

# Diagnostic Commands

This module describes the Cisco IOS XR Software commands to configure diagnostics for system monitoring on the router.

For detailed information about the online diagnostics, refer *Online Diagnostics* module in *System Monitoring Configuration Guide for Cisco 8000 Series Routers*.

To use commands of this module, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using any command, contact your AAA administrator for assistance.

For Cisco Technical Support contact information, see the 'Obtaining Documentation and Submitting a Service Request' section in the Preface.


- show diag , on page 2
- diagnostic monitor interval, on page 5
- diagnostic monitor location disable, on page 6
- diagnostic monitor syslog, on page 7
- diagnostic monitor threshold, on page 8
- show dataplane-health status, on page 9
- show diagnostic trace location, on page 10
- show diagnostic result, on page 11
- monitor dataplane-health, on page 13
- clear packet-trace conditions all, on page 18
- clear packet-trace counters all, on page 19
- packet-trace condition, on page 19
- packet-trace start, on page 21
- packet trace stop, on page 21
- show packet-trace decode, on page 22
- show packet trace description, on page 26
- show packet-trace hw-status, on page 26
- show packet-trace result, on page 28
- show packet trace status, on page 28

# show diag

To display details about the hardware and software on each node in a router, use the **show diag** command in XR EXEC mode.

**show diag** [**location** *node-id*] [**chassis** | **details** | **eeprom** | **fans** | **power-supply** | **summary**]

## Syntax Description

| | |
|---|---|
| **location** *node-id* | (Optional) Displays diagnostic information from the designated node. The *node-id* argument is entered in the *rack/slot/module* notation. |
| chassis | (Optional) Displays detailed diagnostics information for the chassis. |
| details | (Optional) Displays detailed diagnostics information for the current node. |
| eeprom | (Optional) Displays field diagnostics results from the EEPROM. |
| fans | (Optional) Displays information about the fans tray. |
| power-supply | (Optional) Displays information about the power supply. |
| summary | (Optional) Displays summarized diagnostics results for all nodes in the system. |

## Command Default

Diagnostics for all nodes installed in the router are displayed.

Hardware and software information for all nodes installed in the router is displayed

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

The **show diag** command displays detailed information on the hardware components for each node, and on the status of the software running on each node.

## Task ID

| Task ID | Operations |
|---|---|
| sysmgr | read |

## Examples

The following example shows excerpts of output from the **show diag details** command:

```
Router# show diag details
Rack 0-Chassis IDPROM - Cisco 8201 1RU System with 24x400GE QSFP56-DD & 12x100GE QSFP28
 Info
    Controller Family          : 0045
    Controller Type            : 0613
    PID                        : 8201-SYS
    Version Identifier         : V00
    UDI Description            : Cisco 8201 1RU System with 24x400GE QSFP56-DD & 12x100GE
```

```
QSFP28
    Chassis Serial Number      : FOC2325NREU
    Top Assy. Part Number      : 68-6825-06
    Top Assy. Revision         : 09
    PCB Serial Number          : FOC2324NP35
    PCA Number                 : 73-19428-08
    PCA Revision               : 04
    CLEI Code                  : UNASSIGNED
    ECI Number                 : ECI123
    Deviation Number # 1       : 0
    Deviation Number # 2       : 0
    Deviation Number # 3       : 0
    Deviation Number # 4       : 0
    Deviation Number # 5       : 1126
    Manufacturing Test Data    : 00 00 00 00 00 00 00 00
    Calibration Data           : 00000000
    Chassis MAC Address        : 6c8b.d31f.d400
    MAC Addr. Block Size       : 512
    Hardware Revision          : 0.9
    Unknown Field (type 0x00d7): 0
    Device values # 1          : 21 80 84 0c 00 00 00 00

0/RP0/CPU0-Base Board IDPROM - Cisco 8201 1RU System with 24x400GE QSFP56-DD & 12x100GE
QSFP28
 Info
    Controller Family          : 0045
    Controller Type            : 0613
    PID                        : 8201-SYS
    Version Identifier         : V00
    UDI Description            : Cisco 8201 1RU System with 24x400GE QSFP56-DD & 12x100GE
QSFP28
    Chassis Serial Number      : FOC2325NREU
    Top Assy. Part Number      : 68-6825-06
    Top Assy. Revision         : 09
    PCB Serial Number          : FOC2324NP35
    PCA Number                 : 73-19428-08
    PCA Revision               : 04
    CLEI Code                  : UNASSIGNED
    ECI Number                 : ECI123
    Deviation Number # 1       : 0
    Deviation Number # 2       : 0
    Deviation Number # 3       : 0
    Deviation Number # 4       : 0
    Deviation Number # 5       : 1126
    Manufacturing Test Data    : 00 00 00 00 00 00 00 00
    Calibration Data           : 00000000
    Chassis MAC Address        : 6c8b.d31f.d400
    MAC Addr. Block Size       : 512
    Hardware Revision          : 0.9
    Unknown Field (type 0x00d7): 0
    Device values # 1          : 21 80 84 0c 00 00 00 00

HundredGigE0/0/0/0-IDPROM - Cisco QSFP28 100G SR4 Pluggable Optics Module
 Info
    IDPROM Format Revision   : 05
    Hardware Revision        : 1
    PID                      : QSFP-100G-SR4-S
    Version Identifier       : V02
    UDI Description          :
    CLEI Code                : CMUIAL8CAB
    ECI Number               : 0
    Top Assy. Part Number    : AFBR-89CDDZ-CS3
    Top Assy. Revision       : 05
    PCB Serial Number        : AVF2131S02J
```

```
    PCA Number               : N/A
    PCA Revision             : N/A
    Deviation Number # 1     : 0
    Asset ID                 :
    Asset Alias              :
..........
```

The output displayed for the **show diag details** command is the most comprehensive output displayed for **show diag** command variations. All other variations show a subset of the fields displayed except for the **show diag chassis**, **show diag fans**, and **show diag power-supply** commands, which also enable you to display EEPROM information.

```
RP/0/RP0/CPU0:P1#show diag eeprom
Thu Mar 12 18:16:32.436 UTC
Rack 0-Chassis IDPROM - Cisco 8201 1RU Chassis

 00:  1B 5C 04 FF 48 00 45 40 06 13 CB 92 38 32 30 31   .\..H.E@....8201
 10:  2D 53 59 53 00 00 00 00 00 00
 20:  30 30 00 DA 3C 43 69 73 63 6F 20 38 32 30 31 20   00..<Cisco 8201
 30:  31 52 55 20 43 68 61 73 73 69 73 00 00 00 00 00   1RU Chassis.....
 40:  00 00 00 00 00 00 00 00 00 00 00 00 2D 44 44      ............-DD
 50:  20 26 20 31 32 78 31 30 30 47 20 51 53 46 50 32    & 12x100G QSFP2
 60:  38 C2 8B 46 4F 43 32 32 31 37 45 4C 5A 4C 87 44   8..FOC2217ELZL.D
 70:  18 86 04 8D 30 37 20 20 C1 8B 46 4F 43 32 32 31   ....07  ..FOC221
 80:  39 5A 4F 55 47 E2 46 00 49 00 48 7E 05 8A 30 35   9ZOUG.F.I.H~..05
 90:  20 20 C6 8A 55 4E 41 53 53 49 47 4E 45 44 EB 86     ..UNASSIGNED..
 A0:  45 43 49 31 32 33 88 00 00 00 00 88 00 00 00 00   ECI123..........
 B0:  88 00 00 00 00 88 00 00 00 00 88 00 00 02 35 C4   ..............5.
 C0:  08 00 00 00 00 00 00 00 00 86 00 00 00 00 C3 06   ................
 D0:  78 99 52 4C D8 00 43 02 00 41 00 01 D7 44 00 00   x.RL..C..A...D..
 E0:  00 00 C9 08 2F 20 22 08 00 00 00 00 FF FF FF FF   ..../ ".........
 F0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 100:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 110:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 120:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 130:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 140:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 150:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 160:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 170:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 180:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 190:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 1A0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 1B0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 1C0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 1D0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 1E0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
 1F0:  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ................
```

This table describes the significant fields shown in the display.

***Table 1: show diag Field Descriptions***

| Field | Description |
|---|---|
| MAIN | Provides the following general information about the hardware:<br>• Board type<br>• Revision<br>• Device identifier<br>• Serial number |

| Field | Description |
|---|---|
| PCA | Cisco printed circuit assembly (PCA) hardware and revision number. |
| PID | Displays the product identifier (PID) revision for the specified node. |
| VID | Displays the version identifier (VID) for the specified node. |
| CLEI | Displays the common language equipment identifier (CLEI) for the specified node. |
| ECI | Displays the equipment catalog item (ECI) for the specified node. |
| Board State | Displays the current software on the board and whether or not the board is running. |
| PLD | Displays the information about the following programmable logic device (PLD) components on the current module:<br>• Processor<br>• Power<br>• MONLIB |
| SPEED | Displays speed information for the various components of the specified node, in megahertz. |
| MEM Size | Displays the memory size of the specified node, in megabytes. |
| RMA | Displays returned material adjustment (RMA) information for the specified node. |
| DIAGNOSTICS RESULTS | Provides the following information about the last diagnostics test that was run on the specified node:<br>• ENTRY 1<br>• TIMESTAMP—Time stamp for the last diagnostic test that was run on the node.<br>• VERSION<br>• PARAM1<br>• PARAM2<br>• TESTNUM—Identifies the test that was run on the node.<br>• RESULT—Displays whether the last diagnostic test passed or failed.<br>• ERRCODE |

# diagnostic monitor interval

To change the interval at which the online diagnostic tests send packets to the Network Processing Units (NPU) for a specific interval at a specified location, use the **diagnostic monitor interval** command in Config mode. To disable the configuration and restore the system to its original state, use the **no** form of this command.

**diagnostic monitor interval location** *node-id* **test** *test-name* *number-of-days* *hours:minutes:seconds.milliseconds*

**Syntax Description**

| | |
|---|---|
| *node-id* | Specifies a location where diagnostic monitoring was configured. The *node-id* argument is entered in the *rack/slot/module* notation. |
| *test-name* | Name of the diagnostic test. |
| *number-of-days* | Interval between each test run.<br>The *number-of-days* variable specifies the number of days between each test run. |
| *hours:minutes:seconds.milliseconds* | The *hours:minutes:seconds.milliseconds* variable specifies the test interval. Hours is a number in the range from 0 through 23, minutes is a number in the range from 0 through 59, seconds is a number in the range from 0 through 59, and milliseconds is a number in the range of 0 through 999. |

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.5.2/Release 7.3.5 | This command was introduced. |

**Usage Guidelines**

None

**Task ID**

| Task ID | Operations |
|---|---|
| diag | read, write |
| cisco-support | read |

**Examples**

This example shows how to set the diagnostic testing at an interval of 1 hour, 2 minutes, 3 seconds, and 4 milliseconds at location 0/1/CPU0:

```
Router# config
Router(config)# diagnostic monitor interval location 0/1/cpu0 test 1 0 1:2:3.4
```

# diagnostic monitor location disable

To disable automatic diagnostic testing for a specified location, use the **diagnostic monitor location disable** command in Config mode. To enable the diagnostic testing, use the **no** form of this command.

**diagnostic monitor location** *node-id* **test** *test-name* **disable**

**Syntax Description**

| | |
|---|---|
| *node-id* | Specifies a location where diagnostic monitoring was configured. The *node-id* argument is entered in the *rack/slot/module* notation. |

| | |
|---|---|
| *test-name* | Name of the diagnostic test. |
| disable | Disables diagnostic monitoring for a specified location. |

**Command Default**

By default, the automatic diagnostic tests are enabled in the system.

**Command Modes**

XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.5.2/Release 7.3.5 | This command was introduced. |

**Usage Guidelines**

None

**Task ID**

| Task ID | Operations |
|---|---|
| diag | read, write |
| cisco-support | read |

**Examples**

This example shows how to disable the online diagnostic execution at location 0/1/CPU0:

```
Router# config
Router(config)# diagnostic monitor location 0/1/cpu0 test 1 disable
```

# diagnostic monitor syslog

To enable the generation of a system log message when any online diagnosis fails, use the **diagnostic monitor syslog** command in Config mode. To remove the specified command from the configuration file and restore the system to its default condition, use the **no** form of this command.

**diagnostic monitor syslog**

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.5.2/Release 7.3.5 | This command was introduced. |

**Usage Guidelines**

None

**Task ID**

| Task ID | Operations |
|---|---|
| diag | read, write |
| cisco-support | read |

The following example shows how to generate a system log message when any online diagnostic test fails:

```
Router(config)# diagnostic monitor syslog
```

# diagnostic monitor threshold

To set the number of successive failures that triggers the generation of an NP data log, use the **diagnostic monitor threshold** command in Config mode. To remove the specified command from the configuration file and restore the system to its default condition, use the **no** form of this command.

**diagnostic monitor threshold location** *node-id* **test** *test-name* **failure-count** *failures*

**Syntax Description**

| | |
|---|---|
| *node-id* | Specifies a location where diagnostic monitoring was configured. The *node-id* argument is entered in the *rack/slot/module* notation. |
| *test-name* | Specifies the name of the diagnostic test. |
| *failures* | Number of test failures that are allowed. The given range is 1 to 99. |

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.5.2/Release 7.3.5 | This command was introduced. |

**Usage Guidelines**

None

**Task ID**

| Task ID | Operations |
|---|---|
| diag | read, write |
| cisco-support | read |

The following example shows how to set the failure threshold to 35 test failures for test 1 at location 0/1/CPU0:

```
Router# config
Router(config)# diagnostic monitor threshold location 0/1/cpu0 test 1 failure count 35
```

# show dataplane-health status

To check the status of a data plane health test and information on whether the test is still running or if it's completed, along with a summary of the results, use the **show dataplane-health status** command in XR EXEC mode.

**show dataplane-health status**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

XR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.5 | This command was introduced. |

## Usage Guidelines

Use the **show dataplane-health status** command to check the status of a data plane health test and information on whether the test is still running or if it's completed, along with a summary of the results.

## Task ID

| Task ID | Operations |
|---|---|
| system | read |

## Examples

This example displays the status of a data plane health test that is in progress:

```
Router# show dataplane-health status
Dataplane health monitoring in progress..
```

This example displays the status of a data plane health test that is completed and has errors:

```
Router# show dataplane-health status
Dataplane health monitoring completed
Summary of results (Module: fabric):
##############################################################################
Output summary legend:
ERROR: Tests were not run for this slice due to some errors
GOOD: Tests were successful for this slice
LOSS: Packet loss was observed for this slice
CORRUPT: Packet corruption was observed for this slice
##############################################################################
  LC    NP     Slice        GOOD        LOSS     CORRUPT       ERROR
------------------------------------------------------------------------------
------------------------------------------------------------------------------
   1     0         0     2526253           0           0           0
                   1     2527136           0           0           0
                   2     2526235           0           0           0
```

```
          1          0      2527166           0           0           0
                     1      2527217           0           0           0
                     2      2526424           0           0           0
-------------------------------------------------------------------------------
   2      0          0      2526733           0           0           0
                     1      2526948           0           0           0
                     2      2526554           0           0           0
          1          0      2526294           0           0           0
                     1      2526220           0           0           0
                     2      2526085           0           0           0
-------------------------------------------------------------------------------
   3      0          0      2525876           0           0           0
                     1      2526642           0           0           0
                     2      2525957           0           0           0
          1          0      2526491           0           0           0
                     1      2526263           0           0           0
                     2      2526200           0           0           0
          2          0      2526804           0           0           0
                     1      2526135           0           0           0
                     2      2526328           0           0           0
-------------------------------------------------------------------------------
   4      0          0       493934           0       11501           0
                     1            0           0           0           0
                     2            0           0           0           0
          1          0       493605           0       11591           0
                     1            0           0           0           0
                     2            0           0           0           0
-------------------------------------------------------------------------------
   5      0          0       505389           0          30           0
                     1            0           0           0           0
                     2            0           0           0           0
          1          0       505358           0          23           0
                     1            0           0           0           0
                     2            0           0           0           0
-------------------------------------------------------------------------------
   6      0          0      2526307           0           0           0
                     1      2525905           0           0           0
                     2      2526142           0           0           0
          1          0      2526755           0           0           0
                     1      2526603           0           0           0
                     2      2526607           0           0           0
********************************************************************************
Corruption detected:(LC4/0 <-> FC2/0) (LC4/1 <-> FC2/0) (LC5/0 <-> FC3/0) (LC5/1 <-> FC3/0)


********************************************************************************
FAILURES DETECTED IN DATAPATH for fabric mode.
Please run "monitor dataplane-health module no-fabric"
Please check /harddisk:/dph_mon/dataplane_health_fabric_mode_report.txt
********************************************************************************
```

# show diagnostic trace location

To display the logging information of the online diagnostic tests for a specific location, use the **show diagnostic trace location** command in Config mode.

**show diagnostic trace location** *node-id*

**Syntax Description**

| | |
|---|---|
| *node-id* | Specifies a location where diagnostic monitoring was configured. The *node-id* argument is entered in the *rack/slot/module* notation. |

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.5.2/Release 7.3.5 | This command was introduced. |

**Usage Guidelines**

None

**Task ID**

| Task ID | Operations |
|---|---|
| diag | read, write |
| cisco-support | read |

The following example shows the online diagnostic logging information at 0/1/CPU0 location:

```
Router# config
Router(config)# show diagnostic trace location 0/1/CPU0
Apr  1 18:09:38.180 diags/online/packet 0/1/CPU0 t5879 Sending a packet to SPP
Apr  1 18:09:38.180 diags/online/packet 0/1/CPU0 t5879 Sending a packet to SPP
Apr  1 18:09:38.180 diags/online/engineer 0/1/CPU0 t5879 Now sending a pak(seq 1276),
destination slot 1 (card type 0x2), NP 0
Apr  1 18:09:38.180 diags/online/engineer 0/1/CPU0 t5879 Now sending a pak(seq 1276),
destination slot 1 (card type 0x2), NP 1
Apr  1 18:09:38.180 diags/online/engineer 0/1/CPU0 6904# t5879 Slot 1 has 2 NPs for NPU
loopback test, Inactive NP mask: 0x0
Apr  1 18:09:38.180 diags/online/engineer 0/1/CPU0 7456# t5879 Packets sent, time
tick=77148425000000
Apr  1 18:09:38.190 diags/online/gold_message 0/1/CPU0 9188# t5879 0/1/CPU0:
SFNPULoopback{ID=1} Completed Successfully.
Apr  1 18:09:38.190 diags/online/gold_message 0/1/CPU0 9740# t5879 0/1/CPU0: running parallel
 test...
Apr  1 18:09:38.190 diags/online/engineer 0/1/CPU0 8008# t5879 Time took to receive 2 pkts:
 10000000 nsec, timeout val: 500000000 nsec
Apr  1 18:09:38.190 diags/online/engineer 0/1/CPU0 8560# t5879 Successfully verified a
packet, seq. no.: 1276
Apr  1 18:09:38.190 diags/online/engineer 0/1/CPU0 9112# t5879 Successfully verified a
packet, seq. no.: 1276
Apr  1 18:09:38.190 diags/online/engineer 0/1/CPU0 9664# t5879 exp_mask: 0x00000003 mask:
0x00000003, err_mask: 0x00000000
```

# show diagnostic result

To display diagnostic test results, use the **show diagnostic result** command in EXEC mode.

**show diagnostic result location** *node-id* **[detail]**

**Syntax Description**

| | |
|---|---|
| **location** *node-id* | Displays the diagnostic test results for a specified location. The *node-id* argument is entered in the *rack/slot/module* notation. |
| **detail** | (Optional) Specifies detailed results. |

**Command Default**

None

**Command Modes**

Exec mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.5.2/Release 7.3.5 | This command was introduced. |

**Usage Guidelines**

None

**Task ID**

| Task ID | Operations |
|---|---|
| diag | read, write |
| cisco-support | read |

The following example shows the online diagnostic test results at 0/5/CPU0 location:

```
Router#show diagnostic result location 0/5/CPU0

0/5/CPU0:

Overall diagnostic result: PASS
Diagnostic level at card bootup: bypass

Test results: (. = Pass, F = Fail, U = Untested)

1 ) SFNPULoopback --------------------> .


Router#show diagnostic result location 0/5/CPU0 detail

0/5/CPU0:

Overall diagnostic result: PASS
Diagnostic level at card bootup: bypass

Test results: (. = Pass, F = Fail, U = Untested)

_____________________________________________________________________________

1 ) SFNPULoopback --------------------> .

Error code ------------------> 0 (DIAG_SUCCESS)
Total run count -------------> 31
Last test execution time ----> Fri Jun 9 08:28:39 2023
First test failure time -----> n/a
Last test failure time ------> n/a
```

```
Last test pass time ---------> Fri Jun 9 08:28:39 2023
Total failure count ---------> 0
Consecutive failure count ---> 0
```

# monitor dataplane-health

To monitor the health of data plane components including fabric and NPUs, use the **monitor dataplane-health** command in EXEC mode.

**monitor dataplane-health [ module** { **all-modules** | **fabric** | **lookup** | **no-fabric** } **pattern** { *byte-pattern* | **default-patterns** } **duration** *test-duration* **gap** *time-gap* **report** { **detail** | **summary** } **location** { **all** | *node-id* } **stop-on-failure-for-lc** { **false** | **true** } **prompt ]**

## Syntax Description

| | |
|---|---|
| **all-modules** | (Optional) Checks the fabric path, NPU path, and lookup engine for issues. |
| **fabric** | (Optional) Checks the fabric path for issues. |
| **lookup** | (Optional) Checks the lookup engine for issues. |
| **no-fabric** | (Optional) Checks the NPU path for issues. |
| **pattern** { *byte-pattern* \| **default-patterns** } | (Optional) Specifies the data pattern that the utility uses to detect datapath memory corruption. You can either provide a custom byte pattern as a hexadecimal value within the range 0x00 to 0xFF (0 to 255), or use one of the available default patterns: 0x00, 0xF0, 0x0F, 0xFF, or 0x55. |
| **duration** *test-duration* | (Optional) Specifies the duration for which the traffic tests are run for each pattern. The default duration is 10 seconds per pattern. For example, if the default pattern is used, and duration is specified as 10 seconds, the test traffic runs for 50 seconds. Range is 1–60 seconds. |
| **gap** *time-gap* | (Optional) Specifies the time interval between traffic test runs on consecutive NPU slices. Default gap is 5 seconds. Range is 1–30 seconds. |
| **report** { **detail** \| **summary** } | (Optional) Displays the summary or detailed report. By default, the summary report is displayed. Detail option displays more detailed information. In both cases, a detailed report (regardless of the selected option) is saved at the location: `harddisk:/dataplane_health_detail_report.txt`<br>**Note**<br>You must archive the report file before subsequent runs, as this file is overwritten on re-execution of the command. |
| **location** { **all** \| *node-id* } | (Optional) Specifies the line card on which the utility is run. By default, the utility is executed on all LCs in the system. You can also choose a specific LC if necessary. |
| **stop-on-failure-for-lc** { **true** \| **false** } | (Optional) Specifies if the testing must stop or continue when the utility detects an issue. If **true** (default) option is selected, the testing stops when an issue is detected. If **false** is selected, the testing continues to completion even after an issue is detected on the LC. |

| | |
|---|---|
| **prompt** | (Optional) Displays a warning message on the impact of this utility, and prompts for your confirmation to run this utility. If you choose NO to the prompt, tests will not be executed. By default, the utility does not prompt for your confirmation. |

**Command Default**

Monitoring is disabled.

**Command Modes**

EXEC mode

**Command History**

| Release | Modification |
|---|---|
| Release 25.4.1 | Two new keywords, **all-modules** and **lookup**, were added to the **monitor dataplane-health module** command to monitor the data plane health for all paths and only lookup engines respectively. |
| Release 7.3.5 | This command was introduced. |

**Usage Guidelines**

Use the **monitor dataplane-health** command to run the data plane health check utility. Do not use this command on a router that carries live traffic, as this utility affects the system performance. Use this command only on an isolated router.

A router reload is required after the **monitor dataplane-health module all-modules** command to run fabric, NPU, and lookup engine tests and the **monitor dataplane-health module lookup** command to run the lookup engine test are completed.

**Task ID**

| Task ID | Operations |
|---|---|
| system | execute |
| basic-services | |

This example shows how to execute the data plane health check utility for fabric module.

```
RP/0/RP0/CPU0:Router# monitor dataplane-health
Wed Feb 28 15:08:15.659 EST
RP/0/RP0/CPU0:Feb 28 15:08:15.687 EST: dph_mon_bg[337]: %PLATFORM-DPH_MONITOR-6-STARTED :
Dataplane health monitoring started. Please check
harddisk:/dph_mon/dataplane_health_fabric_mode_report.txt for details of the result.
THIS COMMAND IMPACTS SYSTEM PERFORMANCE AND SHOULD IDEALLY BE RUN ON A ROUTER THAT IS
ISOLATED.
Progressive details of the test logged in
/harddisk:/dph_mon/dataplane_health_fabric_mode_report.txt
Previous log file is archived as
/harddisk:/dph_mon/dataplane_health_fabric_mode_report.txt.bak
Please save the archive with a different file name as needed
################################################################################
Module:fabric
Patterns used: 0xf0,0x0f,0x00,0x55,0xff,
Duration per pattern: 10 seconds
Pause time between each slice/pattern test: 2 seconds
Best effort time for test completion: 1044 seconds
Depending on the overall dataplane health state, it may take additional time to complete
################################################################################
```

```
Dataplane health monitoring will run in the background
Wait for completion log message from the process "dph_mon_bg"
   OR
Use "show dataplane-health status" regularly to check for completion

RP/0/RP0/CPU0:Feb 28 15:08:15.687 EST: dph_mon_bg[337]: %PLATFORM-DPH_MONITOR-6-STARTED :
Dataplane health monitoring started. Please check
harddisk:/dph_mon/dataplane_health_fabric_mode_report.txt for details of the result.

RP/0/RP0/CPU0:Feb 28 15:28:54.965 EST: dph_mon_bg[337]: %PLATFORM-DPH_MONITOR-6-COMPLETED
: Dataplane health monitoring completed. Please check
harddisk:/dph_mon/dataplane_health_fabric_mode_report.txt for details of the result.
```

This example shows the output for data plane health check for fabric module.

```
RP/0/RP0/CPU0:Router# show dataplane-health status
Mon Jan 29 23:10:21.564 UTC
Dataplane health monitoring completed
Summary of results (Module: fabric):
################################################################################
Output summary legend:
ERROR: Tests were not run for this slice due to some errors
GOOD: Tests were successful for this slice
LOSS: Packet loss was observed for this slice
CORRUPT: Packet corruption was observed for this slice
################################################################################
  LC    NP     Slice        GOOD        LOSS     CORRUPT       ERROR
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
   0     0         0     2526099           0           0           0
                   1     2526856           0           0           0
                   2     2526529           0           0           0
         1         0     2526590           0           0           0
                   1     2526918           0           0           0
                   2     2526421           0           0           0
         2         0     2526665           0           0           0
                   1     2525818           0           0           0
                   2     2526286           0           0           0
--------------------------------------------------------------------------------
   1     0         0     2526754           0           0           0
                   1     2526328           0           0           0
                   2     2526695           0           0           0
         1         0     2525892           0           0           0
                   1     2526988           0           0           0
                   2     2526215           0           0           0
********************************************************************************
DATAPATH CHECK IS CLEAN (mode: fabric).
********************************************************************************
```

This example shows how to run fabric, NPU, and lookup engine tests together to check the health of these modules. The test includes a warning to reload the router after the test completion.

**Warning** Run the test on a router in a non-production environment as this test impacts the system performance.

```
RP/0/RP0/CPU0:Router# monitor dataplane-health module all-modules
THIS COMMAND IMPACTS SYSTEM PERFORMANCE AND SHOULD IDEALLY BE RUN ON A ROUTER THAT IS
ISOLATED.
THIS COMMAND INCLUDES THE TESTING OF THE LOOKUP ENGINE MODULE.
TESTS ON THE LOOKUP ENGINE REQUIRE A RELOAD AFTER TEST COMPLETION.
```

```
Progressive details of the test logged in
/harddisk:/dph_mon/dataplane_health_fabric_mode_report.txt
Previous log file is archived as
/harddisk:/dph_mon/dataplane_health_fabric_mode_report.txt.bak
Please save the archive with a different file name as needed
################################################################################################
Module:fabric, lookup
Patterns used: 0xf0,0x55,0xff,
Duration per pattern: 10 seconds
Pause time between each slice/pattern test: 2 seconds
Best effort time for test completion: 492 seconds
Depending on the overall dataplane health state, it may take additional time to complete
################################################################################################
Dataplane health monitoring will run in the background
Wait for completion log message from the process "dph_mon_bg"
           OR
Use "show dataplane-health status" regularly to check for completion
RP/0/RP0/CPU0:Router#
```

**Note** Make sure to reload the router once the test is completed.

This example shows the output for data plane health check with fabric, NPU, and lookup engine tests.

```
RP/0/RP0/CPU0:Router# show dataplane-health status
Dataplane health monitoring completed

Datapath test on all requested LC/NPU/slice completed
Summary of results (Module: fabric/lookup):
################################################################################################
Output summary legend:
ERROR: Tests were not run for this slice due to some errors
GOOD: Tests were successful for this slice
LOSS: Packet loss was observed for this slice
CORRUPT: Packet corruption was observed for this slice
LE-FAULTS: Lookup Engine test results - On supported devices, 0 for no-failures,
           or a bit mask of faulty engines; n/a on unsupported devices
################################################################################################
------------------------------------------------------------------------------------------------
  LC     NP      Slice        GOOD        LOSS     CORRUPT       ERROR      LE-FAULTS
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
 LC4      0          0     1663101           0           0           0         0x0000
                     1     1661542           0           0           0         0x0200
                     2     1662258           0           0           0         0x0000
          1          0     1667547           0           0           0         0x0000
                     1     1661658           0           0           0         0x0000
                     2     1661153           0           0           0         0x0000
          2          0     1661087           0           0           0         0x0000
                     1     1661474           0           0           0         0x0000
                     2     1660756           0           0           0         0x0000
          3          0     1661936           0           0           0         0x0000
                     1     1661197           0           0           0         0x0000
                     2     1661644           0           0           0         0x0000
*********************************************************************************#******
DATAPATH CHECK IS CLEAN, LOOKUP ENGINE ERROR(s) DETECTED (mode: fabric/lookup).
*********************************************************************************
```

This example shows how to run lookup engine test. The test includes a warning to reload the router after the test completion.

**Warning** Run the test on a router in a non-production environment as this test impacts the system performance.

```
RP/0/RP0/CPU0:ios# monitor dataplane-health module lookup
Fri Oct 24 13:09:20.906 UTC
THIS COMMAND IMPACTS SYSTEM PERFORMANCE AND SHOULD IDEALLY BE RUN ON A ROUTER THAT IS
ISOLATED.
THIS COMMAND INCLUDES THE TESTING OF THE LOOKUP ENGINE MODULE.
TESTS ON THE LOOKUP ENGINE REQUIRE A RELOAD AFTER TEST COMPLETION.
Progressive details of the test logged in
/harddisk:/dph_mon/dataplane_health_lookup_mode_report.txt
Previous log file is archived as
/harddisk:/dph_mon/dataplane_health_lookup_mode_report.txt.bak
Please save the archive with a different file name as needed
################################################################################
Module:lookup
Best effort time for test completion: 32 seconds
Depending on the overall dataplane health state, it may take additional time to complete
################################################################################
Dataplane health monitoring will run in the background
Wait for completion log message from the process "dph_mon_bg"
           OR
Use "show dataplane-health status" regularly to check for completion
```

**Note** Make sure to reload the router once the test is completed.

This example shows the output for data plane health check with lookup engine test.

```
RP/0/RP0/CPU0:Router# show dataplane-health status

Thur Nov 20 18:06:25.132 UTC
Dataplane health monitoring completed

Datapath test on all requested LC/NPU/slice completed
Summary of results (Module: lookup):
################################################################################
Output summary legend:
LE-FAULTS: Lookup Engine test results - On supported devices, 0 for no-failures,
           or a bit mask of faulty engines; n/a on unsupported devices

################################################################################
--------------------------------------------------------------------------------
  LC    NP     Slice    LE-FAULTS
---------------------------------
---------------------------------
   4     0         0       0x0000
                   1       0x0200
                   2       0x0000
         1         0       0x0000
                   1       0x0000
                   2       0x0000
         2         0       0x0000
                   1       0x0000
                   2       0x0000
         3         0       0x0000
                   1       0x0000
```

```
                    2       0x0000

*************************************************************************
LOOKUP ENGINE ERROR(s) DETECTED (mode: lookup).
*************************************************************************
```

This example shows the output for data plane health check with lookup engine test on unsupported devices.

```
RP/0/RP0/CPU0:Router# show dataplane-health status

Wed Feb 11 20:13:45.429 UTC
Dataplane health monitoring completed

Datapath test on all requested LC/NPU/slice completed
Summary of results (Module: lookup):
#########################################################################################
Output summary legend:
LE-FAULTS: Lookup Engine test results - On supported devices, 0 for no-failures,
           or a bit mask of faulty engines; n/a on unsupported devices
#########################################################################################
----------------------------------------------------------------------------------------
  LC    NP     Slice   LE-FAULTS
----------------------------------------------------------------------------------------
 RP0     0         0         n/a
                   1         n/a
*****************************************************************************************

*****************************************************************************************
DATAPATH CHECK IS CLEAN (mode: lookup).
*****************************************************************************************
```

# clear packet-trace conditions all

To clear all buffered packet-trace conditions, use the **clear packet-trace conditions all** command in EXEC mode.

**clear packet-trace conditions all**

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 25.1.1 | This command was introduced. |

**Usage Guidelines**

You can use this command only when packet tracing is inactive.

**Task ID**

| Task ID | Operations |
|---|---|
| cisco-support | execute |

### Example

The following example shows how clear all buffered packet tracer conditions:

```
Router# clear packet-trace conditions all
```

# clear packet-trace counters all

To clear all packet trace counters and reset the counters to zero, use the **clear packet-trace counters all** command in EXEC mode.

**clear packet-trace counters** *all*

**Syntax Description** This command has no keywords or arguments.

**Command Default** None

**Command Modes** XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 25.1.1 | This command was introduced. |

**Task ID**

| Task ID | Operations |
|---|---|
| cisco-support | execute |

### Example

The following example shows how clear all the packet tracer counters:

```
Router# clear packet-trace counters all
```

# packet-trace condition

To configure the defined conditions and to apply the conditions for packet tracing, use the **packet-trace condition** command in EXEC mode.

**packet-trace condition** { **interface** { **all** | *interface-path-id* } | **n offset** *offset* **value** *value* **mask** *mask* }

**Syntax Description**

| | |
|---|---|
| **interface** *interface-path-id* | Physical interface.<br>**Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router.<br><br>For more information about the syntax for the router, use the question mark (?) online help function. |
| **n** | Specifies the number of offset/value/mask set(s). |
| **offset** *offset* **mask** *mask* **value** *value* | The offset/value/mask triplets that define a flow of interest.<br>**Note** Defining a flow as a set of Offset/Value/Mask triplets allows the packet tracer framework to remain entirely protocol- agnostic. These triplets can represent any segment of any header within the protocol stack. Ensure that both the value and mask do not exceed 4 bytes in size. You can configure conditions at any offset within the first 128 bytes of a given packet, using a value and mask of up to 4 bytes each. |

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 25.1.1 | This command was introduced. |

**Task ID**

| Task ID | Operations |
|---|---|
| cisco-support | read, execute |

### Example

The following example shows how to configure the defined conditions for packet tracing:

```
Router# packet-trace condition interface all
Router# packet-trace condition 1 offset 0 value 0xaabbcc mask 0xffffff
Router# packet-trace condition 2 offset 3 value 0xdd mask 0xff
```

**Note** You must also use the **interface all** condition along with the defined conditions, as this feature is enabled on all NPUs rather a specific interface.

# packet-trace start

To start packet tracing, use the **packet-trace start** command in EXEC mode.

**packet-trace start**

**Syntax Description** This command has no keywords or arguments.

**Command Default** None

**Command Modes** XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 25.1.1 | This command was introduced. |

**Task ID**

| Task ID | Operations |
|---|---|
| cisco-support | execute |

**Example**

The following example shows how to start packet tracing:

```
Router# packet-trace start
```

# packet trace stop

To stop packet tracing, use the **packet-trace stop** command in EXEC mode.

**packet-trace stop**

**Syntax Description** This command has no keywords or arguments.

**Command Default** None

**Command Modes** XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 25.1.1 | This command was introduced. |

**Task ID**

| Task ID | Operations |
|---|---|
| cisco-support | execute |

### Example

The following example shows how to stop packet tracing:

```
Router# packet-trace stop
```

# show packet-trace decode

To view detailed decoded packet trace information for a specific instance and location, use the **show packet-trace decode** command in EXEC mode.

**show packet-trace [ decode** { **all** | **raw** | **rxpp-forwarding** | **rxpp-termination** | **tm-pd** | **txpp-transmit** } **instance** { *instance-id* | *all* } **location** { *node-id* | *path* | *all* } **]**

### Syntax Description

| | |
|---|---|
| **all** | Decodes all packet trace data. |
| **raw** | Dumps raw binary trace data. |
| **rxpp-forwarding** | Decodes RXPP forwarding trace data. |
| **rxpp-termination** | Decodes RXPP termination trace data. |
| **tm-pd** | Decodes TM PD trace data. |
| **txpp-transmit** | Decodes TXPP transmit trace data. |
| **instance** *instance-id* | Displays decoded packet trace information for a given NPU instance. |
| **instance** *all* | Displays decoded packet trace information for all NPU instances. |
| **location** *node-id* | Specifies the node. The *node-id* argument is entered in the *rack/slot/module* notation. Displays the information about a specific node. |
| **location** *all* | Displays decoded information for all nodes. |

### Command Default

None

### Command Modes

XR Config mode

### Command History

| Release | Modification |
|---|---|
| Release 25.3.1 | This command was introduced. |

### Task ID

| Task ID | Operations |
|---|---|
| cisco-support | read, write |

## Example

The following example shows how to view detailed decoded packet trace information for the given instance and location:

```
Router# show packet-trace decode all instance 0 location 0/0/CPU0

Mon Aug 18 21:31:30.699 UTC
Decoding packet trace results for NPU id 0 on node id 0x0
Executing User File: /pkg/bin/packet_trace_decode.py
Writing decoded output to /tmp/decoded.txt and /tmp/decoded.json
######## Captured Packet ########
###[ Ethernet ]###
  dst       = 12:34:56:78:9a:bd
  src       = 00:ca:fe:de:ad:00
  type      = n_802_1Q
###[ 802.1Q ]###
     prio      = 0
     id        = 0
     vlan      = 0
     type      = 0x7103
###[ InjectUp ]###
        type      = 34
        sgt_trust = 0
        padding1  = 0
        phb_src   = 0
        phb_tc    = 3
        phb_dp    = 0
        qos_group = 0
        fwd_qos_tag= 0
        ssp_gid   = 4
        counter_ptr= 84
        padding2  = 0
        ts_opcode = 0
        ts_res2   = 0
        ts_offset = 0
        lm_opcode = 0
        lm_res2   = 0
        lm_offset = 0
        internal  = 0
        ext_type  = 0
###[ Ethernet ]###
           dst       = 04:bd:97:bc:fb:a0
           src       = 00:00:00:00:00:01
           type      = IPv4
###[ IP ]###
              version   = 4
              ihl       = 5
              tos       = 0x0
              len       = 328
              id        = 1
              flags     =
              frag      = 0
              ttl       = 64
              proto     = udp
              chksum    = 0x4f8f
              src       = 10.1.1.10
              dst       = 30.1.1.10
              \options   \
###[ UDP ]###
                 sport     = domain
                 dport     = domain
                 len       = 308
```

```
                  chksum    = 0xc3f7
###[ DNS ]###
                     id        = 24929
                     qr        = 0
                     opcode    = 12
                     aa        = 0
                     tc        = 0
                     rd        = 1
                     ra        = 0
                     z         = 1
                     ad        = 1
                     cd        = 0
                     rcode     = format-error
                     qdcount   = 24929
                     ancount   = 24929
                     nscount   = 24929
                     arcount   = 24929
                     qd        = ''
                     an        = ''
                     ns        = ''
                     ar        = ''
###[ Raw ]###
                        load      = b'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\x00
\x00[illegible]
l3_slp: 0x1
The Interface name for l3_slp 0x1 is HU0/0/0/0

SLICE 0 OUTPUT:
===============

SLICE 1 OUTPUT:
===============

SLICE 2 OUTPUT:
===============

Packet Bytes(up to 128B) :
[illegible]

==== FI Array ====
-------------------------------------------------------------------------------------------------------------
FIELD_NAME                        =>                                VALUE
-------------------------------------------------------------------------------------------------------------
array[9]                          =>                                  0x0
array[8]                          =>                                  0x0
array[7]                          =>                                  0x0
array[6]                          =>                               0x4d00
array[5]                          =>                               0x450f
  offset_in_bytes                                                    0x45
  protocol_type                                       PROTOCOL_TYPE_UDP
array[4]                          =>                               0x3114
  offset_in_bytes                                                    0x31
  protocol_type                                    PROTOCOL_TYPE_IPV4_L4
array[3]                          =>                               0x2301
  offset_in_bytes                                                    0x23
  protocol_type                                   PROTOCOL_TYPE_ETHERNET
array[2]                          =>                               0x121f
  offset_in_bytes                                                    0x12
  protocol_type                                     PROTOCOL_TYPE_INJECT
array[1]                          =>                               0xe48
  offset_in_bytes                                                    0xe
  protocol_type                                       PROTOCOL_TYPE_VLAN
  flags                                                               2
array[0]                          =>                                0x11
```

```
  offset_in_bytes                                    0x0
  protocol_type                  PROTOCOL_TYPE_ETHERNET_VLAN

==== RXPP Termination Input ====

------------------------------------------------------------------------------------------------------------
FIELD_NAME                    =>                                VALUE
------------------------------------------------------------------------------------------------------------
padding_1                     =>                                  0x0
pch_label                     =>                                  0x0
unsch_rcy_code                =>                                  0x0
tx_to_rx_rcy_data             =>                                 0xeb
mtu_violation                 =>                                  0x1
initial_tc                    =>                                  0x0
offset_in_fragment            =>                                  0x0
slice_source_system_port      =>                                 0xc0
processing_code               =>                                  0x0
destination                   =>                                  0x0
use_cache                     =>                                  0x0
single_fragment_packet        =>                                  0x0
flow_signature_on_npuh        =>                                  0x0
phb                           =>                                  0x0
reassembly_context            =>                                0x6fc
learn_enable                  =>                                  0x0
receive_time_from_nppd        =>                                  0x0
rxnpu_recycle_count           =>                                  0x0
rxnpu_recycle_data            =>                                  0x0

==== RXPP Termination Output ====

------------------------------------------------------------------------------------------------------------
FIELD_NAME                    =>                                VALUE
------------------------------------------------------------------------------------------------------------
learn_cmd_and_fwd_app_data    =>                           0xeb800000
lb_command                    =>                                  0x0
offset_in_fragment            =>                                  0x0
slice_source_system_port      =>                                 0xc0
processing_code               =>                                  0x0
destination                   =>                                  0x0
use_cache                     =>                                  0x0
single_fragment_packet        =>                                  0x0
flow_signature_on_npuh        =>                                  0x0
phb                           =>                                  0x0
reassembly_context            =>                                0x6fc
learn_enable                  =>                                  0x0
receive_time_from_nppd        =>                                  0x0
rxnpu_recycle_count           =>                                  0x0
rxnpu_recycle_data            =>                                  0x0

==== RXPP Forwarding Output ====

------------------------------------------------------------------------------------------------------------
FIELD_NAME                    =>                                VALUE
------------------------------------------------------------------------------------------------------------
offset_in_fragment            =>                                  0x0
slice_source_system_port      =>                                 0xc0
processing_code               =>                                  0x0
destination                   =>                               0x74007
use_cache                     =>                                  0x0
single_fragment_packet        =>                                  0x0
flow_signature_on_npuh        =>                                  0x0
phb                           =>                                  0xc
reassembly_context            =>                                0x6fc
learn_enable                  =>                                  0x0
```

```
receive_time_from_nppd        =>                    0x0
rxnpu_recycle_count           =>                    0x0
rxnpu_recycle_data            =>                    0x0
fllb_control_code             =>                    0x0
first_hop_source_ip_learn     =>                    0x0
ethernet_rate_limiter_type    =>                    0x7
fwd_offset_cmd                =>                    0x2
```

# show packet trace description

To view all counters registered with the packet tracer framework along with their descriptions, use the **show packet-trace description** command in EXEC mode.

**show packet-trace description**

**Syntax Description** This command has no keywords or arguments.

**Command Default** None

**Command Modes** XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 25.1.1 | This command was introduced. |

**Task ID**

| Task ID | Operations |
|---|---|
| cisco-support | read |

### Example

The following example shows how to view all counters registered with the packet tracer framework.

```
Router# show packet-trace description
```

# show packet-trace hw-status

To view the packet tracing filter configuration in hardware, use the **show packet-trace hw-status** command in EXEC mode.

**show packet-trace hw-status instance** *instance-id* **location** *node-id*

**Syntax Description**

| | |
|---|---|
| **instance***instance-id* | Displays information about a given instance. |
| **location***node-id* | Specifies a location where packet trace was configured. The *node-id* argument is entered in the *rack/slot/module* notation. |

**Command Default** None

**Command Modes** XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 25.1.1 | This command was introduced. |

**Task ID**

| Task ID | Operations |
|---|---|
| cisco-support | read, write |

### Example

The following example shows how to view packet tracing filter configuration in hardware:

```
Router# show packet-trace hw-status instance 0 location 0/RP0/CPU0

Mon Jul  7 20:57:47.488 UTC
Reading hardware packet trace status for NPU id 0 on node id 0x2000
=== HW filter cfg results for NP0 ===
{
    "Rx Filter Config": [
        {
            "data_offset": 48,
            "data": [
                200
            ],
            "mask": [
                255
            ]
        },
        {
            "data_offset": 0,
            "data": [
                100
            ],
            "mask": [
                255
            ]
        }
    ],
    "Tx Filter Config": [
        {
            "data_offset": 48,
            "data": [
                200
            ],
            "mask": [
                255
            ]
        },
        {
            "data_offset": 0,
            "data": [
                100
            ],
```

```
                "mask": [
                    255
                ]
            }
        ]
}
```

# show packet-trace result

To view the packet tracing result, use the **show packet-trace result** command in EXEC mode.

**show packet-trace result**

**Command Default** None

**Command Modes** XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 25.1.1 | This command was introduced. |

**Task ID**

| Task ID | Operations |
|---|---|
| cisco-support | read, write |

### Example

The following example shows how to view packet tracing results:

```
Router# show packet-trace result

Location     | Source       | Counter      | T | Last-Attribute       | Count
-------------  -------------  -------------  -   -----------------   ---------------
0/RP0/CPU0     NP0            INGRESS        P                           1
0/RP0/CPU0     NP0            EGRESS         P                           1
```

The count value of 1 indicates that the incoming packet has matched the configured filter in both the ingress and egress pipelines.

# show packet trace status

To check the status of tracing (active/incative) and to view the conditions buffered so far by the packet tracing framework, use the **show packet-trace status** command in EXEC mode. To view more detailed status, such as processes that are registered with the packet tracer framework on every card in the router, use the **show packet-trace status detail** command.

**show packet-trace status [detail]**

**Syntax Description** This command has no keywords or arguments.

**Command Default** None

**Command Modes** XR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 25.1.1 | This command was introduced. |

**Task ID**

| Task ID | Operations |
|---|---|
| cisco-support | read |

### Example 1

The following example shows how to view the status of packet tracing.

```
Router# show packet-trace status

Packet Trace Master Process:

  Buffered Conditions:
    Interface HundredGigE0/5/0/6
    1 offset 53 value 0x1 mask 0xff
    2 offset 56 value 0xc0a84d01 mask 0xffffffff
    3 offset 60 value 0xc0a84d02 mask 0xffffffff

  Status: Active
```

The output displays the status as either **Active** or **Inactive**. The status will remain **Inactive** until tracing starts.

### Example 2

The following example shows how to view detailed status of packet tracing.

```
Router# show packet-trace status detail

Packet Trace Master Process:

  Buffered Conditions:

  Status: Inactive

------------------------------------------------------------
Location: 0/RP0/CPU0

Available Counting Modules: 5

  #1 SPP
    Last errors:

  #2 NETIO
    Last errors:
```

```
  #3 UDP
    Last errors:

  #4 TCP
    Last errors:

  #5 ofa_npd_proxy
   Last errors:

Available Marking Modules: 1

  #1 ofa_npd_proxy

    Interfaces: 0

    Conditions: 0

    Last errors:
```

The output displays the status as either **Active** or **Inactive**. The status will remain **Inactive** until tracing starts.