command:

```
RP/0/0/CPU0:router# show im database verbose interface null 0
Mon Nov  9 22:10:37.964 PST

View: OWN - Owner, L3P - Local 3rd Party, G3P - Global 3rd Party,
      LDP - Local Data Plane, GDP - Global Data Plane, RED - Redundancy

Node 0/RP0/CPU0 (0x201)

Interface Null0, ifh 0x00080030 (up, 1500)
  Interface flags:          0x00010097 (IFINDEX|VIRTUAL|CONFIG|VIS|DATA|CONTRO
  Encapsulation:            null
  Interface type:           IFT_NULL
  Views:                    GDP|LDP|G3P|L3P|OWN
  Control location:         0/RP0/CPU0
  Owner Private:            92 bytes
    Flags:                  <none>
    State Transitions:      1
    Dampening Config:       NO
    Shared Locks:           0
    MTU default             1500
    MTU ovh for bc/subif:   0/0
    MTU min/max:            0/0
    MTU avail/child:        0/1500
    MTU actual/notified:    1500/1500
    State (constraint):     UP (UP)
    Callback:               OWN GROUP OWNER - ID 17[-]
    Ctrl Flags:             CFG_RDY|RDY|DNLD|INTF
  Instance ID:              31
  Checkpoint:               48 bytes
  Resource in NetIO:        TRUE

  Protocol        Caps (state, mtu)
  --------        -----------------
  None            null (up, 1500)
    Views:                  LDP|G3P|L3P|OWN
```

```
    Owner Private:          92 bytes
      Flags:                <none>
      MTU min/max:          0/0
      MTU avail/child:      1500/1500
      MTU actual/notified:  1500/1500
      State (constraint):   UP (UP)
      Callback:             OWN GROUP OWNER - ID 17[-]
      Ctrl Flags:           CFG_RDY|RDY|DNLD
    Instance ID:            31
    Checkpoint:             20 bytes
    Resource in NetIO:      TRUE
    Demux limit:            0x00000000
```

This table describes the significant fields shown in the display.

Table 2: show im database Field Descriptions

Field	Description
nodeid	Identifier associated with the node.
Interface	Interface name.
Protocol	Protocol capsulations associated with the interface.
Caps (state, mtu)	Capsulation names with associated state and MTU values.

The following example shows the output of the **show im database** command:

```
RP/0/0/CPU0:router# show im database brief location 0/0/CPU0

View: OWN - Owner, L3P - Local 3rd Party, G3P - Global 3rd Party,
      LDP - Local Data Plane, GDP - Global Data Plane, RED - Redundancy

Node 0/0/CPU0 (0x1)

 Handle   | Name                  | State    | MTU  |#P|#C| Views              |
----------|-----------------------|----------|------|--|--|--------------------|
0x01080020 FI0/0/CPU0              up          8000 11 12 GDP|LDP|L3P|OWN
0x01080060 Gi0/0/0/0               up          9212  3  3 GDP|LDP|L3P|OWN
0x01080080 Gi0/0/0/1               up          1514  3  3 GDP|LDP|L3P|OWN
0x010800a0 Gi0/0/0/2               up          1514  3  3 GDP|LDP|L3P|OWN
0x010800c0 Gi0/0/0/3               down        1514  4  4 GDP|LDP|L3P|OWN
0x010800e0 Gi0/0/0/4               up          1514  3  3 GDP|LDP|L3P|OWN
0x01080100 Gi0/0/0/5               up          1514  3  3 GDP|LDP|L3P|OWN
0x01080120 Gi0/0/0/6               up          1514  8 17 GDP|LDP|L3P|OWN
0x01080140 Gi0/0/0/7               down        1514  6  9 GDP|LDP|L3P|OWN
0x010801c0 Gi0/0/0/6.1             up          1518  4  5 GDP|LDP|L3P|OWN
0x010801e0 Gi0/0/0/6.101           up          1518  5 13 GDP|LDP|L3P|OWN
0x01080200 Gi0/0/0/6.102           up          1518  5 13 GDP|LDP|L3P|OWN
0x01080220 Gi0/0/0/6.103           up          1518  5 13 GDP|LDP|L3P|OWN
0x01080240 Gi0/0/0/6.104           up          1518  5 13 GDP|LDP|L3P|OWN
0x01080260 Gi0/0/0/6.105           up          1518  4 12 GDP|LDP|L3P|OWN
0x01080280 Gi0/0/0/6.106           up          1518  4 12 GDP|LDP|L3P|OWN
0x010802a0 Gi0/0/0/6.107           up          1518  4 12 GDP|LDP|L3P|OWN
0x010802c0 Gi0/0/0/6.108           up          1518  4 10 GDP|LDP|L3P|OWN
0x010802e0 Gi0/0/0/6.109           up          1518  4 10 GDP|LDP|L3P|OWN
0x01080300 Gi0/0/0/6.110           up          1518  4 10 GDP|LDP|L3P|OWN
0x01080320 Gi0/0/0/6.111           up          1518  4 10 GDP|LDP|L3P|OWN
0x01080340 Gi0/0/0/6.112           up          1518  4 10 GDP|LDP|L3P|OWN
0x01080360 Gi0/0/0/6.113           up          1518  4 10 GDP|LDP|L3P|OWN
0x01080380 Gi0/0/0/6.114           up          1518  4 10 GDP|LDP|L3P|OWN
0x010803a0 Gi0/0/0/6.115           up          1518  4 10 GDP|LDP|L3P|OWN
```

```
0x010803c0 Gi0/0/0/6.116          up          1518  4 10 GDP|LDP|L3P|OWN
0x010803e0 Gi0/0/0/6.117          up          1518  4 10 GDP|LDP|L3P|OWN
0x01080400 Gi0/0/0/6.118          up          1518  4 10 GDP|LDP|L3P|OWN
0x01080420 Gi0/0/0/6.119          up          1518  4 10 GDP|LDP|L3P|OWN
0x01080440 Gi0/0/0/6.120          up          1518  4 10 GDP|LDP|L3P|OWN
0x01080460 Gi0/0/0/6.121          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080480 Gi0/0/0/6.122          up          1518  4  6 GDP|LDP|L3P|OWN
0x010804a0 Gi0/0/0/6.123          up          1518  4  6 GDP|LDP|L3P|OWN
0x010804c0 Gi0/0/0/6.124          up          1518  4  6 GDP|LDP|L3P|OWN
0x010804e0 Gi0/0/0/6.125          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080500 Gi0/0/0/6.126          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080520 Gi0/0/0/6.127          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080540 Gi0/0/0/6.128          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080560 Gi0/0/0/6.129          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080580 Gi0/0/0/6.130          up          1518  4  6 GDP|LDP|L3P|OWN
0x010805a0 Gi0/0/0/6.131          up          1518  4  6 GDP|LDP|L3P|OWN
0x010805c0 Gi0/0/0/6.132          up          1518  4  6 GDP|LDP|L3P|OWN
0x010805e0 Gi0/0/0/6.133          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080600 Gi0/0/0/6.134          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080620 Gi0/0/0/6.135          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080640 Gi0/0/0/6.136          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080660 Gi0/0/0/6.137          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080680 Gi0/0/0/6.138          up          1518  4  6 GDP|LDP|L3P|OWN
0x010806a0 Gi0/0/0/6.139          up          1518  4  6 GDP|LDP|L3P|OWN
0x010806c0 Gi0/0/0/6.140          up          1518  4  6 GDP|LDP|L3P|OWN
0x010806e0 Gi0/0/0/6.141          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080700 Gi0/0/0/6.142          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080720 Gi0/0/0/6.143          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080740 Gi0/0/0/6.144          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080760 Gi0/0/0/6.145          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080780 Gi0/0/0/6.146          up          1518  4  6 GDP|LDP|L3P|OWN
0x010807a0 Gi0/0/0/6.147          up          1518  4  6 GDP|LDP|L3P|OWN
0x010807c0 Gi0/0/0/6.148          up          1518  4  6 GDP|LDP|L3P|OWN
0x010807e0 Gi0/0/0/6.149          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080800 Gi0/0/0/6.150          up          1518  4  6 GDP|LDP|L3P|OWN
0x01080820 Gi0/0/0/7.1            down        1518  2  5 GDP|LDP|L3P|OWN
0x01080840 Gi0/0/0/7.2            down        1518  4  6 GDP|LDP|L3P|OWN
0x01080860 Gi0/0/0/7.3            down        1518  3  4 GDP|LDP|L3P|OWN
0x01080880 Gi0/0/0/7.4            down        1518  3  4 GDP|LDP|L3P|OWN
0x010808a0 Gi0/0/0/7.5            down        1518  3  4 GDP|LDP|L3P|OWN
0x010808c0 Gi0/0/0/7.6            down        1518  3  4 GDP|LDP|L3P|OWN
0x010808e0 Gi0/0/0/7.7            down        1518  3  4 GDP|LDP|L3P|OWN
0x01080900 Gi0/0/0/7.8            down        1518  3  4 GDP|LDP|L3P|OWN
0x01080920 Gi0/0/0/7.9            down        1518  3  4 GDP|LDP|L3P|OWN
0x01080940 Gi0/0/0/7.10           down        1518  3  4 GDP|LDP|L3P|OWN
0x01080960 Gi0/0/0/7.11           down        1518  3  4 GDP|LDP|L3P|OWN
0x01100020 Mg0/1/CPU1/0           N/A            -  0  0 GDP
0x01100040 FI0/1/CPU1             N/A            -  0  0 GDP
0x01180020 FI0/1/CPU0             N/A            -  0  0 GDP
0x01180040 Mg0/1/CPU0/0           N/A            -  0  0 GDP
0x01180030 Nu0                    N/A            -  0  0 GDP
0x01180050 En0                    N/A            -  2  2 GDP|LDP
0x01180070 En6tunnel0             N/A            -  2  2 GDP|LDP
0x01180090 Lo0                    N/A            -  0  0 GDP
0x011800b0 Lo1                    N/A            -  0  0 GDP
0x011800d0 Lo2                    N/A            -  0  0 GDP
0x011800f0 Lo3                    N/A            -  0  0 GDP
0x01180110 Lo5                    N/A            -  0  0 GDP
0x01180130 Lo6                    N/A            -  0  0 GDP
0x01180150 Lo7                    N/A            -  0  0 GDP
0x01180170 BE102                  N/A            -  0  0 GDP
0x01180190 BE1080                 N/A            -  3  4 GDP|LDP
0x011801b0 BE1083                 N/A            -  3  4 GDP|LDP
0x011801d0 BE1084                 N/A            -  3  4 GDP|LDP
0x011801f0 BE1085                 N/A            -  5 12 GDP|LDP
0x01180210 BE1085.1               N/A            -  4  6 GDP|LDP
0x01180230 BE1085.102             N/A            -  4  7 GDP|LDP
```

show imds interface brief

To display interface information for the interface manager distribution server (IMDS), use the **show imds interface brief** command in EXEC mode.

show imds interface brief

Syntax Description

This command has no keywords or arguments.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

Use the **show imds interface brief**command to display IMDS interface information. Use the command output to determine if the state, encapsulation being used, maximum transmission unit (MTU), and interface handle (ifhandle) for each interface are as expected.

Task ID

Task ID	Operations
cisco-support	read

Examples

The following example shows the output of the **show imds interface brief**command:

```
RP/0/0/CPU0:router show imds interface brief

IMDS BRIEF INTERFACE DATA (Node 0x201)
  handle            name           flags     state      mtu        encap
---------- ----------------------- ------ ----------- -------- ---------------
0x00080000 FINT0/RP0/CPU0          0x0007 up          8000     91 (fint_base)
0x00080010 Null0                   0x100ab up          1500     17 (null)
0x00080020 MgmtEth0/RP0/CPU0/0     0x1002f up          1514     30 (ether)
0x00080030 Loopback0               0x100ab up          1514     16 (loopback)
0x00080050 Bundle-POS24            0x104ab up          4474     14 (hdlc)
0x00080070 Bundle-Ether28          0x104ab up          1514     30 (ether)
0x00080090 Bundle-Ether28.1        0x10cab up          1500     107 (dot1q)
0x000800b0 Bundle-Ether28.2        0x10cab up          1500     107 (dot1q)
0x000800d0 Bundle-Ether28.3        0x10cab up          1500     107 (dot1q)
0x01180000 FINT0/1/CPU0            0x0007 up          8000     91 (fint_base)
0x01180020 GigabitEthernet0/1/5/0  0x1002f up          1514     30 (ether)
0x01180040 GigabitEthernet0/1/5/1  0x1002f up          1514     30 (ether)
0x01180060 GigabitEthernet0/1/5/2  0x1002f up          1514     30 (ether)
```

```
0x01180080 GigabitEthernet0/1/5/3  0x1002f admin-down  1514     30 (ether)
0x011800a0 GigabitEthernet0/1/5/4  0x1002f down        1514     30 (ether)
0x011800c0 GigabitEthernet0/1/5/5  0x1002f admin-down  1514     30 (ether)
.
.
.
0x01680480 SONET0/6/4/5            0x1006d up          10000    0 (Unknown)
0x016804a0 SonetPath0/6/4/5        0x10005 up          10000    0 (Unknown)
0x016804c0 POS0/6/4/5              0x1002f up          4474     14 (hdlc)
0x016804e0 SONET0/6/4/6            0x1006d up          10000    0 (Unknown)
0x01680500 SonetPath0/6/4/6        0x10005 up          10000    0 (Unknown)
0x01680520 POS0/6/4/6              0x1002f up          4474     14 (hdlc)
0x01680540 SONET0/6/4/7            0x1006d up          10000    0 (Unknown)
0x01680560 SonetPath0/6/4/7        0x10005 down        10000    0 (Unknown)
0x01680580 POS0/6/4/7              0x1002f admin-down  4474     14 (hdlc)
```

This table describes the significant fields shown in the display.

Table 3: show imds interface brief Field Descriptions

Field	Description
name	Interface name.
state	Interface state.
mtu	MTU associated with the interface.
encap	Base encapsulation associated with the interface.

show netio chains

To display Network Input and Output (Netio) chains information for an interface, use the **show netio chains** command in EXEC mode.

show netio chains *interface-type interface-instance* [**location** *node-id*]

Syntax Description

interface-type	Interface type. For more information, use the question mark (?) online help function.
interface-instance	Either a physical interface instance or a virtual interface instance as follows: • Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation. ◦ *rack*: Chassis number of the rack. ◦ *slot*: Physical slot number of the modular services card or line card. ◦ *module*: Module number. A physical layer interface module (PLIM) is always 0. ◦ *port*: Physical port number of the interface. **Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RP0 or RP1) and the module is CPU0. Example: interface MgmtEth0/RP1/CPU0/0. • Virtual interface instance. Number range varies depending on interface type. For more information about the syntax for the router, use the question mark (?) online help function.
location *node-id*	(Optional) Displays Netio chains information for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
Release 3.8.0	This command was introduced.
Release 3.9.0	No modifications.
Release 4.0.0	No modifications.

Usage Guidelines

Task ID

Task ID	Operation
cisco-support	read

Examples

The following example shows the output of the **show netio chains** command:

```
RP/0/0/CPU0:router# show netio chains gigabitEthernet 0/4/0/1

GigabitEthernet0/4/0/1 (handle: 0x05000500, nodeid 0x40) netio chains:
---------------------------------
Base decap chain:
     ether_shim           <130>  <0x79d99950, 0x0807bc84>  <       0,        0>
     ether                <30>  <0x79d7eb14, 0x08079318>  <       0,        0>


Protocol chains:
---------------
<Protocol number> (name)  Stats
  Type  Chain_node       <caps num>  <function, context> <drop pkts, drop bytes>
<7> (arp)   Stats IN: 279 pkts, 16740 bytes; OUT: 279 pkts, 11718 bytes
    Encap:
     ether_shim           <130>  <0x79d99858, 0x081c649c>  <       0,        0>
     l2_adj_rewrite       <86>  <0x7952437c, 0x081c5e4c>  <       0,        0>
     txm_nopull           <60>  <0x79516cd0, 0x0817cbd8>  <       0,        0>
    Decap:
     arp                  <24>  <0x79a9ba14, 0x00000000>  <       0,        0>
    Fixup:
     l2_adj_rewrite       <86>  <0x795236c0, 0x081c5eb8>  <       0,        0>
     txm_nopull           <60>  <0x79516cd0, 0x0817cbd8>  <       0,        0>
<12> (ipv4)   Stats IN: 0 pkts, 0 bytes; OUT: 48 pkts, 9578 bytes
    Encap:
     ipv4                 <26>  <0x79aa2004, 0x0816c204>  <       0,        0>
     ether                <30>  <0x79d7f634, 0x08079318>  <       0,        0>
     ether_shim           <130>  <0x79d99858, 0x081c0ebc>  <       0,        0>
     l2_adj_rewrite       <86>  <0x7952437c, 0x081c280c>  <       0,        0>
     txm_nopull           <60>  <0x79516cd0, 0x0817cbd8>  <       0,        0>
    Decap:
     ipv4                 <26>  <0x79aa2054, 0x00000000>  <       0,        0>
    Fixup:
     l2_adj_rewrite       <86>  <0x795236c0, 0x081c2878>  <       0,        0>
     txm_nopull           <60>  <0x79516cd0, 0x0817cbd8>  <       0,        0>
<13> (mpls)   Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
    Encap:
     mpls                 <25>  <0x79bd5f7c, 0x00000000>  <       0,        0>
     ether                <30>  <0x79d7f634, 0x08079318>  <       0,        0>
     ether_shim           <130>  <0x79d99858, 0x081cf838>  <       0,        0>
     l2_adj_rewrite       <86>  <0x7952437c, 0x081cf52c>  <       0,        0>
     txm_nopull           <60>  <0x79516cd0, 0x0817cbd8>  <       0,        0>
    Decap:
     mpls                 <25>  <0x79bd3130, 0x00000000>  <       0,        0>
    Fixup:
     l2_adj_rewrite       <86>  <0x795236c0, 0x081cf598>  <       0,        0>
     txm_nopull           <60>  <0x79516cd0, 0x0817cbd8>  <       0,        0>
<22> (ether_sock)   Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
    Encap:
     ether_sock           <98>  <0x79d80aac, 0x08079318>  <       0,        0>
     ether_shim           <130>  <0x79d99858, 0x0807bcfc>  <       0,        0>
     l2_adj_rewrite       <86>  <0x7952437c, 0x0807b9a4>  <       0,        0>
     txm_nopull           <60>  <0x79516cd0, 0x0817cbd8>  <       0,        0>
    Decap:
```

```
     ether_sock           <98>  <0x79d80ca8, 0x08079318>  <       0,        0>
    Fixup:
     l2_adj_rewrite       <86>  <0x795236c0, 0x0807ba10>  <       0,        0>
     txm_nopull           <60>  <0x79516cd0, 0x0817cbd8>  <       0,        0>


Protocol SAFI counts:
--------------------

       Protocol        SAFI       Pkts In    Bytes In    Pkts Out   Bytes Out
---------------  ----------    ----------  ----------  ----------  ----------
           ipv4     Unicast      24330016      233944        8412          41
           ipv4   Multicast          3240          60           0           0
           ipv4   Broadcast             0           0           0           0
           ipv6     Unicast             0           0           0           0
           ipv6   Multicast             0           0           0           0


Node drop accounting:
---------------------
No drops
```

Related Commands

Command	Description
show netio clients	Displays Netio clients information.
show netio db	Displays Netio database information.
show netio idb	Displays Netio IDB information.
show netio media registrations	Displays protocol registrations for media changes.
show netio subblock	Displays Netio subblock information.
show netio trace	Displays Netio trace data.

show netio clients

To display Network Input and Output (Netio) clients information, use the **show netio clients** command in EXEC mode.

show netio clients [**location** *node-id*]

Syntax Description

location *node-id*	(Optional) Displays Netio clients information for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
Release 3.8.0	This command was introduced.
Release 3.9.0	No modifications.
Release 4.0.0	No modifications.

Usage Guidelines

Task ID

Task ID	Operation
cisco-support	read

Examples

The following example shows the output of the **show netio clients** command:

```
RP/0/0/CPU0:router# show netio clients location 0/3/2

XIPC: OutputQ [0:0]/[6000] HighOutputQ [0:18]/[2000] PuntbackQ [0:0]/[6000]
XIPC drops/total: OutputQ: 0/0 HighOutputQ: 0/15682677 PuntbackQ: 0/0
Counters (error/total): Output (0/15682677) Puntback (0/0) Jump (0/0)

                   Input              Punt         XIPC InputQ      XIPC PuntQ
ClientID         Drop/Total        Drop/Total      Cur/High/Max    Cur/High/Max
--------------------------------------------------------------------------------
ipv6_icmp           0/0               0/0             0/0/1000        0/0/1000
icmp                0/0               0/0             0/0/1000        0/0/1000
clns                0/0               0/0           L 0/0/1000        0/0/0
                                                    H 0/0/1000
chdlc_socket        0/802651          0/0             0/2/1000        0/0/0
```

```
fr_socket           0/4454002         0/0             0/6/2000        0/0/0
pre_route           0/0               0/0             0/0/1024        0/0/1024
ipv6_io             0/0               0/0             0/0/1000        0/0/1000
ipv6_nd             0/0               0/0             0/0/1000        0/0/1000
l2snoop             0/0               0/0             0/0/1000        0/0/0
icmpv6_unreach_jump        0/0                0/0
arp                 0/0               0/0             0/0/1000        0/0/1000
ppp                 0/10432525        0/0            0/17/1000        0/0/0
mpls_io             0/0               0/0             0/0/1000        0/0/1000
ipv4                0/0               0/0             0/0/1000        0/0/1000
ipv6                0/0               0/0             0/0/1000        0/0/1000

Key:
  L = queue for lower priority packets
  H = queue for higher priority packets
```

Related Commands

Command	Description
show netio chains	Displays Netio chains information.
show netio db	Displays Netio database information.
show netio idb	Displays Netio IDB information.
show netio media registrations	Displays protocol registrations for media changes.
show netio subblock	Displays Netio subblock information.
show netio trace	Displays Netio trace data.

show netio db

To display Network Input and Output (Netio) database information for an interface, use the **show netio db** command in EXEC mode.

show netio db {**caps** | **dll name***dll-name*| **proto**} [**location** *node-id*]

Syntax Description

caps	Displays the capsulations in the Netio database.
dll	Displays the dlls loaded in the Netio database.
name*dll-name*	(Optional) Specifies a DLL name.
proto	Displays the protocol in the Netio database.
location *node-id*	(Optional) Displays Netio database information for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
Release 3.8.0	This command was introduced.
Release 3.9.0	No modifications.
Release 4.0.0	No modifications.

Usage Guidelines

Task ID

Task ID	Operation
cisco-support	read

Examples

The following example shows the output of the **show netio db** command.

```
RP/0/0/CPU0:router# show netio db caps location 0/1/0

Capsulation (ID)              Load Count  DLL Name                    Refcount
```

```
chdlc(13)                                 1 libchdlc_netio.dll                  3
hdlc(14)                                  2 libchdlc_netio.dll                  3
clns(15)                                  2 libclns_netio.dll                   2
ipv4_acl_in(22)                           1 libipv4_netio_acl_filter.dll        2
ipv4_acl_out(23)                          1 libipv4_netio_acl_filter.dll        2
arp(24)                                   1 libipv4_netio.dll                   6
mpls(25)                                 22 libmpls_netio.dll                   3
ipv4(26)                                 18 libipv4_netio.dll                   6
pim_enc(28)                               2 libpim_encaps_netio.dll             1
pim_null(29)                              5 libpim_null_netio.dll               1
ether(30)                                 2 libether_netio.dll                  3
mpls_te(36)                              32 libmpls_netio.dll                   3
txm_nopull(60)                           67 libsched_netio.dll                  1
lpts(81)                                  2 liblpts_netio.dll                   2
ipv6(82)                                  2 libipv6_netio.dll                   5
l2_adj_rewrite(86)                       67 libl2_adj_netio.dll                 1
ipv6_preswitch(90)                        1 libipv6_netio.dll                   5
fint_base(91)                            10 libfint_netio.dll                   1
fint_n2n(92)                              2 libfint_n2n.dll                     2
ether_sock(98)                            2 libether_netio.dll                  3
ipv6_pfilter_in(102)                      1 libipv6_netio_pfilter.dll           2
ipv6_pfilter_out(103)                     1 libipv6_netio_pfilter.dll           2
netio_debug(110)                          1 libnetio_debugnode.dll              1
ipv4_preroute(115)                        2 libipv4_netio.dll                   6
fint_l2transport(125)                     2 libl2fib_netio.dll                  2
ipv6_preroute(128)                        2 libipv6_netio.dll                   5
ether_shim(130)                           4 libether_shim_netio.dll             1
pos_shim(132)                             3 libpos_shim_netio.dll               1
fint_caps_tp(134)                         2 libfint_netio_tp.dll                2
```

Related Commands

Command	Description
show netio chains	Displays Netio chains information.
show netio clients	Displays Netio clients information.
show netio idb	Displays Netio IDB information.
show netio media registrations	Displays protocol registrations for media changes.
show netio subblock	Displays Netio subblock information.
show netio trace	Displays Netio trace data.

show netio idb

To display network input and output (Netio) interface descriptor block (IDB) information for an interface, use the **show netio idb** command in EXEC mode.

show netio idb {*interface-type interface-instance*} [**location** *node-id*]

Syntax Description

interface-type	Interface type. For more information, use the question mark (?) online help function.
interface-instance	Either a physical interface instance or a virtual interface instance as follows: • Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation. ◦ *rack*: Chassis number of the rack. ◦ *slot*: Physical slot number of the modular services card or line card. ◦ *module*: Module number. A physical layer interface module (PLIM) is always 0. ◦ *port*: Physical port number of the interface. **Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RP0 or RP1) and the module is CPU0. Example: interface MgmtEth0/RP1/CPU0/0. • Virtual interface instance. Number range varies depending on interface type. For more information about the syntax for the router, use the question mark (?) online help function.
location *node-id*	(Optional) Displays Netio IDB information for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.8.0	Changed the *interface-type interface-instance* arguments to required ones.

Release	Modification
Release 3.9.0	No modifications.
Release 4.0.0	No modifications.

Usage Guidelines

Use the **show netio idb**command to display control plane information for the software switching path. The output provides useful statistics for determining software forwarding issues.

Task ID

Task ID	Operations
cisco-support	read

Examples

The following example shows the output of the **show netio idb** command:

```
RP/0/0/CPU0:router# show netio idb tenGigE 0/1/1/0 location 0/1/cpu0

TenGigE0/1/1/0 (handle: 0x01180020, nodeid:0x11) netio idb:
---------------------------------
name:                    TenGigE0_1_1_0
interface handle:        0x01180020
interface global index:  2
physical media type:     30
dchain ptr:              <0x482ae8e0>
echain ptr:              <0x482d791c>
fchain ptr:              <0x482d79b8>
driver cookie:           <0x4824ad58>
driver func:             <0x4824ad44>
number of subinterfaces: 4096
subblock array size:     3
DSNCNF:                  0x00000000
interface stats info:
   IN  unknown proto pkts:  0
   IN  unknown proto bytes: 0
   IN  multicast pkts:      0
   OUT multicast pkts:      0
   IN  broadcast pkts:      0
   OUT broadcast pkts:      0
   IN  drop pkts:           0
   OUT drop pkts:           0
   IN  errors pkts:         0
   OUT errors pkts:         0

Chains
--------------------
Base decap chain:
     ether                <30>  <0xfd7aef88, 0x48302824>  <       0,        0>

Protocol chains:
---------------
<Protocol number> (name)  Stats
  Type  Chain_node       <caps num>  <function, context> <drop pkts, drop bytes>
<7> (arp)   Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes

    Encap:
     l2_adj_rewrite       <86>  <0xfcec7a88, 0x4834efec>  <       0,        0>
     queue_fifo           <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull           <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>
```

```
    Decap:
     queue_fifo           <56>  <0xfcedda4c, 0x482dbee4>  <       0,        0>
     arp                  <24>  <0xfd1082cc, 0x00000000>  <       0,        0>
    Fixup:
     l2_adj_rewrite       <86>  <0xfcec745c, 0x00000000>  <       0,        0>
     queue_fifo           <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull           <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>
<12> (ipv4)   Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
    Encap:
     ipv4                 <26>  <0xfd10f41c, 0x482d7724>  <       0,        0>
     ether                <30>  <0xfd7aeb44, 0x48302824>  <       0,        0>
     l2_adj_rewrite       <86>  <0xfcec7a88, 0x4834f104>  <       0,        0>
     queue_fifo           <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull           <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>
    Decap:
     queue_fifo           <56>  <0xfcedda4c, 0x482dbee4>  <       0,        0>
     ipv4                 <26>  <0xfd10f474, 0x00000000>  <       0,        0>
    Fixup:
     l2_adj_rewrite       <86>  <0xfcec745c, 0x00000000>  <       0,        0>
     queue_fifo           <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull           <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>
<22> (ether_sock)   Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
    Encap:
     ether_sock           <98>  <0xfd7b1630, 0x48302824>  <       0,        0>
     l2_adj_rewrite       <86>  <0xfcec7a88, 0x48304c1c>  <       0,        0>
     queue_fifo           <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull           <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>
    Decap:
     queue_fifo           <56>  <0xfcedda4c, 0x482dbee4>  <       0,        0>
     ether_sock           <98>  <0xfd7b1874, 0x48302824>  <       0,        0>
    Fixup:
     l2_adj_rewrite       <86>  <0xfcec745c, 0x00000000>  <       0,        0>
     queue_fifo           <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull           <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>


Protocol SAFI counts:
--------------------

       Protocol        SAFI       Pkts In    Bytes In    Pkts Out   Bytes Out
---------------  ----------    ----------  ----------  ----------  ----------
           ipv4     Unicast             0           0           0           0
           ipv4   Multicast             0           0           0           0
           ipv4   Broadcast             0           0           0           0
           ipv6     Unicast             0           0           0           0
           ipv6   Multicast             0           0           0           0
```

This table describes the significant fields shown in the display.

Table 4: show netio idb Field Descriptions

Field	Description
name	Netio name associated with the interface.
interface handle	Value assigned to the interface by the netio for identification.
IN unknown proto pkts	Number of packets sent to netio that had an unknown protocol type.
IN unknown proto bytes	Number of bytes sent to netio that had an unknown protocol type.
IN multicast pkts	Number of ingress multicast packets for the interface.

Field	Description
OUT multicast pkts	Number of egress multicast packets for the interface.
IN broadcast pkts	Number of ingress broadcast packets for the interface.
OUT broadcast pkts	Number of egress broadcast packets for the interface.
IN drop pkts	Number of ingress dropped packets for the interface.
OUT drop pkts	Number of egress dropped packets for the interface.
IN errors pkts	Number of ingress errored packets for the interface.
OUT errors pkts	Number of egress errored packets for the interface.
Base decap chain	Lowest-level decap chain assigned to the interface.
Protocol chains	Layer 3 protocol chains assigned to the interface.
Type	Layer 3 protocol type.
drop pkts, drop bytes	Dropped packet and byte counters associated with the protocol.
Endcap	Processing steps in the encap chain.
Decap	Processing steps in the decap chain.
Fixup	Processing steps in the fixup chain.
Protocol SAFI counts	Unicast or multicast counts associated with the protocol.
Protocol	Protocol type.
SAFI	Secondary address family identifier type.
Pkts In	Number of packets in for the address family.
Bytes In	Number of bytes in for the address family.
Pkts Out	Number of packets out for the address family.
Bytes Out	Number of bytes out for the address family.

Related Commands

Command	Description
show netio chains	Displays Netio chains information.
show netio clients	Displays Netio clients information.
show netio db	Displays Netio database information.
show netio media registrations	Displays protocol registrations for media changes.
show netio subblock	Displays Netio subblock information.
show netio trace	Displays Netio trace data.

show netio media-registrations

To display Network Input and Output (Netio) protocol registrations for media changes, use the **show netio media-registrations** command in EXEC mode.

show netio media-registrations[**location** *node-id*]

Syntax Description

location *node-id*	(Optional) Displays Netio protocol registrations for media changes for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
Release 3.8.0	This command was introduced.
Release 3.9.0	No modifications.
Release 4.0.0	No modifications.

Usage Guidelines

Task ID

Task ID	Operation
cisco-support	read

Examples

The following example shows the output of the **show netio media-registrations** command:

```
RP/0/0/CPU0:router# show netio media-registrations location 0/2/0

Registrations by L3 for media (change/upgrade) changes
L3 Protocol      Callback    L2 Media
-------------------------------------------
clns             0x795f978c  atm_mux_vc
                             atm_nlpid_vc
                             atm_snap_vc
                             atm_sub
                             dot1q
                             ether
                             fint_base
                             fr_sub_base
                             fr_vc_base
```

```
                                hdlc
                                srp
ipv4             0x79af58e8     atm_mux_vc
                                atm_nlpid_vc
                                atm_snap_vc
                                atm_sub
                                dot1q
                                ether
                                fint_base
                                fr_sub_base
                                fr_vc_base
                                hdlc
                                srp
ipv6             0x796a45e8     atm_mux_vc
                                atm_nlpid_vc
                                atm_snap_vc
                                atm_sub
                                dot1q
                                ether
                                fint_base
                                fr
                                hdlc
                                srp
mpls             0x79c66d14     atm_nlpid_vc
                                atm_snap_vc
                                atm_sub
                                dot1q
                                ether
                                fint_base
                                hdlc
                                ppp
                                srp
lpts             0x79563174     fint_base
ipv6_preroute    0x796a456c     fint_base
```

Related Commands

Command	Description
show netio chains	Displays Netio chains information.
show netio clients	Displays Netio clients information.
show netio db	Displays Netio database information.
show netio idb	Displays Netio IDB information.
show netio subblock	Displays Netio subblock information.
show netio trace	Displays Netio trace data.

show netio subblock

To display Network Input and Output (Netio) subblock information, use the **show netio subblock** command in EXEC mode.

show netio subblock {**idb** {*interface-typeinterface-instance*} | **registrations** } [**location** *node-id*]

Syntax Description

idb	Displays subblock information for an interface.
registrations	Displays all the registered subblocks.
interface-type	Interface type. For more information, use the question mark (?) online help function.
interface-instance	Either a physical interface instance or a virtual interface instance as follows: • Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation. ◦ *rack*: Chassis number of the rack. ◦ *slot*: Physical slot number of the modular services card or line card. ◦ *module*: Module number. A physical layer interface module (PLIM) is always 0. ◦ *port*: Physical port number of the interface. **Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RP0 or RP1) and the module is CPU0. Example: interface MgmtEth0/RP1/CPU0/0. • Virtual interface instance. Number range varies depending on interface type. For more information about the syntax for the router, use the question mark (?) online help function.
location *node-id*	(Optional) Displays Netio subblock information for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
Release 3.8.0	This command was introduced.
Release 3.9.0	No modifications.
Release 4.0.0	No modifications.

Usage Guidelines

Task ID

Task ID	Operation
cisco-support	read

Examples

The following example shows the output of the **show netio subblock** command:

```
RP/0/0/CPU0:router# show netio subblock registrations location 0/2/2

Feature Name      Subblock List  Destroy Func           Handle
  <subblock addr> <intf handle>  <intf name>            <refcnt>
----------------  -------------  ---------------------  --------
ipv6-switch       0x0811cbfc     0x796ae090                     1
    <0x0806a6b0> <0x03000100>   <FINT0_2_CPU0          > <     3>
ether-caps        0x08198ba0     0x79f350b4                     2
    <0x0807aa44> <0x03000600>   <FastEthernet0_2_2_0   > <     3>
    <0x0807aa88> <0x03000700>   <FastEthernet0_2_2_1   > <     3>
    <0x0807aacc> <0x03000800>   <FastEthernet0_2_2_2   > <     3>
    <0x081c2758> <0x03000900>   <FastEthernet0_2_2_3   > <     3>
    <0x081c279c> <0x03000a00>   <FastEthernet0_2_2_4   > <     3>
    <0x081c27e0> <0x03000b00>   <FastEthernet0_2_2_5   > <     3>
    <0x081c2824> <0x03000c00>   <FastEthernet0_2_2_6   > <     3>
    <0x081c2868> <0x03000d00>   <FastEthernet0_2_2_7   > <     4>
fr_control_vc_base_caps  0x081bdf6c     0x7a0209c8                     3
    <0x081c2978> <0x03001a00>   <POS0_2_0_0.0_vc_0     > <     2>
    <0x081c29bc> <0x03001b00>   <POS0_2_0_1.0_vc_0     > <     2>
    <0x081c2a00> <0x03001c00>   <POS0_2_0_0.0_vc_1023  > <     2>
    <0x081c2a44> <0x03001d00>   <POS0_2_0_1.0_vc_1023  > <     2>
fr_vc_base_caps   0x08206424     0x7a020890                     4
    <0x081c2a88> <0x03001e00>   <POS0_2_0_0.1          > <     2>
    <0x081c2acc> <0x03001f00>   <POS0_2_0_1.1          > <     2>
```

Related Commands

Command	Description
show netio chains	Displays Netio chains information.
show netio clients	Displays Netio clients information.
show netio db	Displays Netio database information.

Command	Description
show netio idb	Displays Netio IDB information.
show netio media registrations	Displays protocol registrations for media changes.
show netio trace	Displays Netio trace data.

show netio trace

To display Network Input and Output (Netio) trace information, use the **show netio trace** command in EXEC mode.

show netio trace {**all**| **chains**| **control**| **dpc**| **error**| **interface**| **LC**| **packet**} [*file*| *hexdump*| *last*| *location*| *reverse*| *stats*| *tailf*| *unique*| *verbose*| *wrapping*]

Syntax Description

all	Displays all Netio trace data
chains	Displays Netio chains trace data
control	Displays Netio control trace data
dpc	Displays Netio DPC trace data
error	Displays Netio error trace data
interface	Displays Netio interface trace data
LC	Displays Netio trace information for LC processes data
packet	Displays Netio packet drop error messages trace data
file	(Optional) A specific file name traces in hexadecimal
hexdump	(Optional) Display traces in hexadecimal
last	(Optional) Displays the last n entries
location	(Optional) Displays the card location
reverse	(Optional) Displays the latest traces first
stats	(Optional) Displays statistics
tailf	(Optional) Displays new traces as added
unique	(Optional) Displays unique entries with counts

verbose	(Optional) Displays internal debugging information
wrapping	(Optional) Displays wrapping entries

Command Default

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
Release 3.8.0	This command was introduced.
Release 3.9.0	No modifications.
Release 4.0.0	No modifications.

Usage Guidelines

Task ID

Task ID	Operation
cisco-support	read

Examples

The following example shows the output of the **show netio trace** command:

```
RP/0/0/CPU0:router# show netio trace chains stats location 0/0/CPU0

/net/node0_0_CPU0/dev/shmem/ltrace/netio/chains--- wrapping: inf Mbytes/sec for 1024 entries
361 wrapping entries (1024 possible, 0 filtered, 361 total)
Jan 11 15:04:14.695 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 0 (base), caps 91 (fint_base), op ADD, chain BD, data len 0
Jan 11 15:04:15.070 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 18 (lpts), caps 81 (lpts), op ADD, chain D, data len 4
Jan 11 15:04:16.265 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 18 (lpts), caps 86 (l2_adj_rewrite), op ADD, chain E, data len 0
Jan 11 15:04:16.274 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 18 (lpts), caps 60 (txm_nopull), op ADD, chain E, data len 0
Jan 11 15:04:16.542 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 18 (lpts), caps 86 (l2_adj_rewrite), op ADD, chain F, data len 0
Jan 11 15:04:16.542 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 18 (lpts), caps 60 (txm_nopull), op ADD, chain F, data len 0
Jan 11 15:04:16.542 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
```

```
0x01000100, prot
o 18 (lpts), caps 91 (fint_base), op ADD, chain E, data len 0
Jan 11 15:04:16.542 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 18 (lpts), caps 81 (lpts), op ADD, chain E, data len 4
Jan 11 15:04:16.562 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 6 (fint_n2n), caps 92 (fint_n2n), op ADD, chain D, data len 0
Jan 11 15:04:16.646 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 6 (fint_n2n), caps 86 (l2_adj_rewrite), op ADD, chain E, data len 0
Jan 11 15:04:16.646 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 6 (fint_n2n), caps 60 (txm_nopull), op ADD, chain E, data len 0
Jan 11 15:04:16.646 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 6 (fint_n2n), caps 86 (l2_adj_rewrite), op ADD, chain F, data len 0
Jan 11 15:04:16.646 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 6 (fint_n2n), caps 60 (txm_nopull), op ADD, chain F, data len 0
Jan 11 15:04:16.646 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
o 6 (fint_n2n), caps 91 (fint_base), op ADD, chain E, data len 0
Jan 11 15:04:16.646 netio/chains--- 0/0/CPU0 t1  Chains: update IDB chain, ifhandle
0x01000100, prot
.
.
.
```

Related Commands

Command	Description
show netio chains	Displays Netio chains information.
show netio clients	Displays Netio clients information.
show netio db	Displays Netio database information.
show netio idb	Displays Netio IDB information.
show netio media registrations	Displays protocol registrations for media changes.
show netio subblock	Displays Netio subblock information.

show sysdb connections

To display the client connection information for the system database (SYSDB), use the **show sysdb connections** command in EXEC mode.

show sysdb connections [**detail**| **job**| **path**| **location**| **shared-plane**]

Syntax Description

detail	(Optional) Displays the detailed client connection information.
job *job-id*	(Optional)Specify a Job ID.
path*path-filter*	(Optional) Specify a path filter.
location *node-id*	(Optional)Specify a location. The *node-id* argument is entered in the *rack/slot/module* notation.
shared-plane	(Optional) Displays the shared-plane data.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.0	This command was introduced.

Usage Guidelines

Task ID

Task ID	Operations
sysmgr	read
cisco-support	read

Examples

The following example shows the output of the **show sysdb connections** command.

```
RP/0/0/CPU0:router# show sysdb connections detail location 0/1/CPU0

SysDB Connections:
  "/debug/node/11/LR/sysdb/client/"
```

```
  From:      shmwin_svr (jid 76, nid 0/1/CPU0, tid 1)
  Connid:    00000001 Refcount: 0002 Options:   00000032
  Connected:        Y In trans:    N Verf susp:        N
  Client connid: 00000000
  Connected at: Jul 14 19:31:47.304
  "/debug/node/11/LR/packet/"
  From:      packet (jid 218, nid 0/1/CPU0, tid 1)
  Connid:    00000002 Refcount: 0002 Options:   00000032
  Connected:        Y In trans:    N Verf susp:        N
  Client connid: 00000000
  Connected at: Jul 14 19:31:47.305
  "/debug/node/11/LR/cdm/qsm/"
  From:      qsm (jid 246, nid 0/1/CPU0, tid 4)
  Connid:    00000003 Refcount: 0002 Options:   00000032
  Connected:        Y In trans:    N Verf susp:        N
  Client connid: 00000000
  Connected at: Jul 14 19:31:47.305
  "/debug/node/11/LR/eem/"
  From:      wdsysmon (jid 361, nid 0/1/CPU0, tid 5)
  Connid:    00000005 Refcount: 0002 Options:   00000032
  Connected:        Y In trans:    N Verf susp:        N
Client connid: 00000000
  Connected at: Jul 14 19:31:47.316
  "/debug/node/11/LR/sysmgr/"
  From:      sysmgr (jid 79, nid 0/1/CPU0, tid 7)
  Connid:    00000013 Refcount: 0002 Options:   00000032
...
```

show sysdb trace verification location

To display trace verification information for the system database (SYSDB), use the **show sysdb trace verification location** command in EXEC mode.

show sysdb trace verification location *node-id*

Syntax Description

node-id	Specific node. The *node-id* argument is entered in the *rack*/*slot*/*module* notation.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

Use the **show sysdb trace verification shared-plane location** command to display details of recent verification sysDB transactions and changes on local plane configurations. The command output allows you to confirm that configuration were verified and accepted.

Task ID

Task ID	Operations
sysmgr	read
cisco-support	read

Examples

The following example shows the output of the **show sysdb trace verification shared-plane location** command. The output shows that changes to the SysDB local plane were verified and accepted.

```
RP/0/0/CPU0:router# show sysdb trace verification location 0/3/CPU0

Timestamp             jid       tid   reg handle  connid     action
              path
323 wrapping entries (4096 possible, 299 filtered, 622 total)
Jul  7 20:10:36.212      260       1     90       8782       apply reply
              '__'
Jul  7 20:10:35.476      260       1     90       4912       Apply/abort called
              'cfg/if/act/GigabitEthernet0_3_4_0.1/a/sub_vlan/0x2/________/Gigab
itEthernet0_3_4_0/________'
```

```
Jul  7 20:10:35.475      260       1     90       4912       verify reply: accep
t             '--'
Jul  7 20:10:35.471      260       1     90       4912       Verify called
              'cfg/if/act/GigabitEthernet0_3_4_0.1/a/sub_vlan/0x2/________/Gigab
itEthernet0_3_4_0/________'
Jul  7 20:10:35.471      144       1     4        8782       apply reply
              '--'
Jul  7 20:10:35.471      144       1     4        8782       apply reply
              '--'
Jul  7 20:10:35.471      144       1     4        8782       apply reply
              '--'
Jul  7 20:10:35.471      144       1     4        8782       apply reply
              '--'
Jul  7 20:10:35.471      144       1     4        8782       apply reply
              '--'
Jul  7 20:10:35.471      144       1     4        8782       apply reply
              '--'
Jul  7 20:10:35.471      144       1     4        8782       apply reply
              '--'
Jul  7 20:10:35.471      144       1     4        8782       apply reply
              '--'
Jul  7 20:10:35.470      144       1     4        474        Apply/abort batch e
nded          ''
Jul  7 20:10:35.470      144       1     4        474        Apply/abort called
              'cfg/if/act/GigabitEthernet0_3_4_0/ord_x/im/shutdown'
Jul  7 20:10:35.470      144       1     4        474        Apply/abort called
              'cfg/if/act/GigabitEthernet0_3_4_1/ord_x/im/shutdown'
Jul  7 20:10:35.470      144       1     4        474        Apply/abort called
              'cfg/if/act/GigabitEthernet0_3_4_2/ord_x/im/shutdown'
Jul  7 20:10:35.470      144       1     4        474        Apply/abort called
              'cfg/if/act/GigabitEthernet0_3_4_3/ord_x/im/shutdown'
Jul  7 20:10:35.470      144       1     4        474        Apply/abort called
              'cfg/if/act/GigabitEthernet0_3_4_4/ord_x/im/shutdown'
Jul  7 20:10:35.469      144       1     4        474        Apply/abort called
              'cfg/if/act/GigabitEthernet0_3_4_5/ord_x/im/shutdown'
Jul  7 20:10:35.469      144       1     4        474        Apply/abort called
              'cfg/if/act/GigabitEthernet0_3_4_6/ord_x/im/shutdown'
Jul  7 20:10:35.469      144       1     4        474        Apply/abort called
              'cfg/if/act/GigabitEthernet0_3_4_7/ord_x/im/shutdown'
Jul  7 20:10:35.469      144       1     4        474        Apply/abort batch s
tarted        ''
Jul  7 20:10:35.469      144       1     4        474        verify reply: accep
t             '--'
Jul  7 20:10:35.469      144       1     4        474        verify reply: accep
t             '--'
Jul  7 20:10:35.469      144       1     4        474        verify reply: accep
t             '--'
!
!
!
```

This table describes the significant fields shown in the display.

Table 5: show sysdb trace verification location Field Descriptions

Field	Description
Timestamp	Time of the verification.
jid	Job identifier of the verification.
tid	Thread identifier.
reg handle	Registration handle.
connid	Connection identifier.

Field	Description
action	Action occurring between the sysDB server and client.
apply reply	SysDB notification that the client that an apply action has occurred.
Apply/abort called	SysDB notification for the client that an apply or abort has been called.
verify reply: accept	Verifier has accepted the verification request.

Related Commands

Command	Description
show sysdb connection path shared-plane	Displays system database client connection shared plane data for a specific path.

show sysdb trace verification shared-plane

To display trace verification information for the system database (SYSDB), use the **show sysdb trace verification shared-plane**command in EXEC mode.

show sysdb trace verification shared-plane [**file**| **hexdump**| **last**| **location**| **reverse**| **stats**| **tailf**| **unique**| **verbose**| **wrapping**]

Syntax Description

file	(Optional) Specifies the name of a file.
hexdump	(Optional) Displays the packet contents in hexadecimal format.
last	(Optional) Specifies the last number of packets in the queue to display.
location	(Optional) Displays the card location.
reverse	(Optional) Specifies the new traces as they are added.
stats	(Optional) Displays trace statistics information.
tailf	(Optional) Displays new traces as they are added.
unique	(Optional) Displays a list of unique entries with counts.
verbose	(Optional) Displays internal debugging information.
wrapping	(Optional) Displays wrapping entries of all trace information.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.8.0	All optional arguments were added with their descriptions.

Usage Guidelines

Use the **show sysdb trace verification shared-plane** command to display details of recent verification sysDB transactions and changes on the shared plane. The command output allows you to confirm whether the configuration was verified correctly.

Specifying a path using the | **include** keyword and *path* argument filters the data to display only the sysDB path for the router. Use the **describe** command to determine the path.

Task ID

Task ID	Operations
sysmgr	read
cisco-support	read

Examples

The following example shows the output of the **show sysdb trace verification shared-plane**command. The output shows that changes to the SysDB shared plane were verified and accepted.

```
RP/0/0/CPU0:router# show sysdb trace verification shared-plane | include gl/a/hostname

May 18 19:16:17.143      340       3     210      962        Apply/abort called
              'cfg/gl/a/hostname'
May 18 19:16:17.132      340       3     210      962        Verify called
              'cfg/gl/a/hostname'
May 18 19:16:17.126      340       3     210      962        Apply/abort called
              'cfg/gl/a/hostname'
May 18 19:16:17.109      340       3     210      962        Verify called
              'cfg/gl/a/hostname'
May 18 18:43:16.065      340       3     210      962        register
              'cfg/gl/a/hostname'
May 18 18:41:41.048      340       3     16       362        register
              'cfg/gl/a/hostname'
```

This table describes the significant fields shown in the display.

Table 6: show sysdb trace verification shared-plane Field Descriptions

Field	Description
Apply/abort called	SysDB server has either applied or aborted the action requiring verification.
Verify called	Client has issued a verify request to the sysDB server.
register	Client has registered with sysDB server for verification.

Related Commands

Command	Description
show sysdb connection path shared-plane	Displays sysDB client connection shared plane data for a specific path.

show tbm hardware

To displays tree bitmap hardware-related information, use the **show tbm hardware** command in EXEC mode.

show tbm hardware {**ipv4**| **ipv6**| **mpls**| **vpnv4**| **table-id**| **afi-all**| **sw-only**| **dual**| **egress**| **ingress**} {**unicast**| **multicast**| **safi-all**} {**dual**| **egress**| **ingress**| **sw-only**} {**brief**| **detail**| **lookup**| **prefix prefix-hex-string**} **location** *node-id*

Syntax Description

ipv4	Specifies IP Version 4 address prefixes.
ipv6	Specifies IP Version 6 address prefixes.
mpls	Specifies MPLS-related tree bitmap information.
vpnv4	Specifies VPNv4-related tree bitmap information.
table-id	Specifies tree bitmap information for a specific table ID.
afi-all	Specifies IPv4 and IPv6 commands.
sw-only	Specifies software-only tree bitmap information.
dual	Specifies tree bitmap information for dual, ingress, and egress, modes.
egress	Specifies egress tree bitmap information.
ingress	Specifies ingress tree bitmap information.
unicast	Specifies unicast address prefixes.
multicast	Specifies multicast address prefixes. This option is supported for IPv4 address families.
safi-all	For subaddress family, specifies prefixes for all subaddress families. This option is supported for IPv4 address families.
dual	Specifies ingress and egress tree bitmap information.
brief	Displays brief information.
detail	Displays detailed information.
lookup	Displays key or address information to look up (longest match) in the table.
prefix	Displays prefix-related information.
location *node-id*	Displays tree bitmap hardware-related information for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

Use the **show tbm hardware**command to display hardware-related ingress and egress information for the tree bitmap.

Task ID

Task ID	Operations
cisco-support	read

Examples

The following example shows the output of the **show tbm hardware** command:

```
RP/0/0/CPU0:router# show tbm hardware ipv4 unicast dual detail location 0/6/cpu0

TBM Table Type: IPv4 Unicast
-----------------------------
TBM: number of pulses: 71
TBM: number of Err fix attempts: 0
     No current failures
Past failures: leaf(0), mem(0), mipc(0), flush_mipc(0)
               post_compact(0), pre_compact(0)

PLU Bucket Statistics:
----------------------------
     Bucket 0: 44
     Bucket 1: 44
     Bucket 2: 327
     Bucket 3: 44
     Bucket 4: 44
     Bucket 5: 43
     Bucket 6: 43
     Bucket 7: 45

Ingress PLU Info
----------------
    PLU: Num Writes : 3064
    PLU: Num Copies : 2197

    PLU Memory Channel Statistics:
    ----------------------------
      Number of compactions: 0
         FCRAM0 Chan:      110 (Pages: 5, 1% used)
         FCRAM1 Chan:      125 (Pages: 8, 0% used)
         FCRAM2 Chan:      127 (Pages: 8, 0% used)
         FCRAM3 Chan:      148 (Pages: 8, 0% used)
```

```
         FCRAM4 Chan:      124 (Pages: 8, 0% used)

Egress PLU Info
----------------
    PLU: Num Writes : 3064
    PLU: Num Copies : 2197

    PLU Memory Channel Statistics:
    ------------------------------
      Number of compactions: 0
         FCRAM0 Chan:      110 (Pages: 5, 1% used)
         FCRAM1 Chan:      125 (Pages: 8, 0% used)
         FCRAM2 Chan:      127 (Pages: 8, 0% used)
         FCRAM3 Chan:      148 (Pages: 8, 0% used)
         FCRAM4 Chan:      124 (Pages: 8, 0% used)
```

This table describes the significant fields shown in the display.

Table 7: show tbm hardware Field Descriptions

Field	Description
Past failures	Number of times there was a failure in programming hardware.
PLU: Num Writes	Number of writes to the PLU portion of the hardware.
PLU: Num Copies	Number of copies to the PLU portion of the hardware.
PLU Memory Channel Statistics	Usage levels of each channel in the PLU memory.

show uidb data

To display index data information for the micro-interface descriptor block (uIDB), use the **show uidb data**command in EXEC mode.

show uidb data [shadow] [**ingress**| **egress**] [*interface-type interface-instance*] **location** *node-id*

Syntax Description

shadow	(Optional) Displays uIDB data from shadow copy Route Skill Mapping (RSM) instead of Metro HW.
ingress	(Optional) Displays ingress PSE-related information.
egress	(Optional) Displays egress PSE-related information.
interface-type	(Optional) Interface type. For more information, use the question mark (?) online help function.
interface-instance	Either a physical interface instance or a virtual interface instance as follows: • Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation. ◦ *rack*: Chassis number of the rack. ◦ *slot*: Physical slot number of the modular services card or line card. ◦ *module*: Module number. A physical layer interface module (PLIM) is always 0. ◦ *port*: Physical port number of the interface. **Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RP0 or RP1) and the module is CPU0. Example: interface MgmtEth0/RP1/CPU0/0. • Virtual interface instance. Number range varies depending on interface type. For more information about the syntax for the router, use the question mark (?) online help function.
location *node-id*	(Optional) Displays micro-IDB index data information for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.5.0	Index and operation modes were not supported.
Release 3.7.0	The following keywords were added: • **shadow** • **ingress** • **egress**

Usage Guidelines

Use the **show uidb index** command to display micro-IDB index data information including, from a software perspective, features that are enabled on a selected interface.

Task ID

Task ID	Operations
cisco-support	read

Examples

The following example shows the output of the **show uidb data** command:

```
RP/0/0/CPU0:router# show uidb data shadow ingress gigabitEthernet 0/2/4/4 loc 0/2/CPU0
-----------------------------------------------------------------------------
  Location = 0/2/CPU0
  Ifname/Ifhandle = GigabitEthernet0_2_4_4 / 0x12800a0
  Index = 5
  Pse direction = INGRESS
  ==========================================
  *     (Not programmed in hardware)    *
  ------------------------------------------
  RSM STATUS: 0x7c000000
  -> used: 0x1f
  ->dirty: 0x00
  ->badck: 0x00
  -> prog: DONE
  ->count: 0
  ------------------------------------------
  BUNDLE IFHANDLE: 0
  TUNNEL IFHANDLE: 0
  L2 ENCAP: 3
  ==========================================

  General 16 bytes:
  ------------------------
  IFHANDLE: 0x12800a
  STATUS: 1
  ISSU State: 0
  IPV4 ENABLE: 1
  IPV6 ENABLE: 1
  MPLS ENABLE: 0
  STATS POINTER: 0x7ffd8
```

```
SPRAYER QUEUE: 36
IPV4 MULTICAST: 0
IPV6 MULTICAST: 0
USE TABLE ID IPV4: 0
USE TABLE ID IPV6: 0
USE TABLE ID MPLS: 0
TABLE ID: 0
QOS ENABLE: 0
QOS ID: 0
NETFLOW SAMPLING PERIOD: 0
L2 PKT DROP: 0
L2 QOS ENABLE: 0
SRC FWDING: 0
*[CHECKSUM]*: 0xff70f28c
```

This table describes the significant fields shown in the display.

Table 8: show uidb data Field Descriptions

Field	Description
Location	Node in system where the interface resides.
Ifname/Ifhandle	Name associated with the interface.
SPRAYER QUEUE LSB	Sprayer queue identifier.
ICMP PUNT FLAG	Flag indicating ICMP punts are enabled for the protocol.

Related Commands

Command	Description
show uidb trace, on page 50	Displays UIDB trace data debugging information that helps in troubleshooting the problem.
show uidb data-dump	Displays UIDB data information in hexadecimal format.

show uidb trace

To display trace data information for the micro-interface descriptor block (IDB), use the **show uidb trace** command in EXEC mode.

show uidb trace {**all**| **errors**| **events**| **init**| **rsm**}[**file** *file-name*] **[hexdump]** [**last** *entries*] **[reverse] [stats] [tailf] [unique] [usec] [verbose] [wide] [wrapping]** [**location** {*node-id*| **all**| **mgmt-nodes**}]

Syntax Description

all	Displays all UIDB trace information.
errors	Displays information related to UIDB errors trace.
events	Displays information related to UIDB events trace.
init	Displays information related to UIDB init trace.
rsm	Displays information related to UIDB rsm trace.
file	(Optional) Displays a specific file.
filename	Name of a specific file.
hexdump	(Optional) Displays traces in hexadecimal format.
last	(Optional) Displays trace information for a specific number of entries
entries	Number of entries. Replace entries with the number of entries you want to display. For example, if you enter 5, the display shows the last 5 entries in the trace data. The range is from 1 to 65536.
reverse	(Optional) Displays the latest traces first.
stats	(Optional) Displays the statistics in the command output.
tailf	(Optional) Displays the new traces as they are added in the command output.
usec	(Optional) Displays timestamp w/usec detail.
wide	(Optional) Do not display buffer name, node name, and thread-id.
unique	(Optional) Displays the unique entries with counts in the command output.
verbose	(Optional) Displays the information for internal debugging in the command output.
wrapping	(Optional) Displays the wrapping entries in the command output.
location *node-id*	(Optional) Specifies a node. The *node-id* argument is entered in the *rack/slot/module* notation.

location all	(Optional) Specifies all locations.
location mgmt-nodes	(Optional) Specifies all management nodes.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.
Release 3.5.0	Index and operation modes were not supported.

Usage Guidelines

Task ID

Task ID	Operations
cisco-support	read

Examples

The following example shows the sample output from the **show uidb trace** command:

```
RP/0/0/CPU0:router sh uidb trace init loc 0/6/CPU0
-----------------------------------------------------------------------
28 wrapping entries (512 possible, 0 filtered, 28 total)
Mar 31 02:27:35.368 uidb_svr/initlog 0/6/CPU0 t1  Entering : Event manager init
Mar 31 02:27:36.641 uidb_svr/initlog 0/6/CPU0 t1  Successful : Event manager int
Mar 31 02:27:36.641 uidb_svr/initlog 0/6/CPU0 t1  Entering : Debug init
Mar 31 02:27:36.816 uidb_svr/initlog 0/6/CPU0 t1  Successful : Debug init
Mar 31 02:27:36.816 uidb_svr/initlog 0/6/CPU0 t1  Entering : MIPC bund
Mar 31 02:27:51.695 uidb_svr/initlog 0/6/CPU0 t1  Successful : MIPC bind
Mar 31 02:27:51.695 uidb_svr/initlog 0/6/CPU0 t1  PSE RSM : Init - main() : (50s
Mar 31 02:27:51.803 uidb_svr/initlog 0/6/CPU0 t1  Successful : PSE RSM Init sucd
Mar 31 02:27:51.803 uidb_svr/initlog 0/6/CPU0 t1  Entering : Metro bind
Mar 31 02:27:51.828 uidb_svr/initlog 0/6/CPU0 t1  Successful : Metro bind
Mar 31 02:27:51.828 uidb_svr/initlog 0/6/CPU0 t1  Entering : PLIM ASIC register
Mar 31 02:27:51.922 uidb_svr/initlog 0/6/CPU0 t1  Successful : PLIM ASIC registr
Mar 31 02:27:51.922 uidb_svr/initlog 0/6/CPU0 t1  Entering : UIDB checkpoint int
Mar 31 02:27:51.944 uidb_svr/initlog 0/6/CPU0 t1  Successful : UIDB checkpoint t
Mar 31 02:27:51.944 uidb_svr/initlog 0/6/CPU0 t1  Entering : UIDB shadow memoryt
Mar 31 02:27:51.944 uidb_svr/initlog 0/6/CPU0 t1  Successful : UIDB shadow memot
Mar 31 02:27:51.944 uidb_svr/initlog 0/6/CPU0 t1  Entering : UIDB EDM init
Mar 31 02:27:51.951 uidb_svr/initlog 0/6/CPU0 t1  Successful : UIDB EDM init
Mar 31 02:27:51.951 uidb_svr/initlog 0/6/CPU0 t1  Entering : Checkpoint ingresse
Mar 31 02:27:51.951 uidb_svr/initlog 0/6/CPU0 t1  Successful : Checkpoint ingree
Mar 31 02:27:51.951 uidb_svr/initlog 0/6/CPU0 t1  Entering : Checkpoint egress e
```

Mar 31 02:27:51.951 uidb_svr/initlog 0/6/CPU0 t1 Successful : Checkpoint egrese

Related Commands

Command	Description
show uidb data, on page 47	Displays UIDB index data information.
show uidb data-dump	Displays UIDB data information in hexadecimal format.

show uidb index

To display micro-interface descriptor block (IDB) index information, use the **show uidb index** command in EXEC mode.

show uidb index [*interface-type interface-instance*] **location** *node-id*

Syntax Description

interface-type	(Optional) Interface type. For more information, use the question mark (?) online help function.
interface-instance	Either a physical interface instance or a virtual interface instance as follows: • Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation. ◦ *rack*: Chassis number of the rack. ◦ *slot*: Physical slot number of the modular services card or line card. ◦ *module*: Module number. A physical layer interface module (PLIM) is always 0. ◦ *port*: Physical port number of the interface. **Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RP0 or RP1) and the module is CPU0. Example: interface MgmtEth0/RP1/CPU0/0. • Virtual interface instance. Number range varies depending on interface type. For more information about the syntax for the router, use the question mark (?) online help function.
location *node-id*	Displays UIDB index information for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

Use the **show uidb index**command to display the micro-IDB index assigned by the software.

Task ID

Task ID	Operations
cisco-support	read

Examples

The following example shows the output of the **show uidb index** command:

```
RP/0/0/CPU0:router# show uidb index

--------------------------------------------------------------------------------
  Location Interface-name    Interface-Type       Ingress-index  Egress-index
--------------------------------------------------------------------------
    0/1/CPU0          0                                0                0
    0/1/CPU0 GigabitEthernet0_1_5_0          Main interface        1
    1
    0/1/CPU0 GigabitEthernet0_1_5_1          Main interface        2
    2
    0/1/CPU0 GigabitEthernet0_1_5_2          Main interface        3
    3
    0/1/CPU0 GigabitEthernet0_1_5_3          Main interface        4
    4
    0/1/CPU0 GigabitEthernet0_1_5_4          Main interface        5
    5
    0/1/CPU0 GigabitEthernet0_1_5_5          Main interface        6
    6
    0/1/CPU0 GigabitEthernet0_1_5_6          Main interface        7
    7
    0/1/CPU0 GigabitEthernet0_1_5_7          Main interface        8
    8
    0/1/CPU0 POS0_1_0_0          Main interface        9                9
    0/1/CPU0 POS0_1_4_0          Main interface       10               10
    0/1/CPU0 POS0_1_0_1          Main interface       11               11
    0/1/CPU0 POS0_1_4_1          Main interface       12               12
    0/1/CPU0 POS0_1_0_2          Main interface       13               13
    0/1/CPU0 POS0_1_4_2          Main interface       14               14
    0/1/CPU0 POS0_1_0_3          Main interface       15               15
    0/1/CPU0 POS0_1_4_3          Main interface       16               16
    0/1/CPU0 Bundle-POS24        Bundle Interface  17               17
    0/1/CPU0 Bundle-Ether28  Bundle Interface18 18
    0/1/CPU0 Bundle-Ether28.1           Sub-interface 19               19
    0/1/CPU0 Bundle-Ether28.2           Sub-interface 20               20
    0/1/CPU0 Bundle-Ether28.3           Sub-interface 21               21
    0/6/CPU0          0                                0                0
    0/6/CPU0 GigabitEthernet0_6_5_0          Main interface        1
    1
    0/6/CPU0 GigabitEthernet0_6_5_1          Main interface        2
    2
    0/6/CPU0 GigabitEthernet0_6_5_2          Main interface        3
    3
    0/6/CPU0 GigabitEthernet0_6_5_3          Main interface        4
    4
    0/6/CPU0 GigabitEthernet0_6_5_4          Main interface        5
    5
    0/6/CPU0 GigabitEthernet0_6_5_5          Main interface        6
    6
    0/6/CPU0 GigabitEthernet0_6_5_6          Main interface        7
    7
    0/6/CPU0 GigabitEthernet0_6_5_7          Main interface        8
    8
    0/6/CPU0 POS0_6_0_0          Main interface        9                9
    0/6/CPU0 POS0_6_4_0          Main interface       10               10
```

```
0/6/CPU0 POS0_6_0_1          Main interface       11                11
0/6/CPU0 POS0_6_4_1          Main interface       12                12
0/6/CPU0 POS0_6_0_2          Main interface       13                13
0/6/CPU0 POS0_6_4_2          Main interface       14                14
0/6/CPU0 POS0_6_0_3          Main interface       15                15
0/6/CPU0 POS0_6_4_3          Main interface       16                16
0/6/CPU0 POS0_6_4_4          Main interface       17                17
0/6/CPU0 POS0_6_4_5          Main interface       18                18
0/6/CPU0 POS0_6_4_6          Main interface       19                19
0/6/CPU0 POS0_6_4_7          Main interface       20                20
```

This table describes the significant fields shown in the display.

Table 9: show uidb index Field Descriptions

Field	Description
Location	Node where index is located.
Interface-name	Name of the interface.
Interface-Type	Type of interface.
Ingress-index	Value associated with ingress processing on the interface.
Egress-index	Value associated with egress processing on the interface.

Related Commands

Command	Description
show uidb data, on page 47	Displays micro-interface descriptor block index data information.
show uidb data-dump	Displays micro-interface descriptor block data information in hexadecimal format.

watchdog threshold memory

To configure the value of memory available for each alarm threshold, use the **watchdog threshold memory** command in global configuration or interface configuration mode. To revert to the default threshold memory, use the **no** form of this command.

watchdog memory threshold [**location** *node-id*] **minor** *percentage-memory-available* **severe** *percentage-memory-available* **critical** *percentage-memory-available*

no watchdog memory threshold [**location** *node-id*] **minor** *percentage-memory-available* **severe** *percentage-memory-available* **critical** *percentage-memory-available*

Syntax Description

location *node-id*	Configures the threshold memory for a specified node. The *node-id* argument is entered in the *rack/slot/module* notation.
minor	Specifies the threshold for the minor state.
percentage-memory- available	Memory consumption percentage. Range is from 5 to 40.
severe	Specifies the threshold for the severe state.
critical	Specifies the threshold for the critical state.

Command Default

None

Command Modes

Global configuration

Interface configuration

Command History

Release	Modification
Release 3.4.0	This command was introduced.

Usage Guidelines

Use the **watchdog threshold memory** command to configure the memory thresholds. Threshold values can be applied to all nodes or a specific node using the **location** *node-id* keyword and argument. If the local threshold settings are removed, the local settings return to those set globally. In addition, you can view default and configured thresholds.

This table lists the recommended memory threshold value calculations if the minor threshold is set to 20 percent, the severe threshold is set to 10 percent, and the critical threshold is set to 5 percent.

Table 10: Recommended Memory Threshold Values

Total Available Memory (MB)	Minor Threshold (20 percent of available memory)	Severe Threshold (10 percent of available memory)	Critical Threshold (5 percent of available memory)
128	25.6	12.8	6.4
256	51.2	25.6	12.8
512	102.4	51.2	25.6
1024	204.8	102.4	51.2
2048	409.6	204.8	102.4
4096	819.2	409.6	204.8

Task ID

Task ID	Operations
cisco-support	read, write

Examples

The following example shows how to configure the memory available for each alarm threshold:

```
RP/0/0/CPU0:router #configure
RP/0/0/CPU0:router(config)# watchdog threshold memory location 0/RP0/CPU0 minor 30 severe
20 critical 10
```