ASIC Driver Commands

This module describes the commands used to configure and monitor the application-specific integrated circuit (ASIC) driver on a router running Cisco IOS XR software.

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.

asic-scan egressq (block number)

To run, enable, or disable a generic ASIC scan on a specific egress queue ASIC block ID or error cause, use the **asic-scan egressq** command in EXEC mode.

asic-scan egressq *block_number* [{[*error_cause*] [{**disable** | **enable**}] | **help-cause**}] **location** *node-id*

Syntax Description

block_number	Identifies the block on which to enable or disable the ASIC scan.
error_cause	(Optional) Identifies an error cause in the specified block. When you include the *error_cause* argument in the **asic-scan egressq** command syntax, the command takes place on the specified error cause.
disable	(Optional) Disables the ASIC scan on the specified block or error cause.
enable	(Optional) Enables the ASIC scan on the specified block or error cause.
help_cause	(Optional) Displays a list of all error causes in the specified block.
location *node-id*	Identifies the location of the node hosting the egress queue on which to run, enable, or disable an ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.5.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to display a list of all error causes in the specified block:

```
RP/0/RP0/CPU0:router# asic-scan egressq 10 help-cause location 0/6/CPU0

Total 6 cause(s)

Cause Number          Cause Name
0                     lbm_owl_ovl
1                     scr_m4_ovl
2                     scr_m3_ovl
3                     scr_m2_ovl
4                     scr_m1_ovl
5                     scr_m0_ovl
```

The following example shows how to run a scan on the egress queue ASIC block ID 30 located on the CPU node in slot 1:

```
RP/0/RP0/CPU0:router# asic-scan egressq 30 location 0/6/CPU0
```

asic-scan egressq disable

To disable an ASIC scan on all causes in all blocks on the specified node, use the **asic-scan egressq disable** command in EXEC mode.

asic-scan egressq disable location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node on which to disable the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation.

Note Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.5.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to disable an ASIC scan on all causes in all blocks on the egress queue that is located on the CPU node in slot 6:

```
RP/0/RP0/CPU0:router# asic-scan egressq disable location 0/6/CPU0
```

asic-scan egressq enable

To enable an ASIC scan on all causes in all blocks on the specified node, use the **asic-scan egressq enable** command in EXEC mode.

asic-scan egressq enable location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node on which to enable the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation.

Note Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.5.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to enable an ASIC scan on all causes in all blocks on the egress queue that is located on the CPU node in slot 6:

```
RP/0/RP0/CPU0:router# asic-scan egressq enable location 0/6/CPU0
```

asic-scan egressq help-block

To enable an ASIC scan that lists all the blocks on the specified node, use the **asic-scan egressq help-block** command in EXEC mode.

asic-scan egressq help-block location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node on which to enable the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

None

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.6.0	This command was introduced.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to enable an ASIC scan that lists all block numbers on the egress queue that is located on the CPU node in slot 6:

```
RP/0/RP0/CPU0:router# asic-scan egressq help-block location 0/6/CPU0
```

asic-scan egressq location

To run an ASIC scan on all the egress queues on a specific node, use the **asic-scan egressq location** command in EXEC mode.

asic-scan egressq location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node hosting the egress queues on which to run the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

None

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.5.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to run an ASIC scan on all egress queues located on the CPU node in slot 1:

```
RP/0/RP0/CPU0:router# asic-scan egressq location 0/1/CPU0
```

asic-scan egressq quick-scan

To run a quick ASIC scan on a specific file or on all bits of the chip, use the **asic-scan egressq quick-scan** command in EXEC mode.

asic-scan egressq quick-scan {*filename* | **allscan**} **location** *node-id*

Syntax Description

filename	Name of the .xsvf file on which to run the quick ASIC scan
allscan	Performs a quick ASIC scan on all bits of the chip, including the preread for Metro.
location *node-id*	Identifies the location of the node hosting the egress queue on which to run the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

None

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.5.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to run a quick ASIC scan on all bits of the chip on the egress queue that is located on the CPU node in slot 1:

```
RP/0/RP0/CPU0:router# asic-scan egressq quick-scan allscan location 0/1/CPU0
```

asic-scan pse egress (block number)

To run, enable, or disable a generic ASIC scan on a specific egress packet switching engine (PSE) device block or error cause, use the **asic-scan pse egress** command in EXEC mode.

asic-scan pse egress *block_number* [{[*error_cause*] [{**disable**|**enable**}]|**help-cause**}] **location** *node-id*

Syntax Description

block_number	Identifies the block on which to enable or disable the ASIC scan.
error_cause	(Optional) Identifies an error cause in the specified block. When you include the *error_cause* argument in the **asic-scan pse egress** command syntax, the command takes place on the specified error cause.
disable	(Optional) Disables the ASIC scan on the specified block or error cause.
enable	(Optional) Enables the ASIC scan on the specified block or error cause.
help_cause	(Optional) Displays a list of all error causes in the specified block.
location *node-id*	Identifies the location of the node hosting the egress PSE device on which to run, enable, or disable an ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to display a list of all error causes in the specified block of a PSE device:

```
RP/0/RP0/CPU0:router# asic-scan pse egress 10 help-cause location 0/6/CPU0

Total 6 cause(s)

Cause Number          Cause Name
0                     lbm_owl_ovl
1                     scr_m4_ovl
2                     scr_m3_ovl
3                     scr_m2_ovl
4                     scr_m1_ovl
5                     scr_m0_ovl
```

The following example shows how to run a scan on the PSE ASIC block ID 30 located on the CPU node in slot 1:

```
RP/0/RP0/CPU0:router# asic-scan pse egress 30 location 0/6/CPU0
```

asic-scan pse egress disable

To disable an ASIC scan on all causes in all blocks of a specific packet switching engine (PSE) device, use the **asic-scan pse egress disable** command in EXEC mode.

asic-scan pse egress disable location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node on which to disable the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation.

Note Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to disable an ASIC scan on all causes in all blocks of the egress PSE device that is located on the CPU node in slot 6:

```
RP/0/RP0/CPU0:router# asic-scan pse egress disable location 0/6/CPU0
```

asic-scan pse egress enable

To enable an ASIC scan on all causes in all blocks of a specific egress packet switching engine (PSE) device, use the **asic-scan pse egress enable** command in EXEC mode.

asic-scan pse egress enable location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node that hosts the egress PSE on which to enable the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to enable an ASIC scan on all causes in all blocks on the egress PSE that is located on the CPU node in slot 6:

```
RP/0/RP0/CPU0:router# asic-scan pse egress enable location 0/6/CPU0
```

asic-scan pse egress help-block

To enable an ASIC scan that lists all block numbers of a specific egress packet switching engine (PSE) device, use the **asic-scan pse egress help-block** command in EXEC mode.

asic-scan pse egress help-block location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node that hosts the egress PSE on which to enable the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.6.0	This command was introduced.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to enable an ASIC scan that lists all block numbers on the egress PSE that is located on the CPU node in slot 6:

```
RP/0/RP0/CPU0:router# asic-scan pse egress help-block location 0/6/CPU0
```

asic-scan pse egress location

To run an ASIC scan on a specific packet switching engine (PSE) device, use the **asic-scan pse egress location** command in EXEC mode.

asic-scan pse egress location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node hosting the PSE on which to run the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to run an ASIC scan on all egress PSE located on the CPU node in slot 1:

```
RP/0/RP0/CPU0:router# asic-scan pse egress location 0/1/CPU0
```

asic-scan pse egress quick-scan

To run a quick ASIC scan on a specific file or on all bits of the chip located on a specific packet switching engine (PSE) device, use the **asic-scan pse egress quick-scan** command in EXEC mode.

asic-scan pse egress quick-scan {*filename* | **allscan**} **location** *node-id*

Syntax Description

filename	Name of the .xsvf file on which to run the quick ASIC scan
allscan	Performs a quick ASIC scan on all bits of the chip, including the preread for Metro.
location *node-id*	Identifies the location of the node hosting the egress PSE on which to run the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to run a quick ASIC scan on all bits of the chip on the egress PSE device that is located on the CPU node in slot 1:

```
RP/0/RP0/CPU0:router# asic-scan pse egress quick-scan allscan location 0/1/CPU0
```

asic-scan pse ingress (block number)

To run, enable, or disable a generic ASIC scan on a specific ingress packet switching engine (PSE) device block or error cause, use the **asic-scan pse ingress** command in EXEC mode.

asic-scan pse ingress *block_number* [{[*error_cause*] [{**disable** | **enable**}] | **help-cause**}] **location** *node-id*

Syntax Description

block_number	Identifies the block on which to enable or disable the ASIC scan.
error_cause	(Optional) Identifies an error cause in the specified block. When you include the *error_cause* argument in the **asic-scan pse ingress** command syntax, the command takes place on the specified error cause.
disable	(Optional) Disables the ASIC scan on the specified block or error cause.
enable	(Optional) Enables the ASIC scan on the specified block or error cause.
help-cause	(Optional) Displays a list of all error causes in the specified block.
location *node-id*	Identifies the location of the node hosting the ingress PSE device on which to run, enable, or disable an ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to display a list of all error causes in the specified block of a PSE device:

```
RP/0/RP0/CPU0:router# asic-scan pse ingress 10 help-cause location 0/6/CPU0

 Total 6 cause(s)

 Cause Number          Cause Name
 0                     lbm_owl_ovl
 1                     scr_m4_ovl
 2                     scr_m3_ovl
 3                     scr_m2_ovl
 4                     scr_m1_ovl
 5                     scr_m0_ovl
```

The following example shows how to run a scan on the PSE ASIC block ID 30 located on the CPU node in slot 1:

```
RP/0/RP0/CPU0:router# asic-scan pse ingress 30 location 0/6/CPU0
```

asic-scan pse ingress disable

To disable an ASIC scan on all causes in all blocks of a specific packet switching engine (PSE) device, use the **asic-scan pse ingress disable** command in EXEC mode.

asic-scan pse ingress disable location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node on which to disable the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to disable an ASIC scan on all causes in all blocks of the ingress PSE device that is located on the CPU node in slot 6:

```
RP/0/RP0/CPU0:router# asic-scan pse ingress disable location 0/6/CPU0
```

asic-scan pse ingress enable

To enable an ASIC scan on all causes in all blocks of a specific ingress packet switching engine (PSE) device, use the **asic-scan pse ingress enable** command in EXEC mode.

asic-scan pse ingress enable location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node that hosts the ingress PSE on which to enable the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to enable an ASIC scan on all causes in all blocks on the ingress PSE that is located on the CPU node in slot 6:

```
RP/0/RP0/CPU0:router# asic-scan pse ingress enable location 0/6/CPU0
```

asic-scan pse ingress help-block

To enable an ASIC scan that lists all block numbers of a specific ingress packet switching engine (PSE) device, use the **asic-scan pse ingress help-block** command in EXEC mode.

asic-scan pse ingress help-block location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node that hosts the ingress PSE on which to enable the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.6.0	This command was introduced.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to enable an ASIC scan that lists all block numbers on the ingress PSE that is located on the CPU node in slot 6:

```
RP/0/RP0/CPU0:router# asic-scan pse ingress help-block location 0/6/CPU0
```

asic-scan pse ingress location

To run an ASIC scan on a specific packet switching engine (PSE) device, use the **asic-scan pse ingress location** command in EXEC mode.

asic-scan pse ingress location *node-id*

Syntax Description

location *node-id*	Identifies the location of the node hosting the PSE on which to run the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to run an ASIC scan on all ingress PSE located on the CPU node in slot 1:

```
RP/0/RP0/CPU0:router# asic-scan pse ingress location 0/1/CPU0
```

asic-scan pse ingress quick-scan

To run a quick ASIC scan on a specific file or on all bits of the chip located on a specific packet switching engine (PSE) device, use the **asic-scan pse ingress quick-scan** command in EXEC mode.

asic-scan pse ingress quick-scan {*filename* | **allscan**} **location** *node-id*

Syntax Description

filename	Name of the .xsvf file on which to run the quick ASIC scan
allscan	Performs a quick ASIC scan on all bits of the chip, including the preread for Metro.
location *node-id*	Identifies the location of the node hosting the ingress PSE on which to run the ASIC scan. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	Note was added in the Usage Guidelines section that this command can adversely affect the router operation.

Usage Guidelines

This command should be used only with Cisco support supervision. Incorrect usage of this command can accidentally remove the relevant card from service.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to run a quick ASIC scan on all bits of the chip on the ingress PSE device that is located on the CPU node in slot 1:

```
RP/0/RP0/CPU0:router# asic-scan pse ingress quick-scan allscan location 0/1/CPU0
```

clear controller egressq queue

To clear a specific controller egress queue, use the **clear controller egressq queue** command in EXEC mode.

clear controller egressq queue *queue-id1* [*queue-id2*] **location** *node-id*

Syntax Description

queue-id1	Queue you want to clear. Replace *the queue-id1* argument with a queue number. Range is from 1 through 8191.
queue-id2	(Optional) Last queue in a range of queues to clear. Replace the *queue-id2* argument with a queue number. Range is from 1 through 8191.
location *node-id*	Identifies the location of the node whose egress queue you want to clear. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

If you enter the optional *queue-id2* argument after you enter the *queue-id1* argument, then the *queue-id1* argument specifies the first queue in a range of queues to be cleared. If you enter the *queue-id1* argument without specifying the *queue-id2* argument, then only the queue specified for *queue-id1* is cleared.

Task ID

Task ID	Operations
root-system	read
basic-services	read, write
drivers	read, write

Examples

The following example shows how to clear a specific controller egress queue:

```
RP/0/RP0/CPU0:router# clear controller egress queue 1 location 0/1/CPU0
```

clear controller egressq queue all

To clear all controller egress queues on a specific node, use the **clear controller egressq queue all** command in EXEC mode.

clear controller egressq queue all location *node-id*

Syntax Description

all	Clears all egress queues in the specified location.
location *node-id*	Identifies the location of the egressq queue you want to clear. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	The basic-services and root-systems were removed from this command.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to clear all controller egress queues on a specific node:

```
RP/0/RP0/CPU0:router# clear controller egress queue all location 0/1/CPU0
```

clear controller egressq statistics

To clear the egress queue statistics in a specific location, use the **clear controller egressq statistics** command in EXEC mode.

clear controller egressq statistics location *node-id*

Syntax Description

location *node-id*	Identifies the location of the egressq queue whose statistics you want to clear. The *node-id* argument is expressed in the *rack/slot/module* notation.

Note Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read, write

Examples

The following example shows how to clear all controller egress queue statistics on a specific node:

```
RP/0/RP0/CPU0:router# clear controller egressq statistics location 0/1/CPU0
```

clear controller pse statistics

To clear statistics maintained by the packet switching engine (PSE) on a specific node, use the **clear controller pse statistics** command in EXEC mode.

clear controller pse statistics [{**egress** | **ingress**}]**location** *node-id*

Syntax Description

egress	(Optional) Clears statistics on the egress PSE device only.
ingress	(Optional) Clears statistics on the ingress PSE device only.
location *node-id*	Identifies the location of the node whose PSE device statistics you want to clear. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

If the **egress** or **ingress** keywords are not specified, the **clear controller pse statistics** command takes effect on both device instances.

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	• The **instance 0** and **instance 1** keywords were replaced by the **ingress** and **egress** keywords. • The **clear controller pse statistics** command was removed from the root-system task-ID. • The **clear controller pse statistics** command was first supported under the interface task-ID.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read, write
drivers	read, write

Examples

The following example shows how to clear all PSE statistics on a specific node (0/1/CPU0):

```
RP/0/RP0/CPU0:router# clear controller pse statistics location 0/1/CPU0
```

The following example shows how to clear all egress PSE statistics on a specific node (0/6/CPU0):

```
RP/0/RP0/CPU0:router# clear controller pse statistics egress location 0/6/CPU0
```

Related Commands

Command	Description
show controllers pse summary, on page 124	Displays a summary of packet switching engine information for a specific controller or node.

show controllers cpuctrl cdma channel

To display information about the CPU controller Control Direct-Memory-Access (CDMA) engine, use the **show controllers cpuctrl cdma channel** command in EXEC mode.

show controllers cpuctrl cdma channel {**0** | **1**} **queue** {*queue-id* | **all**} {**active** | **detail**} [**location** *node-id*]

Syntax Description

{**0** | **1**}	Displays CPU controller CDMA engine information for the specified channel. Enter **0** to display CDMA engine information for channel 0, or enter **1** to display CDMA engine information for channel 1.
queue {*queue-id* | **all**}	Displays CDMA information for a specific Direct-Memory-Access (DMA) queue, or for all DMA queues on the specified channel. • Enter the **queue all keywords** to display CDMA information for all DMA queues on the specified channel. • Enter the **queue**keyword and *queue-id argument* to display CDMA information for a specific queue. Replace the *queue-id* argument with a queue number. Range is from 1 through 7.
active	Displays descriptions for active DMA queues only.
detail	Displays descriptions for any DMA queues, regardless of whether they are active.
location *node-id*	(Optional) Identifies the location of the node whose CPU controller CDMA information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	• The **show controllers cpuctrl cdma** command was replaced by the **show controllers cpuctrl cdma channel** command. • The **show controllers cpuctrl cdma channel** command was first supported under the interface task-ID.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows partial output from the **show controllers cpuctrl cdma channel** command with the **active** keyword.

```
RP/0/RP0/CPU0:router# show controllers cpuctrl cdma channel all active

Mon Nov  9 23:13:08.235 PST
-------------------------------------------------------
DMA queue:
Channel: 0     queue: 0        state: Inactive

-------------------------------------------------------
DMA queue:
Channel: 0     queue: 1        state: Active

       OS Interrupt Level =        108   Cpuctrl Int Level =         12
          OS Run Priority =         45       client handle =          1
              ISR context = 0x9c35c49c  Pakman/Bufman Inst = bufman/misc
 client callback function = 0x4c60df28    cleanup function = 0x4c60ded8
                  Pakmode = 0x00000002           Pollflags = 0x00000000
   Total DMA transactions =          3  Queue create count =          1
         DMA transactions =          3   Bytes transferred =      67584

     DMA Out of Desc errs =          0             DMA IWA =          0
     DMA transaction errs =          0

Descriptor list base addr = 0xe4037300    Physical address = 0x76037300
               list_size =         32  Active descriptors =          0
           current_index =          3    tx_enqueue_index =          3

-------------------------------------------------------
DMA queue:
Channel: 0     queue: 2        state: Inactive

-------------------------------------------------------
DMA queue:
Channel: 0     queue: 3        state: Active

       OS Interrupt Level =        107   Cpuctrl Int Level =         11
          OS Run Priority =         30       client handle =          0
              ISR context = 0x9c35cd84  Pakman/Bufman Inst = bufman/mipc
 client callback function = 0x4c60df28    cleanup function = 0x4c60ded8
                  Pakmode = 0x00000005           Pollflags = 0x00000000
   Total DMA transactions =     115622  Queue create count =          1
         DMA transactions =     115622   Bytes transferred =   10213420

     DMA Out of Desc errs =          0             DMA IWA =          0
     DMA transaction errs =          0

Descriptor list base addr = 0xe4051100    Physical address = 0x76051100
               list_size =        256  Active descriptors =          0
           current_index =        166    tx_enqueue_index =        166

-------------------------------------------------------
DMA queue:
Channel: 0     queue: 4        state: Active

       OS Interrupt Level =        113   Cpuctrl Int Level =         17
          OS Run Priority =         45       client handle =          5
              ISR context = 0x9c35c748  Pakman/Bufman Inst = bufman/misc
 client callback function = 0x4c60df28    cleanup function = 0x4c60ded8
                  Pakmode = 0x00000002           Pollflags = 0x00000000
   Total DMA transactions =        321  Queue create count =          1
```

```
         DMA transactions =        321   Bytes transferred =      42594

     DMA Out of Desc errs =          0             DMA IWA =          0
     DMA transaction errs =          0

Descriptor list base addr = 0xe40d2068    Physical address = 0x760d2068
                list_size =       1023  Active descriptors =          0
            current_index =        321    tx_enqueue_index =        321

------------------------------------------------------
DMA queue:
Channel: 0     queue: 5        state: Active

       OS Interrupt Level =        111   Cpuctrl Int Level =         15
          OS Run Priority =         45       client handle =          3
              ISR context = 0x9c35cad8  Pakman/Bufman Inst = bufman/misc
 client callback function = 0x4c60df28    cleanup function = 0x4c60ded8
                  Pakmode = 0x00000002           Pollflags = 0x00000000
   Total DMA transactions =        376  Queue create count =          1
         DMA transactions =        376   Bytes transferred =       7674

     DMA Out of Desc errs =          0             DMA IWA =          0
     DMA transaction errs =          0

Descriptor list base addr = 0xe4047110    Physical address = 0x76047110
                list_size =        128  Active descriptors =          0
            current_index =        120    tx_enqueue_index =        120

------------------------------------------------------
DMA queue:
Channel: 0     queue: 6        state: Active

       OS Interrupt Level =        112   Cpuctrl Int Level =         16
          OS Run Priority =         45       client handle =          6
              ISR context = 0x9c35c9f4  Pakman/Bufman Inst = bufman/misc
 client callback function = 0x4c60df28    cleanup function = 0x4c60ded8
                  Pakmode = 0x00000002           Pollflags = 0x00000000
   Total DMA transactions =        306  Queue create count =          1
         DMA transactions =        306   Bytes transferred =     394340

     DMA Out of Desc errs =          0             DMA IWA =          0
     DMA transaction errs =          0

Descriptor list base addr = 0xe40d7068    Physical address = 0x760d7068
                list_size =       1023  Active descriptors =          0
            current_index =        306    tx_enqueue_index =        306

------------------------------------------------------
DMA queue:
Channel: 0     queue: 7        state: Active

       OS Interrupt Level =        110   Cpuctrl Int Level =         14
          OS Run Priority =         45       client handle =          4
              ISR context = 0x9c35c580  Pakman/Bufman Inst = bufman/misc
 client callback function = 0x4cad1130    cleanup function = 0x4cad058c
                  Pakmode = 0x00000002           Pollflags = 0x00000000
   Total DMA transactions =     140344  Queue create count =          1
         DMA transactions =     140344   Bytes transferred = 2344779856

     DMA Out of Desc errs =          0             DMA IWA =          0
     DMA transaction errs =          0

Descriptor list base addr = 0xe40cd068    Physical address = 0x760cd068
                list_size =       1023  Active descriptors =          0
```

```
            current_index =        193    tx_enqueue_index =        193

------------------------------------------------------
DMA queue:
Channel: 1     queue: 0        state: Inactive

------------------------------------------------------
DMA queue:
Channel: 1     queue: 1        state: Active

       OS Interrupt Level =        108   Cpuctrl Int Level =         12
          OS Run Priority =         45       client handle =          1
              ISR context = 0x9c35c49c  Pakman/Bufman Inst = bufman/misc
 client callback function = 0x4c60df28    cleanup function = 0x4c60ded8
                 Pakmode = 0x00000002           Pollflags = 0x00000000
   Total DMA transactions =          3  Queue create count =          1
         DMA transactions =          3   Bytes transferred =      67584

     DMA Out of Desc errs =          0             DMA IWA =          0
     DMA transaction errs =          0

Descriptor list base addr = 0xe40370f0    Physical address = 0x760370f0
                list_size =         32  Active descriptors =          0
            current_index =          3    tx_enqueue_index =          3

------------------------------------------------------
DMA queue:
Channel: 1     queue: 2        state: Inactive

------------------------------------------------------
DMA queue:
Channel: 1     queue: 3        state: Active

       OS Interrupt Level =        106   Cpuctrl Int Level =         10
          OS Run Priority =         30       client handle =          0
              ISR context = 0x9c35ce68  Pakman/Bufman Inst = bufman/mipc
 client callback function = 0x4c60df28    cleanup function = 0x4c60ded8
                 Pakmode = 0x00000005           Pollflags = 0x00000000
   Total DMA transactions =     114238  Queue create count =          1
         DMA transactions =     114238   Bytes transferred =    8686924

     DMA Out of Desc errs =          0             DMA IWA =          0
     DMA transaction errs =          0

Descriptor list base addr = 0xe4052110    Physical address = 0x76052110
                list_size =        256  Active descriptors =          0
            current_index =         62    tx_enqueue_index =         62
--More--
```

This table describes the significant fields shown in the display.

Table 1: show controllers cpuctrl cdma channel Field Descriptions

Field	Description
DMA queue	Identifies the DMA[1] queue.
Channel	Identifies the channel whose DMA queue is displayed. 0 is the ingress channel, and 1 is the egress channel
queue	Identifies the queue.

Field	Description
state	Current state of the queue.
OS Interrupt Level	Current interrupt level for the queue.
Cpuctrl Interrupt Level	Current interrupt level for the CPU controller.
OS Run Priority	Run priority level for this queue.
client handle	Internal identifier for the Cisco client.
ISR context	Internal information about the location of the ISR2 pointer.
Pakman/Bufman Instance	Internal information about the location of the Pakman and Bufman Instance.
client callback function pointer	Internal information about the client callback function pointer.
cleanup function	Internal information about the client cleanup function pointer.
Queue Created 1 times	Number of times this queue was regenerated.
Pakmode	Information about internal data structures and parameters.
Pollflags	Specifies whether the CDMA queue uses a polling or interrupt-driven approach for detecting CDMA operation completion notification. **Note** Currently, CDMA queues use interrupt driven completion only. PDMA queues use interrupt-driven and polling completion.
Total DMA transactions	Number of DMA transactions in the queue.
Queue create count	Number of times this queue was regenerated.
DMA transactions	Number of DMA transactions in the queue.
Bytes transferred	Number of bytes that have been transferred by the Control Direct-Memory-Access engine.
DMA Out of Desc errs	Number of DMA errors in the queue.
CDMA transactions	Number of CDMA transactions in the queue.
DMA IWA	Number of IWA bytes that have been processed by the Direct-Memory-Access engine.
DMA transaction errs	Number of DMA transactions that had errors.
Descriptor list base addr	Internal information about the location of the descriptor list.
Physical address	Physical address of the CPU memory that holds the descriptors in the ring used by the CDMA queue hardware.
list_size	Total number of descriptors in the ring used by the CDMA queue hardware.

Field	Description
Active descriptors	Number of descriptors that have transactions that are not cleaned after being notified of their completion. Note that the hardware may not yet have completed these transactions.
current_index	Points to the next descriptor that the hardware is expected to complete.
tx_enqueue_index	Points to the descriptor that will be added to the next operation request.
Index	Location of the descriptor in the ring.
Shadow	Internal field that manages requests that have been split into multiple descriptors.
Hdr	Internal field that manages requests that have been split into multiple descriptors.
Flags	Internal field that manages requests that have been split into multiple descriptors.
Descriptor	Descriptor heading.
Width	Width of the data on the ASIC side in bits. The DMA stride is rounded up to the next power of two bytes that contains this number of bits.
Bufsize	Size of the buffer used for the transfer.
Xfersize	Number of bytes on the CPU memory that are occupied by the transfer.
Memaddr	36 bit physical address of the CPU memory in the transfer.
Squidaddr	40 bit address of the ASIC register or memory in the transfer.

[1] Direct Memory Access

Related Commands

Command	Description
show controllers cpuctrl clients, on page 35	Displays information about all CPU controller clients on the router, or for specific CPU controller clients.

show controllers cpuctrl clients

To display information about all CPU controller clients on the router, or for specific CPU controller clients, use the **show controllers cpuctrl clients** command in EXEC mode.

show controllers cpuctrl clients {**all** | **cdma** *clients* | **pdma** *clients*} {**active** | **detail**} [**location** *node-id*]

Syntax Description

all	Displays a summary information for all clients on the router.
cdma *clients*	Displays information about Control Direct-Memory-Access (CDMA) clients only. Replace the *clients* argument with one of the following keywords: • **dcc**—Displays information for the DCC client. • **egressq**—Displays information for the egressq client. • **fabricq**—Displays information for the Fabricq ASIC client. • **fia**—Displays information for the Fabric Interface ASIC (FIA) client. • **ingressq**—Displays information for the ingressq ASIC client. • **jacketcard**—Displays information for the jacketcard client. • **mipc**—Displays information for the Metro Inter-Process-Communication (MIPC) client. • **pla768**—Displays information for the ASIC client for OC-768. • **plaspa**—Displays information for the ASIC client for the SPA. • **plim**—Displays information for the PLIM client. • **plimasic**—Displays information for the PLIM ASIC client. • **pse**—Displays information for the PSE client.
pdma *clients*	Displays information for Packet Direct-Memory-Access (PDMA) clients only. Replace *clients* with one of the following keywords: • **bfd**—Displays information for the client bidirectional forwarding detection (BFD) PDMA packet. • **diag**—Displays information for the PDMA client called DIAG packet. • **fabio**—Displays information for the FABIO PDMA packet client. • **fia**—Displays information for the fabric interface ASIC packet PDMA client. • **frr**—Displays information for the fast reroute (FRR) packet PDMA client. • **gsp**—Displays information for the Group Services Packet (GSP) PDMA client. • **mipc**—Displays information for the MIPC packet PDMA client. • **mstats**—Displays information for the MSTATS packet PDMA client. • **netflow**—Displays information for the NetFlow packet PDMA client.
active	Displays descriptions for active queues only.
detail	Displays descriptions for any queues, regardless of whether or not they are active.
location *node-id*	(Optional) Identifies the location of the node whose CPU controller information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default No default behavior or values

Command Modes EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	The **show controllers cpuctrl clients** command was first supported under the interface task-ID.
Release 3.6.0	The following keywords were added: • **bfd** • **fabio** • **pla768** • **plaspa** The **netio** keyword was removed.

Usage Guidelines No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples The following example shows how to display information about all the CPU controller clients:

```
RP/0/RP0/CPU0:router# show controllers cpuctrl clients all

Mon Nov  2 15:27:45.524 UTC
-------------------------------------------------------------
DMA client:     client_name: MIPC       client_handle:0
isr_count: 2    queue_count:   2        client_pid:    90186

-------------------------------------------------------------
DMA client:     client_name: Pse        client_handle:1
isr_count: 1    queue_count:   2        client_pid:    90186

-------------------------------------------------------------
DMA client:     client_name: DCC        client_handle:2
isr_count: 0    queue_count:   0        client_pid:    0

-------------------------------------------------------------
DMA client:     client_name: PlimAsic   client_handle:3
isr_count: 1    queue_count:   1        client_pid:    86063

-------------------------------------------------------------
DMA client:     client_name: Egressq    client_handle:4
isr_count: 1    queue_count:   1        client_pid:    94283

-------------------------------------------------------------
DMA client:     client_name: Ingressq   client_handle:5
isr_count: 1    queue_count:   1        client_pid:    65592
```

```
-------------------------------------------------------------
DMA client:     client_name: Fabricq    client_handle:6
isr_count: 1    queue_count:   1        client_pid:    65591

-------------------------------------------------------------
DMA client:     client_name: PLIM       client_handle:7
isr_count: 0    queue_count:   0        client_pid:    0

-------------------------------------------------------------
DMA client:     client_name: FIA        client_handle:8
isr_count: 0    queue_count:   0        client_pid:    65595

-------------------------------------------------------------
DMA client:     client_name: JacketCard client_handle:9
isr_count: 0    queue_count:   0        client_pid:    81991

-------------------------------------------------------------
PKT client:     client_name: FRR        client_handle:0
isr_count: 1    queue_count:   2        client_pid:    73794

-------------------------------------------------------------
PKT client:     client_name: GSP        client_handle:1
isr_count: 2    queue_count:   4        client_pid:    73794

-------------------------------------------------------------
PKT client:     client_name: IPC        client_handle:2
isr_count: 1    queue_count:   1        client_pid:    24602

-------------------------------------------------------------
PKT client:     client_name: MIPC       client_handle:3
isr_count: 1    queue_count:   2        client_pid:    90186

-------------------------------------------------------------
PKT client:     client_name: MSTATS     client_handle:4
isr_count: 2    queue_count:   2        client_pid:    94284

-------------------------------------------------------------
PKT client:     client_name: NETFLOW    client_handle:5
isr_count: 2    queue_count:   2        client_pid:    147612

-------------------------------------------------------------
PKT client:     client_name: FABIO      client_handle:6
isr_count: 1    queue_count:   13       client_pid:    65589

-------------------------------------------------------------
PKT client:     client_name: FIA        client_handle:7
isr_count: 0    queue_count:   0        client_pid:    0

-------------------------------------------------------------
PKT client:     client_name: DIAG       client_handle:8
isr_count: 1    queue_count:   3        client_pid:    118896

-------------------------------------------------------------
PKT client:     client_name: BFD        client_handle:9
isr_count: 1    queue_count:   2        client_pid:    118883

-------------------------------------------------------------
DMA client:     client_name: MIPC       client_handle:0
isr_count: 0    queue_count:   0        client_pid:    0

-------------------------------------------------------------
DMA client:     client_name: Pse        client_handle:1
isr_count: 0    queue_count:   0        client_pid:    0
```

```
------------------------------------------------------------
DMA client:     client_name: DCC        client_handle:2
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: PlimAsic   client_handle:3
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: Egressq    client_handle:4
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: Ingressq   client_handle:5
isr_count: 1    queue_count:   1        client_pid:    122941

------------------------------------------------------------
DMA client:     client_name: Fabricq    client_handle:6
isr_count: 1    queue_count:   1        client_pid:    122939

------------------------------------------------------------
DMA client:     client_name: PLIM       client_handle:7
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: FIA        client_handle:8
isr_count: 0    queue_count:   0        client_pid:    122943

------------------------------------------------------------
DMA client:     client_name: JacketCard client_handle:9
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
PKT client:     client_name: FRR        client_handle:0
isr_count: 1    queue_count:   2        client_pid:    122952

------------------------------------------------------------
PKT client:     client_name: GSP        client_handle:1
isr_count: 2    queue_count:   4        client_pid:    122952

------------------------------------------------------------
PKT client:     client_name: IPC        client_handle:2
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
PKT client:     client_name: MIPC       client_handle:3
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
PKT client:     client_name: MSTATS     client_handle:4
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
PKT client:     client_name: NETFLOW    client_handle:5
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
PKT client:     client_name: FABIO      client_handle:6
isr_count: 1    queue_count:   4        client_pid:    122935

------------------------------------------------------------
PKT client:     client_name: FIA        client_handle:7
isr_count: 0    queue_count:   0        client_pid:    0
```

```
------------------------------------------------------------
PKT client:     client_name: DIAG       client_handle:8
isr_count: 1    queue_count:   2        client_pid:    147600

------------------------------------------------------------
PKT client:     client_name: BFD        client_handle:9
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: MIPC       client_handle:0
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: Pse        client_handle:1
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: DCC        client_handle:2
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: PlimAsic   client_handle:3
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: Egressq    client_handle:4
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: Ingressq   client_handle:5
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: Fabricq    client_handle:6
isr_count: 1    queue_count:   1        client_pid:    147511

------------------------------------------------------------
DMA client:     client_name: PLIM       client_handle:7
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: FIA        client_handle:8
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
DMA client:     client_name: JacketCard client_handle:9
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
PKT client:     client_name: FRR        client_handle:0
isr_count: 1    queue_count:   2        client_pid:    155718

------------------------------------------------------------
PKT client:     client_name: GSP        client_handle:1
isr_count: 2    queue_count:   4        client_pid:    155718

------------------------------------------------------------
PKT client:     client_name: IPC        client_handle:2
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
PKT client:     client_name: MIPC       client_handle:3
isr_count: 0    queue_count:   0        client_pid:    0
```

```
------------------------------------------------------------
PKT client:     client_name: MSTATS     client_handle:4
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
PKT client:     client_name: NETFLOW    client_handle:5
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
PKT client:     client_name: FABIO      client_handle:6
isr_count: 1    queue_count:   4        client_pid:    147508

------------------------------------------------------------
PKT client:     client_name: FIA        client_handle:7
isr_count: 0    queue_count:   0        client_pid:    0

------------------------------------------------------------
PKT client:     client_name: DIAG       client_handle:8
isr_count: 1    queue_count:   2        client_pid:    180341

------------------------------------------------------------
PKT client:     client_name: BFD        client_handle:9
isr_count: 0    queue_count:   0        client_pid:    0
```

This table describes the significant fields shown in the display.

Table 2: show controllers cpuctrl clients Field Descriptions

Field	Description
client	Client type. Value is DMA[2] or PKT.
client_name	Name of the client.
client_handle	Internal client identifier.
isr_count	ISR[3] counters.
queue_count	Queue counters.
client_pid	Client PID[4].

[2] Direct Memory Access
[3] Interrupt Service Routine
[4] Process Identifier

Related Commands

Command	Description
show controllers cpuctrl summary, on page 57	Displays summarized information about all the ASICs accessed through the CPU controller ASICs on the router or on a specific node.

show controllers cpuctrl devices

To display information about the CPU controller devices on the router, use the **show controllers cpuctrl devices** command in EXEC mode.

show controllers cpuctrl devices *device-name* {**pdma** *queue* {**all** | **tx** | **rx**} {**active** | **detail**} | **pio**} [**location** *node-id*]

Syntax Description

device-name	Displays information about a specific CPU controller device. Replace the *device-name argument* with one of the following device names: • **egressq**—Displays information about the egressq ASIC instance 0. • **epse**—Displays information about the egress packet switching engine (PSE) ASIC. • **fabricq instance 0**—Displays information about the fabricq ASIC instance 0. • **fabricq instance 1**—Displays information about the fabricq ASIC instance 1. • **fabricq instance 2**—Displays information about the fabricq ASIC instance 2. • **fabricq instance 3**—Displays information about the fabricq ASIC instance 3. • **fia instance 0**—Displays information about the Fabric Interface ASIC (FIA) instance 0. • **fia instance 1**—Displays information about the FIA instance 1. • **ingressq**—Displays information about the ingressq ASIC. • **ipse**—Displays information about the ingress packet switching engine (PSE) ASIC. • **jacketcard instance 0**—Displays information about the jacketcard instance 0 ASIC. • **jacketcard instance 1**—Displays information about the jacketcard instance 1 ASIC. • **jacketcard instance 2**—Displays information about the jacketcard instance 2 ASIC. • **plim asic instance 0**—Displays information about the PLIM ASIC instance 0. • **plim asic instance 1**—Displays information about the PLIM ASIC instance 1. • **plim asic instance 2** —Displays information about the PLIM ASIC instance 2. • **plim asic instance 3**—Displays information about the PLIM ASIC instance 3. • **plim fpga**—Displays information about the field-programmable gate array (FPGA) PLIM.
pdma *queue* **all**	Displays transmit and receive Packet Direct-Memory-Access (PDMA) information for the specified queue. Replace the *queue* argument with a queue number. The range is from 0 to 7.
pdma *queue* **tx**	Displays transmit PDMA information only for the specified queue. Replace the *queue* argument with a queue number. The range is from 0 to7.
pdma *queue* **rx**	Displays receive PDMA information only for the specified queue. Replace the *queue* argument with a queue number. The range is from 0 to 7.
pdma all	Displays transmit and receive PDMA information for all queues.
active	Displays descriptions for active queues only.
detail	Displays detailed descriptions for any queues, regardless of whether they are active.
pio	Displays transmit and receive Polled I/O (PIO) information for the specified queue.

location *node-id*	(Optional) Identifies the location of the node whose CPU controller information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	The **show controllers cpuctrl devices** command was first supported under the interface task-ID.
Release 3.6.0	The following keywords were changed: • **fabricq0** changed to **fabric instance 0** • **fabricq1** changed to **fabric instance 1** • **fabricq2** changed to **fabric instance 2** • **fabricq3** changed to **fabric instance 3** • **fia0** changed to **fia instance 0** • **fia1** changed to **fia instance 1** • **jacketcard0** changed to **jacketcard instance 0** • **jacketcard1** changed to **jacketcard instance 1** • **jacketcard2** changed to **jacketcard instance 2** • **plimasic0** changed to **plim asic instance 0** • **plimasic1** changed to **plim asic instance 1** • **plimasic2** changed to **plim asic instance 2** • **plimasic3** changed to **plim asic instance 3** • **plimfpga** changed to **plim fpga**

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display transmit and receive PDMA information for all active queues on the egressq ASIC instance 0:

```
RP/0/RP0/CPU0:router# show controllers cpuctrl devices egressq pdma queue all active location
 0/6/CPU0

------------------------------------------------------------
TX PKT queue:
cpuctrl net Port: 7 (Egressq   )     queue: 0        state: Active
```

```
       OS Interrupt Level =        127   Cpuctrl Int Level =         31
          OS Run Priority =         10       client handle =          6
              ISR context = 0x7810c1c8  Pakman/Bufman Inst = pakman/server
 client callback function = 0x48200298    cleanup function = 0x482002bc
                 Pakmode = 0x00000001          Pollflags = 0x00000000
Total Packets transmitted =     660089  Queue create count =          1
      Packets transmitted =     660089   Bytes transmitted =   17166002

Tx Out of Descriptor errs =          0             Tx IWA =          0
         Tx oversize errs =          0  Tx EgressQ q0 errs =          0

Descriptor list base addr = 0xec348068    Physical address = 0x30348068
               list_size =       1024  Active descriptors =          0
           current_index =        633    tx_enqueue_index =        633

-------------------------------------------------------
TX PKT queue:
cpuctrl net Port: 0 (Egressq   )     queue: 1        state: Inactive

-------------------------------------------------------
TX PKT queue:
cpuctrl net Port: 0 (Egressq   )     queue: 2        state: Inactive

-------------------------------------------------------
TX PKT queue:
cpuctrl net Port: 0 (Egressq   )     queue: 3        state: Inactive

-------------------------------------------------------
TX PKT queue:
cpuctrl net Port: 0 (Egressq   )     queue: 4        state: Inactive

-------------------------------------------------------
TX PKT queue:
cpuctrl net Port: 0 (Egressq   )     queue: 5        state: Inactive

-------------------------------------------------------
TX PKT queue:
cpuctrl net Port: 7 (Egressq   )     queue: 6        state: Active

       OS Interrupt Level =        118   Cpuctrl Int Level =         22
          OS Run Priority =         10       client handle =          2
              ISR context = 0x7810cf24  Pakman/Bufman Inst = bufman/ipc
 client callback function = 0xfc71d604    cleanup function = 0xfc71d6b8
                 Pakmode = 0x00000005          Pollflags = 0x00000000
Total Packets transmitted =          0  Queue create count =          1
      Packets transmitted =          0   Bytes transmitted =          0

Tx Out of Descriptor errs =          0             Tx IWA =          0
         Tx oversize errs =          0  Tx EgressQ q0 errs =          0

Descriptor list base addr = 0xec07a110    Physical address = 0x3007a110
               list_size =        256  Active descriptors =          0
           current_index =          0    tx_enqueue_index =          0

-------------------------------------------------------
TX PKT queue:
cpuctrl net Port: 7 (Egressq   )     queue: 7        state: Active

       OS Interrupt Level =        119   Cpuctrl Int Level =         23
          OS Run Priority =         40       client handle =          9
              ISR context = 0x7810d008  Pakman/Bufman Inst = pakman/sever
 client callback function = 0xfc71d604    cleanup function = 0xfc71d6b8
                 Pakmode = 0x00000001          Pollflags = 0x00000000
Total Packets transmitted =          0  Queue create count =          1
```

```
      Packets transmitted =          0   Bytes transmitted =          0

Tx Out of Descriptor errs =          0              Tx IWA =          0
         Tx oversize errs =          0  Tx EgressQ q0 errs =          0

Descriptor list base addr = 0xed63a068    Physical address = 0x3163a068
                list_size =       1024  Active descriptors =          0
            current_index =          0    tx_enqueue_index =          0

-------------------------------------------------------
RX PKT queue:
cpuctrl net Port: 7 (Egressq   )     queue: 0        state: Active

       OS Interrupt Level =        121   Cpuctrl Int Level =         25
          OS Run Priority =         10       client handle =          8
              ISR context = 0x7810d0ec  Pakman/Bufman Inst = bufman/misc
 client callback function = 0xfc71d550    cleanup function = 0xfc71d6b8
                  Pakmode = 0x00000001           Pollflags = 0x00000000
 Requested Rx Buffer Size =       1024  Packet switchcount =         20
    Actual Rx Buffer Size =       1648               Pool =          4
                      MTU =      12188     MTU Descriptors =          8
   Total Packets received =      71080  Queue create count =          1
         Packets received =      71080      Bytes received =  858219920

        Rx No Buffer errs =          0  NoBufferLimit errs =          0
 Rx No Packet Header errs =          0    Packet Form errs =          0
           Rx Packet errs =          0
       Rx Intr Stall errs =          0   Rx Intr Drop errs =          0

Descriptor list base addr = 0xec05c940    Physical address = 0x3005c940
                list_size =        128  Active descriptors =          0
            current_index =         64    tx_enqueue_index =         64

--More--
```

This table describes the significant fields shown in the display.

Table 3: show controllers cpuctrl devices Field Descriptions

Field	Description
PKT queue	Displays whether the packet queue is TX[5] or RX[6].
cpuctrl net Port	Identifies the CPU controller port.
queue	Identifies the queue whose CPU controller device information is displayed.
client handle	Internal Cisco client identifier.
queue state	Current state of the queue. The queue can be *Active* or *Inactive*.

[5] transmit

[6] receive

Related Commands

Command	Description
show controllers cpuctrl summary, on page 57	Displays summarized information about all the ASICs accessed through the CPU controller ASICs on the router or on a specific node.

show controllers cpuctrl error

To display the squid error information about the CPU controller, use the **show controllers cpuctrl error** command in EXEC mode.

show controllers cpuctrl error [**location** *node-id*] [**detail**]

Syntax Description

location *node-id*	(Optional) Identifies the location of the node whose internal CPU controller information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.
detail	(Optional) Displays detailed squid error information about the CPU controller.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.8.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display squid error information about the CPU controller:

```
RP/0/RP0/CPU0:router# show controllers cpuctrl error detail

Errors for node : 0/4/CPU1

    INTERNAL ERRORS:
    ===============
                 Error Interrupts =       2000

================================================================================
RP/0/RP0/CPU0:router#show controllers cpuctrl error detail
Tue Jul 21 04:15:02.632 DST

Errors for node : 0/4/CPU1

    INTERNAL ERRORS:
    ===============
                 Error Interrupts =       2000
        Spurious Error Interrupts =          0
                SN overflow count =          0
                PM overflow count =          0
               PCIX overflow count =          0
                ISN overflow count =          0
```

```
                Port overflow count =          0
                 Log overflow count =          0

==============================================================================
```

show controllers cpuctrl internal

To display internal information about the CPU controller, use the **show controllers cpuctrl internal** command in EXEC mode.

show controllers cpuctrl internal [**location** *node-id*]

Syntax Description

location *node-id*	(Optional) Identifies the location of the node whose internal CPU controller information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	The **show controllers cpuctrl internal** command was first supported under the interface task-ID.
Release 3.6.0	The location *node-id* keyword and argument were changed from required to optional.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display internal information about the CPU controller:

```
RP/0/RP0/CPU0:router# show controllers cpuctrl internal

Cpuctrl Internal Info for node 0/0/CPU0:
        Error Interrupts = 0    Spurious Error Interrupts = 0
        PCI Error Overflows = 0         PCI PM Error Overflows = 0
        PCIX Error Overflows = 0        Internal Access PCI Overflows = 0
        Port Error Overflows = 0        Error Log Overflows = 0
        cpuctrl Config Reg = 0x8357ffff  cpuctrl Physical Offset = 0x80000000
        cpuctrl Window Size = 0x40000000 cpuctrl Port Window Size = 0x04000000
        cpuctrl SHMem Size = 0x00800000  cpuctrl SHMem Used = 0x00223ee8
        cpuctrl version info: Squid FPGA v2.07 Fri Jan 23 16:21:01 2004 ykoren


Cpuctrl Internal Info for node 0/3/CPU0:
        Error Interrupts = 0    Spurious Error Interrupts = 0
        PCI Error Overflows = 0         PCI PM Error Overflows = 0
        PCIX Error Overflows = 0        Internal Access PCI Overflows = 0
        Port Error Overflows = 0        Error Log Overflows = 0
```

```
        cpuctrl Config Reg = 0x8357ffff  cpuctrl Physical Offset = 0x80000000
        cpuctrl Window Size = 0x40000000 cpuctrl Port Window Size = 0x04000000
        cpuctrl SHMem Size = 0x00800000  cpuctrl SHMem Used = 0x00223ee8
        cpuctrl version info: Squid FPGA v2.07 Fri Jan 23 16:21:01 2004 ykoren


Cpuctrl Internal Info for node 0/RP0/CPU0:
        Error Interrupts = 0    Spurious Error Interrupts = 0
        PCI Error Overflows = 0         PCI PM Error Overflows = 0
        PCIX Error Overflows = 0        Internal Access PCI Overflows = 0
        Port Error Overflows = 0        Error Log Overflows = 0
        cpuctrl Config Reg = 0x803f007f  cpuctrl Physical Offset = 0x80000000
        cpuctrl Window Size = 0x40000000 cpuctrl Port Window Size = 0x04000000
        cpuctrl SHMem Size = 0x00800000  cpuctrl SHMem Used = 0x00223ee8
        cpuctrl version info: Squid FPGA v2.07 Fri Feb 6 17:49:22 2004 ykoren

Cpuctrl Internal Info for node 0/RP1/CPU0:
        Error Interrupts = 0    Spurious Error Interrupts = 0
        PCI Error Overflows = 0         PCI PM Error Overflows = 0
        PCIX Error Overflows = 0        Internal Access PCI Overflows = 0
        Port Error Overflows = 0        Error Log Overflows = 0
        cpuctrl Config Reg = 0x003f007f  cpuctrl Physical Offset = 0x80000000
        cpuctrl Window Size = 0x40000000 cpuctrl Port Window Size = 0x04000000
        cpuctrl SHMem Size = 0x00800000  cpuctrl SHMem Used = 0x00223ee8
        cpuctrl version info: Squid FPGA v2.05 Wed Sep 3 17:37:47 2003 ykoren
```

This table describes the significant fields shown in the display.

Table 4: show controllers cpuctrl internal Field Descriptions

Field	Description
Error Interrupts	Total of error interrupts that have occurred on this node.
Spurious Error Interrupts	Current number interrupts that have occurred on this node due to spurious errors.
PCI Error Overflows	Number of times the PCI1 error buffer overflowed on the node.
PCI PM Error Overflows	Number of times PCI PM2 error buffer overflowed on this node.
PCIX Error Overflows	Number of times the PCI-X error buffer overflowed on this node.
Internal Access PCI Overflows	Number of times the Internal Access PCI buffer overflowed on this node.
Port Error Overflows	Number of times the port error buffer overflowed on this node.
Error Log Overflows	Number of times the error log buffer overflowed on this node.
cpuctrl Config Reg	CPU controller configuration register, expressed in hexadecimal format.
cpuctrl Physical Offset	CPU controller physical offset, expressed in hexadecimal format.
cpuctrl Window Size	CPU controller window size, expressed in hexadecimal format.
cpuctrl Port Window Size	CPU controller port window size, expressed in hexadecimal format.
cpuctrl SHMem Size	CPU controller shared memory size, expressed in hexadecimal format.

Field	Description
cpuctrl SHMem Used	CPU controller shared memory that has been used already, expressed in hexadecimal format.
cpuctrl version info	Provides version information for the CPU controller. The information displayed is: • Squid version • Date of the last version installation or upgrade

[7] [8]

Related Commands

Command	Description
show controllers cpuctrl summary, on page 57	Displays summarized information about all the ASICs accessed through the CPU controller ASICs on the router or on a specific node.

[7] 1. Peripheral Component Interconnect
[8] 2. port manager

show controllers cpuctrl ports

To display port information for the specified CPU controller ASIC, use the **show controllers cpuctrl ports** command in EXEC mode.

show controllers cpuctrl ports *asic_id* {**pdma queue** {*queue-id* | **all**} **direction** {**all** | **rx** | **tx**} {**active** | **detail**} | **pio**} [**location** *node-id*]

Syntax Description

asic_id	Identifies the CPU controller ASIC whose port information you want to display. Replace the *asic_id* argument with one of the following keywords: • **cpuctrl**—Displays port information for the CPU controller ASIC instance 0. • **egressq**—Displays port information for instance 0 of the egressq ASIC. • **epse**—Displays port information for the egress packet switching engine (EPSE) ASIC. • **fabricq instance 0**—Displays port information for instance 0 of the fabricq ASIC. • **fabricq instance 1**—Displays port information for instance 1 of the fabricq ASIC. • **fabricq instance 2**—Displays port information for instance 2 of the fabricq ASIC. • **fabricq instance 3**—Displays port information for instance 3 of the fabricq ASIC. • **fia instance 0**—Displays port information for instance 0 of the fabric Interface ASIC (FIA). • **fia instance 1**—Displays port information for instance 1 of the FIA. • **ingressq**—Displays port information for the ingressq ASIC. • **ipse**—Displays port information for the ingress packet switching engine (PSE) ASIC. • **jacketcard instance 0**—Displays port information for instance 0 of the jacketcard ASIC. • **jacketcard instance 1**—Displays port information for instance 1 of the jacketcard ASIC. • **jacketcard instance 2**—Displays port information for instance 2 of the jacketcard ASIC. • **pla768**—Displays port information for the OC768 plim ASIC. • **plaspa instance 0**—Displays port information for instance 0 of the SPA plim ASIC. • **plaspa instance 1**—Displays port information for instance 1 of the SPA plim ASIC. • **plim asic instance 0**—Displays port information for instance 0 of the plim ASIC. • **plim asic instance 1**—Displays port information for instance 1 of the plim ASIC. • **plim asic instance 2**—Displays port information for instance 2 of the plim ASIC. • **plim asic instance 3**—Displays port information for instance 3 of the plim ASIC.
pdma queue *queue-id*	Displays transmit and receive Packet Direct-Memory-Access (PDMA) information for the specified queue.
pdma queue all	Displays transmit and receive PDMA information for all queues. The information displayed pertains to the ASIC indicated in the **show controllers cpuctrl ports** command.
direction all	Displays transmit and receive Packet Direct-Memory-Access (PDMA) information for the specified queue. The information displayed pertains to the ASIC you specified for the *asic_id* argument.

direction rx	Displays receive PDMA information only for the specified queue. The information displayed pertains to the ASIC you specified for the *asic_id* argument.
direction tx	Displays transmit PDMA information only for the specified queue. The information displayed pertains to the ASIC you specified for the *asic_id* argument.
active	Displays descriptions for active queues only.
detail	Displays descriptions for any queues, regardless of whether they are active.
pio	Displays transmit and receive Polled I/O (PIO) information for the specified queue.
location *node-id*	(Optional) Identifies the location of the node whose CPU controller port information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	The **show controllers cpuctrl ports** command was first supported under the interface task-ID. The **queue** and **direction** keywords were added to the **show controllers cpuctrl ports** command.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

This example shows how to display port information for the transmit and receive PIO queues on the EPSE ASIC:

```
RP/0/RP0/CPU0:router# show controllers cpuctrl ports epse pio

client name: PSE                device instance: 1
cpuctrl net port: 6             pci base: 0x98000000
map reg: 0x6076a12c             OS interrupt level: 102
Cpuctrl interrupt level: 6      OS Run priority: 6
config reg: 0x00000000          channels reg: 0x00000000
int_cause_asic_mask: 0x00000010 int_cause_error_mask: 0x00000020
int_cause_link_error_mask: 0x00000040
crc_errors: 0   sync_errors: 0  reframe_events: 0


client name: PSE                device instance: 1
cpuctrl net port: 6             pci base: 0x98000000
```

```
map reg: 0x6076a12c             OS interrupt level: 102
Cpuctrl interrupt level: 6      OS Run prority: 6
config reg: 0x00000000          channels reg: 0x00000000
int_cause_asic_mask: 0x00000010 int_cause_error_mask: 0x00000020
int_cause_link_error_mask: 0x00000040
crc_errors: 0   sync_errors: 0  reframe_events: 0


device PSE instance 1 is not detected on node 201/
```

This table describes the significant fields shown in the display.

Table 5: show controllers cpuctrl ports Field Descriptions

Field	Description
client name	Identifies the client whose port CPU controller information is displayed.
device instance	The client device instance for which the port CPU controller information is displayed.
cpuctrl net port	Identifies the CPU controller net port.
pci base	PCI[9] base.
map reg	Client map register.
OS interrupt level	Level of interrupt configured for the port. Interrupts are triggered by arrival of a packet that causes the CPU to postpone other tasks and handle the packet.
Cpuctrl interrupt level	Level of CPU controller interrupt configured for the port.
OS Run priority	Software priority.
config reg	Configuration register, expressed in hexadecimal format.
channels reg	Channel register, expressed in hexadecimal format.
int_cause_asic_mask	Internal ASIC masking information.
int_cause_error_mask	Internal error masking information.
int_cause_link_error_mask	Internal link error masking information.
crc_errors	Number of CRC[10] errors that have occurred on this port.
sync_errors	Number of synchronization errors that have occurred on this port.
reframe_events	Number of reframe events that have occurred on this port.

[9] Peripheral Component Interconnect

[10] cyclic redundancy check

Related Commands

Command	Description
show controllers cpuctrl summary, on page 57	Displays summarized information about all the ASICs accessed through the CPU controller ASICs on the router or on a specific node.

show controllers cpuctrl ports cpuctrl pio

To display polled I/O (PIO) information for the specified CPU controller ASIC, use the **show controllers cpuctrl ports cpuctrl pio** command in EXEC mode.

show controllers cpuctrl ports *asic_id* **cpuctrl pio** [**location** *node-id*]

Syntax Description

asic_id	Identifies the CPU controller ASIC whose port information you want to display. Replace the *asic_id* argument with one of the following keywords: • **cpuctrl** —Displays port information for the CPU controller ASIC instance 0. • **egressq** —Displays port information for instance 0 of the egressq ASIC. • **epse** —Displays port information for the egress packet switching engine (EPSE) ASIC. • **fabricq instance 0** —Displays port information for instance 0 of the fabricq ASIC. • **fabricq instance 1** —Displays port information for instance 1 of the fabricq ASIC. • **fabricq instance 2** —Displays port information for instance 2 of the fabricq ASIC. • **fabricq instance 3** —Displays port information for instance 3 of the fabricq ASIC. • **fia instance 0** —Displays port information for instance 0 of the fabric Interface ASIC (FIA). • **fia instance 1** —Displays port information for instance 1 of the FIA. • **ingressq** —Displays port information for the ingressq ASIC. • **ipse** —Displays port information for the ingress packet switching engine (PSE) ASIC. • **jacketcard instance 0** —Displays port information for instance 0 of the jacketcard ASIC. • **jacketcard instance 1** —Displays port information for instance 1 of the jacketcard ASIC. • **jacketcard instance 2** —Displays port information for instance 2 of the jacketcard ASIC. • **pla768** —Displays port information for the OC768 plim ASIC. • **dma**—Displays port PDMA information. • **io**—Displays port Polled I/O (PIO) information. • **plaspa instance 0** —Displays port information for instance 0 of the SPA plim ASIC. • **plaspa instance 1** —Displays port information for instance 1 of the SPA plim ASIC. • **plim asic instance 0** —Displays port information for instance 0 of the plim ASIC. • **plim asic instance 1** —Displays port information for instance 1 of the plim ASIC. • **plim asic instance 2** —Displays port information for instance 2 of the plim ASIC. • **plim asic instance 3** —Displays port information for instance 3 of the plim ASIC.
location *node-id*	(Optional) Identifies the location of the node whose CPU controller port information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.

Release	Modification
Release 3.5.0	The **show controllers cpuctrl ports** command was first supported under the interface task-ID. The **queue** and **direction** keywords were added to the **show controllers cpuctrl ports** command.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display port information for the transmit and receive PIO queues on the EPSE ASIC:

```
RP/0/RP0/CPU0:router# show controllers cpuctrl ports cpuctrl pio location 0/1/CPU0

client name: Cpuctrl            device instance: 0
cpuctrl net port: 0             pci base: 0x80000000
map reg: 0x00000000             OS interrupt level: 96
Cpuctrl interrupt level: 0      OS Run prority: 0
config reg: 0x00000000          channels reg: 0x00000000
int_cause_asic_mask: 0x00000000 int_cause_error_mask: 0x00000000

int_cause_link_error_mask: 0x00000000
crc_errors: 0   sync_errors: 0  reframe_events: 0
```

Related Commands

Command	Description
show controllers cpuctrl ports, on page 51	Displays port information for the specified CPU controller ASIC.
show controllers cpuctrl summary, on page 57	Displays summarized information about all the ASICs accessed through the CPU controller ASICs on the router or on a specific node.

show controllers cpuctrl summary

To display summarized information about all the ASICs accessed through the CPU controller ASICs on the router or on a specific node, use the **show controllers cpuctrl summary** command in EXEC mode.

show controllers cpuctrl summary [**location** *node-id*]

Syntax Description

location *node-id*	(Optional) Identifies the location of the node whose CPU controller ASIC information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	The **show controllers cpuctrl summary** command was first supported under the interface task-ID.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display summarized information about all the ASICs accessed through the CPU controller ASICs on a specific node:

```
RP/0/RP0/CPU0:router# show controllers cpuctrl summary location 0/0/CPU0

Cpuctrl discovered 11 device on node 0/0/CPU0:
Cpuctrl HW version string for this node is:
Squid FPGA v2.07 Fri Jan 23 16:21:01 2004 ykoren

------------------------------------------------------------
device_name:         Fabricq    device instance:    0
Cpuctrl net port:    3          pci_base:            0x8c000000

------------------------------------------------------------
device_name:         Fabricq    device instance:    1
Cpuctrl net port:    4          pci_base:            0x90000000

------------------------------------------------------------
device_name:         Ingressq   device instance:    0
Cpuctrl net port:    8          pci_base:            0xa0000000

------------------------------------------------------------
```

```
device_name:         Egressq    device instance:    0
Cpuctrl net port:    7          pci_base:            0x9c000000

-------------------------------------------------------------
device_name:         FIA        device instance:    0
Cpuctrl net port:    1          pci_base:            0x84000000

-------------------------------------------------------------
device_name:         FIA        device instance:    1
Cpuctrl net port:    2          pci_base:            0x88000000

-------------------------------------------------------------
device_name:         Cpuctrl    device instance:    0
Cpuctrl net port:    0          pci_base:            0x80000000

-------------------------------------------------------------
device_name:         PSE        device instance:    1
Cpuctrl net port:    6          pci_base:            0x98000000

 --More--
```

```
RP/0/RP0/CPU0:router# show controllers cpuctrl summary location 0/5/CPU0

Cpuctrl discovered 2 device on node 0/5/CPU0

-------------------------------------------------
 Device Name: NPU     Device Instance: 16
Cpuctrl Port: 16     PCI Base Address: 0x1000000000
            :         PCI Window Size: 0x00000000

-------------------------------------------------
 Device Name: NPU     Device Instance: 0
Cpuctrl Port: 216    PCI Base Address: 0xd800000000
            :         PCI Window Size: 0x00000000
```

This table describes the significant fields shown in the display.

Table 6: show controllers cpuctrl summary Field Descriptions

Field	Description
device_name	Identifies the device whose CPU controller information is displayed.
device instance	Identifies the CPU device whose information is displayed.
Cpuctrl net port	Identifies the CPU controller network port.
pci_base	PCI[11] base, in hexadecimal format.

[11] Peripheral Component Interconnect

Related Commands

Command	Description
show controllers cpuctrl clients, on page 35	Displays information about all CPU controller clients on the router, or for specific CPU controller clients.
show controllers cpuctrl devices, on page 41	Displays information about the CPU controller devices on the router.

Command	Description
show controllers cpuctrl ports, on page 51	Displays port information for the specified CPU controller ASIC.

show controllers cpuctrl trace

To display the trace information about all the ASICs accessed through the CPU controller ASICs on the router or on a specific node, use the **show controllers cpuctrl trace** command in EXEC mode.

show controllers cpuctrl trace [{**all** | **client** | **queue** | **server**}]

Syntax Description

all	Displays the trace information of all the devices.
client	Displays all the trace information of all the cdma or pdma clients.
queue	Displays all the trace information of all the ASIC queues.
server	Displays all the server trace information.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.8.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display statistics information of all the CDMA CPU controller ASIC queues:

```
RP/0/RP0/CPU0:router# show controllers cpuctrl trace queue all cdma stats

/dev/shmem/ltrace/squid/queue/dma/chan0/queue4: 278 searches, 462 elements, 1.7h
/dev/shmem/ltrace/squid/queue/dma/chan0/queue6: 51 searches, 85 elements, 1.7 eh
/dev/shmem/ltrace/squid/queue/dma/chan0/queue4 unique: 3.584 Mbytes/sec for 64 s
/dev/shmem/ltrace/squid/queue/dma/chan0/queue6 unique: inf Mbytes/sec for 64 ens
10 unique entries (128 possible, 0 filtered)
Jul 14 19:29:06.548 squid/queue/dma/chan0/queue6 0/RP0/CPU0 1# t1  INI:DMA:000:0
Jul 14 19:29:06.936 squid/queue/dma/chan0/queue4 0/RP0/CPU0 1# t1  INI:DMA:000:0
Jul 14 19:29:11.192 squid/queue/dma/chan0/queue4 0/RP0/CPU0 1# t5  BOT:HWC:011:0
Jul 14 20:50:56.795 squid/queue/dma/chan0/queue6 0/RP0/CPU0 17# t12 TOP:DMA:0101
Jul 14 20:50:56.803 squid/queue/dma/chan0/queue6 0/RP0/CPU0 17# t5  BOT:HWC:0120
```

```
Jul 14 20:50:56.803 squid/queue/dma/chan0/queue6 0/RP0/CPU0 14# t5  BOT:HWC:0101
Jul 14 20:50:56.803 squid/queue/dma/chan0/queue6 0/RP0/CPU0 2# t5  BOT:HWC:011:1
Jul 14 20:54:34.970 squid/queue/dma/chan0/queue4 0/RP0/CPU0 92# t12 TOP:DMA:0101
Jul 14 20:54:34.970 squid/queue/dma/chan0/queue4 0/RP0/CPU0 92# t5  BOT:HWC:010b
Jul 14 20:54:34.970 squid/queue/dma/chan0/queue4 0/RP0/CPU0 92# t5  BOT:HWC:0120
/dev/shmem/ltrace/squid/queue/dma/chan0/queue4 wrapping: inf Mbytes/sec for 128s
/dev/shmem/ltrace/squid/queue/dma/chan0/queue6 wrapping: inf Mbytes/sec for 128s
176 wrapping entries (256 possible, 0 filtered, 329 total)
Jul 14 19:29:06.548 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  INI:DMA:000: Cr0
Jul 14 19:29:08.586 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DM0
Jul 14 19:29:08.586 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:011: DM0
Jul 14 19:29:13.214 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:29:13.217 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DM0
Jul 14 19:29:13.217 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:13.219 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:29:13.224 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DM1
Jul 14 19:29:13.224 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:13.233 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:29:13.237 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DM2
Jul 14 19:29:13.237 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:13.241 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:29:13.243 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DM3
Jul 14 19:29:13.243 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:13.249 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:29:13.253 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DM4
Jul 14 19:29:13.253 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:13.265 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:29:13.269 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DM5
Jul 14 19:29:13.269 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:19.212 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM1
Jul 14 19:29:19.212 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM1
Jul 14 19:29:19.212 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM1
Jul 14 19:29:19.218 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DM6
Jul 14 19:29:19.218 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:19.218 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:19.218 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:19.218 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:29:19.226 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DM9
Jul 14 19:29:19.226 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:19.235 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:29:19.241 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DMa
Jul 14 19:29:19.241 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:19.249 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:29:19.256 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DMb
Jul 14 19:29:19.256 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:29:19.262 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:29:19.269 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DMc
Jul 14 19:29:19.269 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:04.148 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM2
Jul 14 19:30:04.148 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:04.149 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:04.151 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM3
Jul 14 19:30:04.151 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:05.160 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:05.161 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM4
Jul 14 19:30:05.161 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:05.163 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:05.168 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM5
Jul 14 19:30:05.168 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:06.172 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:06.175 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM6
Jul 14 19:30:06.175 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:06.175 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:06.182 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM7
```

```
Jul 14 19:30:06.182 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:07.189 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:07.189 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM8
Jul 14 19:30:07.189 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:07.189 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:07.190 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM9
Jul 14 19:30:07.190 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:08.192 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:08.193 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMa
Jul 14 19:30:08.193 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:08.194 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:08.194 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMb
Jul 14 19:30:08.194 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:09.198 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:09.199 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMc
Jul 14 19:30:09.199 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:09.199 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:09.200 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMd
Jul 14 19:30:09.200 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:10.207 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:10.211 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMe
Jul 14 19:30:10.211 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:10.215 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:10.221 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMf
Jul 14 19:30:10.221 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:11.231 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:11.234 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM0
Jul 14 19:30:11.234 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:11.240 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:11.247 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM1
Jul 14 19:30:11.247 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:12.253 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:12.254 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM2
Jul 14 19:30:12.254 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:12.254 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:12.254 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM3
Jul 14 19:30:12.254 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:13.259 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:13.261 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM4
Jul 14 19:30:13.261 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:13.263 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:13.264 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM5
Jul 14 19:30:13.264 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:14.268 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:14.272 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM6
Jul 14 19:30:14.272 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:14.272 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:14.273 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM7
Jul 14 19:30:14.273 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:15.275 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:15.276 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM8
Jul 14 19:30:15.276 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:15.277 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:15.278 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM9
Jul 14 19:30:15.278 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:16.285 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:16.293 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMa
Jul 14 19:30:16.293 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:16.301 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:16.308 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMb
Jul 14 19:30:16.308 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:17.320 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:17.324 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMc
Jul 14 19:30:17.324 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
```

```
Jul 14 19:30:17.328 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:17.332 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMd
Jul 14 19:30:17.332 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:18.338 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:18.339 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMe
Jul 14 19:30:18.339 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:18.340 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:18.340 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMf
Jul 14 19:30:18.340 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:19.342 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:19.346 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM0
Jul 14 19:30:19.346 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:19.350 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:19.354 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM1
Jul 14 19:30:19.354 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:30:20.360 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t1  TOP:DMA:010: DM0
Jul 14 19:30:20.364 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM2
Jul 14 19:30:20.364 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:32:06.165 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 19:32:06.166 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM3
Jul 14 19:32:06.166 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:32:06.614 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 19:32:06.622 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM4
Jul 14 19:32:06.622 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:32:26.670 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 19:32:26.675 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM5
Jul 14 19:32:26.675 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:32:26.878 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 19:32:26.887 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM6
Jul 14 19:32:26.887 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:32:26.927 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 19:32:26.935 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM7
Jul 14 19:32:26.935 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 19:32:27.144 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 19:32:27.150 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM8
Jul 14 19:32:27.150 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 20:50:56.806 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 20:50:56.806 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 20:50:56.816 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DMd
Jul 14 20:50:56.816 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 20:50:56.816 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 20:50:56.817 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 20:50:56.825 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 20:50:56.825 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 20:50:56.825 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 20:50:56.825 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:010: DM1
Jul 14 20:50:56.825 squid/queue/dma/chan0/queue6 0/RP0/CPU0 t5  BOT:HWC:011: DM1
Jul 14 20:51:30.093 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 20:51:30.117 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DM9
Jul 14 20:51:30.117 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 20:54:00.889 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 20:54:00.890 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMa
Jul 14 20:54:00.890 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
Jul 14 20:54:34.993 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t12 TOP:DMA:010: DM1
Jul 14 20:54:34.993 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:010: DMb
Jul 14 20:54:34.993 squid/queue/dma/chan0/queue4 0/RP0/CPU0 t5  BOT:HWC:012: Us0
```

The following example shows how to display trace information for internal unique entries of the CPU controller:

show controllers cpuctrl vcdma

To display the information about all the show commands in the VCDMA library, use the **show controllers cpuctrl vcdma** command in EXEC mode.

show controllers cpuctrl vcdma [{**cdma** | **trace**}]

Syntax Description

cdma	Displays the information related to a specific or all cdma channels.
trace	Displays the trace information of a specified controller ASIC.
all	Displays the trace information of all the controller ASICs.
client	Displays the trace information of the cdma clients.
queue	Displays the trace information of all the ASIC queues.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.8.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display the cdma channel information of the active queue descriptors accessed by the CPU controller ASICs:

```
RP/0/RP0/CPU0:router# show controllers cpuctrl vcdma cdma channel all active

No queue present for channel 0, queue 0 on 0/1/CPU0
No queue present for channel 0, queue 1 on 0/1/CPU0
No queue present for channel 0, queue 2 on 0/1/CPU0
No queue present for channel 0, queue 3 on 0/1/CPU0
No queue present for channel 0, queue 4 on 0/1/CPU0
No queue present for channel 0, queue 5 on 0/1/CPU0
No queue present for channel 0, queue 6 on 0/1/CPU0

------------------------------------------------------
VCDMA queue:
```

```
Channel: 0     queue: 7       vqueue state: 0x1

            client handle =          4          client pid =      81997
      client event handle = 0x5001a998     client callback = 0x4c9d13e8
     descriptor ring size =       1024
       Queue create count =          1  Queue delete count =          0
      Total VCDMA Batches =     200227       VCDMA Batches =     200227
   Total DMA transactions =     211808    DMA transactions =     211808
    Total bytes =           3645826297     bytes =           3645826297
   Total Out of desc errs =          0    Out of desc errs =          0
VCDMA Batches active on this queue: 0

No queue present for channel 1, queue 0 on 0/1/CPU0
No queue present for channel 1, queue 1 on 0/1/CPU0
No queue present for channel 1, queue 2 on 0/1/CPU0
No queue present for channel 1, queue 3 on 0/1/CPU0
No queue present for channel 1, queue 4 on 0/1/CPU0
No queue present for channel 1, queue 5 on 0/1/CPU0
No queue present for channel 1, queue 6 on 0/1/CPU0
No queue present for channel 1, queue 7 on 0/1/CPU0
No queue present for channel 0, queue 0 on 0/6/CPU0
No queue present for channel 0, queue 1 on 0/6/CPU0
No queue present for channel 0, queue 2 on 0/6/CPU0
No queue present for channel 0, queue 3 on 0/6/CPU0
No queue present for channel 0, queue 4 on 0/6/CPU0
No queue present for channel 0, queue 5 on 0/6/CPU0
No queue present for channel 0, queue 6 on 0/6/CPU0

-----------------------------------------------------
VCDMA queue:
Channel: 0     queue: 7       vqueue state: 0x1

            client handle =          4          client pid =      81980
      client event handle = 0x5001a998     client callback = 0x4c9ba3e8
     descriptor ring size =       1024
       Queue create count =          1  Queue delete count =          0
      Total VCDMA Batches =     200238       VCDMA Batches =     200238
   Total DMA transactions =     211819    DMA transactions =     211819
    Total bytes =           3646313721     bytes =           3646313721
   Total Out of desc errs =          0    Out of desc errs =          0
VCDMA Batches active on this queue: 0

No queue present for channel 1, queue 0 on 0/6/CPU0
No queue present for channel 1, queue 1 on 0/6/CPU0
No queue present for channel 1, queue 2 on 0/6/CPU0
No queue present for channel 1, queue 3 on 0/6/CPU0
No queue present for channel 1, queue 4 on 0/6/CPU0
No queue present for channel 1, queue 5 on 0/6/CPU0
No queue present for channel 1, queue 6 on 0/6/CPU0
No queue present for channel 1, queue 7 on 0/6/CPU0
```

show controllers egressq eio links

To display Elastic I/O (EIO) information for the egress queueing ASIC, use the **show controllers egressq eio links** command in EXEC mode.

show controllers egressq eio links {*link-id* | **all**} **location** *node-id*

Syntax Description

link-id	Identifies the EIO link whose egress queueing ASIC information you want to display. Replace the *link-id* argument with a link identifier. Range is from 0 to 4294967295.
all	Displays egress queueing ASIC information for all links on the specified node.
location *node-id*	Identifies the location of the egress queue manager whose EIO link information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display information for all EIO links on a node's egress queueing ASIC:

```
RP/0/RP0/CPU0:router# show controllers egressq eio links all location 0/2/CPU0

             Node: 0/2/CPU0:
----------------------------------------

EIO links:
------------------------------------------------------------------------
ASIC Id     Peer Id     Type Link-Id Attempts Accept Failed State
------------------------------------------------------------------------
EGRESSQ_0   PSE_1       RX   11      1        1      0      EIO_LINK_TRAINED
EGRESSQ_0   PSE_1       TX   64      0        0      0      EIO_LINK_TRAINED
EGRESSQ_0   PLIM-ASIC_0 TX   60      0        0      0      EIO_LINK_TRAINED
EGRESSQ_0   PLIM-ASIC_1 TX   61      0        0      0      EIO_ADD_LINK
EGRESSQ_0   PLIM-ASIC_2 TX   62      0        0      0      EIO_ADD_LINK
```

```
EGRESSQ_0    PLIM-ASIC_3 TX    63       0         0       0       EIO_ADD_LINK
```

This table describes the significant fields shown in the display.

Table 7: show controllers egressq eio links Field Descriptions

Field	Description
ASIC Id	Identifies the ASIC that is associated with the link whose information is displayed.
Peer Id	Identifies the ASIC peer.
Type	Indicates whether the AISC is receiving (RX) or transmitting (TX).
Link-Id	Link identifier.
Attempts	Number of attempts made to send or receive information on the specified link.
Accept	Number of consecutive EIO requests for the link made by the PSE driver to the peer that were accepted.
Failed	Number of consecutive EIO requests for the link that failed.
State	Current link state.

Related Commands

Command	Description
show controllers egressq group, on page 68	Displays information about egress queue groups.
show controllers egressq port, on page 74	Displays egress queue information for a port, or for several ports.

show controllers egressq group

To display information about egress queue groups, use the **show controllers egressq group** command in EXEC mode.

show controllers egressq group {*group1* [*group2*] | **all** | **limits** {**max** | **min**}} **location** *node-id*

Syntax Description

group1	First group whose information you want to display. Range is from 1 to 2047.
group2	(Optional) Last group whose information you want to display. Range is from 1 to 2047.
all	Indicates that you want to display information about all egress queue groups on a specific node.
limits	Indicates that you want to display the limits table. Follow the keyword with **max** to display the maximum limits table, or **min** to display the minimum limits table.
max	Display the maximum limits table.
min	Display the minimum limits table.
location *node-id*	Identifies the location of the egress queue whose group information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

If you do not specify a group number for the *group2* argument, information is displayed only for the egress queue you specified for the *group1* argument. If you specify *group1* and *group2* , information is displayed for all groups within the range from *group1* through *group2* .

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display information for a specific range of egress queue groups on the CPU node in slot 2:

```
RP/0/RP0/CPU0:router# show controllers egressq group 1 3 location 0/2/CPU0
```

```
---------------------------------
          Group 1
---------------------------------
 Port                : 0
 Priority            : Low
 Max LB Tokens       : 7812
 Max LB Limit Index  : 38
 Min LB Tokens       : 0
 Min LB Limit Index  : 0
 Quantum             : 27
 Default Queue       : 1
 High Priority Queue : N/A
 Low Priority Queue  : 1
---------------------------------
---------------------------------
          Group 2
---------------------------------
 Port                : 1
 Priority            : Low
 Max LB Tokens       : 7812
 Max LB Limit Index  : 38
 Min LB Tokens       : 0
 Min LB Limit Index  : 0
 Quantum             : 27
 Default Queue       : 2
 High Priority Queue : N/A
 Low Priority Queue  : 2
---------------------------------
---------------------------------
          Group 3
---------------------------------
 Port                : 256
 Priority            : Low
 Max LB Tokens       : 7812
 Max LB Limit Index  : 38
 Min LB Tokens       : 0
 Min LB Limit Index  : 0
 Quantum             : 27
 Default Queue       : 3
 High Priority Queue : N/A
 Low Priority Queue  : 3
---------------------------------
```

This table describes the significant fields shown in the display.

Table 8: show controllers egressq group Field Descriptions

Field	Description
Port	Port identifier.
Priority	Group priority.
Max LB Tokens	Maximum number of port LB tokens.
Max LB Limit Index	Maximum LB limit.
Min LB Tokens	Minimum number of port LB tokens.
Min LB Limit Index	Minimum LB limit.

Field	Description
Quantum	Quantum value for this group.
Default Queue	Default queue for this group.
High Priority Queue	High priority queue identifier.
Low Priority Queue	Low priority queue identifier.

Related Commands

Command	Description
show controllers egressq eio links, on page 66	Displays Elastic I/O (EIO) information for the egress queueing ASIC.
show controllers egressq port, on page 74	Displays egress queue information for a port, or for several ports.

show controllers egressq interface

To display information about interfaces associated with an egress queue, use the **show controllers egressq interface** command in EXEC mode.

show controllers egressq interface [{*type interface-path-id* | **all**}] **location** *node-id*

Syntax Description

type interface-path-id	Identifies a physical interface or a virtual interface. **Note** Use the **show interfaces** command to see a list of all possible interfaces currently configured on the router.
all	Indicates that you want to display egress queue information for all interfaces in the specified **location**.
location *node-id*	Identifies the location of the interface whose egress queue information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following example shows how to display egress queue information for all configured interfaces on the router:

```
RP/0/RP0/CPU0:router# show controllers egressq interface all

---------------------------------
          Interface POS0/0/0/15
---------------------------------
          Port 776
---------------------------------
 Max LB Tokens       : 18720
 Max LB Limit Index  : 41
 Quantum             : 27
 Default Group       : 24
 High Priority Group : N/A
 Low Priority Group  : 24
---------------------------------
```

```
---------------------------------
          Interface POS0/0/0/14
---------------------------------
          Port 780
---------------------------------
 Max LB Tokens       : 18720
 Max LB Limit Index  : 41
 Quantum             : 27
 Default Group       : 23
 High Priority Group : N/A
 Low Priority Group  : 23
---------------------------------
---------------------------------
          Interface POS0/0/0/13
---------------------------------
          Port 778
---------------------------------
 Max LB Tokens       : 18720
 Max LB Limit Index  : 41
 Quantum             : 27
 Default Group       : 22
 High Priority Group : N/A
 Low Priority Group  : 22
---------------------------------
---------------------------------
          Interface POS0/0/0/12
---------------------------------
          Port 782
---------------------------------
 Max LB Tokens       : 18720
 Max LB Limit Index  : 41
 Quantum             : 27
 Default Group       : 21
 High Priority Group : N/A
 Low Priority Group  : 21
---------------------------------
---------------------------------
          Interface POS0/0/0/11
---------------------------------
          Port 520
---------------------------------
 Max LB Tokens       : 18720
 Max LB Limit Index  : 41
 Quantum             : 27
 Default Group       : 20
 High Priority Group : N/A
 Low Priority Group  : 20
---------------------------------
---------------------------------
          Interface POS0/0/0/10
---------------------------------
          Port 524
---------------------------------
 Max LB Tokens       : 18720
 Max LB Limit Index  : 41
 Quantum             : 27
 Default Group       : 19
 High Priority Group : N/A
 Low Priority Group  : 19
---------------------------------
---------------------------------
          Interface POS0/0/0/9
---------------------------------
          Port 522
```

```
---------------------------------
 Max LB Tokens       : 18720
 Max LB Limit Index  : 41
 Quantum             : 27
 Default Group       : 18
 High Priority Group : N/A
 Low Priority Group  : 18
---------------------------------
---------------------------------
          Interface POS0/0/0/8
---------------------------------
          Port 526
---------------------------------
 Max LB Tokens       : 18720
 Max LB Limit Index  : 41
 Quantum             : 27
 Default Group       : 17
 High Priority Group : N/A
 Low Priority Group  : 17
---------------------------------
---------------------------------
```

This table describes the significant fields shown in the display.

Table 9: show controllers egressq interface Field Descriptions

Field	Description
Interface	Interface identifier, in the <*type*>*rack/slot/module/port* format.
Port	Port to which the specified interface belongs.
Max LB Tokens	Maximum number of port LB tokens.
Max LB Limit Index	Maximum LB limit.
Quantum	Average number of bytes in the interface egress queue.
Default Group	Default group for this interface.
High Priority Group	High priority group identifier.
Low Priority Group	Low priority group identifier.

Related Commands

Command	Description
show controllers egressq eio links, on page 66	Displays Elastic I/O (EIO) information for the egress queueing ASIC.
show controllers egressq group, on page 68	Displays information about egress queue groups.
show controllers egressq port, on page 74	Displays egress queue information for a port, or for several ports.

show controllers egressq port

To display egress queue information for a port, or for several ports, use the **show controllers egressq port** command in EXEC mode.

show controllers egressq port {*port1* [*port2*] | **all** | **limits max**} **location** *node-id*

Syntax Description

port1	Identifies the port whose egress queue information you want to display. Replace the *port1* argument with a port identifier. Range is from 0 to 1023.
port2	Identifies the last port whose egress queue information you want to display. Replace the *port2* argument with a port identifier. Range is from 0 to 1023.
all	Indicates that you want to display egress queue information for all ports on the node in the specified location.
limits max	Indicates that you want to display the maximum limits table.
location *node-id*	Identifies the location of the egress queue whose group information you want to display. The *node-id* argument is expressed 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

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following example shows how to display egress queue information for ports 1, 2, and 3 on the CPU node in slot 2:

```
RP/0/RP0/CPU0:router# show controllers egressq port 1 3 location 0/2/CPU0

---------------------------------
          Port 1
---------------------------------
 Max LB Tokens       : 7812
 Max LB Limit Index  : 38
```

```
 Quantum             : 27
 Default Group       : 2
 High Priority Group : N/A
 Low Priority Group  : 2
---------------------------------


---------------------------------
          Port 2
---------------------------------
Port is not allocated
---------------------------------
          Port 3
---------------------------------
Port is not allocated
```

This table describes the significant fields shown in the display.

Table 10: show controllers egressq port Field Descriptions

Field	Description
Port	Port identifier.
Max LB Tokens	Maximum number of port LB tokens.
Max LB Limit Index	Maximum LB limit.
Quantum	Quantum value for this interface.
Default Group	Default group.
High Priority Group	High priority group identifier.
Low Priority Group	Low priority group identifier.

Related Commands

Command	Description
show controllers egressq eio links, on page 66	Displays Elastic I/O (EIO) information for the egress queueing ASIC.
show controllers egressq group, on page 68	Displays information about egress queue groups.

show controllers egressq queue

To display information about a specific egress queue, or a range of egress queues, use the **show controllers egressq queue** command in EXEC mode.

show controllers egressq queue {**queue1** [**queue2**] | **from-group** {**group-id** | **all**} | **from-interface** {**type instance** | **all**} | **from-port** {**port-id** | **all**} | **limits** {**max** | **min**} | **all**} **location node-id**

Syntax Description

queue1	Identifies the first port whose egress queue information you want to display. Replace the *queue1* argument with a queue identifier. Range is from 1 through 8191.
queue2	(Optional) Identifies the last port whose egress queue information you want to display. Replace the *queue2* argument with a queue identifier. Range is from 1 through 8191.
from-group *group-id*	Displays egress queue information for a specific port group. Replace the *group-id* argument with the number that identifies the port group whose information is to be displayed. Range is from 1 through 2047.
from-group all	Displays egress queue information for all port groups on the specified location.
from-interface *type instance*	Displays egress queue information for a specific interface. Replace the *type instance* argument with a physical interface ID or a virtual interface ID. **Note** Use the **show interfaces** command to see a list of all possible interfaces currently configured on the router.
from-interface all	Displays egress queue information for all interfaces on the specified location.
from-port *port-id*	Displays egress queue information for a specific port. Replace the *port-id* argument with a port ID. Range is from 0 to 1023.
from-port all	Displays egress queue information for all ports on the specified location.
limits max	Displays the maximum limits table.
l**imits min**	Displays the minimum limits table.
all	Displays detailed information about all egress queues on the specified location.
location *node-id*	Identifies the location of the node whose egress queue information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.2	This command was introduced.

Release	Modification
Release 3.5.0	The **context** keyword and *max_elements* argument were removed from the **show controllers egressq queue** command. The following keywords and arguments were added to the **show controllers egressq queue** command: • **from-group** { *group-id* | **all** } • **from-interface** { *type instance* | **all** } • **from-port** { *port-id* | **all**}

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display information about egress queues 1 and 2 on the CPU node in slot 2:

```
RP/0/RP0/CPU0:router# show controllers egressq queue 1 2 location 0/2/CPU0
----------------------------------
          Queue 1
----------------------------------
 Group                 : 1
 Priority              : Low
 Max LB Tokens         : 7812
 Max LB Limit Index    : 38
 Min LB Tokens         : 0
 Min LB Limit Index    : 0
 Quantum               : 27
 Instantaneous length  : 0
 Length high watermark : 0
----------------------------------
----------------------------------
          Queue 2
----------------------------------
 Group                 : 2
 Priority              : Low
 Max LB Tokens         : 7812
 Max LB Limit Index    : 38
 Min LB Tokens         : 0
 Min LB Limit Index    : 0
 Quantum               : 27
 Instantaneous length  : 0
 Length high watermark : 0
```

This table describes the significant fields shown in the display.

Table 11: show controllers egressq queue Field Descriptions

Field	Description
Group	Group identifier.
Priority	Group priority.
Max LB Tokens	Maximum number of port LB tokens.
Max LB Limit Index	Maximum LB limit.
Min LB Tokens	Minimum number of port LB tokens.
Min LB Limit Index	Minimum LB limit.
Quantum	Quantum value for this queue.
Instantaneous length	Egress queue instantaneous length.
Length high watermark	Length high watermark register.

Related Commands

Command	Description
show controllers egressq eio links, on page 66	Displays Elastic I/O (EIO) information for the egress queueing ASIC.
show controllers egressq group, on page 68	Displays information about egress queue groups.
show controllers egressq port, on page 74	Displays egress queue information for a port, or for several ports.

show controllers egressq statistics

To display egress queue manager statistics, use the **show controllers egressq statistics** command in EXEC mode.

show controllers egressq statistics location *node-id*

Syntax Description

location *node-id*	Identifies the location of the egress queue whose statistics you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.2	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following example shows how to display egress queue manager statistics:

```
RP/0/RP0/CPU0:router# show controllers egressq statistics location 0/2/CPU0

-------------------------------------------------
             Egressq Statistics
-------------------------------------------------
egressq ASIC version: 1
egressq ASIC state: Normal
plimasic link0 output packets: 0
plimasic link0 output bytes: 0
plimasic link1 output packets: 0
plimasic link1 output bytes: 0
plimasic link2 output packets: 0
plimasic link2 output bytes: 0
plimasic link3 output packets: 0
plimasic link3 output bytes: 0
cpuctrl input packets: 0
cpuctrl output bytes: 1433250
pse input packets: 29250
pse dropped packets: 0
cpuctrl dropped packets: 0
```

This table describes the significant fields shown in the display.

Table 12: show controllers egressq statistics Field Descriptions

Field	Description
egressq ASIC version	Version identifier for the egress queue ASIC.
egressq ASIC state	Current state of the egress queue ASIC.
plimasic link0 output packets	Number of output packets on link 0 of the ASIC.
plimasic link0 output bytes	Output bytes on link 0 of the ASIC.
plimasic link1 output packets	Number of output packets on link 1 of the ASIC.
plimasic link1 output bytes	Number of output packets on link 1 of the ASIC.
plimasic link2 output packets	Number of output packets on link 2 of the ASIC
plimasic link2 output bytes	Number of output packets on link 2 of the ASIC.
plimasic link3 output packets	Number of output packets on link 3 of the ASIC.
plimasic link3 output bytes	Number of output packets on link 3 of the ASIC.
cpuctrl input packets	Number of CPU controller input packets received on this ASIC.
cpuctrl output bytes	CPU controller output bytes transmitted on this ASIC.
pse input packets	Number of PSE input packets received on this ASIC.
pse dropped packets	Number of PSE packets that were dropped by this ASIC due to errors.
cpuctrl dropped packets	Number of CPU controller packets that were dropped by this ASIC due to errors.

Related Commands

Command	Description
show controllers egressq eio links, on page 66	Displays Elastic I/O (EIO) information for the egress queueing ASIC.
show controllers egressq group, on page 68	Displays information about egress queue groups.
show controllers egressq port, on page 74	Displays egress queue information for a port, or for several ports.

show controllers egressq trace

To display the internal trace buffer information for Egressq on a specific controller or node, use the **show controllers egressq trace** command in EXEC mode.

show controllers egressq trace [{**all** | **c2c** | **cdma** | **detail** | **errors** | **func** | **info** | **interrupt** | **periodic**}] **location** *node-id*

Syntax Description

all	(Optional) Displays the internal trace (Itrace) information for all of the egressq errors and events.
c2c	(Optional) Displays the internal trace for C2C or EIO link processing information.
cdma	(Optional) Displays the internal trace information for egressq cdma/vcdma processing events.
detail	(Optional) Displays the detailed processing internal trace data for egressq events.
errors	(Optional) Displays the internal trace information for all kinds of egressq processing errors.
func	(Optional) Displays the internal trace data for egressq function entry/exit events.
info	(Optional) Displays the internal trace data for general information with processing events.
interrupt	(Optional) Displays the internal trace data for general information with interrupt events.
periodic	(Optional) Displays the internal trace data for general information with periodic events.
location *node-id*	Identifies the location of the egress queue whose statistics you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.9.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
cisco-support	read
interface	read

Examples

The following example shows how to display the internal trace information of Egressq for a specific controller instance:

```
RP/0/RP0/CPU0:router# show controllers egressq trace info reverse location 0/2/CPU0

516 wrapping entries (16384 possible, 0 filtered, 516 total)
Oct 16 14:26:47.165 egressq/info 0/2/CPU0 t1  SHARQ-INFO: sharq_rsm_send_dma - Complete
VCDMA requests (succeed)
Oct 16 14:26:47.151 egressq/info 0/2/CPU0 t1  SHARQ-INFO: sharq_rsm_send_dma - Start sending
 VCDMA requests (elems 15)
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: Sharq remapping
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: port 526 default group is set to
 16
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: group 16 default queues are set
to LPQ 23 HPQ 24
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: queue1: 23 (priority 2) & queue2:
 24 (priority2 1) are created under group1 16 and group2 16 respectively
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: Sharq ID is:  3
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: Set queue params: queue 24,
max_lb_token 0, max_lb_index 33, min_lb_token 39063, min_lb_index: 57, weight 10, quantum
27
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: queue 24 weight is set to 10
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: queue 24 burst size is set to max
 312504, min 312504
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: queue 24 BW is set to max
10000128Kbps, min 10000128Kbps
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: queue 24 is allocated for group
16 / port 526
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: queue1 23 (priority 2) is created
 for group1 16
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: Sharq ID is:  3
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: Set queue params: queue 23,
max_lb_token 39063, max_lb_index 43, min_lb_token 0, min_lb_index: 0, weight 10, quantum
27
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: queue 23 weight is set to 10
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: queue 23 burst size is set to max
 312504, min 0
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: queue 23 BW is set to max
10000128Kbps, min 0Kbps
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: queue 23 is allocated for group
16 / port 526
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: Set group params: group 16,
max_lb_token 39023, max_lb_index 43, min_lb_token 40, min_lb_index: 34, quantum 27, priority
 2
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: group 16 weight is set to 10
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: group 16 burst size is set to max
 312504, min 320
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: group 16 BW is set to max
10000128Kbps, min 10240Kbps
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: group 16 is allocated for port
526
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: Set port params: port 526,
max_lb_token 39063, max_lb_index 43, quantum 27, default_group: 4294967295, first_group
4294967295, hp-group 4294967295
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: port 526 weight is set to 10
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: Port 526 Max burst is set to
312504
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SHARQ-INFO: port 526 MaxBW is set to
10000128Kbps
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SERVER-INFO: Create interfaces - intf type
1, port 526, default_priority 2
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SERVER-INFO: added interface: ifh 0x01280120,
```

```
 if TenGigE0/2/0/7, port 526, type 1, local_if 7, spa 0
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SERVER-INFO: Dispatch create interfaces -
handle 0x1904a, numOfParamEntries 1, tuningBestEffort 1, bulk_req 0x9bffda58, req_params
0x9bffda74
Oct 16 14:26:47.150 egressq/info 0/2/CPU0 t1  SERVER-INFO: egressq_read_msg_buffer -Received
 message length is 80
```

show controllers egressq resources

To display the Egressq resource usage on a specific controller or node, use the **show controllers egressq resources** command inEXEC mode.

show controllers egressq resources location *node-id*

Syntax Description

location *node-id*	Identifies the location of the egress queue whose statistics you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.9.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following example shows how to display the internal trace information of Egressq for a specific controller instance:

```
RP/0/RP0/CPU0:router# show controllers egressq resources location 0/1/CPU0

Resource          Total     Used     Free
-----------------------------------------
Ports              1024       29      995
Groups             2048       30     2018
Queues             8192       50     8142
```

show controllers ingressq backpressure

To display the ingressq backpressure details, on a specific controller or node, use the **show controllers ingressq backpressure** command in EXEC mode.

show controllers ingressq backpressure {**all** | **bpmem** | **hiccup** | **maptab**}

Syntax Description

all	Displays backpressure details for all controllers.
bpmem	Displays backpresssure bitmap memory content.
hiccup	Displays hiccup details.
maptab	Displays FGID mapping tables.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 4.0.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following example shows how to display the internal trace information of Egressq for a specific controller instance:

```
RP/0/RP0/CPU0:router# show controllers ingressq backpressure all
```

show controllers ingressq block

To display the ingressq asic block details, on a specific controller or node, use the **show controllers ingressq block** command in EXEC mode.

show controllers ingressq block {**brm** | **cri** | **dcm** | **dqp** | **dqs** | **eproc** | **fbm** | **fqm** | **help** | **internal** | **isq** | **lns** | **mti** | **pfi** | **pgi** | **pii** | **pmi** | **qdm** | **seg** | **spi** | **sqe** | **ssm** | **tfi**}

Syntax Description

brm	Displays Barrier Manager (BRM) details.
cri	Displays Crab Interface (CRI) statistics.
dcm	Displays Discard Manager (DCM) details.
dqp	Displays Destination Queue (DQP) details.
dqs	Displays Destination Queue Scheduler (DQS) details.
eproc	Displays EPROC block data.
fbm	Displays Free Block Manager (FBM) details.
fqm	Displays Free Queue Manager (FQM) deyails.
help	Displays supported block names.
internal	Displays ingressq internal datastructure.
isq	Displays Input Shape Queue (ISQ) details.
lns	Displays Link Scheduler (LNS) details.
mti	Displays pse interface control data.
pfi	Displays Plane Fabric Interface (PFI) details.
pgi	Displays Pogo Interface (PGI) details.
pii	Displays Packet Ingress Interface (PII) control data.
pmi	Displays Packet Memory Interface (PMI) control data.
qdm	Displays QDM details.
seg	Displays segmentation unit details.
spi	Displays ingressq interface control data.
sqe	Displays Shape Queue Engine (SQE) details.
ssm	Displays Show Start Manager (SSM) details.
tfi	Displays To Fabric Interface (TFI) CLI.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 4.0.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following example shows how to display the internal trace information of Egressq for a specific controller instance:

```
RP/0/RP0/CPU0:router# show controllers ingressq block isq
```

show controllers plim asic ether queues

To display the ethernet PLIM asic information, use the **show controllers plim asic ether queues** command in EXEC mode.

show controllers plim asic ether queues location [{*node-idname*}]

Syntax Description

location [*node-id* | *name*]	Identifies the location of the ethernet. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 4.0.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following example shows how to display the internal trace information of Egressq for a specific controller instance:

```
RP/0/RP0/CPU0:router# show controllers plim asic ether queues location 0/1/CPU0
```

show controllers plim asic pla

To display physical layer interface module (PLIM) ASIC line-card information for the plane card, use the **show controllers plim asic pla** command in EXEC mode.

show controllers plim asic pla {**clients** | **eio** {**link_id** | **all**} **error** | **packet mbp instance number** | **pointers** | **sbp**} [**location** *node-id*]

Syntax Description

clients	Displays information for PLA clients.
eio	Displays information about EIO commands for PLA.
link_id	Displays PLA information for a specific link ID. Range is from 0 to 4294967295.
all	Displays PLA information for all links.
error	Displays PLA error information.
packet	Displays information for the dump packet.
mbp	Displays MBP buffer information.
instance number	Displays the PLA instance number for a specific instance. Range is from 0 to 3.
pointers	Displays information for PLA pointers.
sbp	Displays sharp back-pressure (SBP) table.
location *node-id*	Identifies the location of the node whose PLIM ASIC information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.8.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
drivers	read

Task ID	Operations
cisco-support	read

Examples

The following example shows how to display PLIM ASIC PLA information for an interface on a router:

```
RP/0/RP0/CPU0:router# show controllers plim asic pla eio links?

<0-4294967295>  link id
all             show all links
```

show controllers plim asic pla768

To display physical layer interface module (PLIM) ASIC line-card information for the OC-768 card, use the **show controllers plim asic pla768** command in EXEC mode.

show controllers plim asic pla768 {**ports** | **eio** {**link_id** | **all**}} [**location** *node-id*]

Syntax Description

ports	Displays port mapping information.
eio	Displays information about Elastic I/O (EIO) commands for the Reindeer ASIC.
link_id	Displays OC-768 information for a specific link ID. Range is from 0 to 4294967295.
all	Displays PLA information for all links.
location *node-id*	Identifies the location of the node whose PLIM ASIC information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.8.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
drivers	read
cisco-support	read

Examples

The following example shows how to display PLIM ASIC PLA768 information for an interface:

```
RP/0/RP0/CPU0:router# show controllers plim asic pla768 ?

eio    Show eio commands for the Reindeer ASIC
ports  Port mapping information
```

show controllers plim asic plaspa

To display physical layer interface module (PLIM) ASIC line-card information for the shared port adapters (SPA), use the **show controllers plim asic plaspa** command in EXEC mode.

show controllers plim asic plaspa {**eio** {**link_id** | **all**} | **error** | **ifhandle** | **instance** | **num-list** | **queues** | **sbp** | **txport** | **uidb** {**entry instance** | **map instance** | **table instance**}} [**location** *node-id*]

Syntax Description

eio *link_id*	Displays EIO commands for PLASPA information for a specific link ID. Range is from 0 to 4294967295.
all	Displays PLA information for all links.
error	Displays PLASPA error information.
ifhandle	Displays the name associated with the interface.
instance	Displays instance information of the PLASPA ASIC. Range is from 0 to1.
num-inst	Displays number of instances of the PLASPA ASIC.
queues	Displays PLASPA queue information.
sbp	Displays egress queue back-pressure table information.
txport	Displays egressq port information. Range of shared port adapters (SPA) instance is from 0 to 5.
uidb	Displays PLASPA uidb hash table information.
location *node-id*	Identifies the location of the node whose PLIM ASIC information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.8.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read

Task ID	Operations
drivers	read
cisco-support	read

Examples

The following example shows how to display PLIM ASIC PLASPA information for an interface on the router:

```
RP/0/RP0/CPU0:router# show controllers plim asic plaspa eio links all location 0/3/CPU0


Wed Sep 10 04:52:27.452 UTC

             Node: 0/3/CPU0:
----------------------------------------

EIO links:
--------------------------------------------------------------------
ASIC Id     Peer Id     Type Link-Id Attempts Accept Failed State
--------------------------------------------------------------------
PLASPA_1    PSE_0       TX   2       0        0      0      EIO_LINK_TRAINED
PLASPA_1    EGRESSQ_0   RX   61      1        1      0      EIO_LINK_TRAINED
```

show controllers plim asic statistics

To display physical layer interface module (PLIM) ASIC statistics for a specific node or interface, use the **show controllers plim asic statistics** command in EXEC mode.

show controllers plim asic statistics {**interface** *type interface-path-id* | **summary**} [**location** *node-id*]

Syntax Description

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.
summary	Displays a summarized information for PLIM ASICs on a specified node, or for all interfaces on the router.
location *node-id*	Identifies the location of the node whose PLIM ASIC information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
root-system	read

Examples

The following example shows how to display PLIM ASIC statistics information for a POS interface:

```
RP/0/RP0/CPU0:router# show controllers plim asic statistics interface POS 0/2/0/0

                                Node: 0/2/CPU0
------------------------------------------------------------------------------
POS0/2/0/0 Tx Statistics
```

```
------------------------------------------------------------------------------
TotalOctets       : 78904040            TotalPkts         : 1622308
Unicastpkts       : 1622308             MulticastPkts     : 0
BroadcastPkts     : 0                   <64Octets         : 1610433
64Octets          : 0                   65to127Octets     : 11875
128to255Octets    : 0                   256to511Octets    : 0
512to1023Octets   : 0                   1024to1518Octets  : 0
1519to1548Octets  : 0                   1549to9216Octets  : 0
>9216Octet        : 0                   BadCRCPkts        : 0
802.1QPkts        : 0                   Underrun          : 0
Runt              : 0                   Giant             : 0
PausePkts         : 0                   Jabbers           : 0
DeferralAbort     : 0                   LateCollision     : 0
CollisionAbort    : 0                   OneCollision      : 0
MultiCollision    : 0                   TotalCollisions   : 0
TotalDefer        : 0                   LateCollisionAbort: 0
LengthAbort       : 0                   TxBP count        : 0

POS0/2/0/0 Rx Statistics
------------------------------------------------------------------------------
 --More-- failed to get stats
TotalOctets       : 91010808
TotalPkts         : 1815571             UnicastPkts       : 1815571
MulticastPkts     : 0                   BroadcastPkts     : 0
64Octets          : 63846               65to127Octets     : 11844
128to255Octets    : 7                   256to511Octets    : 0
512to1023Octets   : 0                   1024to1518Octets  : 0
1519to1548Octets  : 0                   1549to9216Octets  : 0
>9216Octets       : 0                   BadCRCPkts        : 0
BadCodedPkts      : 0                   Runt              : 0
ShortPkts         : 1739874             802.1QPkts        : 0
Drop              : 0                   PausePkts         : 0
ControlPkts       : 0                   Jabbers           : 0
BadPreamble       : 0

POS0/2/0/0 Drop
------------------------------------------------------------------------------
RxFiFO Drop       : 0                   PAR Tail Drop     : 0
TxFIFO Drop       : 0
```

This table describes the significant fields shown in the display.

Table 13: show controllers plim asic statistics Field Descriptions

Field	Description
TotalOctets	Number of octets received or transmitted on the interface.
TotalPkts	Number of total packets received or transmitted on the interface.
UnicastPkts	Number of unicast packets received or transmitted on the interface.
MulticastPkts	Number of multicast packets received or transmitted on the interface. Received packets were directed to the multicast address.
BroadcastPkts	Number of good broadcast packets received or transmitted. Received packets were directed to the broadcast address.
64Octets	Number of packets (including bad packets) received or transmitted that were less than 64 octets in length (excluding framing bits but including FCS octets).

Field	Description
64Octets	Number of packets (including bad packets) received or transmitted that were 64 octets in length (excluding framing bits but including FCS octets).
65to127Octets	Number of packets (including bad packets) received or transmitted that were between 65 and 127 octets in length inclusive (excluding framing bits but including FCS octets).
128to255Octets	Number of packets (including bad packets) received or transmitted that were between 128 and 255 octets in length inclusive (excluding framing bits but including FCS octets).
256to511Octets	Number of packets (including bad packets) received or transmitted that were between 256 and 511 octets in length inclusive (excluding framing bits but including FCS octets).
512to1023Octets	Number of packets (including bad packets) received or transmitted that were between 512 and 1023 octets in length inclusive (excluding framing bits but including FCS octets).
1024to1518Octets	Number of packets (including bad packets) received or transmitted that were between 1024 and 1518 octets in length inclusive (excluding framing bits but including FCS octets).
1519to1548Octets	Number of packets (including bad packets) received or transmitted that were between 1519 and 1548 octets in length inclusive (excluding framing bits but including FCS octets).
1549to9216Octets	Number of packets (including bad packets) received or transmitted that were between 1549 and 9216 octets in length inclusive (excluding framing bits but including FCS octets).
>9216Octet	Number of packets (including bad packets) received or transmitted that were greater than 9216 octets in length (excluding framing bits but including FCS octets).
BadCRCPkts	Number of packets received or transmitted that had a length (excluding framing bits, but including FCS octets) of between 64 and 1518 octets, inclusive, but had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS error) or a bad FCS with a non integral number of octets (alignment error).
802.1QPkts	Number of 802.1QPkts received or transmitted on the interface.
Underrun	Number of packets that were not retrieved quickly enough from shared memory to be transmitted or received.
Runt	Number of packets received or transmitted that were less than 64 octets long (excluding framing bits, but including FCS octets) and were otherwise well formed.
Giant	Number of packets received or transmitted that were longer than 1518 octets (excluding framing bits, but including FCS octets) and were otherwise well formed.
PausePkts	Number of pause packets transmitted/received on the interface. Pause packets that tell remote devices to delay sending more packets for a specified period of time.

Field	Description
Jabbers	Number of packets received or transmitted that were longer than 1518 octets (excluding framing bits but including FCS octets) and had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS error) or a bad FCS with a non-integral number of octets (assigned error).
DeferralAbort	Number of deferral terminations that occurred on this segment.
LateCollision	Number of late collisions on this segment
CollisionAbort	Number of collisions that were terminated.
OneCollision	Number of single collisions that occurred on this segment.
MultiCollision	Number multiple collisions that occurred on this segment.
TotalCollisions	Number of collisions on this segment.
TotalDefer	Number of deferrals on this segment.
LateCollisionAbort	Number of late collision terminations that occurred on this segment.
LengthAbort	Number of length terminations that occurred on this segment.
TxBP count	Number of transmit BP on this segment.
Rx Statistics	Indicates the statistics that follow were received by the interface.
TX statistics	Indicates the statistics that follow were transmitted by the interface.
RxFiFO Drop	Displays the receive FIFO drop information.
PAR Tail Drop	Displays PAR tail drop information.
TxFIFO Drop	Displays transmitted FIFO drop information.

show controllers plim asic summary

To display summarized physical layer interface module (PLIM) ASIC information for a specific node or interface, use the **show controllers plim asic** command in EXEC mode.

show controllers plim asic summary [**location** *node-id*]

Syntax Description

location *node-id*	Identifies the location of the node whose PLIM ASIC information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Note Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
root-system	read

Examples

The following example shows how to display summarized PLIM ASIC information for all locations:

```
RP/0/RP0/CPU0:router# show controllers plim asic summary

Node: 0/1/CPU0
--------------------------------------
Instance# 0    Summary info:
--------------------------------------
Name       : PLASPA    Version  : 2

Port 0
Jacket slot: 1         SPA type : SPA_NAME_UNKNOWN
Port 1
Jacket slot: 3         SPA type : SPA_NAME_UNKNOWN
Port 2
Jacket slot: 0         SPA type : 4xOC3 POS SPA

Instance# 1    Summary info:
--------------------------------------
Name       : PLASPA    Version  : 2

Port 0
```

```
Jacket slot: 2         SPA type : SPA_NAME_UNKNOWN
Port 1
Jacket slot: 4         SPA type : 4xOC48 POS/RPR HHSPA
Port 2
Jacket slot: 5         SPA type : 8xGE SPA


IFName     : POS0/1/0/0
Inst#      : 0         Port     : 2
RxLPORT    : 0x80      TxLPORT  : 0x48
Uidb       : 0x2       Key      : 0x80
Hkey       : 209       Hkey idx : 0

IFName     : POS0/1/0/1
Inst#      : 0         Port     : 2
RxLPORT    : 0x81      TxLPORT  : 0x49
Uidb       : 0x4       Key      : 0x81
Hkey       : 28        Hkey idx : 0

IFName     : POS0/1/0/2
Inst#      : 0         Port     : 2
RxLPORT    : 0x82      TxLPORT  : 0x4a
Uidb       : 0x6       Key      : 0x82
Hkey       : 183       Hkey idx : 0
```

This table describes the significant fields shown in the display.

Table 14: show controllers plim asic summary Field Descriptions

Field	Description
Node	Node whose information is displayed. Information is displayed for each node's SPA and its interfaces.
Instance	PLIM ASIC identifier. This is the PLIM ASIC associated with the specified location.
Summary info (for SPA)	Displays the following info for all SPAs installed in the router: • Name—Identifies the SPA.whose information is displayed. • Version—Version identifier for the PLIM ASIC. • Jacket slot—Identifies the slot containing the jacket card for the specified SPA. • SPA type —Describes the SPA whose information is displayed. • Port—Port associated with the PLIM ASIC. • Inst#—SPA ASIC instance Identifier.
Summary info (for interfaces)	Displays the following info for all interfaces associated with the specified SPA: • Intf name—Identifies the SPA.whose information is displayed. • Inst#—ASIC associated with this interface. • Port—Port associated with the PLIM ASIC. • RxLPORT—Receive port, in hexadecimal format. • TxLPORT—Transmit port, in hexadecimal format. • Uidb—UIDB[12] assigned by the software, in hexadecimal format. • Key—AISC key, in hexadecimal format. • Hkey—ASIC registry key. • Hkey idx—ASIC registry key index.

[12] Universal interface descriptor block

show controllers pse eio links

To display packet switching engine (PSE) information for Elastic I/O (EIO) links, use the **show controllers pse eio links** command in EXEC mode.

show controllers pse eio links {*link-number* | **all**} [**location** *node-id*]

Syntax Description

links *link-number*	Displays PSE information for the specified link. Replace the *link_number* argument with a link number in the range from 0 to 4294967295. **Note** Use the **show controllers pse eio links all location** *node-id* command to display all available EIO link IDs on a node.
links all	Displays PSE information for all links on the specified node.
location *node-id*	(Optional) Displays all EIO links on a specific node only. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.6.0	The *link-number* argument range of 1 through 4294967295 has changed to 0 to 4294967295.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
drivers	read
interface	read

Examples

The following is sample output from the **show controllers pse eio links** command for a specific EIO link:

```
RP/0/RP0/CPU0:router# show controllers pse eio links 1

             Node: 0/0/CPU0:
----------------------------------------

EIO link: 1
```

```
-------------------------------------------------------------------
ASIC Id     Peer Id     Type Link-Id Attempts Accept Failed State
-------------------------------------------------------------------
PSE_0       PLASPA_0    RX   1       1         1        0      EIO_LINK_TRAINED
-------------------------------------------------------------------
Macro-id Windage Diag status
-------------------------------------------------------------------
0        4       0x0000
1        4       0x0000
2        4       0x0000
3        4       0x0000
-------------------------------------------------------------------
```

The following is sample output from the show controllers pse eio links command for all EIO links:

```
RP/0/RP0/CPU0:router#show controllers pse eio links all

             Node: 0/0/CPU0:
----------------------------------------

EIO links:
-------------------------------------------------------------------
ASIC Id     Peer Id     Type Link-Id Attempts Accept Failed State
-------------------------------------------------------------------
PSE_1       FABRICQ_0   RX   50      1        1      0      EIO_LINK_TRAINED
PSE_1       FABRICQ_1   RX   51      1        1      0      EIO_LINK_TRAINED
PSE_1       EGRESSQ_0   RX   64      1        1      0      EIO_LINK_TRAINED
PSE_1       EGRESSQ_0   TX   11      0        0      0      EIO_LINK_TRAINED
PSE_0       PLASPA_0    RX   1       1        1      0      EIO_LINK_TRAINED
PSE_0       PLASPA_1    RX   2       1        1      0      EIO_LINK_TRAINED
PSE_0       INGRESSQ_0  RX   24      1        1      0      EIO_LINK_TRAINED
PSE_0       INGRESSQ_0  TX   10      0        0      0      EIO_LINK_TRAINED
```

This table describes the significant fields shown in the display.

Table 15: show controllers pse eio links Field Descriptions

Field	Description
Node	Node whose PSE information is displayed.
EIO link	EIO link whose information is displayed.
ASIC Id	ASIC associated with this EIO link.
Peer Id	ASIC of the remote peer of this EIO link.
Type	Indicates whether this link is receiving or transmitting (re)training requests.
Link-Id	Unique ID assigned by the system to the EIO link.
Attempts	Number of consecutive EIO training requests for the link made by the PSE driver to the peer. **Note** Typically, the PSE driver at the receive end of a link initiates the training of that EIO link. Therefore, the EIO links whose PSE driver is at the transmit end show a value of *0* for the *Attempts* field, even though those links are in the EIO_LINK_TRAINED state.

Field	Description
Accept	Number of PSE driver requests for training that were accepted by the peer. When a PSE driver request is accepted, training is started. **Note** Typically, the PSE driver at the receive end of a link initiates the training of that EIO link. Therefore, the EIO links whose PSE driver is at the transmit end show a value of *0* for the *Accept* field, even though those links are in the EIO_LINK_TRAINED state.
Failed	Number of times the training for this link failed for any reason.
State	Current state of the EIO link.
Macro-id Windage Diag status	Windage value used for the training, and the final status of the training for that windage. **Note** The Macro-id Windage Diag status field is displayed only when you ask EIO info for specific link.

Related Commands

Command	Description
show controllers pse mp, on page 111	displays packet switching engine information for the maintenance processor on a specific controller or node.
show controllers pse summary, on page 124	Displays a summary of packet switching engine information for a specific controller or node.

show controllers pse ipc

To display packet switching engine (PSE) device information for interprocess communication (IPC) connections, or for a specific IPC controller, use the **show controllers pse ipc** command in EXEC mode.

show controllers pse ipc {**client-connection** {*connID* | **all**} | **cpuctrlif**} [{**egress** | **ingress**}] [**location** *node-id*]

Syntax Description

client-connection *connID*	Displays IPC information about the specified client connection. Replace the *connID* argument with the client connection ID. Range is from 0 to 255.
client-connection all	Displays IPC information about the all client connection.
cpuctrlif	Displays IPC information about the specified PSE IPC CPU controller instance.
egress	(Optional) Displays IPC information for the egress PSE device. **Note** Follow the **egress**argument with the location *node-id* keyword and argument to restrict the command to a specific node containing the specified device instance.
ingress	(Optional) Displays IPC information for the ingress PSE device. **Note** Follow the **ingress** argument with the location *node-id* keyword and argument to restrict the command to a specific node containing the specified device instance.
location *node-id*	(Optional) Identifies the location of the node whose PSE device information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router. **Note** Include the **egress** or **ingress**argument before thelocation *node-id* keyword and argument to restrict the command to a specific device instance on the specified node.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	The **instance** { **0** | **1** } keywords were replaced by the **egress** and **ingress** keywords.
Release 3.6.0	The *connID* argument range of 1 to 255 has changed to 0 to 255.

Usage Guidelines

To display client connection IPC information for a specific device instance, include the **egress** or **ingress** argument with the command, as shown in the following examples:

```
show controllers pse ipc client-connection connId egress
```

```
show controllers pse ipc client-connection connId ingress
```

To display client connection IPC information for a specific node location, include the **location** *node-id* keyword and argument with the command, as shown in the following example:

```
show controllers pse ipc client-connection connId location node-id
```

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following command shows how to display PSE information for all IPC client-connections:

```
RP/0/RP0/CPU0:router# show controllers pse ipc client-connection all

Mon Nov  2 15:20:06.228 UTC

             Node: 0/0/CPU0:
----------------------------------------

PSE IPC Conn Info:
-------------------------------------------------
ConnId:         0 (Backdoor)
ConnOptions:    0x1
Clinfos:        0 (Notif) 0 (Reply)
ConnStateFlags: 0
ConnNotifFlags: 0
SendReqs:       0 (iHP) 5976 (iNP) 0 (eHP) 11404 (eNP)
SendFiltered:   0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
SendFailed:     0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
RecvReqs:       0
RecvReqTouts:   0
RecvDatTouts:   2
RecvQState:     0/16


PSE IPC Conn Info:
-------------------------------------------------
ConnId:         1 (uidb-mipc)
ConnOptions:    0
Clinfos:        0x50074180 (Notif) 0 (Reply)
ConnStateFlags: 0
ConnNotifFlags: 0xc0c2
SendReqs:       0 (iHP) 189 (iNP) 0 (eHP) 190 (eNP)
SendFiltered:   0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
SendFailed:     0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
RecvReqs:       0
RecvReqTouts:   0
RecvDatTouts:   0
RecvQState:     0/16
```

```
PSE IPC Conn Info:
-------------------------------------------------
ConnId:         2 (stats_svr)
ConnOptions:    0
Clinfos:        0x50074f94 (Notif) 0 (Reply)
ConnStateFlags: 0
ConnNotifFlags: 0x1003002
SendReqs:       0 (iHP) 72 (iNP) 0 (eHP) 327 (eNP)
SendFiltered:   0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
SendFailed:     0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
RecvReqs:       0
RecvReqTouts:   0
RecvDatTouts:   0
RecvQState:     0/16


PSE IPC Conn Info:
-------------------------------------------------
ConnId:         3 (TCAM Mgr)
ConnOptions:    0
Clinfos:        0x50073d9c (Notif) 0 (Reply)
ConnStateFlags: 0
ConnNotifFlags: 0x30e06
SendReqs:       0 (iHP) 291 (iNP) 0 (eHP) 163 (eNP)
SendFiltered:   0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
SendFailed:     0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
RecvReqs:       21
RecvReqTouts:   0
RecvDatTouts:   0
RecvQState:     0/16


PSE IPC Conn Info:
-------------------------------------------------
ConnId:         4 (hfr_pm)
ConnOptions:    0
Clinfos:        0x500778e0 (Notif) 0 (Reply)
ConnStateFlags: 0
ConnNotifFlags: 0xc0006
SendReqs:       0 (iHP) 21 (iNP) 0 (eHP) 5 (eNP)
SendFiltered:   0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
SendFailed:     0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
RecvReqs:       0
RecvReqTouts:   0
RecvDatTouts:   0
RecvQState:     0/16


PSE IPC Conn Info:
-------------------------------------------------
ConnId:         5 (PLU-TLU-Mgr-118869)
ConnOptions:    0
Clinfos:        0x500785b0 (Notif) 0 (Reply)
ConnStateFlags: 0
ConnNotifFlags: 0x1fe
SendReqs:       0 (iHP) 59 (iNP) 0 (eHP) 779 (eNP)
SendFiltered:   0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
SendFailed:     0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
RecvReqs:       126
RecvReqTouts:   0
RecvDatTouts:   0
RecvQState:     0/16
```

```
PSE IPC Conn Info:
-------------------------------------------------
ConnId:         6 (PLU-TLU-Mgr-118878)
ConnOptions:    0
Clinfos:        0x50079218 (Notif) 0 (Reply)
ConnStateFlags: 0
ConnNotifFlags: 0x1fe
SendReqs:       0 (iHP) 72038 (iNP) 0 (eHP) 65852 (eNP)
SendFiltered:   0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
SendFailed:     0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
RecvReqs:       0
RecvReqTouts:   0
RecvDatTouts:   0
RecvQState:     0/16


PSE IPC Conn Info:
-------------------------------------------------
ConnId:         7 (PLU-TLU-Mgr-118875)
ConnOptions:    0
Clinfos:        0x50079ea4 (Notif) 0 (Reply)
ConnStateFlags: 0
ConnNotifFlags: 0x1fe
SendReqs:       0 (iHP) 3 (iNP) 0 (eHP) 0 (eNP)
SendFiltered:   0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
SendFailed:     0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
RecvReqs:       0
RecvReqTouts:   0
RecvDatTouts:   0
RecvQState:     0/16


PSE IPC Conn Info:
-------------------------------------------------
ConnId:         8 (PLU-TLU-Mgr-118876)
ConnOptions:    0
Clinfos:        0x5007ab30 (Notif) 0 (Reply)
ConnStateFlags: 0
ConnNotifFlags: 0x1fe
SendReqs:       0 (iHP) 21 (iNP) 0 (eHP) 18 (eNP)
SendFiltered:   0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
SendFailed:     0 (iHP) 0 (iNP) 0 (eHP) 0 (eNP)
RecvReqs:       0
RecvReqTouts:   0
RecvDatTouts:   0
RecvQState:     0/16
```

This table describes the significant fields shown in the display.

Table 16: show controllers pse ipc client-connection Field Descriptions

Field	Description
ConnId	Connection identifier.
ConnOptions	Connection bind time options requested by the client creating this connection.
Clinfos	Client information (clinfo) index number. The memory database is organized as a table of clinfos, which are indexed by the region id and maintained in shared memory.
ConnStateFlags	Bitmask containing the current state of the connection.

Field	Description
ConnNotifFlags	Bitmask of the types of notifications that the client creating this connection has registered for.
SendReqs	Number of PSE IPC send requests that client has made on this connection.
SendFiltered	Number of send requests that were filtered out (dropped) from the PSE driver due to any internal filter policy.
SendFailed	Number of send requests that could not be handled by the PSE driver due to error conditions such as hardware I/O failures, lack of buffering space, and so forth.
RecvReqs	Number of data receive requests that the client has made on this connection.
RecvReqTouts	Number of data receive requests from the client that timed out because no data was received from PSE device.
RecvDatTouts	Number of times data was received from PSE device, but timed out because there was no receive request from client to pick up that data from the connection.
RecvQState	Current state of the internal data receive FIFO queue. The internal data receive FIFO queue buffers data arriving from the PSE device for the connection, until the client sends a receive request for that data. The RecvQState is expressed in the *X/Y* notation, where *X* is current depth of the queue, and *Y* is the maximum depth allowed for the queue.

The following command shows how to display PSE information for a specific IPC controller connection:

```
RP/0/RP0/CPU0:router# show controllers pse ipc cpuctrlif

             Node: 0/0/CPU0:
----------------------------------------

Pse IPC cpuctrl if Information
------------------------------------------------
Cpuctrl interface is up
Normal Priority Tx I/F:
        BufBase 0x112190  BufSize 32768 PadBufp 0xec0af118 Seqnum 854
        RORegNum 10 RORegVal 0x49f0 WORegNum 2 WORegVal 0x49f0
        CDMACh 0 CDMAQ 3
        HoldQ 0/512 CDMAPendQ 0/128 UnackQ 0/256
        NumEnqs 854 NumRejs 0 NumDmaReqFails 0
        NumDmaPendFails 0 NumMbxSpcFails 0 NumHldqOvflows 0
Normal Priority Rx I/F:
        NumPDMAEntries 64 NumEnqs 15 NumRejs 0

Pse IPC cpuctrl if Information
------------------------------------------------
Cpuctrl interface is up
Normal Priority Tx I/F:
        BufBase 0x112190  BufSize 32768 PadBufp 0xec0af14c Seqnum 424
        RORegNum 10 RORegVal 0x1c60 WORegNum 2 WORegVal 0x1c60
        CDMACh 1 CDMAQ 3
        HoldQ 0/512 CDMAPendQ 0/128 UnackQ 0/256
        NumEnqs 424 NumRejs 0 NumDmaReqFails 0
        NumDmaPendFails 0 NumMbxSpcFails 0 NumHldqOvflows 0
Normal Priority Rx I/F:
```

```
        NumPDMAEntries 64 NumEnqs 2 NumRejs 0

             Node: 0/3/CPU0:
----------------------------------------

Pse IPC cpuctrl if Information
-----------------------------------------------
Cpuctrl interface is up
Normal Priority Tx I/F:
        BufBase 0x112190  BufSize 32768 PadBufp 0xec0af118 Seqnum 833
        RORegNum 10 RORegVal 0xdb8 WORegNum 2 WORegVal 0xdb8
        CDMACh 0 CDMAQ 3
        HoldQ 0/512 CDMAPendQ 0/128 UnackQ 0/256
        NumEnqs 833 NumRejs 0 NumDmaReqFails 0
        NumDmaPendFails 0 NumMbxSpcFails 0 NumHldqOvflows 0
Normal Priority Rx I/F:
        NumPDMAEntries 64 NumEnqs 9 NumRejs 0

Pse IPC cpuctrl if Information
-----------------------------------------------
Cpuctrl interface is up
Normal Priority Tx I/F:
        BufBase 0x112190  BufSize 32768 PadBufp 0xec0af14c Seqnum 409
        RORegNum 10 RORegVal 0x5f70 WORegNum 2 WORegVal 0x5f70
        CDMACh 1 CDMAQ 3
        HoldQ 0/512 CDMAPendQ 0/128 UnackQ 0/256
        NumEnqs 409 NumRejs 0 NumDmaReqFails 0
        NumDmaPendFails 0 NumMbxSpcFails 0 NumHldqOvflows 0
Normal Priority Rx I/F:
        NumPDMAEntries 64 NumEnqs 2 NumRejs 0
```

This table describes the significant fields shown in the display.

Table 17: show controllers pse ipc cpuctrlif Field Descriptions

Field	Description
Cpuctrl interface is up Cpuctrl interface is down	Whether the connection is currently up or down.

Field	Description
Normal Priority Tx I/F	Displays the following transmit buffer information for the connection: • BufBase • BufSize • PadBufp • Seqnum • RORegNum • RORegVal • WORegNum • WORegVal • CDMACh • CDMAQ • HoldQ • CDMAPendQ • UnackQ • NumEnqs • NumRejs • NumDmaReqFails • NumDmaPendFails • NumMbxSpcFails • NumHldqOvflows
Normal Priority Rx I/F:	Displays the following receive buffer information for the connection: • NumPDMAEntries • NumEnqs • NumRejs

Related Commands

Command	Description
show controllers pse eio links, on page 101	Displays packet switching engine information for Elastic I/O links.
show controllers pse mp, on page 111	displays packet switching engine information for the maintenance processor on a specific controller or node.
show controllers pse summary, on page 124	Displays a summary of packet switching engine information for a specific controller or node.

show controllers pse mp

To display packet switching engine (PSE) information for the maintenance processor on a specific controller or node, use the **show controllers pse mp** command in EXEC mode.

show controllers pse mp [{**egress** | **ingress**}] [**location** *node-id*]

Syntax Description

egress	(Optional) Displays information for the egress PSE device only. **Note** Follow the **egress** argument with the **location** *node-id* keyword and argument to restrict the command to a specific node.
ingress	(Optional) Displays information for the ingress PSE device only. **Note** Follow the **ingress** argument with the **location** *node-id* keyword and argument to restrict the command to a specific node.
location *node-id*	(Optional) Identifies the location of the node whose PSE device information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router. **Note** Include the **instance** {**0** | **1**} argument before the**location** *node-id* keyword and argument to restrict the command to a specific device instance on the specified node.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.

Usage Guidelines

The **instance** and **location** optional keywords are not mutually exclusive. The **instance** keyword directs the command to a specific PSE device, and the **location** keyword directs the command to that device or devices on the specified modular services card. You can specify the **instance** and **location** options together in the same command. If you do not specify the **instance** {**0** | **1**} and **location** *node-id* keywords and argument, the **show controllers pse mp** command takes effect on both device instances on all modular services cards.

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following command shows how to display PSE maintenance processor information for a specific controller instance:

```
RP/0/RP0/CPU0:router# show controllers pse mp instance 1

             Node: 0/0/CPU0:
----------------------------------------

PSE 1, MP Info:
-----------------------------------------------
MIPC Subtask Context Information
        High Priority MIPC Mbox Info
                NumProc 0 NumUnsupp 0
                NxtSeqNum 0 NumOOSeq 0
        Norm Priority MIPC Mbox Info
                NumProc 429 NumUnsupp 0
                NxtSeqNum 425 NumOOSeq 0
        NumMsgsGtred 93 NumDMAErrDrops 5 NumGtrDisDrops 2
MSTAT Subtask Context Information
        NumSegs 9 CntrsPerBrst 93 MinCycleMsecs 4000
        ---------  Seg[0] config -----------------
        SegId 0 SegBeg 0 SegEnd 3071
        ExpBeg 0 ExpEnd 0 ExpRate 0 CycBefExp 0
        ---------  Seg[1] config -----------------
        SegId 1 SegBeg 3072 SegEnd 74239
        ExpBeg 3072 ExpEnd 3072 ExpRate 0 CycBefExp 0
        ---------  Seg[2] config -----------------
        SegId 2 SegBeg 74240 SegEnd 107007
        ExpBeg 74240 ExpEnd 74240 ExpRate 0 CycBefExp 0
        ---------  Seg[3] config -----------------
        SegId 3 SegBeg 107008 SegEnd 262143
        ExpBeg 107008 ExpEnd 107008 ExpRate 0 CycBefExp 0
        ---------  Seg[4] config -----------------
        SegId 4 SegBeg 262144 SegEnd 265215
        ExpBeg 262144 ExpEnd 262144 ExpRate 0 CycBefExp 0
        ---------  Seg[5] config -----------------
        SegId 5 SegBeg 265216 SegEnd 336383
        ExpBeg 265216 ExpEnd 265216 ExpRate 0 CycBefExp 0
        ---------  Seg[6] config -----------------
        SegId 6 SegBeg 336384 SegEnd 442367
        ExpBeg 336384 ExpEnd 336385 ExpRate 1 CycBefExp 1
        ---------  Seg[7] config -----------------
        SegId 7 SegBeg 442368 SegEnd 458751
        ExpBeg 442368 ExpEnd 442368 ExpRate 0 CycBefExp 0
        ---------  Seg[8] config -----------------
        SegId 8 SegBeg 458752 SegEnd 491519
        ExpBeg 458752 ExpEnd 458752 ExpRate 0 CycBefExp 0
        CurrTokens 9 MaxTokens 44 MinTokens 11
        CurrSeg 5 CurrCntr 289954 CntrsExp 66532
        PktsExp 33266 DMAErrDrops 0 REDDrops 0
        REDDelays 0 GtrDisDelays 0
        CfgUpdsApp 1 SegUpdsApp 10 UpdsRej 0

             Node: 0/3/CPU0:
----------------------------------------

PSE 1, MP Info:
-----------------------------------------------
MIPC Subtask Context Information
        High Priority MIPC Mbox Info
                NumProc 0 NumUnsupp 0
                NxtSeqNum 0 NumOOSeq 0
```

```
        Norm Priority MIPC Mbox Info
                NumProc 416 NumUnsupp 0
                NxtSeqNum 412 NumOOSeq 0
        NumMsgsGtred 71 NumDMAErrDrops 4 NumGtrDisDrops 3
MSTAT Subtask Context Information
        NumSegs 9 CntrsPerBrst 93 MinCycleMsecs 4000
        ---------  Seg[0] config -----------------
        SegId 0 SegBeg 0 SegEnd 3071
        ExpBeg 0 ExpEnd 0 ExpRate 0 CycBefExp 0
        ---------  Seg[1] config -----------------
        SegId 1 SegBeg 3072 SegEnd 74239
        ExpBeg 3072 ExpEnd 3072 ExpRate 0 CycBefExp 0
        ---------  Seg[2] config -----------------
        SegId 2 SegBeg 74240 SegEnd 107007
        ExpBeg 74240 ExpEnd 74240 ExpRate 0 CycBefExp 0
        ---------  Seg[3] config -----------------
        SegId 3 SegBeg 107008 SegEnd 262143
        ExpBeg 107008 ExpEnd 107008 ExpRate 0 CycBefExp 0
        ---------  Seg[4] config -----------------
        SegId 4 SegBeg 262144 SegEnd 265215
        ExpBeg 262144 ExpEnd 262144 ExpRate 0 CycBefExp 0
        ---------  Seg[5] config -----------------
        SegId 5 SegBeg 265216 SegEnd 336383
        ExpBeg 265216 ExpEnd 265216 ExpRate 0 CycBefExp 0
        ---------  Seg[6] config -----------------
        SegId 6 SegBeg 336384 SegEnd 442367
        ExpBeg 336384 ExpEnd 336415 ExpRate 1 CycBefExp 0
        ---------  Seg[7] config -----------------
        SegId 7 SegBeg 442368 SegEnd 458751
        ExpBeg 442368 ExpEnd 442368 ExpRate 0 CycBefExp 0
        ---------  Seg[8] config -----------------
        SegId 8 SegBeg 458752 SegEnd 491519
        ExpBeg 458752 ExpEnd 458752 ExpRate 0 CycBefExp 0
        CurrTokens 5 MaxTokens 44 MinTokens 11
        CurrSeg 6 CurrCntr 398633 CntrsExp 1064256
        PktsExp 33258 DMAErrDrops 0 REDDrops 0
        REDDelays 0 GtrDisDelays 0
        CfgUpdsApp 1 SegUpdsApp 10 UpdsRej 0
```

This table describes the significant fields shown in the display.

Table 18: show controllers pse mp Field Descriptions

Field	Description
MIPC Subtask Context Information	MIPC subtask information from the following mailboxes (queues): • High Priority MIPC Mbox • Norm Priority MIPC Mbox **Note** The MIPC MBox is a chunk of the MP DMEM that receives MIPC messages. The Norm Priority mailbox has buffer of 32 KB, while the High Priority Mbox has an 8 KB buffer.

Field	Description
High Priority MIPC Mbox Info	Mailbox (or queue) that receives high-priority MIPC messages. The following information is displayed for the High Priority MIPC Mbox: • NumProc—Number of MIPC messages that have been processed. • NumUnsupp—Number of unsupported MIPC messages. • NxtSeqNum—Sequence number of the message with respect to other messages sent on the same MIPC channel. • NumOOSeq—Number of MIPC messages that were received out of sequence.
Norm Priority MIPC Mbox info	Mailbox (or queue) that receives normal-priority MIPC messages. The following information is displayed for the High Priority MIPC Mbox: • NumProc—Number of MIPC messages that have been processed. • NumUnsupp—Number of unsupported MIPC messages. • NxtSeqNum—Identifies the sequence number of the message with respect to other messages sent on the same MIPC channel. • NumOOSeq—Number of MIPC messages that were received out of sequence.
NumMsgsGtred	Number of MIPC gather messages that have been processed.
NumDMAErrDrops	Number of MIPC gather messages that were dropped due to DMA errors.
NumGtrDisDrops	Number of MIPC gather messages that were dropped due to distribution errors.
MSTAT Subtask Context Information	MSTAT subtask information from the following queues: • High Priority MIPC Mbox • Norm Priority MIPC Mbox The MIPC MBox is a chunk of the MP DMEM that receives MIPC messages. The Norm Priority mailbox has buffer of 32 KB, while the High Priority Mbox has an 8 KB buffer.

Related Commands

Command	Description
show controllers pse eio links, on page 101	Displays packet switching engine information for Elastic I/O links.
show controllers pse summary, on page 124	Displays a summary of packet switching engine information for a specific controller or node.

show controllers pse qfp classification

To display ACL class group ID, which is used to view Ternary Content Addressable Memory(TCAM) entry, use the **show controllers pse qfp classification** command in EXEC mode.

show controllers pse qfp classification class-group-manager class-group all location *node-ID*

Syntax Description

class-group-manager	Displays the class group manager.
class-group	Displays the class group.
all	Displays the ACL class group ID for all class groups.
location *node-ID*	Displays the ACL class group ID for a specific location. The node-ID argument is entered in the rack/slot/module notation.

Command Default

None

Command Modes

EXEC mode

Command History

Release	Modification
Release 4.3.1	This command was introduced

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operation
sonet sdh or dwdm	read
interface	read
root-system	read

Example

This command shows how to display ACL class group ID, which is used to view TCAM entry:

```
RP/0/RP0/CPU0:router# show controllers pse qfp classification class-group-manager class-group
 all location 0/2/cpu0

QFP classification class client all group
  class-group [acl:1]
  class-group [acl:2]
  class-group [acl:3]
  class-group [acl:4]
  class-group [lpts:2]
  class-group [lpts:3]
```

```
class-group [lpts:4]
class-group [li:1]
class-group [li:2]
```

Related Commands

Command	Description
show controllers pse qfp classification feature-manager, on page 117	Displays TCAM entries.

show controllers pse qfp classification feature-manager

To display Ternary Content Addressable Memory(TCAM) entries, use the **show controllers pse qfp classification feature-manager** command in EXEC mode.

show controllers pse qfp classification feature-manager class-group ce-data client *class-group-ID interface-path-id* **location** *node-ID*

Syntax Description

class-group	Displays the class group.
ce-data	Displays the ce-data.
client	Specifies type of client. The available options are: • acl - ACL client • cce - CCE client, includes QoS, Firewall, and FPM • dbg - Conditional debug client • ess - ESS client • fw-pam - Firewall PAM(Port-Application-Mapping) client • ipsec - IPSec client • lpts - LPTS client • nat - NAT client • pbr - PBR client • tc - ESS TC client • v6comp - V6 Compression client • wccp - WCCP client
class-group-ID	Displays the class group ID for all class groups.

interface-path-id	Either a physical interface instance or a virtual interface instance as follows: • Physical interface instance. Naming notation is *rack/slot/module/port*; 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 (RSP0) and the module is CPU0. Example: interface MgmtEth0/RSP0/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 the ACL class group ID for a specific location. The node-id argument is entered in the rack/slot/module notation.

Command Default

None

Command Modes

EXEC mode

Command History

Release	Modification
Release 4.3.1	This command was introduced

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operation
sonet sdh or dwdm	read
interface	read
root-system	read

Example

This command shows how to display TCAM entries:

```
RP/0/RP0/CPU0:router# show controllers pse qfp classification feature-manager class-group
ce-data acl 2 interface pos0/2/1/2.1 detail location 0/2/cPU0
```

```
QFP TCAM information for qfp_num 0

CPP classification class group TCAM
class-group [acl-cg:2] (classes: 2, total number of vmrs: 2)
key name: 160_01  value size: 160 result size: 4
  region id: 1  vmr id: 2 number of vmrs: 2
  Value : : 33000001 06000000 00500050 29000001 00020001
   Mask : : ffffffff ff000000 ffffffff ffffffff ffff0009
Result : : 01000002 89c54800 00000000 00000000

  Value : : 00000000 00000000 00000000 00000000 00020001
   Mask : : 00000000 00000000 00000000 00000000 ffff0009
Result : : 02000002 89c54810 00000000 00000000


QFP TCAM information for qfp_num 1

CPP classification class group TCAM
class-group [acl-cg:2] (classes: 2, total number of vmrs: 2)
key name: 160_01  value size: 160 result size: 4
  region id: 1  vmr id: 1 number of vmrs: 2
  Value : : 33000001 06000000 00500050 29000001 00010001
   Mask : : ffffffff ff000000 ffffffff ffffffff ffff0009
Result : : 01000002 8986d800 00000000 00000000

  Value : : 00000000 00000000 00000000 00000000 00010001
   Mask : : 00000000 00000000 00000000 00000000 ffff0009
Result : : 02000002 8986d810 00000000 00000000
```

Related Commands

Command	Description
show controllers pse qfp classification, on page 115	Display ACL class group ID which is used to dump TCAM entries.

show controllers pse statistics

To display packet switching engine (PSE) statistics for a specific controller instance, or for a specific node, use the **show controllers pse statistics**command in EXEC mode.

show controllers pse statistics [**all**] [{**egress** | **ingress**}] [**location** *node-id*]

Syntax Description

all	(Optional) Displays all counters.
egress	(Optional) Displays statistics for the egress PSE device only. **Note** Follow the **egress** argument with the **location** *node-id* keyword and argument to restrict the command to a specific node.
ingress	(Optional) Displays statistics for the ingress PSE device only. **Note** Follow the ingress argument with the **location** *node-id* keyword and argument to restrict the command to a specific node.
location *node-id*	(Optional) Identifies the location of the node whose PSE device information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router. **Note** Include the **egress** or **ingress**keyword before the **location** *node-id* keyword and argument to restrict the command to a specific device instance on the specified node.

Command Default

If you do not specify the **egress** or **ingress** and **location** *node-id* keywords and argument, the **show controllers pse statistics** command displays statistical information for both device instances on all modular services cards.

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	The **instance** { **0**	**1**} keywords were replaced by the **egress** and **ingress** keywords.
Release 3.6.0	The **all** keyword was changed from required to optional.

Usage Guidelines

The optional **egress** or **ingress** and **location** keywords are not mutually exclusive. The **egress** and **ingress** keywords direct the command to specific PSE device, and the **location** keyword directs the command to that device or devices on the specified modular services card. You can specify the **egress** or **ingress** and **location** options together in the same command. If you do not specify the **egress** or **ingress** and **location** *node-id* keywords and argument, the **show controllers pse statistics** command displays statistical information for both device instances on all modular services cards.

GigabitEthernet shared port adapters(SPA) support all types of ethernet packets. A line card drops a packet only if the packet is of an invalid ethernet type. This eventually increases the L2 protocol unknown count in the output of the **show controller pse statistics ingress location node-ID** command.

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following command shows how to display PSE statistics:

```
RP/0/RP0/CPU0:router# show controllers pse statistics

Node 0/0/CPU0 Ingress PSE Stats
--------------------------------

Punt Stats                                Punted          Policed & Dropped
----------                                ------          -----------------
L2 low priority                           8383            0
L2 control                                133708          0
CDP                                       145926          0
ARP                                       8389            0
Bundle Control                            156877          0
IPv4 TTL expiration                       39179           0
IPv4 BFD async                            128348286       0
IPv4 BFD echo                             6543965         0
ACL log                                   39142667        0
IPv6 link local                           511927          0
IPv6 BFD async                            1380652214      0
EOAM CFM CCM pkts                         57390870        0
EOAM EFM pkts                             956527          0
SPA IPC punt                              2551214         0

Drop Stats                                Dropped
----------                                -------
IFIB policer drop                         225
Service lookup miss                       2137
IPv4 not enabled                          1
IPv4 interface down                       5
IPv4 MC not enabled                       60380
IPv6 not enabled                          1
EOAM EFM feature disable drop             176

Debug Stats                               Count
-----------                               -----
PPE idle counter                          84330433181953


Node 0/0/CPU0 Egress PSE Stats
--------------------------------

Punt Stats                                Punted          Policed & Dropped
----------                                ------          -----------------
IPv4 L2LI punt                            1               0
ACL log                                   1               0
IPv6 L2LI punt                            9               0

Drop Stats                                Dropped
----------                                -------
```

```
Pre-route no adjacency in PIT            8

Debug Stats                              Count
-----------                              -----
PPE idle counter                         84334688870964
Recirculate UIDB index                   31864
```

The following command shows how to display PSE statistics for a specific controller instance:

```
RP/0/RP0/CPU0:router# show controllers pse statistics instance 0


Node 0/0/CPU0 Ingress PSE Stats
--------------------------------

Punt Stats                               Punted          Policed & Dropped
----------                               ------          -----------------
L2 low priority                          8383            0
L2 control                               133708          0
CDP                                      145932          0
ARP                                      8389            0
Bundle Control                           156883          0
IPv4 TTL expiration                      39182           0
IPv4 BFD async                           128354734       0
IPv4 BFD echo                            6543965         0
ACL log                                  39144634        0
IPv6 link local                          511927          0
IPv6 BFD async                           1380721157      0
EOAM CFM CCM pkts                        57393762        0
EOAM EFM pkts                            956575          0
SPA IPC punt                             2551214         0

Drop Stats                               Dropped
----------                               -------
IFIB policer drop                        225
Service lookup miss                      2137
IPv4 not enabled                         1
IPv4 interface down                      5
IPv4 MC not enabled                      60385
IPv6 not enabled                         1
EOAM EFM feature disable drop            176

Debug Stats                              Count
-----------                              -----
PPE idle counter                         84334518624455
```

This table describes the significant fields shown in the display.

Table 19: show controllers pse statistics Field Descriptions

Field	Description
Node	Identifies the node whose PSE statistics are displayed. The node ID is expressed in the *rack/slot/module* notation.

Field	Description
PSE 0, Statistics Info	Displays all statistics maintained by the PSE.

Related Commands

Command	Description
show controllers pse eio links, on page 101	Displays packet switching engine information for Elastic I/O links.
show controllers pse ipc, on page 104	Displays packet switching engine device information for interprocess communication connections, or for a specific IPC controller.
show controllers pse mp, on page 111	displays packet switching engine information for the maintenance processor on a specific controller or node.
show controllers pse summary, on page 124	Displays a summary of packet switching engine information for a specific controller or node.

show controllers pse summary

To display a summary of packet switching engine (PSE) information for a specific controller or node, use the **show controllers pse summary** command in EXEC mode.

show controllers pse summary [{**egress** | **ingress**}] [**location** *node-id*]

Syntax Description

egress	(Optional) Displays information for the egress PSE device only. **Note** Follow the **egress** argument with the **location** *node-id* keyword and argument to restrict the command to a specific node.
ingress	(Optional) Displays information for the ingress PSE device only. **Note** Follow the **ingress** keyword with the **location** *node-id* keyword and argument to restrict the command to a specific node.
location *node-id*	(Optional) Identifies the location of the node whose PSE device information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router. **Note** Include the **instance** {**0**	**1** } argument before the **location** *node-id* keyword and argument to restrict the command to a specific device instance on the specified node.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced.
Release 3.5.0	The **instance** { **0**	**1**} keywords were replaced by the **egress** and **ingress** keywords.

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following command shows how to display a summary of PSE information for a specific controller instance:

```
RP/0/RP0/CPU0:router# show controllers pse summary ingress

Node: 0/1/CPU0:
----------------------------------------

Ingress PSE, Summary Info:
-------------------------------------------------
IBM P/N     :   LotNum      :
SerialNum   : 0xffffff  BadPPECls   : 0000
Version     : 3         CpuctrlPort : 5
DeviceState : 0 (UP)
StartupOpts : 00000000  MmappedBase : 0x61111000
ClsDisMask  : 0000      NFusedPPEs  : 4   (0 hwf, 4 swf)
MPUcodeName : /pkg/ucode/crs/pse/ingress_mp_v3.mucode
PPEUcodeName: /pkg/ucode/crs/pse/metro_ingress_turbo_plim2_v3.mucode
INTR-Status : 00000000  INTR-Enable : 0x7ffffe
NColdResets : 1         NWarmResets : 0
NPPEUcDlds  : 1         NResetRetry : 0
NIntrtps    : 1         NIntrptThrot: 0

             Node: 0/6/CPU0:
----------------------------------------

Ingress PSE, Summary Info:
-------------------------------------------------
IBM P/N     :   LotNum      :
SerialNum   : 0xffffff  BadPPECls   : 0000
Version     : 3         CpuctrlPort : 5
DeviceState : 0 (UP)
StartupOpts : 00000000  MmappedBase : 0x61111000
ClsDisMask  : 0000      NFusedPPEs  : 4   (0 hwf, 4 swf)
MPUcodeName : /pkg/ucode/crs/pse/ingress_mp_v3.mucode
PPEUcodeName: /pkg/ucode/crs/pse/metro_ingress_turbo_plim2_v3.mucode
INTR-Status : 00000000  INTR-Enable : 0x7ffffe
NColdResets : 1         NWarmResets : 0
NPPEUcDlds  : 1         NResetRetry : 0
NIntrtps    : 1         NIntrptThrot: 0
```

This table describes the significant fields shown in the display.

Table 20: show controllers pse summary Field Descriptions

Field	Description
Node	Node whose PSE information is displayed.
IBM P/N	IBM part number.
SerialNum	Serial number of the node.
BadPPECls	Number of bad PPECIs on the PSE.
Version	PSE version.
CpuctrlPort	Cpuctrl port associated with the PSE instance.
DeviceState	Whether the node is up (active) or down (inactive).
StartupOpts	Internal information about the PSE startup options.

Field	Description
MmappedBase	Internal mapping information.
ClsDisMask	Internal masking information.
NFusedPPEs	Internal PPE information.
MPUcodeName	Information about the MPU code.
PPEUcodeName	Information about the PPEU code.
INTR-Status	Internal status information.
INTR-Enable	Internal enable information.
NColdResets	Number of cold resets experienced by the router.
NWarmResets	Number of warm resets experienced by the router.
NPPEUcDlds	Internal information about the PSE.
NResetRetry	Number of times the router attempted to reset itself.
NIntrtps	Internal information about interrupts on the PSE.
NIntrptThrot	Internal throttling information for the PSE.

Related Commands

Command	Description
show controllers pse eio links, on page 101	Displays packet switching engine information for Elastic I/O links.
show controllers pse ipc, on page 104	Displays packet switching engine device information for interprocess communication connections, or for a specific IPC controller.
show controllers pse mp, on page 111	displays packet switching engine information for the maintenance processor on a specific controller or node.
show controllers pse summary, on page 124	Displays a summary of packet switching engine information for a specific controller or node.

show controllers pse tcam

To display the ternary content addressable memory (TCAM) manager module information for a packet switching engine (PSE) on a specific controller or node, use the **show controllers pse tcam** command in EXEC mode.

show controllers pse tcam [{**connections** | **contents** | **lookup** | **prefix-compression** | **region-addrs** | **region-list** | **rlb-range-map** | **rsm** | **summary** | **trace**}] [{**egress** | **ingress**}] [**location** *node-id*]

Syntax Description

connections	(Optional) Displays the summary information for the client and server connections.
contents	(Optional)Displays the contents of TCAM entries/registers and range logic block (RLB) entries.
lookup	(Optional) Displays the results of a lookup operation in a specific bank.
prefix-compression	(Optional) Displays the failed prefixes in the IPv6 prefix compression.
region-addrs	(Optional) Displays the CAM addresses used by a specific region.
region-list	(Optional) Displays region handles for a feature in a specific bank.
rlb-range-map	(Optional) Displays the mapping of specified port range to fields in TCAM entries.
rsm	(Optional) Display resource shadow memory (RSM) data.
summary	(Optional) Displays the summary of CAM free space or entries for a region.
trace	(Optional) Displays the TCAM manager trace data.
egress	(Optional) Displays information for the egress PSE device only. **Note** Follow the **egress** argument with the **location** *node-id* keyword and argument to restrict the command to a specific node.
ingress	(Optional) Displays information for the ingress PSE device only. **Note** Follow the **ingress** keyword with the **location** *node-id* keyword and argument to restrict the command to a specific node.
location *node-id*	(Optional) Identifies the location of the node whose PSE device information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation. **Note** Use the **show platform** command to see the location of all nodes installed in the router. **Note** Include the **instance** {**0** | **1** } argument before the **location** *node-id* keyword and argument to restrict the command to a specific device instance on the specified node.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.8.0	This command was introduced.
Release 5.1.3	The **lpts-ipv6-compression** keyword was added under **region-list** and **prefix-compression** . The show command output was updated to display local packet transport services (LPTS) IPv6 compression information.

Usage Guidelines

The optional **egress** or **ingress** and **location** keywords are not mutually exclusive. The **egress** and **ingress** keywords direct the command to specific PSE device, and the **location** keyword directs the command to that device or devices on the specified modular services card. You can specify the **egress** or **ingress** and **location** options together in the same command.

Task ID

Task ID	Operations
acl	read
interface	read
drivers	read

Examples

The following command shows how to display a summary of PSE TCAM information for a specific controller instance:

```
RP/0/RP0/CPU0:router# show controllers pse tcam summary ingress location 0/1/cpu0

TCAM Device Information for Ingress Metro, CAM channel 0:
Device size: 18M (256K array entries of 72-bits), 260014 available
Current mode of operation: Turbo
Software Initialization:
        Memory management state: complete
        Range block state: complete
        IPv6 prefix compression state: complete
Hardware Initialization:
        Device registers: complete
        CAM/SRAM Memory: complete
        Default entries for applications: complete
        Range Logic Block registers: complete
        IPv6 prefix compression region: complete
Feature specific information:
        packet filtering (id 0):
                Owner client id: 3.   Limit 260096 cells
                Total 1 regions using 76 CAM cells
        QoS (id 1):
                Owner client id: 1.   Limit 260096 cells
                Duplication enabled in Turbo mode into CAM channel 1
        Fab QoS (id 2):
                Owner client id: 1.   Limit 260096 cells
                Duplication enabled in Turbo mode into CAM channel 1
        ipv6 prefix compress (id 10):
                Owner client id: 13.   Limit 260096 cells
                Total 1 regions using 2 CAM cells
```

```
                Entry duplication enabled in Turbo and Feature modes into CAM c1
        tcam_mgr (id 11):
                Owner client id: 13.   Limit 260096 cells
                Total 1 regions using 4 CAM cells
        L2FIB (id 12):
                Owner client id: 14.   Limit 260096 cells
                Total 1 regions using 2048 CAM cells
        LI (id 13):
                Owner client id: 3.   Limit 262144 cells
                Total 1 regions using 0 CAM cells
                Duplication enabled in Turbo mode into CAM channel 1
TCAM Device Information for Ingress Metro, CAM channel 1:
Device size: 18M (256K array entries of 72-bits), 261760 available
Current mode of operation: Turbo
Software Initialization:
        Memory management state: complete
        Range block state: complete
        IPv6 prefix compression state: complete
Hardware Initialization:
        Device registers: complete
        CAM/SRAM Memory: complete
        Default entries for applications: complete
        Range Logic Block registers: complete
        IPv6 prefix compression region: complete
Feature specific information:
        Pre-IFIB (id 4):
                Owner client id: 10.   Limit 260096 cells
                Total 3 regions using 302 CAM cells
        ipv6 prefix compress (id 10):
                Owner client id: 13.   Limit 260096 cells
                Total 1 regions using 2 CAM cells
        tcam_mgr (id 11):
                Owner client id: 13.   Limit 260096 cells
                Total 2 regions using 80 CAM cells
        L2FIB (id 12):
                Owner client id: 14.   Limit 260096 cells
                Total 1 regions using 0 CAM cells
        LI (id 13):
                Owner client id: 3.   Limit 262144 cells
                Total 1 regions using 0 CAM cells
```

The following command shows how to display LPTS IPv6 compression information of PSE TCAM at a specific controller instance:

```
RP/0/RP0/CPU0:router# show controllers pse tcam region-list ingress lpts-ipv6-compression
location 0/1/cpu0
Number of regions for feature 36 in Ingress PSE, bank 0 is: 1
Region:            LPTS IPv6 pref compr ACL ID 0x000f id_len 4 region ID 7 region duplicate
 ID 7 entries 7

TCAM logical region information:
Ingress PSE, CAM bank 0, region name LPTS IPv6 pref compr
region ID 7 region duplicate ID 7
for feature lpts ipv6 prefix compress owner-id 15
Region size 7 entries (Max allowed 16000) of 144 bits (Result entry size 16 bytes)
ACL ID 0x   f of width 4 bits allocated from bucket (calculated) 0x6f msb 0x60
CAM cells in Ingress PSE, bank 0 used by region 7:
entry 0x01d02-0x01d0f, offset 0 - 13, result 0x01d02-0x01d0f, map 1 result allocated
Total cells: 14
Frame index: 7
Frame index bitmap: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x80
Total Frame used: 1

Number of regions for feature 36 in Ingress PSE, bank 1 is: 1
```

```
Region:            LPTS IPv6 pref compr ACL ID 0x000f id_len 4 region ID 7 region duplicate
 ID 0 entries 7

TCAM logical region information:
Ingress PSE, CAM bank 1, region name LPTS IPv6 pref compr
region ID 7 region duplicate ID 0
for feature lpts ipv6 prefix compress owner-id 15
Region size 7 entries (Max allowed 16000) of 144 bits (Result entry size 16 bytes)
ACL ID 0x   f of width 4 bits allocated from bucket (calculated) 0x6f msb 0x60
CAM cells in Ingress PSE, bank 1 used by region 7:
entry 0x01802-0x01803, offset 0 - 1, result 0x01802-0x01803, map 1 result allocated
entry 0x0180e-0x01819, offset 2 - 13, result 0x0180e-0x01819, map 1 result allocated
Total cells: 14
Frame index: 6
Frame index bitmap: 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x40
Total Frame used: 1
```

show controllers pse utilization

To display the packet processing engine (PPE) utilization information in the packet switching engine (PSE) on a specific controller or node, use the **show controllers pse utilization** command in EXEC mode. The PPE processes each packet individually within the PSE ASIC and there are multiple PPEs present in the PSE.

show controllers pse utilization [{**egress** | **ingress**}]

Syntax Description

egress	(Optional) Displays the PPE utilization information for the egress PSE device only. **Note** Follow the **egress** argument with the **location** *node-id* keyword and argument to restrict the command to a specific node.
ingress	(Optional) Displays PPE utilization information for the ingress PSE device only. **Note** Follow the **ingress** keyword with the **location** *node-id* keyword and argument to restrict the command to a specific node.
location *node-id*	(Optional) Identifies the location of the node whose PPE utilization information you want to display. The *node-id* argument is expressed in the *rack/slot/module* notation.

Command Default

No default behavior or values

Command Modes

EXEC mode

Command History

Release	Modification
Release 3.8.0	This command was introduced.

Usage Guidelines

The optional **egress** or **ingress** and **location** keywords are not mutually exclusive. The **egress** and **ingress** keywords direct the command to specific PSE device, and the **location** keyword directs the command to that device or devices on the specified modular services card. You can specify the **egress** or **ingress** and **location** options together in the same command.

Task ID

Task ID	Operations
interface	read
drivers	read

Examples

The following command shows how to display the PPE utilization information of all the ingress locations:

```
RP/0/RP0/CPU0:router# show controllers pse utilization ingress location all

PPE Utilization
```

```
        Node            Ingress   Egress
=======================================
        0/1/CPU0:          0.0
        0/6/CPU0:          0.1
```

The following command shows how to display the PPE utilization information of all the nodes having access to a PSE:

```
RP/0/RP0/CPU0:router# show controllers pse utilization

PPE Utilization
        Node            Ingress   Egress
=======================================
        0/1/CPU0:          0.0       0.0
        0/6/CPU0:          0.0       1.0
```

show packet-memory

To display information for packet memory, use the **show packet-memory** command inEXEC mode.

show packet-memory [{**clients** | **corrupt** | **failures** | **hssd** | **ifinput** | **ifoutput** | **internal** | **inuse** | **job** | **mutex** | **old** | **reserved** | **summary** | **trace** | **watch**}] [**location** *node-id*]

Syntax Description

clients	(Optional) Displays the packet manager clients.
corrupt	(Optional) Displays the information about corrupted packets.
failures	(Optional) Displays the packet buffer, header, hardware buffer allocation failures.
hssd	(Optional) Displays High Speed Small Data (HSSD).
ifinput	(Optional) Displays packets from a specific interface.
ifoutput	(Optional) Displays packets to a specific interface.
internal	(Optional) Displays the packet memory along with actual number of particles in free list.
inuse	(Optional) Displays the total number of packets in use
job	(Optional) Displays the number of packets owned by a specific process.
mutex	(Optional) Displays the pakman mutex monitoring configuration.
old	(Optional) Displays the total number of packets older than one minute.
reserved	(Optional) Displays the reserved memory information.
summary	(Optional) Displays the packet memory usage summary information.
trace	(Optional) Displays the packet-memory traces.
watch	(Optional) Displays the pakman watch configuration.
location *node-id*	(Optional) Displays detailed packet memory information for the designated node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

Displays information about all packet memory.

Command Modes

EXEC mode

Command History

Release	Modification
Release 2.0	This command was introduced .

Release	Modification
Release 3.9.0	Included the following keywords: • clients • corrupt • failures • fsv • hssd • ifinput • ifoutput • internal • inuse • job • mutex • old • reserved • summary • trace • watch

Usage Guidelines

The **show packet-memory** command can be used to display the total number of packet and particle headers, along with the packet memory that is currently allocated in the system.

Task ID

Task ID	Operations
basic-services	read

Examples

The following example shows how to display packet memory information:

```
RP/0/RP0/CPU0:router# show packet-memory

Packet memory statistics :
==========================
Packet headers
total: 32000, free: 32000, size: 448
Particle Pools(8)
Pool(0):total: 8000, free: 8000, size: 256
fallback: 0, region: 0
Pool(1):total: 4000, free: 3968, size: 512
fallback: 1, region: 0
Pool(2):total: 16, free: 16, size: 512
fallback: 2, region: 0
Pool(3):total: 8000, free: 7936, size: 768
fallback: 3, region: 0
Pool(4):total: 12800, free: 9172, size: 1648
fallback: 4, region: 0
Pool(5):total: 320, free: 320, size: 2560
fallback: 5, region: 0
Pool(6):total: 1600, free: 1088, size: 4608
fallback: 6, region: 0
Pool(7):total: 640, free: 640, size: 6240
```

```
fallback: 7, region: 0
Particle clone
total: 8000, free: 8000, size: 256
Packet Feature Specific Variable (FSV)
total: 16000, free: 16000, size: 88
Packet trace
total: 16384, free: 16384, size: 40
```

This table describes the significant fields shown in the display.

Table 21: show packet memory Field Descriptions

Field	Description
Packet headers	Data structure that defines and controls an aggregation of data structures, collectively known as a packet. Includes information about every packet in the system.
Particle Pools	Data structure that describes a particle and may be chained to other particles in a linked list. Includes information about the actual data of the packet and other particle headers in this packet if present in this packet.
Particle clone	Duplicate particle header that points to a previously allocated particle. Differs from a particle header in that a particle clone shares the particle with another particle header.
Packet Feature Specific Variable (FSV)	Scratch pad shared among the features in the packet path, listing hangs of the packet header.
Packet trace	Data associated with the packet header to help tracing a packet in the system.