debug iapp through debug ip ftp

debug iapp

Use the debug iapp privileged EXEC command to begin debugging of IAPP operations. Use the **no**form of this command to stop the debug operation.

[no] debug iapp {**packets**| **event**| **error**}

Syntax Description

packets	Displays IAPP packets sent and received by the access point. Link test packets are not displayed
event	Displays significant IAPP events
error	Displays IAPP software and protocol errors

Command Default

This command has no default setting.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(11)JA	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

This example shows how to begin debugging of IAPP packets:

```
SOAP-AP# debug iapp packet
```

This example shows how to begin debugging of IAPP events:

```
SOAP-AP# debug iapp events
```

This example shows how to begin debugging of IAPP errors:

```
SOAP-AP# debug iapp errors
```

Related Commands

Command	Description
show debugging	Displays all debug settings

debug idmgr

To enable debugging for the identity manager (IDMGR), use the **debug idmgr** command in privileged EXEC mode. To disable debugging for the IDMGR, use the **no** form of this command.

debug idmgr {**core**| **data**| **db**| **elog**| **flow local**}

Syntax Description

core	Specifies debugging for the Layer 2 (L2) access core process flow.
data	Specifies debugging for data handling.
db	Specifies debugging for database interaction.
elog	Specifies debugging for event logging.
flow local	Specifies debugging for remote and local interaction.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(2)S	This command was introduced.

Usage Guidelines

You can use the **debug idmgr** command to debug errors such as missing or incorrect attributes in a session or Accounting, Authentication, and Authorization (AAA) records.

Usage Guidelines

The following is sample output from the **debug idmgr** command:

```
Router# debug idmgr core
IDMGR core process flow debugging is on
Router# debug idmgr data
IDMGR data handling debugging is on
Router# debug idmgr db
IDMGR database interaction debugging is on
R1# debug idmgr elog
IDMGR event logging debugging is on
R1# debug idmgr flow local
IDMGR local process flow debugging is on
2w6d: %SYS-5-CONFIG_I: Configured from console by console
2w6d: IDMGR: Enabled core flow debugging
2w6d: IDMGR: Enabled local flow debugging
2w6d: IDMGR: Enabled DB interaction debugging
2w6d: IDMGR:(07EC4890) got an Session Assert Request
2w6d: IDMGR:(07EC4890) Local processing Session Assert Request
2w6d: IDMGR: Set field session-handle 2281701385(88000009) in idmgr db record
```

```
2w6d: IDMGR: Set field aaa-unique-id 16(00000010) in idmgr db record
2w6d: IDMGR: Set field composite-key in idmgr db record
2w6d: IDMGR: Set field idmgr-data in idmgr db record
2w6d: IDMGR:(07EC4890) Adding new record 07640138 for session handle 88000009 to Session
DB
2w6d: IDMGR: Enabled core flow debugging
2w6d: IDMGR: Enabled local flow debugging
2w6d: IDMGR: Enabled DB interaction debugging
2w6d: IDMGR:(07EC4890) got an Session Update Event
2w6d: IDMGR:(07EC4890) Local processing Session Update Event
2w6d: IDMGR:(07EC4890) Search for session record
2w6d: IDMGR: Set field session-handle 2281701385(88000009) in search record
2w6d: IDMGR:(07EC4890) Found match for session handle 88000009
2w6d: IDMGR:(07EC4890) Found record in search get, returning 07640138
2w6d: IDMGR: releasing memory for search record field with type session-handle
2w6d: IDMGR: Set field idmgr-mask 4294967295(FFFFFFFF) in search record
2w6d: IDMGR: releasing memory for search record field with type idmgr-mask
Router#
2w6d: IDMGR:(07EC4890) Updating attribute authen-status in datalist
2w6d: IDMGR:(07EC4890) Updated record 07640138 for 88000009 to Session DB
```

Related Commands

Command	Description
show subscriber session	Displays information about subscriber sessions on an ISG.

debug if-mgr efp-ext

To enable debugging for the interface manager (IF-MGR) Ethernet flow point (EFP) extension, use the **debug if-mgr efp-ext**command in privileged EXEC mode. To turn off debugging for the IF-MGR EFP extension, use the **no**form of this command.

debug if-mgr {**errors**| **trace**} **efp-ext**

no debug if-mgr {**errors**| **trace**} **efp-ext**

Syntax Description

errors	Specifies debugging for IF-MGR EFP extension errors.
trace	Specifies debugging for IF-MGR EFP extension traces.

Command Default

Debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRD1	This command was introduced.

Usage Guidelines

Before you issue the **debug if-mgr efp-ext**command, consider the high volume of output that debug commands usually generate and the amount of time the debugging operation may take.

Examples

The following example shows how to enable debugging for IF-MGR EFP extension errors:

```
Router> enable
Router# debug if-mgr errors efp-ext
Router#
```

debug ima

To display debugging messages for inverse multiplexing over AMT (IMA) groups and links, use the **debug ima**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ima

no debug ima

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging for IMA groups is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(5)XK	This command was modified.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following example shows output when you enter the **debug ima** command while adding two ATM links to an IMA group. Notice that the group has not yet been created with the **interface atm** *slot* **/ima** *group-number* command, so the links are not activated yet as group members. However, the individual ATM links are deactivated.

```
Router# debug ima

IMA network interface debugging is on
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface atm1/0
Router(config-if)# ima-group 1
Router(config-if)#
01:35:08:IMA shutdown atm layer of link ATM1/0
01:35:08:ima_clear_atm_layer_if ATM1/0
01:35:08:IMA link ATM1/0 removed in firmware
01:35:08:ima_release_channel:ATM1/0 released channel 0.
01:35:08:Bring up ATM1/4 that had been waiting for a free channel.
01:35:08:IMA:no shut the ATM interface.
01:35:08:IMA allocate_channel:ATM1/4 using channel 0.
01:35:08:IMA config_restart ATM1/4
01:35:08:IMA
adding link 0 to Group ATM1/IMA1ATM1/0 is down waiting for IMA group 1 to be activated
01:35:08:Link 0 was added to Group ATM1/IMA1
01:35:08:ATM1/0 is down waiting for IMA group 1 to be created.
01:35:08:IMA send AIS on link ATM1/0
01:35:08:IMA Link up/down Alarm:port 0, new status 0x10, old_status 0x1.
```

```
01:35:10:%LINK-3-UPDOWN:Interface ATM1/4, changed state to up
01:35:10:%LINK-3-UPDOWN:Interface ATM1/0, changed state to down
01:35:11:%LINEPROTO-5-UPDOWN:Line protocol on Interface ATM1/4, changed state to up
01:35:11:%LINEPROTO-5-UPDOWN:Line protocol on Interface ATM1/0, changed state to down
Router(config-if)# int atm1/1
Router(config-if)# ima-group 1
Router(config-if)#
01:37:19:IMA shutdown atm layer of link ATM1/1
01:37:19:ima_clear_atm_layer_if ATM1/1
01:37:19:IMA link ATM1/1 removed in firmware
01:37:19:ima_release_channel:ATM1/1 released channel 1.
01:37:19:Bring up ATM1/5 that had been waiting for a free channel.
01:37:19:IMA:no shut the ATM interface.
01:37:19:IMA allocate_channel:ATM1/5 using channel 1.
01:37:19:IMA config_restart ATM1/5
01:37:19:IMA adding link 1 to Group ATM1/IMA1ATM1/1 is down waiting for IMA group 1 to be
activated
01:37:19:Link 1 was added to Group ATM1/IMA1
01:37:19:ATM1/1 is down waiting for IMA group 1 to be created.
01:37:19:IMA send AIS on link ATM1/1
01:37:19:IMA Link up/down Alarm:port 1, new status 0x10, old_status 0x1.
Router(config-if)#
01:37:21:%LINK-3-UPDOWN:Interface ATM1/5, changed state to up
01:37:21:%LINK-3-UPDOWN:Interface ATM1/1, changed state to down
01:37:22:%LINEPROTO-5-UPDOWN:Line protocol on Interface ATM1/5, changed state to up
01:37:22:%LINEPROTO-5-UPDOWN:Line protocol on Interface ATM1/1, changed state to down
```

Related Commands

Command	Description
debug backhaul-session-manager set	Displays debugging messages for ATM errors, and reports specific problems such as encapsulation errors and errors related to OAM cells.
debug events	Displays debugging messages for ATM events, and reports specific events such as PVC setup completion, changes in carrier states, and interface rates.

debug installer

To enable debugs in the installer, use the **debug installer** command in Privileged EXEC mode. To disable debugging use the **no** form of the command.

debug installer [**all**| **process**| **issu**| **common**]

Syntax Description

all	Enables all installer debugs
process	Enables all the debugs inside Installer process
issu	Enables all the debugs inside the installer's Bash provisioning scripts
common	Enables all the debugs inside the installer common code

Command Default

No debugs enabled

Command Modes

Privileged EXEC

Command History

Release	Modification
IOS XE 3.2.0 SE	Command introduced.

Privileged EXEC

Usage Guidelines

The debug output for the above commands is displayed to the console and/or the IOS logging buffer.

It's always a good idea to turn on **debug installer all** when troubleshooting installer related problems

Examples

To enable all installer debugs, perform the following:

```
        infra-p2-3#debug installer all
All installer debugging is on
```

Related Commands

Command	Description
show version	To display information about the currently loaded software along with hardware and device information, use the **show version** command.

debug interface

To display interface descriptor block debugging messages, use the **debug interface**command in privileged EXEC mode. To disable the debugging messages, use the **no** form of this command.

debug interface *type number*

no debug interface *type number*

Syntax Description

type number	Interface type and number. In the case of an ATM interface, you get the following options once you enter the interface type and number: • **vc** --Displays information about the virtual circuit. • [*vpi* /]*vci*--Specifies the virtual channel identifier (VCI) or virtual path identifier/virtual channel identifier (VPI/VCI) pair, if the interface to be debugged is an ATM-encapsulated interface. Valid values for *vpi* are 0 to 255. Valid values for *vci* are 1 to 65535.

Command Default

By default, debugging messages are not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(4)T	This command was introduced.
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.

Examples

The following is sample output from the **debug interface** command:

```
Router# debug interface ATM 1/0 vc 0/5
Condition 1 set
*Jan 31 19:36:38.399: ATM VC Debug: Condition 1, atm-vc 0/5 AT1/0 triggered, count 1
```

Related Commands

Command	Description
debug interface counters exceptions	Displays a message when a recoverable exceptional condition happens during the computation of the interface packet and data rate statistics.
debug interface counters protocol memory	Displays the memory operations (create and free) of protocol counters on interfaces and debugging messages during memory operations.

debug interface counters exceptions

To display a message when a recoverable exceptional condition happens during the computation of the interface packet and data rate statistics, use the **debug interface counters exceptions** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug interface counters exceptions

no debug interface counters exceptions

Syntax Description

This command has no arguments or keywords.

Command Default

By default, the debugging messages are not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(4)T	This command was introduced.
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.

Usage Guidelines

Use the **debug interface counters exceptions** command to debug problems where the packet counter values or rates have unexpected values. The command helps to flag interfaces whose packet counter values have decreased in number. This condition can occur if a packet is counted and then dropped. This command helps you to determine if the input and output rate statistics are adjusted to display a zero value versus an unexpected value. It is also possible for zero values to be displayed if an interface is running at or close to its maximum capacity due to interface statistics being viewed as negative values.

This message is rate limited to one message per minute. If multiple interfaces are having unexpected counter statistic issues, then a message is displayed only for the first interface that experiences a problem within a minute.

Examples

The following is sample output from the **debug interface counters exceptions** command when backward-going counters are detected. The output is self-explanatory.

```
Router# debug interface counters exceptions
IF-4-BACKWARD_COUNTERS: Corrected for backward rx_bytes counters (561759 -> 526385) on
Multilink1
IF-4-BACKWARD_COUNTERS: Corrected for backward tx_bytes counters (288114 -> 268710) on
Multilink1
IF-4-BACKWARD_COUNTERS: Corrected for backward tx_bytes counters (2220 -> 0) on
Virtual-Access4
```

Related Commands

Command	Description
debug interface	Displays the interface descriptor block debugging messages.
debug interface counters protocol memory	Displays the memory operations (create and free) of protocol counters on interfaces and debugging messages during memory operations.

debug interface counters protocol memory

To display the memory operations (create and free) of protocol counters on interfaces and debugging messages during memory operations, use the **debug interface counters protocol memory**command in privileged EXEC mode. To disable the debugging output, use the **no** form of this command.

debug interface counters protocol memory

no debug interface counters protocol memory

Syntax Description

This command has no arguments or keywords.

Command Default

By default, the debugging messages are not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(4)T	This command was introduced.
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.

Examples

The following is sample output from the **debug interface counters protocol memory**command. The output is self-explanatory.

```
Router# debug interface counters protocol memory
interface counter protocol memory operations debugging is on
*Jan 11 11:34:08.154: IDB_PROTO: Ethernet0/0 created CDP
*Jan 11 11:35:08.154: IDB_PROTO: Ethernet0/0 reset CDP
```

Related Commands

Command	Description
debug interface	Displays the interface descriptor block debugging messages.
debug interface counters exceptions	Displays a message when a recoverable exceptional condition happens during the computation of the interface packet and data rate statistics.

debug interface states

To display intermediary messages when an interface's state transitions, use the **debug interface states** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug interface states

no debug interface states

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(11)T	This command was introduced.
12.2(44)S	This command was integrated into Cisco IOS Release 12.2(44)S.

Usage Guidelines

This command helps to debug interface state transition problems and includes the following interface state related message outputs:

- BRIDGE_ADJ--bridging database and Spanning tree protocol (STP) port state adjustment
- CSTATE_REQ--carrier state change request
- CSTATE_TMR--carrier timer state change
- LSTATE_REQ--line protocol state change request
- LSTATE_TMR--line protocol timer state change
- ROUTE_ADJ--route adjustment
- TRANS_ADJ--state transition adjustment

The debug information can be restricted to display state transitions on an interface basis using the **debug condition interface** command.

Caution Because the **debug interface states** command is a global debug command for all the interfaces in the router, in some cases such as with online insertion and removal (OIR) this command generates a substantial amount of output, depending on the number of interfaces hosted on the shared port adapter (SPA) or the line card. Use the **debug condition interface** command instead for debugging an interface state transition problem.

Examples

The following is sample output from the **debug interface states**command when the **shutdown** command is executed on an interface. The output is self-explanatory.

```
Router# debug interface states
interface state transitions debugging is on
Router# debug condition interface fast0/0
Condition 1 set
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
*Sep  1 12:24:46.294: [IDB Fa0/0 UARUYY] LSTATE_REQ: Entry
*Sep  1 12:24:46.294: [IDB Fa0/0 UARUYY] LSTATE_REQ: timers not running
*Sep  1 12:24:46.294: [IDB Fa0/0 UARUYY] LSTATE_REQ: Exit
Router(config)# interface fast0/0
Router(config-if)# shut
Router(config-if)#
*Sep  1 12:24:56.294: [IDB Fa0/0 UARUYY] LSTATE_REQ: Entry
*Sep  1 12:24:56.294: [IDB Fa0/0 UARUYY] LSTATE_REQ: timers not running
*Sep  1 12:24:56.294: [IDB Fa0/0 UARUYY] LSTATE_REQ: Exit
*Sep  1 12:24:57.162: [IDB Fa0/0 UARUYY] CSTATE_REQ: Entry, requested
state: A
*Sep  1 12:24:57.162: [IDB Fa0/0 UARUYY] CSTATE_REQ: starting ctimer (2000)
*Sep  1 12:24:57.162: [IDB Fa0/0 AURUYY] CSTATE_REQ: state assign
*Sep  1 12:24:57.162: [IDB Fa0/0 AURUYY] LSTATE_REQ: Entry
*Sep  1 12:24:57.162: [IDB Fa0/0 AURUYY] LSTATE_REQ: Exit
*Sep  1 12:24:57.162: [IDB Fa0/0 AURUYY] CSTATE_REQ: Exit
*Sep  1 12:24:57.162: [IDB Fa0/0 AURUYY] CSTATE_REQ: Entry, requested
state: A
*Sep  1 12:24:57.162: [IDB Fa0/0 AURUYY] CSTATE_REQ: state assign
*Sep  1 12:24:57.162: [IDB Fa0/0 AURUYY] LSTATE_REQ: Entry
*Sep  1 12:24:57.162: [IDB Fa0/0 AURUYY] LSTATE_REQ: Exit
*Sep  1 12:24:57.162: [IDB Fa0/0 AURUYY] CSTATE_REQ: Exit
*Sep  1 12:24:57.166: [IDB Fa0/0 AURUnY] TRANS_ADJ: Entry
*Sep  1 12:24:57.166: [IDB Fa0/0 AURUnn] TRANS_ADJ: propagating change
to subifs
*Sep  1 12:24:57.170: [IDB Fa0/0 AURUnn] TRANS_ADJ: Exit
*Sep  1 12:24:57.170: [IDB Fa0/0 AURUnn] ROUTE_ADJ: Entry
*Sep  1 12:24:57.170: [IDB Fa0/0 AURUnn] ROUTE_ADJ: Exit
*Sep  1 12:24:57.170: [IDB Fa0/0 AURUnn] BRIDGE_ADJ: Entry
*Sep  1 12:24:57.170: [IDB Fa0/0 AURUnn] BRIDGE_ADJ: Exit
*Sep  1 12:24:59.162: [IDB Fa0/0 AURUnn] CSTATE_TMR: Entry
*Sep  1 12:24:59.162: [IDB Fa0/0 AURUnn] CSTATE_TMR: netidb=Fa0/0,
linestate: n
*Sep  1 12:24:59.162: [IDB Fa0/0 AURUnn] LSTATE_REQ: Entry
*Sep  1 12:24:59.162: [IDB Fa0/0 AURUnn] LSTATE_REQ: timers not running
*Sep  1 12:24:59.162: [IDB Fa0/0 AURUnn] LSTATE_REQ: starting lineproto
timer
*Sep  1 12:24:59.162: [IDB Fa0/0 AURUnn] LSTATE_REQ: Exit
*Sep  1 12:24:59.162: [IDB Fa0/0 AURUnn] CSTATE_TMR: transition detected
*Sep  1 12:24:59.162: %ENTITY_ALARM-6-INFO: ASSERT INFO Fa0/0 Physical
Port Administrative State Down
*Sep  1 12:24:59.162: [IDB Fa0/0 AURUnn] TRANS_ADJ: Entry
*Sep  1 12:24:59.162: [IDB Fa0/0 AURUnn] TRANS_ADJ: Exit
*Sep  1 12:24:59.162: [IDB Fa0/0 AURUnn] CSTATE_TMR: Exit
*Sep  1 12:25:00.162: [IDB Fa0/0 AURUnn] LSTATE_TMR: Entry
*Sep  1 12:25:00.162: [IDB Fa0/0 AURUnn] LSTATE_TMR: not spoofing,
current state: n
*Sep  1 12:25:00.162: [IDB Fa0/0 AURUnn] LSTATE_TMR: informing line
```

```
state transitions
*Sep  1 12:25:00.162: [IDB Fa0/0 AURUnn] TRANS_ADJ: Entry
*Sep  1 12:25:00.162: [IDB Fa0/0 AURUnn] TRANS_ADJ: Exit
*Sep  1 12:25:00.162: [IDB Fa0/0 AURUnn] ROUTE_ADJ: Entry
*Sep  1 12:25:00.162: [IDB Fa0/0 AURUnn] ROUTE_ADJ: Exit
*Sep  1 12:25:00.162: [IDB Fa0/0 AURUnn] LSTATE_TMR: Exit
```

Related Commands

Command	Description
debug condition interface	Limits output for some debug commands on the basis of the interface, VC, or VLAN.

debug interface(vasi)

To display debugging information for the VRF-Aware Service Infrastructure (VASI) interface descriptor block, use the **debug interface** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug interface {**vasileft| vasiright**} *number*

no debug interface {**vasileft| vasiright**} *number*

Syntax Description

vasileft	Displays information about vasileft interface.
vasiright	Displays information about vasiright interface.
number	Identifier of the VASI interface. The range is from 1 to 256.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 2.6	This command was introduced.

Examples

The following is sample output from the **debug interface**command:

```
Router# debug interface vasileft 100
Condition 1 set
```

Related Commands

interface (vasi)	Configures a VASI virtual interface.
debug adjacency (vasi)	Displays debugging information for the VASI adjacency.
debug vasi	Displays debugging information for the VASI.
show vasi pair	Displays the status of a VASI pair.

debug iosd issu

To enable all the debugs inside the IOS issu_iosd and iosvrp_issu_upgrade subsystems, use the **debug iosd issu** command in Privileged EXEC mode. To disable debugging use the **no** form of the command.

debug iosd issu

Command Default

Debugs not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
IOS XE 3.2.0 SE	Command introduced.

Privileged EXEC

Usage Guidelines

No command variables

It's always a good idea to turn on **debug iosd issu** when troubleshooting installer related problems

Related Commands

Command	Description
show version	To display information about the currently loaded software along with hardware and device information, use the **show version** command.

debug ip access-list hash-generation

To display debugging information about access control list (ACL) hash-value generation (for ACL Syslog entries), use the **debug ip access-list hash-generation** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip access-list hash-generation

no debug ip access-list hash-generation

Syntax Description

This command has no arguments or keywords.

Command Default

This command is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(22)T	This command was introduced.

Usage Guidelines

Use this command when configuring an access control entry (ACE) to view the router-generated hash values for the ACE.

This command displays the input and output for the hash-generation mechanism. The input is the ACE text and ACL name. The output is an MD5 algorithm-derived, 4-byte value.

Examples

The following example shows sample debug output displayed when configuring ACL hash-value generation.

Note The example in this section shows sample output for a numbered access list. However, you can configure ACL hash-value generation for both numbered and named access lists, and for both standard and extended access lists.

```
Router#
*Aug 9 00:24:31.765: %SYS-5-CONFIG_I: Configured from console by console
Router# debug ip access-list hash-generation
Syslog hash code generation debugging is on
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip access-list logging hash-generation
Router(config)# access-list 101 permit tcp host 10.1.1.1 host 10.1.1.2 log
Router(config)#
*Aug 9 00:25:31.661: %IPACL-HASHGEN: Hash Input: 101 extended permit 6 host 20.1.1.1 host
20.1.1.2 Hash Output: 0xA363BB54
Router(config)# exit
Router#
```

Related Commands

Command	Description
ip access-list logging hash-generation	Enables the generation of hash-values for access control entries in the system messaging logs.
show ip access-list	Displays the contents of all current access lists.

debug ip access-list intstats

To display information about whether or not the interface-level statistics of an access list were created, updated, cleared or deleted successfully, use the **debug ip access-list intstats**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip access-list intstats

no debug ip access-list intstats

Syntax Description

This command has no arguments or keywords.

Command Default

No default behaviors or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(6)T	This command was introduced.

Examples

The following is sample output from the **debug ip access-list intstats**command:

```
Router# debug ip access-list intstats
Router# enable
Router# configure terminal
Router(config)#interface e0/0
Router(config-if)#ip access-group 100 in
*Oct 29 08:52:16.763: IPACL-INTSTATS: ACL swsb created
*Oct 29 08:52:16.763: IPACL-INTSTATS: ACL header stats structure created
*Oct 29 08:52:16.763: IPACL-INTSTATS: I/P stats table created
*Oct 29 08:52:16.763: IPACL-INTSTATS: Statsid bitmap created
*Oct 29 08:52:16.763: IPACL-INTSTATS: Done with static ACEs
Router(config-if)#ip access-group 100 out
*Oct 29 08:52:19.435: IPACL-INTSTATS: O/P stats table created
*Oct 29 08:52:19.435: IPACL-INTSTATS: Done with static ACEs
```

debug ip access-list turboacl

To display debugging information about turbo access control lists (ACLs), use the **debug ip access-list turboacl**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip access-list turboacl

no debug ip access-list turboacl

Syntax Description

This command has no arguments or keywords.

Command Default

No default behaviors or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2	This command was introduced.
12.3(3)T	This command was modified to include support for turbo ACLs.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The **debug ip access-list turboacl**command is useful for debugging problems associated with turbo ACLs. Turbo ACLs compile the ACLs into a set of lookup tables, while maintaining the first packet matching requirements. Packet headers are used to access these tables in a small, fixed, number of lookups, independent of the existing number of ACL entries.

Examples

The following is sample output from the **debug ip access-list turboacl**command:

```
Router# debug ip access-list turboacl

*Aug 20 00:41:17.843 UTC:Miss at index 73, 19
*Aug 20 00:41:17.843 UTC:Adding dynamic entry, update = 1
*Aug 20 00:41:17.843 UTC:Miss at index 21, 39
*Aug 20 00:41:17.847 UTC:Adding dynamic entry, update = 1
*Aug 20 00:41:17.847 UTC:Miss at index 116, 42
*Aug 20 00:41:17.851 UTC:Adding dynamic entry, update = 1
*Aug 20 00:41:17.851 UTC:Miss at index 119, 28
*Aug 20 00:41:17.851 UTC:Adding dynamic entry, update = 1
*Aug 20 00:41:17.855 UTC:Miss at index 116, 42
*Aug 20 00:41:17.855 UTC:Adding dynamic entry, update = 1
*Aug 20 00:41:17.855 UTC:Miss at index 92, 20
*Aug 20 00:41:17.855 UTC:Adding dynamic entry, update = 1
*Aug 20 00:41:17.855 UTC:Miss at index 119, 28
```

```
*Aug 20 00:41:17.855 UTC:Adding dynamic entry, update = 1
*Aug 20 00:41:17.855 UTC:Miss at index 56, 29
*Aug 20 00:41:17.859 UTC:Adding dynamic entry, update = 1
*Aug 20 00:41:17.859try, update = 1
*Aug 20 00:41:19.959 UTC:Miss at index 29, 41
*Aug 20 00:41:19.959 UTC:Adding dynamic entry, update = 1
*Aug 20 00:41:19.959 UTC:Miss at index 29, 38
```

The table below describes the significant fields shown in the display.

Table 1: debug ip access-list turboacl Field Descriptions

Field	Description
Aug 20 00:41:17.843 UTC	Date and Coordinated Universal Time (UTC) the command was used to debug the turbo ACL.
Miss at index 73, 19	Location in the compiled access list tables where a new packet lookup does not match an existing entry.
Adding dynamic entry, update = 1	Action taken to add a new entry in the compiled access list tables as a result of a packet being processed.

debug ip admission consent

To display authentication proxy consent page information on the router, use the **debug ip admission consent** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip admission consent [**events**| **errors**| **messages**]

no debug ip admission consent

Syntax Description

errors	(Optional) Displays only error messages.
events	(Optional) Displays only event-related messages.
messages	(Optional) Displays only packet-related messages.

Command Default

If an option is not selected, all debug messages are displayed.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(15)T	This command was introduced.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

```
Router# debug ip admission consent errors

IP Admission Consent Errors debugging is on
Router# debug ip admission consent events

IP Admission Consent Events debugging is on
Router# debug ip admission consent messages

IP Admission Consent Messages debugging is on
Router#
Router# show debugging
IP Admission Consent:
IP Admission Consent Errors debugging is on
IP Admission Consent Events debugging is on
IP Admission Consent Messages debugging is on
```

debug ip admission eapoudp

To display information about Extensible Authentication Protocol over User Datagram Protocol (UDP) (EAPoUDP) network admission control events, use the **debug ip admission eapoudp**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip admission eapoudp

no debug ip admission eapoudp

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC #

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Examples

The following sample output from the **debug ip admission eapoudp** command shows information about network admission control using EAPoUDP. In the command output, the term “posture” refers to the credentials (for example, antivirus state or version of Cisco IOS software) of the host system.

```
Router# debug ip admission eapoudp

Posture validation session created for client mac= 0001.027c.f364 ip= 10.0.0.1
Total Posture sessions= 1 Total Posture Init sessions= 1
*Apr  9 19:39:45.684: %AP-6-POSTURE_START_VALIDATION: IP=10.0.0.1|
Interface=FastEthernet0/0.420
*Apr  9 19:40:42.292: %AP-6-POSTURE_STATE_CHANGE: IP=10.0.0.1| STATE=POSTURE ESTAB
*Apr  9 19:40:42.292: auth_proxy_posture_parse_aaa_attributes:
CiscoDefined-ACL name= #ACSACL#-IP-HealthyACL-40921e54
Apr  9 19:40:42.957: %AP-6-POSTURE_POLICY: Apply access control list
(xACSACLx-IP-HealthyACL-40921e54) policy for host (10.0.0.1)
```

The fields in the display are self-explanatory.

Related Commands

Command	Description
show ip admission	Displays IP admission control cache entries or the running admission control configuration.

debug ip auth-proxy

To display the authentication proxy configuration information on the router, use the **debug ip auth-proxy**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip auth-proxy {**detailed| ftp| function-trace| object-creation| object-deletion| telnet| timers**}

no debug ip auth-proxy

Syntax Description

detailed	Displays details of the TCP events during an authentication proxy process. The details are generic to all FTP, HTTP, and Telnet protocols.
ftp	Displays FTP events related to the authentication proxy.
function-trace	Displays the authentication proxy functions.
object-creation	Displays additional entries to the authentication proxy cache.
object-deletion	Displays deletion of cache entries for the authentication proxy.
telnet	Displays Telnet-related authentication proxy events.
timers	Displays authentication proxy timer-related events.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.3(1)	The **detailed** keyword was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use the **debug ip auth-proxy** command to display authentication proxy activity.

> **Note** The **function-trace** debugging information provides low-level software information for Cisco technical support representatives. No output examples are provided for this keyword option.

Examples

The following examples illustrate the output of the **debug ip auth-proxy** command. In these examples, debugging is on for object creations, object deletions, HTTP, and TCP.

In this example, the client host at 192.168.201.1 is attempting to make an HTTP connection to the web server located at 192.168.21.1. The HTTP debugging information is on for the authentication proxy. The output shows that the router is setting up an authentication proxy entry for the login request:

```
00:11:10: AUTH-PROXY creates info:
cliaddr - 192.168.21.1, cliport - 36583
seraddr - 192.168.201.1, serport - 80
ip-srcaddr 192.168.21.1
pak-srcaddr 0.0.0.0
```

Following a successful login attempt, the debugging information shows the authentication proxy entries created for the client. In this example, the client is authorized for SMTP (port 25), FTP data (port 20), FTP control (port 21), and Telnet (port 23) traffic. The dynamic access control list (ACL) entries are included in the display.

```
00:11:25:AUTH_PROXY OBJ_CREATE:acl item 61AD60CC

00:11:25:AUTH-PROXY OBJ_CREATE:create acl wrapper 6151C7C8 -- acl item 61AD60CC
00:11:25:AUTH-PROXY Src 192.168.162.216 Port [0]
00:11:25:AUTH-PROXY Dst 192.168.162.220 Port [25]
00:11:25:AUTH_PROXY OBJ_CREATE:acl item 6151C908

00:11:25:AUTH-PROXY OBJ_CREATE:create acl wrapper 6187A060 -- acl item 6151C908
00:11:25:AUTH-PROXY Src 192.168.162.216 Port [0]
00:11:25:AUTH-PROXY Dst 192.168.162.220 Port [20]
00:11:25:AUTH_PROXY OBJ_CREATE:acl item 61A40B88

00:11:25:AUTH-PROXY OBJ_CREATE:create acl wrapper 6187A0D4 -- acl item 61A40B88
00:11:25:AUTH-PROXY Src 192.168.162.216 Port [0]
00:11:25:AUTH-PROXY Dst 192.168.162.220 Port [21]
00:11:25:AUTH_PROXY OBJ_CREATE:acl item 61879550

00:11:25:AUTH-PROXY OBJ_CREATE:create acl wrapper 61879644 -- acl item 61879550
00:11:25:AUTH-PROXY Src 192.168.162.216 Port [0]
00:11:25:AUTH-PROXY Dst 192.168.162.220 Port [23]
```

The next example shows the debug output following a **clear ip auth-proxy cache** command to clear the authentication entries from the router. The dynamic ACL entries are removed from the router.

```
00:12:36:AUTH-PROXY OBJ_DELETE:delete auth_proxy cache 61AD6298
00:12:36:AUTH-PROXY OBJ_DELETE:delete create acl wrapper 6151C7C8 -- acl item 61AD60CC
00:12:36:AUTH-PROXY OBJ_DELETE:delete create acl wrapper 6187A060 -- acl item 6151C908
00:12:36:AUTH-PROXY OBJ_DELETE:delete create acl wrapper 6187A0D4 -- acl item 61A40B88
00:12:36:AUTH-PROXY OBJ_DELETE:delete create acl wrapper 61879644 -- acl item 61879550
```

The following example shows the timer information for a dynamic ACL entry. All times are expressed in milliseconds. The *first laststart* is the time that the ACL entry is created relative to the startup time of the router. The *lastref* is the time of the last packet to hit the dynamic ACL relative to the startup time of the router. The *exptime* is the next expected expiration time for the dynamic ACL. The *delta* indicates the remaining time before the dynamic ACL expires. After the timer expires, the debugging information includes a message indicating that the ACL and associated authentication proxy information for the client have been removed.

```
00:19:51:first laststart 1191112

00:20:51:AUTH-PROXY:delta 54220 lastref 1245332 exptime 1251112
00:21:45:AUTH-PROXY:ACL and cache are removed
```

The following example is sample output with the **detailed** keyword enabled:

```
00:37:50:AUTH-PROXY:proto_flag=5, dstport_index=1
00:37:50: SYN SEQ 245972 LEN 0
00:37:50:dst_addr 192.168.127.2 src_addr 192.168.27.1 dst_port 21 src_port 4347
00:37:50:AUTH-PROXY:auth_proxy_half_open_count++ 1
00:37:50:AUTH-PROXY:proto_flag=5, dstport_index=1
00:37:50: ACK 1820245643 SEQ 245973 LEN 0
00:37:50:dst_addr 192.168.127.2 src_addr 192.168.27.1 dst_port 21 src_port 4347
00:37:50:clientport 4347 state 0
00:37:50:AUTH-PROXY:incremented proxy_proc_count=1
00:37:50:AUTH-PROXY:proto_flag=5, dstport_index=1
00:37:50: ACK 1820245674 SEQ 245973 LEN 0
00:37:50:dst_addr 192.168.127.2 src_addr 192.168.27.1 dst_port 21 src_port 4347
00:37:50:clientport 4347 state 0
00:37:57:AUTH-PROXY:proto_flag=5, dstport_index=1
00:37:57: PSH ACK 1820245674 SEQ 245973 LEN 16
00:37:57:dst_addr 192.168.127.2 src_addr 192.168.27.1 dst_port 21 src_port 4347
00:37:57:clientport 4347 state 0
00:37:57:AUTH-PROXY:proto_flag=5, dstport_index=1
00:37:57: ACK 1820245699 SEQ 245989 LEN 0
00:37:57:dst_addr 192.168.127.2 src_addr 192.168.27.1 dst_port 21 src_port 4347
00:37:57:clientport 4347 state 0
00:38:01:AUTH-PROXY:proto_flag=5, dstport_index=1
00:38:01: PSH ACK 1820245699 SEQ 245989 LEN 16
00:38:01:dst_addr 192.168.127.2 src_addr 192.168.27.1 dst_port 21 src_port 4347
00:38:01:clientport 4347 state 0
00:38:01:AUTH-PROXY:Authenticating user ryan
00:38:01:AUTH-PROXY:Session state is INIT.Not updating stats
00:38:01:AUTH-PROXY:Session state is INIT.Not updating stats
00:38:01:AUTH-PROXY:Sent AAA request successfully
00:38:01:AUTH-PROXY:Sent password successfully
00:38:01:AUTH-PROXY:processing authorization data
00:38:01:AUTH-PROXY:Sending accounting start.unique-id 2
00:38:01:AUTH-PROXY:Session state is INIT.Not updating stats
00:38:01:AUTH-PROXY:Session state is INIT.Not updating stats
00:38:01:AUTH-PROXY:wait complete on watched boolean stat=0
00:38:01:AUTH-PROXY:src ip addr is 192.168.127.2, dstaddr=192.168.27.1
00:38:01: SYN ACK 2072458992 SEQ 4051022445 LEN 0
00:38:01:AUTH-PROXY:src ip addr is 192.168.127.2, dstaddr=192.168.27.1
00:38:01: PSH ACK 2072458992 SEQ 4051022446 LEN 49
00:38:02:AUTH-PROXY:src ip addr is 192.168.127.2, dstaddr=192.168.27.1
00:38:02: ACK 2072459003 SEQ 4051022495 LEN 0
00:38:02:AUTH-PROXY:src ip addr is 192.168.127.2, dstaddr=192.168.27.1
00:38:02: PSH ACK 2072459003 SEQ 4051022495 LEN 33
00:38:02:AUTH-PROXY:src ip addr is 192.168.127.2, dstaddr=192.168.27.1
00:38:02: ACK 2072459014 SEQ 4051022528 LEN 0
00:38:02:AUTH-PROXY:src ip addr is 192.168.127.2, dstaddr=192.168.27.1
00:38:02: PSH ACK 2072459014 SEQ 4051022528 LEN 26
00:38:03:AUTH-PROXY:proto_flag=5, dstport_index=1
00:38:03: ACK 1820245725 SEQ 246005 LEN 0
00:38:03:dst_addr 192.168.127.2 src_addr 192.168.27.1 dst_port 21 src_port 4347
00:38:03:clientport 4347 state 3
7200b#
```

Related Commands

Command	Description
show debug	Displays the debug options set on the router.

debug ip auth-proxy ezvpn

To display information related to proxy authentication behavior for web-based activation, use the **debug ip auth-proxy ezvpn**command in privileged EXEC mode. To turn off debugging, use the **no** form of this command.

debug ip auth-proxy ezvpn

no debug ip auth-proxy ezvpn

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not turned on.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS 12.2SX family of releases. Support in a specific 12.2SX release is dependent on your feature set, platform, and platform hardware.

Usage Guidelin

Caution Using this command may result in considerable output if simultaneous authentications are taking place.

Examples

The following is output from the **debug ip auth-proxy ezvpn** command. The output displays the proxy authentication behavior of a web-based activation.

```
Router# debug ip auth-proxy ezvpn
*Dec 20 20:25:11.006: AUTH-PROXY: New request received by EzVPN WebIntercept from
10.4.205.205
*Dec 20 20:25:17.150: AUTH-PROXY:GET request received
*Dec 20 20:25:17.150: AUTH-PROXY:Authentication scheme is 401
*Dec 20 20:25:17.362: AUTH-PROXY:Authorization information not present in GET request
*Dec 20 20:25:17.362: AUTH-PROXY: Allocated on credinfo for connect at 0x81EF1A84
*Dec 20 20:25:17.362: AUTH-PROXY: Posting CONNECT request to EzVPN
*
Dec 20 20:25:17.362: EZVPN(tunnel22): Received CONNECT from 10.4.205.205!
*Dec 20 20:25:17.366: EZVPN(tunnel22): Current State: CONNECT_REQUIRED
*Dec 20 20:25:17.366: EZVPN(tunnel22): Event: CONNECT
```

The output in the display is self-explanatory.

Related Commands

Command	Description
xauth userid mode	Specifies how the Cisco Easy VPN Client handles Xauth requests or prompts from the server.

debug ip bgp

To display information related to processing of the Border Gateway Protocol (BGP), use the **debug ip bgp command in**privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip bgp [*ip-address*| **addpath**| **dampening**| **events**| **in**| **keepalives**| **out**| **updates**| **vpnv4**| **mpls**]

no debug ip bgp [*ip-address*| **addpath**| **dampening**| **events**| **in**| **keepalives**| **out**| **updates**| **vpnv4**| **mpls**]

Cisco 10000 Series Router

debug ip bgp [*ip-address*| **dampening**| **events**| **in**| **keepalives**| **out**| **updates**| **vpnv4**| **mpls**| **all**| **groups**| **import**| **ipv4**| **ipv6**]

no debug ip bgp [*ip-address*| **dampening**| **events**| **in**| **keepalives**| **out**| **updates**| **vpnv4**| **mpls**| **all**| **groups**| **import**| **ipv4**| **ipv6**]

Syntax Description

ip-address	(Optional) The BGP neighbor IP address.
addpath	(Optional) Displays BGP additional path events.
dampening	(Optional) Displays BGP dampening.
events	(Optional) Displays BGP events.
in	(Optional) Displays BGP inbound information.
keepalives	(Optional) Displays BGP keepalives.
out	(Optional) Displays BGP outbound information.
updates	(Optional) Displays BGP updates.
vpnv4	(Optional) Displays Virtual Private Network version 4 (VPNv4) Network Layer Reachability Information (NLRI).
mpls	(Optional) Displays Multiprotocol Label Switching (MPLS) information.
all	(Optional) Displays all address family information.
groups	(Optional) Displays BGP configuration and update groups information.
import	(Optional) Displays BGP import routes to a VPN routing and forwarding (VRF) instance across address family information.

ipv4	(Optional) Displays BGP IPv4 address family information.
ipv6	(Optional) Displays BGP IPv6 address family information.

Command Modes

Privileged EXEC(#)

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS Release 12.0(21)ST. The **mpls** keyword was added.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(13)T	The **mpls** keyword was added.
12.2(17b)SXA	This command was integrated into Cisco IOS Release 12.2(17b)SXA.
12.0(27)S	The command output was modified to show explicit-null label information.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2(33)SRE	This command was modified. The **addpath** keyword was added.
12.2(33)XNE	This command was integrated into Cisco IOS Release 12.2(33)XNE.
Cisco IOS XE Release 2.5	This command was integrated into Cisco IOS XE Release 2.5.

Usage Guidelines

Use this command with the **updates** and **mpls** keywords to display explicit-null label information. The optional arguments in, out, keepalives, updates, and events provide verbose output to the debug ip bgp command. The sequence in which the optional arguments are provided affects the behavior of the command. The non peer specific commands override the peer-specific commands.

Examples

Following is the sample output from the **debug ip bgp** command used with vpnv4 keyword:

```
Router# debug ip bgp vpnv4
```

```
03:47:14:vpn:bgp_vpnv4_bnetinit:100:2:10.0.0.0/8
03:47:14:vpn:bnettable add:100:2:10.0.0.0/8
03:47:14:vpn:bestpath_hook route_tag_change for vpn2:10.0.0.0/255.0.0.0(ok)
03:47:14:vpn:bgp_vpnv4_bnetinit:100:2:10.0.0.0/8
03:47:14:vpn:bnettable add:100:2:10.0.0.0/8
03:47:14:vpn:bestpath_hook route_tag_change for vpn2:10.0.0.0/255.0.0.0(ok)
03:47:14:vpn:bgp_vpnv4_bnetinit:100:2:10.0.0.0/8
03:47:14:vpn:bnettable add:100:2:10.0.0.0/8
03:47:14:vpn:bestpath_hook route_tag_chacle ip bgp *nge for vpn2:10.0.0.0/255.0.0.0(ok)
```

The following example shows sample output, including the explicit-null label, from the d**ebug ip bgp updates** and the **debug ip bgp mpls** commands:

```
Router# debug ip bgp updates
BGP updates debugging is on
Router# debug ip bgp mpls
BGP MPLS labels debugging is on

Router#
01:33:53: BGP(0): route 10.10.10.10/32 up
01:33:53: BGP(0): nettable_walker 10.10.10.10/32 route sourced locally
01:33:53: BGP: adding MPLS label to 10.10.10.10/32
01:33:53: BGP: check on 10.10.10.10/8 in LDP - ok
01:33:53: BGP: label imp-null allocated via LDP
01:33:53: BGP-IPv4: send exp-null label for 10.10.10.10/32
01:33:53: BGP-IPv4: Send prefix 10.10.10.10/32, label exp-null    !explicit-null label being
 sent
01:33:53: BGP(0): 10.10.10.11 send UPDATE (format) 10.10.10.10/32, next 10.10.10.12, metric
 0, path , mpls label 0 !label value is 0
01:33:53: BGP(0): updgrp 1 - 10.10.10.12 enqueued 1 updates, average/maximum size (bytes)
61/61
```

Following example shows a sample output from the debug ip bgp command when various arguments are provided in a particular sequence:

```
Router# debug ip bgp 209.165.200.225
Router# debug ip bgp 209.165.200.225 updates
Router# debug ip bgp keepalives
Router# debug ip bgp events
Router# debug ip bgp in
Router# debug ip bgp out

Router# show debug
IP routing:
  BGP debugging is on (outbound) for address family: IPv4 Unicast
  BGP events debugging is on
  BGP keepalives debugging is on
  BGP updates debugging is on (outbound) for address family: IPv4 Unicast
```

The behavior of the command changes when the arguments are provided in a different sequence

```
Router# debug ip bgp keepalives
Router# debug ip bgp events
Router# debug ip bgp in
Router# debug ip bgp out
Router# debug ip bgp 209.165.200.225
Router# debug ip bgp 209.165.200.225 updates

Router# show debug
IP routing:
  BGP debugging is on for neighbor 209.165.200.225 for address family: IPv4 Unicast
  BGP events debugging is on for neighbor 209.165.200.225
  BGP keepalives debugging is on for neighbor 209.165.200.225 for address family: IPv4
Unicast
  BGP updates debugging is on for neighbor 209.165.200.225 for address family: IPv4 Unicast
```

debug ip bgp groups

To display information related to the processing of Border Gateway Protocol (BGP) update-groups, use the **debug ip bgp update** privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip bgp groups [*index-group*| *ip-address*]

no debug ip bgp groups

Syntax Description

index-group	(Optional) Specifies that update-group debugging information for the corresponding index number will be displayed. The range of update-group index numbers is from 1 to 4294967295.
ip-address	(Optional) Specifies that update-group debugging information for a single peer will be displayed.

Command Default

No information about BGP update-groups is displayed.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The output of this command displays information about update-group calculations and the addition and removal of update-group members. Information about peer-groups, peer-policy, and peer-session templates will also be displayed in the output of this command as neighbor configurations change.

Note The output of this command can be very verbose. This command should not be deployed in a production network unless you are troubleshooting a problem.

When a change to outbound policy occurs, the router automatically recalculates update-group memberships and applies the changes by triggering an outbound soft reset after a 1-minute timer expires. This behavior is designed to provide the network operator with time to change the configuration if a mistake is made. You can manually enable an outbound soft reset before the timer expires by entering the **clear ip bgp** *ip-address* **soft out** command.

Note In Cisco IOS Release 12.0(25)S, 12.3(2)T, and prior releases the update group recalculation delay timer is set to 3 minutes.

Examples

The following sample output from the **debug ip bgp groups** command shows that peering has been established with neighbor 10.4.9.8 and update-group calculations are occurring for this member:

```
Router# debug ip bgp groups

5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.8 flags 0x0 cap 0x0 and updgrp 1 fl0
5w4d: BGP-DYN(0): Created update-group(0) flags 0x0 cap 0x0 from neighbor 10.4.0
5w4d: BGP-DYN(0): Adding  neighbor 10.4.9.8 flags 0x0 cap 0x0, to update-group 0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 Up
```

The following sample output from the **debug ip bgp groups** command shows the recalculation of update-groups after the **clear ip bgp groups** command was issued:

```
Router# debug ip bgp groups

5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.5 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.5 flags 0x0 cap 0x0 and updgrp 2 fl0
5w4d: BGP-DYN(0): Update-group 2 flags 0x0 cap 0x0 policies same as 10.4.9.5 fl0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.8 flags 0x0 cap 0x0 and updgrp 2 fl0
5w4d: BGP-DYN(0): Update-group 2 flags 0x0 cap 0x0 policies same as 10.4.9.8 fl0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.21 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.21 flags 0x0 cap 0x0 and updgrp 1 f0
5w4d: BGP-DYN(0): Update-group 1 flags 0x0 cap 0x0 policies same as 10.4.9.21 f0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.5 Up
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.21 Up
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 Up
```

The table below describes the significant fields shown in the display.

Table 2: debug ip bgp groups Field Descriptions

Field	Description
%BGP-5-ADJCHANGE:	A BGP neighbor has come Up or gone Down. The IP address of the neighbor is specified in the output string.
BGP-DYN(0):	This line is displayed when a neighbor adjacency is established. The BGP dynamic update group algorithm analyzes the policies of the new neighbor and then adds the neighbor to the appropriate BGP update group.

Related Commands

Command	Description
clear ip bgp	Resets a BGP connection or session.
clear ip bgp update-group	Clears BGP update-group member sessions.
show ip bgp replication	Displays BGP update-group replication statistics.
show ip bgp update-group	Displays information about BGP update-groups.

debug ip bgp igp-metric ignore

To display information related to the system ignoring the Interior Gateway Protocol (IGP) metric during best path selection, use the **debug ip bgp igp-metric ignore**command in privileged EXEC mode. To disable such debugging output, use the **no** form of the command.

debug ip bgp igp-metric ignore

no debug ip bgp igp-metric ignore

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.4S	This command was introduced.

Usage Guidelines

You might use this command if the path you expected to be chosen as the best path at the shadow RR was not chosen as such. That could be because the **bgp bestpath igp-metric ignore** command makes the best path algorithm choose the same best path as the primary RR if they are not co-located.

Examples

The following example turns on debugging of events related to the system ignoring the IGP metric during bestpath selection:

```
Router# debug ip bgp igp-metric ignore
```

Related Commands

Command	Description
bgp bestpath igp-metric ignore	Specifies that the system ignore the Interior Gateway Protocol (IGP) metric during best path selection.

debug ip bgp import

To display debugging information related to importing IPv4 prefixes from the BGP global routing table into a VRF table or exporting from a VRF table into the BGP global table, use the **debug ip bgp import** command in privileged EXEC mode. To disable the display of such debugging information, use the **no** form of this command.

debug ip bgp import {**events**| **updates** [*access-list*| *expanded-access-list*]}

no debug ip bgp import {**events**| **updates** [*access-list*| *expanded-access-list*]}

Syntax Description

events	Displays messages related to IPv4 prefix import events.
updates	Displays messages related to IPv4 prefix import updates.
access-list	(Optional) Number of the access list used to filter debugging messages. The range is from 1 to 199.
expanded-access-list	(Optional) Number of the expanded access list used to filter debugging messages. The range is from 1300 to 2699.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(29)S	This command was introduced.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.3(14)T	This command was integrated into Cisco IOS Release 12.3(14)T.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
15.2(4)S	This command was modified. The output now includes information for the BGP Support for IP Prefix Export from a VRF to the Global Table feature.
Cisco IOS XE Release 3.7S	This command was modified. The output now includes information for the BGP Support for IP Prefix Export from a VRF to the Global Table feature.

Usage Guidelines

Use this command to display debugging information related to the BGP Support for IP Prefix Import from Global Table into a VRF Table feature or the BGP Support for IP Prefix Export from a VRF Table into Global Table feature. The former feature provides the capability to import IPv4 unicast prefixes from the global routing table into a Virtual Private Network (VPN) routing/forwarding (VRF) instance table using an import route map. The latter feature provides the capability to export IPv4 or IPv6 prefixes from a VRF table into the global table using an export route map.

Examples

The following example configures IPv4 prefix import debugging messages for both import events and import updates to be displayed on the console of the router:

```
Router# debug ip bgp import events

BGP import events debugging is on
Router# debug ip bgp import updates
BGP import updates debugging is on for access list 3
00:00:50: %BGP-5-ADJCHANGE: neighbor 10.2.2.2 Up
00:01:06: BGP: reevaluate IPv4 Unicast routes in VRF academic
00:01:06: BGP: 0 routes available (limit: 1000)
00:01:06: BGP: import IPv4 Unicast routes to VRF academic
00:01:06: BGP(2)-VRF(academic): import pfx 100:1:10.30.1.0/24 via 10.2.2.2
00:01:06: BGP: accepted 8 routes (limit: 1000)
00:01:06: BGP: reevaluate IPv4 Multicast routes in VRF multicast
00:01:06: BGP: 0 routes available (limit: 2)
00:01:06: BGP: import IPv4 Multicast routes to VRF multicast
00:01:06: %BGP-4-AFIMPORT: IPv4 Multicast prefixes imported to multicast vrf reached the
limit 2
00:01:06: BGP: accepted 2 routes (limit: 2)
00:01:06: BGP: reevaluate IPv4 Unicast routes in VRF BLUE
00:01:06: BGP: 0 routes available (limit: 1000)
00:01:06: BGP: import IPv4 Unicast routes to VRF BLUE
00:01:06: BGP: accepted 3 routes (limit: 1000)
```

The table below describes the significant fields shown in the display.

Table 3: debug ip bgp import Field Descriptions

Field	Description
BGP: accepted 2 routes (limit: 2)	Number of routes imported into the VRF, and the default or user-defined prefix import limit.
BGP: reevaluate IPv4 Unicast routes in VRF BLUE	Prefix was imported during BGP convergence and is being reevaluated for the next scan cycle.
BGP: 0 routes available (limit: 1000)	Number of routes available from the import source, and the default or user-defined prefix import limit.
BGP: import IPv4 Unicast routes to VRF BLUE	Import map and prefix type (unicast or multicast) that is being imported into the specified VRF.

The following is a sample debug message for the IP prefix export from a VRF table to global table:

```
Device# debug ip bgp import events
```

```
*Jul 12 10:06:48.357: BGP GBL-IMP: vpn1:VPNv4 Unicast:base 1:1:192.168.4.0/24
-> global:IPv4 Unicast:base Creating importing net.
  4.4.4.4 (metric 11) from 4.4.4.4 (4.4.4.4)
   Origin IGP, metric 0, localpref 100, valid, internal, best
   Extended Community: RT:1:1
   mpls labels in/out nolabel/16
```

Related Commands

Command	Description
clear ip bgp	Resets a BGP connection.
export map (VRF table to global table)	Exports IP prefixes from a VRF table to the global routing table based on a route map.
import map	Imports IP prefixes from the global routing table to a VRF table based on a route map.

debug ip bgp range

To display debugging information related to Border Gateway Protocol (BGP) dynamic subnet range neighbors, use the **debug ip bgp range** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip bgp range [detail]

no debug ip bgp range

Syntax Description

detail	(Optional) Specifies that detailed debugging information about BGP dynamic subnet range neighbors will be displayed.

Command Default

No debugging information about BGP dynamic subnet range neighbors is displayed.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SXH	This command was introduced.
15.0(1)S	This command was integrated into Release 15.0(1)S.
Cisco IOS XE Release 3.1S	This command was integrated into Cisco IOS XE Release 3.1S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

The output of this command displays information about the identification and creation of BGP dynamic subnet range neighbors. BGP dynamic neighbors are configured using a range of IP addresses and BGP peer groups. After a subnet range is configured for a BGP peer group, and a TCP session is initiated for an IP address in the subnet range, a new BGP neighbor is dynamically created as a member of that group. The new BGP neighbor will inherit any configuration or templates for the group.

Examples

The following output shows that the **debug ip bgp range** command has been entered and a BGP neighbor at 192.168.3.2 has been dynamically created using the subnet range 192.168.0.0/16. This new neighbor is a member of the peer group named group192.

```
Router# debug ip bgp range
bgprange_debug = 1, sense = 1
BGP dynamic Range debugging is on
!
```

```
*Mar 26 20:05:13.251: BGP:DN: Created a new neighbor *192.168.3.2
in range 192.168.0.0/16, peer-group group192,count = 1
```

The following sample output from the **debug ip bgp range detail** command shows more detailed debugging of the addition of dynamic BGP neighbors:

```
Router# debug ip bgp range detail
bgprange_debug = 1, sense = 1
BGP dynamic Range debugging is on with detail (Dynamic Range neighbors details only)
!
*Mar 26 20:09:12.311: BGP:DN: ACCEPT an OPEN from 192.168.1.2 valid range
0x32123D8:192.168.0.0/16,tcb 0x32114C0
!
*Mar 26 20:09:12.331: BGP: 192.168.1.2 passive open to 192.168.1.1
*Mar 26 20:09:12.331: BGP:DN: ACCEPTED an OPEN from 192.168.1.2 valid range
0x32123D8:192.168.0.0/16,tcb 0x3494040
!
*Mar 26 20:09:12.331: BGP:DN: Created a new neighbor *192.168.1.2
in range 192.168.0.0/16, peer-group group192,count = 2
```

The table below describes the significant field shown in the display.

Table 4: debug ip bgp range Field Descriptions

Field	Description
BGP:DN:	A potential dynamic BGP neighbor has been identified as opening a TCP session with an IP address in a subnet associated with a BGP peer group. BGP accepts the session and creates a new neighbor. The new neighbor becomes a member of the peer group associated with its subnet range.

Related Commands

Command	Description
bgp listen	Configures BGP dynamic neighbor parameters.
clear ip bgp peer-group	Clears BGP peer group member sessions.
show ip bgp peer-group	Displays information about BGP peer groups.

debug ip bgp sso

To display Border Gateway Protocol (BGP)-related stateful switchover (SSO) events or debugging information for BGP-related interactions between the active Route Processor (RP) and the standby RP, use the **debug ip bgp sso**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip bgp sso {**events| transactions**} **[detail]**

no debug ip bgp sso {**events| transactions**} **[detail]**

Syntax Description

events	Displays BGP-related SSO failures.
transactions	Displays debugging information for failed BGP-related interactions between the active RP and the standby RP.
detail	(Optional) Displays detailed debugging information about successful BGP-related SSO operations and successful BGP-related interactions between the active and the standby RP.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(28)SB	This command was introduced.
12.2(33)SRB1	This command was integrated into Cisco IOS Release 12.2(33)SRB1.
15.0(1)S	This command was integrated into Cisco IOS Release 15.0(1)S.
Cisco IOS XE 3.1S	This command was integrated into Cisco IOS XE Release 3.1S.

Usage Guidelines

The **debug ip bgp sso**command is used to display BGP-related SSO events or debugging information for BGP-related interactions between the active RP and the standby RP. This command is useful for monitoring or troubleshooting BGP sessions on a provider edge (PE) router during an RP switchover or during a planned In-Service Software Upgrade (ISSU).

Examples

The following is sample output from the **debug ip bgp sso** command with the **events** keyword. The following output indicates that the 10.34.32.154 BGP session is no longer SSO capable.

```
*Mar 28 02:29:43.526: BGPSSO: 10.34.32.154 reset SSO and decrement count
```

Tip Use the **show ip bgp vpnv4 all neighbors**command to display the reason that the SSO-capable BGP session has been disabled.

The following is sample output from the **debug ip bgp sso** command with the **transactions** keyword. The following output shows an SSO notification indicating that the SSO capability is pending for 602 BGP neighbors. This notification is generated as the state between the active and standby RP is being synchronized during the bulk synchonization phase of SSO initialization. During this phase, the Transmission Control Blocks (TCBs) must be synchronized with the TCBs on the standby RP before SSO initialization is complete.

```
*Mar 28 02:32:12.102: BGPSSO: tcp sso notify pending for 602 nbrs
```

debug ip bgp updates

To display information about the processing of Border Gateway Protocol (BGP) updates, use the **debug ip bgp updates** command in privileged EXEC mode. To disable the display of BGP update information, use the **no** form of this command.

debug ip bgp updates [*access-list*| *expanded-access-list*] [**in**| **out**] **[events] [refresh]**

no debug ip bgp updates [*access-list*| *expanded-access-list*] [**in**| **out**] **[events] [refresh]**

Syntax Description

access-list	(Optional) Number of access list used to filter debugging messages. The range that can be specified is from 1 to 199.
expanded-access-list	(Optional) Number of expanded access lists used to filter debugging messages. The range that can be specified is from 1300 to 2699.
in	(Optional) Specifies debugging messages for inbound BGP update information.
out	(Optional) Specifies debugging messages for outbound BGP update information.
events	(Optional) Specifies debugging messages for BGP update events.
refresh	(Optional) Specifies debugging messages for BGP update refresh.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(31)SB	This command was modified. The **refresh** keyword was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debug ip bgp updates** command. The output shows that the BGP session was cleared between neighbor 10.4.9.21 and the local router 10.4.9.4. There are no field description tables for this command because the debugging output from this command depends on the subsequent commands that are entered.

```
Router# debug ip bgp updates
5w2d: %SYS-5-CONFIG_I: Configured from console by console
5w2d: BGP: 10.4.9.21 went from Idle to Active
5w2d: BGP: 10.4.9.21 open active, delay 7032ms
5w2d: BGP: 10.4.9.21 open active, local address 10.4.9.4
5w2d: BGP: 10.4.9.21 went from Active to OpenSent
5w2d: BGP: 10.4.9.21 sending OPEN, version 4, my as: 101
5w2d: BGP: 10.4.9.21 send message type 1, length (incl. header) 45
5w2d: BGP: 10.4.9.21 rcv message type 1, length (excl. header) 26
5w2d: BGP: 10.4.9.21 rcv OPEN, version 4
5w2d: BGP: 10.4.9.21 rcv OPEN w/ OPTION parameter len: 16
5w2d: BGP: 10.4.9.21 rcvd OPEN w/ optional parameter type 2 (Capability) len 6
5w2d: BGP: 10.4.9.21 OPEN has CAPABILITY code: 1, length 4
5w2d: BGP: 10.4.9.21 OPEN has MP_EXT CAP for afi/safi: 1/1
5w2d: BGP: 10.4.9.21 rcvd OPEN w/ optional parameter type 2 (Capability) len 2
5w2d: BGP: 10.4.9.21 OPEN has CAPABILITY code: 128, length 0
5w2d: BGP: 10.4.9.21 OPEN has ROUTE-REFRESH capability(old) for all address-fams
5w2d: BGP: 10.4.9.21 rcvd OPEN w/ optional parameter type 2 (Capability) len 2
5w2d: BGP: 10.4.9.21 OPEN has CAPABILITY code: 2, length 0
5w2d: BGP: 10.4.9.21 OPEN has ROUTE-REFRESH capability for all address-families
5w2d: BGP: 10.4.9.21 went from OpenSent to OpenConfirm
5w2d: BGP: 10.4.9.21 went from OpenConfirm to Established
5w2d: %BGP-5-ADJCHANGE: neighbor 10.4.9.21 Up
5w2d: BGP(0): 10.4.9.21 computing updates, afi 0, neighbor version 0, table ver0
5w2d: BGP(0): 10.4.9.21 update run completed, afi 0, ran for 0ms, neighbor vers1
5w2d: BGP(0): 10.4.9.21 initial update completed
```

The following is sample output from the **debug ip bgp updates out**command. The output shows that the local router is sending updates with the cost community:

```
Router# debug ip bgp updates out
*Mar 15 01:41:23.515:BGP(0):10.0.0.5 computing updates, afi 0, neighbor version 0, table
version 64, starting at 0.0.0.0
*Mar 15 01:41:23.515:BGP(0):10.0.0.5 send UPDATE (format) 0.0.0.0/0, next 10.0.0.2, metric
0, path , extended community Cost:igp:1:100
*Mar 15 01:41:23.515:BGP(0):10.0.0.5 send UPDATE (format) 10.2.2.0/24, next 10.20.20.10,
metric 0, path 10, extended community Cost:igp:8:22
*Mar 15 01:41:23.515:BGP(0):10.0.0.5 send UPDATE (format) 10.13.13.0/24, next 10.0.0.8,
metric 0, path
```

The following is sample output from the **debug ip bgp updates in**command. The output shows that the local router is receiving updates with the cost community:

```
Router# debug ip bgp updates in
*Jan  6 01:27:09.111:BGP(2):10.0.0.8 rcvd UPDATE w/ attr:nexthop 10.0.0.8, origin ?,
localpref 100, metric 0, path 10, extended community RT:100:1 Cost:igp:10:10
Cost:igp:11:11
```

debug ip bgp vpnv4 checkpoint

To display the events for the Virtual Routing and Forwarding (VRF) checkpointing system between the active and standby Route Processors, use the debug ip bgp vpnv4 checkpoint command in privileged EXEC mode. To disable the display of these events, use the **no** form of this command.

debug ip bgp vpnv4 checkpoint

no debug ip bgp vpnv4 checkpoint

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series router.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Examples

The following example shows command output on the active Route Processor:

```
Router# debug ip bgp vpnv4 checkpoint
3d18h: %HA-5-SYNC_NOTICE: Config sync started.
3d18h: vrf-nsf: vrf vpn2 tableid 1 send OK
3d18h: vrf-nsf: vrf tableid bulk sync complete msg send OK
3d18h: vrf-nsf: CF send ok
3d18h: vrf-nsf: CF send ok
3d18h: %HA-5-SYNC_NOTICE: Config sync completed.
3d18h: %HA-5-SYNC_NOTICE: Standby has restarted.
3d18h: %HA-5-MODE: Operating mode is sso, configured mode is sso.
```

Related Commands

Command	Description
debug ip bgp vpnv4 nsf	Displays the nonstop forwarding events for the VRF table-id synchronization subsystem between the active and standby route processors.

debug ip bgp vpnv4 nsf

To display the nonstop forwarding events for the VRF table-id synchronization subsystem between the active and standby Route Processors, use the debug ip bgp vpnv4 nsf command in privileged EXEC mode. To disable the display of these events, use the **no** form of this command.

debug ip bgp vpnv4 nsf

no debug ip bgp vpnv4 nsf

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(25)S	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series router.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Examples

The following example shows the command output on the active Route Processor:

```
Router# debug ip bgp vpnv4 nsf
MPLS VPN NSF Processing debugging is on
Router(config)# ip vrf vpn3
3d18h: vrf-nsf: vrf vpn3 tableid 2 send rpc OK
Router(config-vrf)# no ip vrf vpn3
% IP addresses from all interfaces in VRF vpn3 have been removed
3d18h: vrf-nsf:  rx vrf tableid delete complete msg, tid = 2, name = vpn3
```

The following example shows the command output on the standby Route Processor:

```
Router# debug ip bgp vpnv4 nsf
MPLS VPN NSF Processing debugging is on
00:05:21: vrf-nsf: rx vrf tableid rpc msg, tid = 2, name = vpn3
% IP addresses from all interfaces in VRF vpn3 have been removed
00:06:22: vrf-nsf: vrf vpn3 tableid 2 , delete complete, send OK
```

Related Commands

Command	Description
debug ip bgp vpnv4 checkpoint	Display the events for the VRF checkpointing system between the active and standby Route Processors.

debug ip bgp vpnv4 unicast

To display debugging messages for Virtual Private Network version 4 (VPNv4) unicast routes, use the **debug ip bgp vpnv4 unicast** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip bgp vpnv4 unicast {**checkpoint| csc| import| keepalives| labelmode| updates**}

no debug ip bgp vpnv4 unicast {**checkpoint| csc| import| keepalives| labelmode| updates**}

Syntax Description

checkpoint	Displays virtual routing and forwarding (VRF) nonstop forwarding (NSF) checkpoint messages and events.
csc	Displays VRF processing messages for a Carrier Supporting Carrier (CSC) VPN.
import	Displays VRF import processing messages.
keepalives	Displays Border Gateway Protocol (BGP) keepalives.
labelmode	Displays VRF label mode processing.
updates	Displays BGP updates processing for Unicast VPNv4 address family.

Command Default

Debugging of VPNv4 unicast routes is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(5)T	This command was introduced.
XE Release 2.2	The **labelmode** keyword was added.
12.2(33)SRD	This command was integrated into Cisco IOS Release 12.2(33)SRD.

Examples

The following example enables debugging of MPLS VPN label mode processing:

```
Router# debug ip bgp vpnv4 unicast labelmode
MPLS VPN Label mode processing debugging is on
```

```
Router# config terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
% This command is an unreleased and unsupported feature
Router(config)#
*Oct 18 11:35:01.159: vpn: changing the label mode (Enable: per-vrf) for all-vrfs
*Oct 18 11:35:01.459: vpn: label mode change, bnet walk complete.
*Oct 18 11:35:01.459: BGP: VPNv4 Unicast label mode changed
Router(config)#^Z
Router#
*Oct 18 11:35:21.995: %SYS-5-CONFIG_I: Configured from console by console
Router# show debug

Tag VPN:
  MPLS VPN Label mode processing debugging is on
Router#
```

Related Commands

Command	Description
show ip vrf detail	Displays assigned label mode for the VRF.

debug ip bgp vpnv6 unicast

To display debugging messages for Virtual Private Network version 6 (VPNv6) unicast routes, use the **debug ip bgp vpnv6 unicast** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip bgp vpnv6 unicast {**csc**| **import**| **keepalives**| **labelmode**| **topology**| **updates**}

no debug ip bgp vpnv6 unicast {**csc**| **import**| **keepalives**| **labelmode**| **topology**| **updates**}

Syntax Description

csc	Displays VPN routing and forwarding (VRF) processing messages for a Carrier Supporting Carrier (CSC) VPN.
import	Displays VRF import processing messages.
keepalives	Displays Border Gateway Protocol (BGP) keepalives.
labelmode	Displays VRF label mode processing.
topology	Displays the routing topology instance.
updates	Displays BGP updates processing for the unicast VPNv6 address family.

Command Default

Debugging of VPNv6 unicast routes is not enabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRD	This command was introduced.

Examples

The following example enables debugging of MPLS VPN label mode processing:

```
Router# debug ip bgp vpnv6 unicast labelmode
MPLS VPN Label mode processing debugging is on
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mpls label mode vrf vpn1 protocol bgp-vpnv6 per-vrf
% Command accepted but obsolete, unreleased or unsupported; see documentation.
Router(config)#
6d03h: vpn: changing the label mode (Enable: per-vrf) for vrf vpn1, address family ipv6
6d03h: vpn: setting pervrfaggr label 18 for vrf vpn1:2001:DB8:1:2::/96
```

```
6d03h: vpn: setting pervrfaggr label 18 for vrf vpn1:2001:DB8:2::1/128
6d03h: vpn: pervrfaggr, withdraw and free local label 19 for vpn1:2001:DB8:CE1::1/128
6d03h: vpn: setting pervrfaggr label 18 for vrf vpn1:2001:DB8:CE1::1/128
6d03h: vpn: label mode change, bnet walk complete.
6d03h: BGP: VPNv6 Unicast label mode changed
Router(config)# end
```

Related Commands

Command	Description
show vrf detail	Displays assigned label mode for the VRF.

debug ip casa affinities

To display debugging messages for affinities, use the **debug ip casa affinities**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip casa affinities

no debug ip casa affinities

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging for affinities is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debug ip casa affinities**command:

```
Router# debug ip casa affinities
16:15:36:Adding fixed affinity:
16:15:36:    10.10.1.1:54787 -> 10.10.10.10:23 proto = 6
16:15:36:Updating fixed affinity:
16:15:36:    10.10.1.1:54787 -> 10.10.10.10:23 proto = 6
16:15:36:    flags = 0x2, appl addr = 10.10.3.2, interest = 0x5/0x100
16:15:36:    int ip:port = 10.10.2.2:1638, sequence delta = 0/0/0/0
16:15:36:Adding fixed affinity:
16:15:36:    10.10.10.10:23 -> 10.10.1.1:54787 proto = 6
16:15:36:Updating fixed affinity:
16:15:36:    10.10.10.10:23 -> 10.10.1.1:54787 proto = 6
16:15:36:    flags = 0x2, appl addr = 0.0.0.0, interest = 0x3/0x104
16:15:36:    int ip:port = 10.10.2.2:1638, sequence delta = 0/0/0/0
```

The table below describes the significant fields shown in the display.

Table 5: debug ip casa affinities Field Descriptions

Field	Description
Adding fixed affinity	Adding a fixed affinity to affinity table.
Updating fixed affinity	Modifying a fixed affinity table with information from the services manager.

Field	Description
flags	Bit field indicating actions to be taken on this affinity.
fwd addr	Address to which packets will be directed.
interest	Services manager that is interested in packets for this affinity.
int ip:port	Services manager port to which interest packets are sent.
sequence delta	Used to adjust TCP sequence numbers for this affinity.

debug ip casa packets

To display debugging messages for packets, use the **debug ip casa packets**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip casa packets

no debug ip casa packets

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging for packets is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debug ip casa packets**command:

```
Router# debug ip casa packets
16:15:36:Routing CASA packet - TO_MGR:
16:15:36:    10.10.1.1:55299 -> 10.10.10.10:23 proto = 6
16:15:36:    Interest Addr:10.10.2.2   Port:1638
16:15:36:Routing CASA packet - FWD_PKT:
16:15:36:    10.10.1.1:55299 -> 10.10.10.10:23 proto = 6
16:15:36:    Fwd Addr:10.10.3.2
16:15:36:Routing CASA packet - TO_MGR:
16:15:36:    10.10.10.10:23 -> 10.10.1.1:55299 proto = 6
16:15:36:    Interest Addr:10.10.2.2   Port:1638
16:15:36:Routing CASA packet - FWD_PKT:
16:15:36:    10.10.10.10:23 -> 10.10.1.1:55299 proto = 6
16:15:36:    Fwd Addr:0.0.0.0
16:15:36:Routing CASA packet - TICKLE:
16:15:36:    10.10.10.10:23 -> 10.10.1.1:55299 proto = 6
16:15:36:    Interest Addr:10.10.2.2   Port:1638  Interest Mask:SYN
16:15:36:    Fwd Addr:0.0.0.0
16:15:36:Routing CASA packet - FWD_PKT:
16:15:36:    10.10.1.1:55299 -> 10.10.10.10:23 proto = 6
16:15:36:    Fwd Addr:10.10.3.2
```

The table below describes the significant fields shown in the display.

Table 6: debug ip casa packets Field Descriptions

Field	Description
Routing CASA packet - TO_MGR	Forwarding Agent is routing a packet to the services manager.
Routing CASA packet - FWD_PKT	Forwarding Agent is routing a packet to the forwarding address.
Routing CASA packet - TICKLE	Forwarding Agent is signaling services manager while allowing the packet in question to take the appropriate action.
Interest Addr	Services manager address.
Interest Port	Port on the services manager where packet is sent.
Fwd Addr	Address to which packets matching the affinity are sent.
Interest Mask	Services manager that is interested in packets for this affinity.

debug ip casa wildcards

To display debugging messages for wildcards, use the **debug ip casa wildcards**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip casa wildcards

no debug ip casa wildcards

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging for wildcards is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debug ip casa wildcards**command:

```
Router# debug ip casa wildcards
16:13:23:Updating wildcard affinity:
16:13:23:    10.10.10.10:0 -> 0.0.0.0:0 proto = 6
16:13:23:    src mask = 255.255.255.255, dest mask = 0.0.0.0
16:13:23:    no frag, not advertising
16:13:23:    flags = 0x0, appl addr = 0.0.0.0, interest = 0x8107/0x8104
16:13:23:    int ip:port = 10.10.2.2:1638, sequence delta = 0/0/0/0
16:13:23:Updating wildcard affinity:
16:13:23:    0.0.0.0:0 -> 10.10.10.10:0 proto = 6
16:13:23:    src mask = 0.0.0.0, dest mask = 255.255.255.255
16:13:23:    no frag, advertising
16:13:23:    flags = 0x0, appl addr = 0.0.0.0, interest = 0x8107/0x8102
16:13:23    int ip:port = 10.10.2.2:1638, sequence delta = 0/0/0/0
```

The table below describes the significant fields shown in the display.

Table 7: debug ip casa wildcards Field Descriptions

Field	Description
src mask	Source of connection.
dest mask	Destination of connection.
no frag, not advertising	Not accepting IP fragments.

Field	Description
flags	Bit field indicating actions to be taken on this affinity.
fwd addr	Address to which packets matching the affinity will be directed.
interest	Services manager that is interested in packets for this affinity.
int ip: port	Services manager port to which interest packets are sent.
sequence delta	Used to adjust sequence numbers for this affinity.

debug ip cef

To troubleshoot various Cisco Express Forwarding events, use the **debug ip cef** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip cef {**drops** [**rpf** [*access-list*]] [*access-list*]| **receive** [*access-list*]| **events** [*access-list*]| **interface**| **dialer**}

no debug ip cef {**drops** [**rpf** [*access-list*]] [*access-list*]| **receive** [*access-list*]| **events** [*access-list*]| **interface**| **dialer**}

Specific to Interprocess Communication (IPC) Records

debug ip cef {**ipc**| **interface-ipc**| **prefix-ipc** [*access-list*]}

no debug ip cef {**ipc**| **interface-ipc**| **prefix-ipc** [*access-list*]}

Cisco 10000 Series Routers Only

debug ip cef {**drops** [**rpf** [*access-list*]] [*access-list*]| **receive** [*access-list*]| **events** [*access-list*]}

no debug ip cef {**drops** [**rpf** [*access-list*]] [*access-list*]| **receive** [*access-list*]| **events** [*access-list*]}

Cisco 10000 Series Routers Only--Specific to IPC Records

debug ip cef ipc

no debug ip cef ipc

Syntax Description

drops	Records dropped packets.
rpf	(Optional) Records the result of the Reverse Path Forwarding (RPF) check for packets.
access-list	(Optional) Limits debugging collection to packets that match the list.
receive	Records packets that are ultimately destined to the router and packets destined to a tunnel endpoint on the router. If the decapsulated tunnel is IP, the packets are Cisco Express Forwarding switched; otherwise the packets are process switched.
events	Records general Cisco Express Forwarding events.
interface	Records IP Cisco Express Forwarding interface events.
dialer	Records IP Cisco Express Forwarding interface events for dialer interfaces.

ipc	Records information related to IPC in Cisco Express Forwarding. Possible types of events are the following: • IPC messages received out of sequence • Status of resequenced messages • Status of buffer space for IPC messages • Transmission status of IPC messages • Throttle requests sent from a line card to the Route Processor
interface-ipc	Records IPC updates related to interfaces. Possible reporting includes an interface coming up or going down and updates to fibhwidb and fibidb.
prefix-ipc	Records updates related to IP prefix information. Possible updates include the following: • Debugging of IP routing updates in a line card • Reloading of a line card with a new table • Updates related to exceeding the maximum number of routes • Control messages related to Forwarding Information Base (FIB) table prefixes

Command Default

This command is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.2GS	This command was introduced.
11.1CC	Support for multiple platforms was added.
12.0(5)T	The **rpf** keyword was added.
12.2(4)T	The **dialer** keyword was added.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.

Release	Modification
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

This command gathers additional information for the handling of Cisco Express Forwarding interface, IPC, or packet events.

Note For packet events, we recommend that you use an access control list (ACL) to limit the messages recorded.

Examples

The following is sample output from the **debug ip cef rpf**commandfor a packet that is dropped when it fails the RPF check. IP address 172.17.249.252 is the source address, and Ethernet 2/0/0 is the input interface.

```
Router# debug ip cef drops rpf
IP CEF drops for RPF debugging is on
00:42:02:CEF-Drop:Packet from 172.17.249.252 via Ethernet2/0/0 -- unicast rpf check
```

The following is sample output for Cisco Express Forwarding packets that are not switched using information from the FIB table but are received and sent to the next switching layer:

```
Router# debug ip cef receive
IP CEF received packets debugging is on
00:47:52:CEF-receive:Receive packet for 10.1.104.13
```

The table below describes the significant fields shown in the display.

Table 8: debug ip cef receive Field Descriptions

Field	Description
CEF-Drop:Packet from 172.17.249.252 via Ethernet2/0/0 -- unicast rpf check	A packet from IP address 172.17.249.252 is dropped because it failed the RPF check.
CEF-receive:Receive packet for 10.1.104.13	Cisco Express Forwarding has received a packet addressed to the router.

The following is sample output from the **debug ip cef dialer**commandfor a legacy dialer:

```
Router# debug ip cef dialer
00:19:50:CEF-Dialer (legacy):add link to 10.10.10.2 via Dialer1 through BRI0/0:1
00:19:50:CEF-Dialer:adjacency added:0x81164850
00:19:50:CEF-Dialer:adjacency found:0x81164850; fib->count:1
00:19:50:CEF-Dialer:setup loadinfo with 1 paths
```

The following is sample output from the **debug ip cef dialer** command for a dialer profile:

```
Router# debug ip cef dialer
00:31:44:CEF-Dialer (profile dynamic encap (not MLP)):add link to 10.10.10.2 via Dialer1
through Dialer1
00:31:44:CEF-Dialer:adjacency added:0x81164850
00:31:44:CEF-Dialer:adjacency found:0x81164850; fib->count:1
```

The table below describes the significant fields shown in the display.

Table 9: debug ip cef dialer Field Descriptions

Field	Description
CEF-Dialer (legacy):add link to 10.10.10.2 via Dialer1 through BRI0/0:1	A link was added to IP address 10.10.10.2 for legacy Dialer1 through physical interface BRI0/0:1.
CEF-Dialer (profile dynamic encap (not MLP)):add link to 10.10.10.2 via Dialer1 through Dialer1	A link was added to IP address 10.10.10.2 for dialer profile Dialer1 through Dialer1.

Related Commands

Command	Description
ip cef	Enables Cisco Express Forwarding on the RPC card.
show ip cef	Displays entries in the FIB or displays a summary of the FIB.

debug ip cef accounting non-recursive

To troubleshoot Cisco Express Forwarding accounting records, use the **debug ip cef accounting non-recursive**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip cef accounting non-recursive

no debug ip cef accounting non-recursive

Syntax Description

This command has no arguments or keywords.

Command Default

This command is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.1CC	This command was introduced.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

This command records accounting events for nonrecursive prefixes when the **ip cef accounting non-recursive** command is enabled in global configuration mode.

Examples

The following is sample output from the **debug ip cef accounting non-recursive** command:

```
Router# debug ip cef accounting non-recursive
03:50:19:CEF-Acct:tmstats_binary:Beginning generation of tmstats
ephemeral file (mode binary)
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF2000
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF1EA0
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF17C0
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF1D40
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF1A80
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF0740
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF08A0
```

```
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF0B60
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF0CC0
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF0F80
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF10E0
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF1240
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF13A0
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF1500
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF1920
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF0E20
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF1660
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF05E0
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF0A00
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF1BE0
03:50:19:CEF-Acct:snapshoting loadinfo 0x63FF0480
03:50:19:CEF-Acct:tmstats_binary:aggregation complete, duration 0 seconds
03:50:21:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:24:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:24:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:27:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:29:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:32:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:35:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:38:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:41:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:45:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:48:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:49:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:52:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:55:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:57:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:57:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:57:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:57:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:57:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:57:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:57:CEF-Acct:tmstats_binary:writing 45 bytes
03:50:57:CEF-Acct:tmstats_binary:tmstats file written, status 0
```

The table below describes the significant fields shown in the display.

Table 10: debug ip cef accounting non-recursive Field Descriptions

Field	Description
Beginning generation of tmstats ephemeral file (mode binary)	Tmstats file is being created.
CEF-Acct:snapshoting loadinfo 0x63FF2000	Baseline counters are being written to the tmstats file for each nonrecursive prefix.
CEF-Acct:tmstats_binary:aggregation complete, duration 0 seconds	Tmstats file creation is complete.
CEF-Acct:tmstats_binary:writing 45 bytes	Nonrecursive accounting statistics are being updated to the tmstats file.
CEF-Acct:tmstats_binary:tmstats file written, status 0	Update of the tmstats file is complete.

Related Commands

Command	Description
debug ip cef	Troubleshoots various Cisco Express Forwarding events.
ip cef accounting	Enables Cisco Express Forwarding network accounting.
show ip cef	Displays entries or a summary of the FIB table.

debug ip cef fragmentation

To report fragmented IP packets when Cisco Express Forwarding is enabled, use the **debug ip cef fragmentation**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command:

debug ip cef fragmentation

no debug ip cef fragmentation

Syntax Description

This command has no arguments or keywords.

Command Default

This command is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(14)S	This command was introduced.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

This command is used to troubleshoot fragmentation problems when Cisco Express Forwarding switching is enabled.

Examples

The following is sample output from the **debug ip cef fragmentation**command:

```
Router# debug ip cef fragmentation
00:59:45:CEF-FRAG:no_fixup path:network_start 0x5397CF8E datagramstart 0x5397CF80 data_start
 0x397CF80 data_block 0x397CF40 mtu 1000 datagramsize 1414 data_bytes 1414
00:59:45:CEF-FRAG:send frag:datagramstart 0x397CF80 datagramsize 442 data_bytes 442
00:59:45:CEF-FRAG:send frag:datagramstart 0x38BC266 datagramsize 1006 data_bytes 1006
00:59:45:CEF-FRAG:no_fixup path:network_start 0x5397C60E datagramstart 0x5397C600 data_start
 0x397C600 data_block 0x397C5C0 mtu 1000 datagramsize 1414 data_bytes 1414
```

```
00:59:45:CEF-FRAG:send frag:datagramstart 0x397C600 datagramsize 442 data_bytes 442
00:59:45:CEF-FRAG:send frag:datagramstart 0x38BC266 datagramsize 1006 data_bytes 1006
```

The table below describes the significant fields shown in the display.

Table 11: debug ip cef fragmentation Field Descriptions

Field	Description
no_fixup path	A packet is being fragmented in the no_fixup path.
network_start 0x5397CF8E	Memory address of the IP packet.
datagramstart 0x5397CF80	Memory address of the encapsulated IP packet.
data_start 0x397CF80	For particle systems, the memory address where data starts for the first packet particle.
data_block 0x397C5C0	For particle systems, the memory address of the first packet particle data block.
mtu 1000	Maximum transmission unit of the output interface.
datagramsize 1414	Size of the encapsulated IP packet.
data_bytes 1414	For particle systems, the sum of the particle data bytes that make up the packet.
send frag	Fragment is being forwarded.

Related Commands

Command	Description
debug ip cef	Troubleshoots various Cisco Express Forwarding events.

debug ip cef hash

To record Cisco Express Forwarding load sharing hash algorithm events, use the **debug ip cef hash**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip cef hash

no debug ip cef hash

Syntax Description

This command has no arguments or keywords.

Command Default

This command is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0(12)S	This command was introduced.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB. This command is not supported on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.This command is not supported on the Cisco 7600 router.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

Use this command when changing the load sharing algorithm to display the hash table details.

Examples

The following is sample output from the **debug ip cef hash**command with IP Cisco Express Forwarding load algorithm tunnel information:

```
Router# debug ip cef hash
01:15:06:%CEF:ip cef load-sharing algorithm tunnel 0
01:15:06:%CEF:Load balancing algorithm:tunnel
01:15:06:%CEF:Load balancing unique id:1F2BA5F6
01:15:06:%CEF:Destroyed load sharing hash table
01:15:06:%CEF:Sending hash algorithm id 2, unique id 1F2BA5F6 to slot 255
```

The following lines show IP Cisco Express Forwarding load algorithm universal information:

```
01:15:28:%CEF:ip cef load-sharing algorithm universal 0
01:15:28:%CEF:Load balancing algorithm:universal
01:15:28:%CEF:Load balancing unique id:062063A4
01:15:28:%CEF:Creating load sharing hash table
01:15:28:%CEF:Hash table columns for valid max_index:
01:15:28:12: 9  7  7 4  4 10  0  7 10  4  5  0  4  7  8  4
01:15:28:15: 3 10 10  4 10  4  0  7  1  7 14  6 13 13 11 13
01:15:28:16: 1  3  7 12  4 14  8  7 10  4  1 12  8 15  4  8
01:15:28:%CEF:Sending hash algorithm id 3, unique id 062063A4 to slot 255
```

The table below describes the significant fields shown in the display.

Table 12: debug ip cef hash Field Descriptions

Field	Description
ip cef load-sharing algorithm tunnel 0	Echo of the user command.
Load balancing algorithm:tunnel	Load sharing algorithm is set to tunnel.
Load balancing unique id:1F2BA5F6	ID field in the command is usually 0. In this instance, the router chose a pseudo random ID of 1F2BA5F6.
Destroyed load sharing hash table	Purge the existing hash table.
Sending hash algorithm id 2, unique id 1F2BA5F6 to slot 255	Algorithm is being distributed.
Creating load sharing hash table	Hash table is being created.
Hash table columns for valid max_index:	Generated hash table.

Related Commands

Command	Description
debug ip cef	Troubleshoots various Cisco Express Forwarding events.
debug ip cef rrhash	Records Cisco Express Forwarding removal of receive hash events.

debug ip cef rrhash

To record Cisco Express Forwarding removal of receive hash events, use the **debug ip cef rrhash**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip cef rrhash

no debug ip cef rrhash

Syntax Description

This command has no arguments or keywords.

Command Default

This command is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(2)T	This command was introduced.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB. This command is not supported on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA. This command is not supported on the Cisco 7600 routers.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

Use this command to verify the removal of receive hash events when you are shutting down or deleting an interface.

Examples

The following is sample output from the **debug ip cef rrhash**command:

```
Router# debug ip cef rrhash
00:27:15:CEF:rrhash/check:found 10.1.104.7 on down idb [ok to delete]
00:27:15:CEF:rrhash/check:found 10.1.104.0 on down idb [ok to delete]
00:27:15:CEF:rrhash/check:found 10.1.104.255 on down idb [ok to delete]
00:27:15:CEF:rrhash/check:found 10.1.104.7 on down idb [ok to delete]
00:27:15:CEF:rrhash/check:found 10.1.104.7 on down idb [ok to delete]
00:27:15:CEF:rrhash/check:found 10.1.104.0 on down idb [ok to delete]
00:27:15:CEF:rrhash/check:found 10.1.104.255 on down idb [ok to delete]
00:27:15:CEF:rrhash/check:found 10.1.104.7 on down idb [ok to delete]
```

The table below describes the significant fields shown in the display.

Table 13: debug ip cef rrhash Field Descriptions

Field	Description
rrhash/check	Verify address is on the receive list.
found 10.1.104.7 on down idb [ok to delete]	Found a valid address on the receive list for a shutdown interface that can be deleted.

Related Commands

Command	Description
debug ip cef	Troubleshoots various Cisco Express Forwarding events.
debug ip cef hash	Records Cisco Express Forwarding removal of receive hash events.

debug ip cef subblock

To troubleshoot Cisco Express Forwarding subblock events, use the **debug ip cef subblock**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip cef subblock [**id** {**all**| **hw** *hw-id*| **sw** *sw-id*}] [**xdr** {**all**| **control**| **event**| **none**| **statistic**}]

no debug ip cef subblock [**id** {**all**| **hw** *hw-id*| **sw** *sw-id*}] [**xdr** {**all**| **control**| **event**| **none**| **statistic**}]

Syntax Description

id	(Optional) Subblock types.
all	(Optional) All subblock types.
hw *hw-id*	(Optional) Hardware subblock and identifier.
sw *sw-id*	(Optional) Software subblock and identifier.
xdr	(Optional) Exernal Data Representation (XDR) message types.
control	(Optional) All XDR message types.
event	(Optional) Event XDR messages only.
none	(Optional) No XDR messages.
statistic	(Optional) Statistic XDR messages.

Command Default

This command is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.0S	This command was introduced.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Release	Modification
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

This command is used to record Cisco Express Forwarding subblock messages and events.

Examples

The following is sample output from the **debug ip cef subblock**command:

```
Router# debug ip cef subblock
00:28:12:CEF-SB:Creating unicast RPF subblock for FastEthernet6/0
00:28:12:CEF-SB:Linked unicast RPF subblock to FastEthernet6/0.
00:28:12:CEF-SB:Encoded unit of unicast RPF data (length 16) for FastEthernet6/0
00:28:12:CEF-SB:Sent 1 data unit to slot 6 in 1 XDR message
```

Examples

The following is sample output from the **debug ip cef subblock**command:

```
Router# debug ip cef subblock
00:28:12:CEF-SB:Creating unicast RPF subblock for FastEthernet6/0/0
00:28:12:CEF-SB:Linked unicast RPF subblock to FastEthernet6/0/0.
00:28:12:CEF-SB:Encoded unit of unicast RPF data (length 16) for FastEthernet6/0/0
00:28:12:CEF-SB:Sent 1 data unit to slot 6 in 1 XDR message
```

The table below describes the significant fields shown in the display.

Table 14: debug ip cef subblock Field Descriptions

Field	Description
Creating unicast RPF subblock for FastEthernet6/0/0	Creating an Unicast Reverse Path Forwarding (Unicast RPF) interface descriptor subblock.
Linked unicast RPF subblock to FastEthernet6/0/0	Linked the subblock to the specified interface.
Encoded unit of unicast RPF data (length 16) for FastEthernet6/0/0	Encoded the subblock information in an XDR.
Sent 1 data unit to slot 6 in 1 XDR message	Sent the XDR message to a line card through the IPC.

Related Commands

Command	Description
debug ip cef	Troubleshoots various Cisco Express Forwarding events.

debug ip cef table

To enable the collection of events that affect entries in the Cisco Express Forwarding tables, use the **debug ip cef table**command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip cef table [*access-list*| **consistency-checkers**]

no debug ip cef table [*access-list*| **consistency-checkers**]

Syntax Description

access-list	(Optional) Controls collection of consistency checker parameters from specified lists.
consistency-checkers	(Optional) Sets consistency checking characteristics.

Command Default

This command is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
11.2GS	This command was introduced.
11.1CC	Support was added for multiple platforms.
12.0(15)S	The **consistency-checkers** keyword was added.
12.2(2)T	This command was integrated into Cisco IOS Release 12.2(2)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB and implemented on the Cisco 10000 series routers.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

This command is used to record Cisco Express Forwarding table events related to the Forwarding Information Base (FIB) table. Possible types of events include the following:

- Routing updates that populate the FIB table
- Flushing of the FIB table
- Adding or removing of entries to the FIB table
- Table reloading process

Examples

The following is sample output from the **debug ip cef table** command:

```
Router# debug ip cef table
01:25:46:CEF-Table:Event up, 10.1.1.1/32 (rdbs:1, flags:1000000)
01:25:46:CEF-IP:Checking dependencies of 0.0.0.0/0
01:25:47:CEF-Table:attempting to resolve 10.1.1.1/32
01:25:47:CEF-IP:resolved 10.1.1.1/32 via 10.1.104.1 to 10.1.104.1 Ethernet2/0/0
01:26:02:CEF-Table:Event up, default, 0.0.0.0/0 (rdbs:1, flags:400001)
01:26:02:CEF-IP:Prefix exists - no-op change
```

Examples

The following is sample output from the **debug ip cef table** command:

```
Router# debug ip cef table
01:25:46:CEF-Table:Event up, 10.1.1.1/32 (rdbs:1, flags:1000000)
01:25:46:CEF-IP:Checking dependencies of 0.0.0.0/0
01:25:47:CEF-Table:attempting to resolve 10.1.1.1/32
01:25:47:CEF-IP:resolved 10.1.1.1/32 via 10.1.104.1 to 10.1.104.1  GigabitEthernet2/0/0
01:26:02:CEF-Table:Event up, default, 0.0.0.0/0 (rdbs:1, flags:400001)
01:26:02:CEF-IP:Prefix exists - no-op change
```

The table below describes the significant fields shown in the display.

Table 15: debug ip cef table Field Descriptions

Field	Description
CEF-Table	Indicates a table event.
Event up, 10.1.1.1/32	IP prefix 10.1.1.1/32 is being added.
rdbs:1	Event is from routing descriptor block 1.
flags:1000000	Indicates the network descriptor block flags.
CEF-IP	Indicates a Cisco Express Forwarding IP event.
Checking dependencies of 0.0.0.0/0	Resolves the next hop dependencies for 0.0.0.0/0.
attempting to resolve 10.1.1.1/32	Resolves the next hop dependencies.
resolved 10.1.1.1/32 via 10.1.104.1 to 10.1.104.1 Ethernet2/0/0	Next hop to IP prefix 10.1.1.1/32 is set and is added to the table.
Event up, default, 0.0.0.0/0 Prefix exists - no-op change	Indicates no table change is necessary for 0.0.0.0/32.

Related Commands

Command	Description
cef table consistency-check	Enables Cisco Express Forwarding consistency checker table values by type and parameter.
clear cef table	Clears the Cisco Express Forwarding tables.
clear ip cef inconsistency	Clears Cisco Express Forwarding inconsistency statistics and records found by the Cisco Express Forwarding consistency checkers.
debug cef	Enables the display of information about Cisco Express Forwarding events.
debug ip cef	Troubleshoots various Cisco Express Forwarding events.
show cef table consistency-check	Displays Cisco Express Forwarding consistency checker table values.
show ip cef inconsistency	Displays Cisco Express Forwarding IP prefix inconsistencies.

debug ip ddns update

To enable debugging for Dynamic Domain Name System (DDNS) updates, use the **debug ip ddns update**command in privileged EXEC mode. To disable the debugging, use the **no** form of this command.

debug ip ddns update

no debug ip ddns update

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(8)YA	This command was introduced.
12.3(14)T	This command was integrated into Cisco IOS Release 12.3(14)T.

Examples

Use the **debug ip ddns update** command to verify that your configurations are working properly. The following sample configurations are shown for demonstration of possible debug output that could display for each configuration.

Examples

The following scenario has a client configured for IETF DDNS updating of address (A) Resource Records (RRs) during which a Dynamic Host Configuration Protocol (DHCP) server is expected to update the pointer (PTR) RR. The DHCP client discovers the domain name system (DNS) server to update using an Start of Authority (SOA) RR lookup since the IP address to the server to update is not specified. The DHCP client is configured to include an fully qualified domain name (FQDN) DHCP option and notifies the DHCP server that it will be updating the A RRs.

```
!DHCP Client Configuration
ip ddns update method testing
 ddns
interface Ethernet1
 ip dhcp client update dns
 ip ddns update testing
 ip address dhcp
end
!DHCP Server Configuration
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Debug Output Enabled
Router# debug ip ddns update
00:14:39: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.4, mask
 255.0.0.0, hostname canada_reserved
00:14:39: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.4
00:14:39: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:14:42: DHCPC: Server performed PTR update
```

```
00:14:42: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.4
00:14:42: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:14:42: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:14:42: DDNS:   Zone = hacks
00:14:42: DDNS:   Prerequisite: canada_reserved.hacks not in use
00:14:42: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.4
00:14:42: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:14:42: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.4 finished
00:14:42: DYNDNSUPD: Another update completed (total outstanding=0)
```

Examples

The following scenario has the client configured for IETF DDNS updating of both A and DNS RRs and requesting that the DHCP server update neither. The DHCP client discovers the DNS server to update using an SOA RR lookup since the IP address to the server to update is not specified. The DHCP client is configured to include an FQDN DHCP option that instructs the DHCP server to not update either A or PTR RRs.

```
!DHCP Client Configuration
ip dhcp-client update dns server none
ip ddns update method testing
 ddns both
interface Ethernet1
 ip ddns update testing
 ip address dhcp
end
!DHCP Server Configuration
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Debug Output Enabled
Router# debug ip ddns update
00:15:33: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.5, mask
 255.0.0.0, hostname canada_reserved
00:15:33: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.5
00:15:33: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:15:36: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.5
00:15:36: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:15:36: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:15:36: DDNS:   Zone = 10.in-addr.arpa
00:15:36: DDNS:   Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:15:36: DDNS:   Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:15:36: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:15:36: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:15:36: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:15:36: DDNS:   Zone = hacks
00:15:36: DDNS:   Prerequisite: canada_reserved.hacks not in use
00:15:36: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.5
00:15:36: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:15:36: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.5 finished
00:15:36: DYNDNSUPD: Another update completed (total outstanding=0)
```

Examples

The following scenario has the client configured for IETF DDNS updating of both A and DNS RRs and requesting that the DHCP server update neither. The DHCP client explicitly specifies the server to update. The DHCP client is configured to include an FQDN DHCP option that instructs the DHCP server not to update either A or PTR RRs. The configuration is performed using the **ip dhcp client update dns** command. The DHCP server is configured to override the client request and update both A and PTR RR anyway.

```
!DHCP Client Configuration
ip dhcp client update dns server none
ip ddns update method testing
 ddns both
interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing
 ip address dhcp
```

```
end
!DHCP Server Configuration
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns both override
!Debug Output Enabled on DHCP Client
Router# debug ip ddns update
00:16:30: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.6, mask
 255.0.0.0, hostname canada_reserved
00:16:30: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.6
00:16:30: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:16:33: DHCPC: Server performed both updates
```

Examples

The following scenario has the client configured for IETF DDNS updating of both A and DNS RRs and requesting the DHCP server to update neither. The DHCP client is configured to include an FQDN DHCP option which instructs the DHCP server not to update either A or PTR RRs. The DHCP server is configured to allow the client to update whatever RR it chooses.

```
!DHCP Client Configuration
ip dhcp client update dns server non
ip ddns update method testing
 ddns both
interface Ethernet1
 ip dhcp client update dns server none
 ip ddns update testing host 172.19.192.32
 ip address dhcp
end
!DHCP Server Configuration
ip dhcp pool test
 network 10.0.0.0 255.0.0.0
 update dns
!Debug Output Enabled on DHCP Client
Router# debug ip ddns update
00:17:52: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.0.0.7, mask
 255.0.0.0, hostname canada_reserved
00:17:52: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.6
00:17:52: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:17:55: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.7
00:17:55: DYNDNSUPD: Adding DNS mapping for canada_reserved.hacks <=> 10.0.0.7 server
10.19.192.32
00:17:55: DDNS: Enqueuing new DDNS update 'canada_reserved.hacks' <=> 10.0.0.7 server
10.19.192.32
00:17:55: DDNS: Zone name for '7.0.0.11.in-addr.arpa.' is '11.in-addr.arpa'
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS:   Zone = 11.in-addr.arpa
00:17:55: DDNS:   Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:17:55: DDNS:   Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Zone name for '10.0.0.11.in-addr.arpa.' is '10.in-addr.arpa'
00:17:55: DDNS: Using server 10.19.192.32
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS:   Zone = 11.in-addr.arpa
00:17:55: DDNS:   Prerequisite: 10.0.0.11.in-addr.arpa. not in use
00:17:55: DDNS:   Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Dynamic DNS Update 1 (PTR) for host canada_reserved.hacks returned 6
(YXDOMAIN)
00:17:55: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:17:55: DDNS:   Zone = 11.in-addr.arpa
00:17:55: DDNS:   Update: delete 10.0.0.11.in-addr.arpa. all PTR RRs
00:17:55: DDNS:   Update: add 10.0.0.11.in-addr.arpa. IN PTR canada_reserved.hacks
00:17:55: DDNS: Dynamic DNS Update 2 (PTR) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS:   Zone = hacks
00:17:55: DDNS:   Prerequisite: canada_reserved.hacks not in use
00:17:55: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 0 (NOERROR)
```

```
00:17:55: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.7 finished
00:17:55: DYNDNSUPD: Another update completed (total outstanding=1)
00:17:55: DDNS: Zone name for 'canada_reserved.hacks' is 'hacks'
00:17:55: DDNS: Using server 10.19.192.32
00:17:55: DDNS: Dynamic Update 1: (sending to server 10.19.192.32)
00:17:55: DDNS:   Zone = hacks
00:17:55: DDNS:   Prerequisite: canada_reserved.hacks not in use
00:17:55: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 1 (A) for host canada_reserved.hacks returned 6 (YXDOMAIN)
00:17:55: DDNS: Dynamic Update 2: (sending to server 10.19.192.32)
00:17:55: DDNS:   Zone = hacks
00:17:55: DDNS:   Update: delete canada_reserved.hacks all A RRs
00:17:55: DDNS:   Update: add canada_reserved.hacks IN A 10.0.0.7
00:17:55: DDNS: Dynamic DNS Update 2 (A) for host canada_reserved.hacks returned 0 (NOERROR)
00:17:55: DDNS: Update of 'canada_reserved.hacks' <=> 10.0.0.7 finished
00:17:55: DYNDNSUPD: Another update completed (total outstanding=0)
```

Examples

In the following scenario, the debug output displays the internal host table updates when the default domain name is hacks. The update method named test specifies that the internal Cisco IOS software host table should be updated. Configuring the update method as “test” should be used when the address on the Ethernet interface 0/0 changes. The hostname is configured for the update on this interface.

```
!Cisco IOS Software Configuration
ip domain name hacks
ip ddns update method test
 internal
interface ethernet0/0
 ip ddns update test hostname test2
 ip addr dhcp
!Debug Output Enabled
Router# debug ip ddns update
*Jun 4 03:11:10.591: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet0/0 assigned DHCP address
10.0.0.5, mask 255.0.0.0, hostname test2
*Jun 4 03:11:10.591: DYNDNSUPD: Adding DNS mapping for test2.hacks <=> 10.0.0.5
*Jun 4 03:11:10.591: DYNDNSUPD: Adding internal mapping test2.hacks <=> 10.0.0.5
```

Using the **show hosts** command displays the newly added host table entry.

```
Router# show hosts
Default domain is hacks
Name/address lookup uses domain service
Name servers are 255.255.255.255
Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined
Host                      Port Flags      Age Type   Address(es)
test2.hacks               None (perm, OK) 0   IP    10.0.0.5
```

Shutting down the interface removes the host table entry.

```
interface ethernet0/0
 shutdown
*Jun 4 03:14:02.107: DYNDNSUPD: Removing DNS mapping for test2.hacks <=> 10.0.0.5
*Jun 4 03:14:02.107: DYNDNSUPD: Removing mapping test2.hacks <=> 10.0.0.5
```

Using the **show hosts** command confirms that the entry has been removed.

```
Router# show hosts
Default domain is hacks
Name/address lookup uses domain service
Name servers are 255.255.255.255
Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined
Host                      Port Flags      Age Type   Address(es)
```

Examples

In the following scenario, the debug output shows the HTTP-style DDNS updates. The sample configuration defines a new IP DDNS update method named dyndns that configures a URL to use when adding or changing an address. No URL has been defined for use when removing an address since DynDNS.org does not use such a URL for free accounts. A maximum update interval of 28 days has been configured, which specifies that updates should be sent at least every 28 days. Configuring the new “dyndns” update method should be used for Ethernet interface 1.

```
!DHCP Client Configuration
ip ddns update method dyndns
 http
    add http://test:test@<s>/nic/update?system=dyndns&hostname=<h>&myip=<a>
    interval max 28 0 0 0
interface ethernet1
 ip ddns update hostname test.dyndns.org
 ip ddns update dyndns host members.dyndns.org
 ip addr dhcp
!Debugging Enabled
Router# debug ip ddns update
00:04:35: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet1 assigned DHCP address 10.32.254.187,
 mask 255.255.255.240, hostname test.dyndns.org
00:04:35: DYNDNSUPD: Adding DNS mapping for test.dyndns.org <=> 10.32.254.187 server
63.208.196.94
00:04:35: DYNDNSUPD: Sleeping for 3 seconds waiting for interface Ethernet1 configuration
to settle
00:04:38: HTTPDNS: Update add called for test.dyndns.org <=> 10.32.254.187
00:04:38: HTTPDNS: Update called for test.dyndns.org <=> 10.32.254.187
00:04:38: HTTPDNS: init
00:04:38: HTTPDNSUPD: Session ID = 0x7
00:04:38: HTTPDNSUPD: URL =
'http://test:test@63.208.196.94/nic/update?system=dyndns&hostname=test.dyndns.org&myip=10.32.254.187'
00:04:38: HTTPDNSUPD: Sending request
00:04:40: HTTPDNSUPD: Response for update test.dyndns.org <=> 10.32.254.187
00:04:40: HTTPDNSUPD: DATA START
good 10.32.254.187
00:04:40: HTTPDNSUPD: DATA END, Status is Response data received, successfully
00:04:40: HTTPDNSUPD: Call returned SUCCESS for update test.dyndns.org <=> 10.32.254.187
00:04:40: HTTPDNSUPD: Freeing response
00:04:40: DYNDNSUPD: Another update completed (outstanding=0, total=0)
00:04:40: HTTPDNSUPD: Clearing all session 7 info
!28 days later, the automatic update happens.
00:05:39: DYNDNSUPD: Adding DNS mapping for test.dyndns.org <=> 10.32.254.187 server
63.208.196.94
00:05:39: HTTPDNS: Update add called for test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNS: Update called for test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNS: init
00:05:39: HTTPDNSUPD: Session ID = 0x8
00:05:39: HTTPDNSUPD: URL =
'http://test:test@63.208.196.94/nic/update?system=dyndns&hostname=test.dyndns.org&myip=10.32.254.187'
00:05:39: HTTPDNSUPD: Sending request
00:05:39: HTTPDNSUPD: Response for update test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNSUPD: DATA START
nochg 10.32.254.187
00:05:39: HTTPDNSUPD: DATA END, Status is Response data received, successfully
00:05:39: HTTPDNSUPD: Call returned SUCCESS for update test.dyndns.org <=> 10.32.254.187
00:05:39: HTTPDNSUPD: Freeing response
00:05:39: DYNDNSUPD: Another update completed (outstanding=0, total=0)
00:05:39: HTTPDNSUPD: Clearing all session 8 info
```

The table below describes the significant fields shown in the output.

Table 16: debug ip ddns update Field Descriptions

Field	Description
HTTPDNSUPD	Reflects the method of update. In this case, the update method is HTTP.
HTTPDNSUPD: URL =	URL that is used to update the DNS.

Related Commands

Command	Description
debug dhcp	Displays debugging information about the DHCP client and monitors the status of DHCP packets.
debug ip dhcp server	Enables DHCP server debugging.
host (host-list)	Specifies a list of hosts that will receive DDNS updates of A and PTR RRs.
ip ddns update hostname	Enables a host to be used for DDNS updates of A and PTR RRs.
ip ddns update method	Specifies a method of DDNS updates of A and PTR RRs and the maximum interval between the updates.
ip dhcp client update dns	Enables DDNS updates of A RRs using the same hostname passed in the hostname and FQDN options by a client.
ip dhcp-client update dns	Enables DDNS updates of A RRs using the same hostname passed in the hostname and FQDN options by a client.
ip dhcp update dns	Enables DDNS updates of A and PTR RRs for most address pools.
ip host-list	Specifies a list of hosts that will receive DDNS updates of A and PTR RRs.
show ip ddns update	Displays information about the DDNS updates.
show ip ddns update method	Displays information about the DDNS update method.
show ip dhcp server pool	Displays DHCP server pool statistics.
show ip host-list	Displays the assigned hosts in a list.

Command	Description
update dns	Dynamically updates a DNS with A and PTR RRs for some address pools.

debug ip dfp agent

To display debugging messages for the Dynamic Feedback Protocol (DFP) agent subsystem, use the **debug ip dfp**command in user EXEC or privileged EXEC mode. To stop debugging output, use the **no** form of this command.

debug ip dfp agent

no debug ip dfp agent

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

User EXEC or privileged EXEC mode

Command History

Release	Modification
12.1(8a)E	This command was introduced.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

This command displays debugging messages for the DFP agent subsystem.

See the following caution before using debug commands:

Caution Because debugging output is assigned a high priority in the CPU process, it can render the system unusable. For this reason, use debug commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to use debug commands during periods of lower network flows and fewer users. Debugging during these periods reduces the effect these commands have on other users on the system.

Examples

The following example configures a DFP agent debugging session:

```
Router# debug ip dfp agent
DFP debugging is on
```

The following example stops all debugging:

```
Router# no debug all
All possible debugging has been turned off
```

debug ip dhcp server

To enable Cisco IOS Dynamic Host Configuration Protocol (DHCP) server debugging, use the **debug ip dhcp server** command in privileged EXEC mode. To disable DHCP server debugging, use the **no** form of this command.

debug ip dhcp server {**events**| **packets**| **linkage**| **class**}

no debug ip dhcp server {**events**| **packets**| **linkage**| **class**}

Syntax Description

events	Reports server events, such as address assignments and database updates.
packets	Decodes DHCP receptions and transmissions.
linkage	Displays database linkage information, such as parent-child relationships in a radix tree.
class	Displays DHCP class-based information.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(1)T	This command was introduced.
12.2(13)ZH	The **class** keyword was added.
12.3(4)T	The **class** keyword was integrated into Cisco IOS Release 12.3(4)T.
12.3(11)T	The output was enhanced to show the static mappings.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following example shows a combination of DHCP server events and decoded receptions and transmissions:

```
Router# debug ip dhcp server events
```

```
Router# debug ip dhcp server packets
```

```
DHCPD:DHCPDISCOVER received from client 0b07.1134.a029 through relay 10.1.0.253.
DHCPD:assigned IP address 10.1.0.3 to client 0b07.1134.a029.
DHCPD:Sending DHCPOFFER to client 0b07.1134.a029 (10.1.0.3).
DHCPD:unicasting BOOTREPLY for client 0b07.1134.a029 to relay 10.1.0.253.
DHCPD:DHCPREQUEST received from client 0b07.1134.a029.
DHCPD:Sending DHCPACK to client 0b07.1134.a029 (10.1.0.3).
```

```
DHCPD:unicasting BOOTREPLY for client 0b07.1134.a029 to relay 10.1.0.253.
DHCPD:checking for expired leases.
```

The following example shows database linkage information:

```
Router# debug ip dhcp server linkage

DHCPD:child pool:10.1.0.0 / 255.255.0.0 (subnet10.1)
DHCPD:parent pool:10.0.0.0 / 255.0.0.0 (net10)
DHCPD:child pool:10.0.0.0 / 255.0.0.0 (net10)
DHCPD:pool (net10) has no parent.
DHCPD:child pool:10.1.0.0 / 255.255.0.0 (subnet10.1)
DHCPD:parent pool:10.0.0.0 / 255.0.0.0 (net10)
DHCPD:child pool:10.0.0.0 / 255.0.0.0 (net10)
DHCPD:pool (net10) has no parent.
```

The following example shows when a DHCP class is removed:

```
Router# debug ip dhcp server class
DHCPD:deleting class CLASS1
```

The following example shows the debug output when the configured pattern does not match:

```
Router# debug ip dhcp server class

DHCPD:Searching for a match to 'relay-information
0106000 4000202020208000600009e80b8800' in class CLASS1
DHCPD:Searching for a match to 'relay-information 01060004000202020208000600009e80b8800' in
 class CLASS1
DHCPD:Searching for a match to 'relay-information 0106000
```

The following example shows the debug output when you unconfigure a DHCP pattern in a DHCP class and then configure the pattern in the DHCP class:

```
Router# debug ip dhcp server class

DHCPD:pattern 'relay-information 123456' removed from class CLASS1
DHCPD:Added pattern 'relay-information 010600040002020202 0800060009e80b8800' for class
CLASS1
```

The following example shows the debug output when the configured pattern does match:

```
Router# debug ip dhcp server class

DHCPD:Searching for a match to 'relay-information
0106000 4000202020208000600009e80b8800' in class CLASS1
DHCPD:input pattern 'relay-information 010600040002020202 0800060009e80b8800' matches class
 CLASS1
DHCPD:input matches class CLASS1
```

The following example shows the debug output when static mappings are configured:

```
Router# debug ip dhcp server
Loading abc/static_pool from 10.19.192.33 (via Ethernet0): !
[OK - 333 bytes]
*May 26 23:14:21.259: DHCPD: contacting agent tftp://10.19.192.33/abc/static_pool (attempt
 0)
*May 26 23:14:21.467: DHCPD: agent tftp://10.19.192.33/abc/static_pool is responding.
*May 26 23:14:21.467: DHCPD: IFS is ready.
*May 26 23:14:21.467: DHCPD: reading bindings from
tftp://10.19.192.33/abc/static_pool.
*May 26 23:14:21.707: DHCPD: read 333 / 1024 bytes.
*May 26 23:14:21.707: DHCPD: parsing text line “*time* Apr 22 2002 11:31 AM”
*May 26 23:14:21.707: DHCPD: parsing text line “”
*May 26 23:14:21.707: DHCPD: parsing text line
!IP address Type Hardware address Lease expiration.
*May 26 23:14:21.707: DHCPD: parsing text line
“10.9.9.1/24 id 0063.6973.636f.2d30.3036.302e.3437”
*May 26 23:14:21.707: DHCPD: creating binding for 10.9.9.1
*May 26 23:14:21.707: DHCPD: Adding binding to radix tree (10.9.9.1)
*May 26 23:14:21.707: DHCPD: Adding binding to hash tree
*May 26 23:14:21.707: DHCPD: parsing text line
```

```
"10.9.9.4 id 0063.7363.2d30.3036.302e.3762.2e39.3634.632d"
*May 26 23:14:21.711: DHCPD: creating binding for 10.9.9.4
*May 26 23:14:21.711: DHCPD: Adding binding to radix tree (10.9.9.4)
*May 26 23:14:21.711: DHCPD: Adding binding to hash tree
*May 26 23:14:21.711: DHCPD: parsing text line "Infinite"
*May 26 23:14:21.711: DHCPD: parsing text line ""
*May 26 23:14:21.711: DHCPD: parsing text line
!IP address Interface-index Lease expiration VRF.
*May 26 23:14:21.711: DHCPD: parsing text line "*end*"
*May 26 23:14:21.711: DHCPD: read static bindings from tftp://10.19.192.33/smith/static_pool.
```

Related Commands

Command	Description
debug dhcp	Displays debugging information about the DHCP client and monitors the status of DHCP packets.
debug ip ddns update	Enables debugging for DDNS updates.
host (host-list)	Specifies a list of hosts that will receive DDNS updates of A and PTR RRs.
ip ddns update hostname	Enables a host to be used for DDNS updates of A and PTR RRs.
ip ddns update method	Specifies a method of DDNS updates of A and PTR RRs and the maximum interval between the updates.
ip dhcp client update dns	Enables DDNS updates of A RRs using the same hostname passed in the hostname and FQDN options by a client on an interface.
ip dhcp-client update dns	Enables DDNS updates of A RRs using the same hostname passed in the hostname and FQDN options by a client.
ip dhcp update dns	Enables DDNS updates of A and PTR RRs for most address pools.
ip host-list	Specifies a list of hosts that will receive DDNS updates of A and PTR RRs.
show ip ddns update	Displays information about the DDNS updates.
show ip ddns update method	Displays information about the DDNS update method.
show ip dhcp server pool	Displays DHCP server pool statistics.
show ip host-list	Displays the assigned hosts in a list.
update dns	Dynamically updates a DNS with A and PTR RRs for some address pools.

debug ip dhcp server redundancy

To display debugging information about DHCP server and relay agent redundancy events, use the **debug ip dhcp server redundancy**command in privileged EXEC mode. To disable the display of debugging output, use the **no** form of this command.

debug ip dhcp server redundancy

no debug ip dhcp server redundancy

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging output is disabled for DHCP server and relay agent redundancy events.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(31)SB2	This command was introduced.

Usage Guidelines

Use this command with caution. Many bindings being synchronized between the active and standby Route Processor (RP) can trigger a large amount of debugging output.

Examples

The following example displays debug messages regarding DHCP server and relay agent redundancy events. The last line (and only that line) is output when the **debug ip dhcp server redundancy** command is enabled. The line indicates that a binding update message has been sent to the standby for the IP address 10.0.0.2 in the pool named “test.”

```
Router# debug ip dhcp server redundancy
*Mar 22 10:32:21: DHCPD: assigned IP address 10.0.0.2 to client
0063.6973.636f.2d30.3030.342e.3465.6130.2e30.3831.632d.4661.312f.302e.31.
*Mar 22 10:32:21:  DHCPD: lease time = 3600
*Mar 22 10:32:21: DHCPD: dhcpd_lookup_route: host = 10.0.0.2
*Mar 22 10:32:21: DHCPD: dhcpd_lookup_route: index = 0
*Mar 22 10:32:21: DHCPD: dhcpd_create_and_hash_route: host = 10.0.0.2
*Mar 22 10:32:21: DHCPD: dhcpd_create_and_hash_route index = 0
*Mar 22 10:32:21: DHCPD: dhcpd_add_route: lease = 3600
*Mar 22 10:32:21: DHCPD: dynamic sync completed for 10.0.0.2 in pool test
```

Related Commands

Command	Description
debug dhcp redundancy	Displays debugging information about DHCP proxy client redundancy events.

debug ip dhcp server snmp

To enable DHCP server Simple Network Management Protocol (SNMP) debugging, use the **debug ip dhcp server snmp** command in privileged EXEC mode. To disable DHCP server SNMP debugging, use the **no** form of this command.

debug ip dhcp server snmp

no debug ip dhcp server snmp

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(33)SRC	This command was introduced.

Examples

The following example shows how to enable debugging and display DHCP server SNMP debugging events:

```
Router# debug ip dhcp server snmp
```

```
00:18:01: DHCPD SNMP: pool 'pool1' 'high' utilization trap is ignored
00:18:18: DHCPD SNMP: pool 'pool1' 'low' utilization trap is ignored
00:20:46: DHCPD SNMP: subnet 4.1.1.0 'high' utilization trap is ignored
00:21:03: DHCPD SNMP: subnet 4.1.1.0 'low' utilization trap is ignored
00:18:01: DHCPD SNMP: subnet trap is not enabled
00:37:32: DHCPD SNMP: pool trap is not enabled
00:37:57: DHCPD SNMP: interface trap is not enabled
00:27:27: DHCPD SNMP: duplicate trap is not enabled
```

debug ip dns name-list

To enable debugging output for Domain Name System (DNS) name list events, use the **debug ip dns name-list** command in privileged EXEC mode. To disable debugging output for DNS name list events, use the **no** form of this command.

debug ip dns name-list

no debug ip dns name-list

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging output is disabled for DNS name lists.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(9)T	This command was introduced.

Usage Guidelines

This command enables the writing of DNS name list event messages to system message logging (syslog) output. A DNS name list event can be either of the following:

- The addition or removal of a DNS name list entry (a hostname pattern and action to perform on an incoming DNS query for a hostname that matches the pattern). To add or remove a DNS name list entry, use the **ip dns name-list** command.
- The removal of a DNS name list.

Note The addition of a DNS name list is reported as an addition of a name list entry.

To display which debugging options are enabled (DNS name list, DNS view, or DNS view list), use the **show debugging** command. To display the syslog history statistics and buffer contents, use the **show logging** command. To display a particular DNS name list or all configured name lists, use the **show ip dns name-list** command.

Examples

The following sample output from the **debug ip dns name-list** command shows the hostname pattern www.example.com being added to DNS name list 1 as a permit clause. Next, the hostname patterns www.example1.com and www.example2.com are added to DNS name list 2 as deny clauses and permit clauses, respectively. Finally, the hostname pattern www.example1.com is removed from DNS name list 2.

```
Router# debug ip dns name-list
```

```
DNS Name-list debugging is on
.
.
.
Router# show debugging

DNS Name-list debugging is on
.
.
.
Router# show logging

.
.
.
*May 16 14:54:44.326: DNS_NAMELIST: adding permit 'WWW.EXAMPLE' to name-list 1
*May 16 14:54:44.910: DNS_NAMELIST: adding deny 'WWW.EXAMPLE1.COM' to name-list 2
*May 16 14:54:45.202: DNS_NAMELIST: adding permit 'WWW.EXAMPLE2.COM' to name-list 2
*May 16 19:32:20.881: DNS_NAMELIST: removing 'WWW.EXAMPLE1.COM' from name-list 2
```

Related Commands

Command	Description
ip dns name-list	Defines a list of pattern-matching rules in which each rule permits or denies the use of a DNS view list member to handle a DNS query based on whether the query hostname matches the specified regular expression.
show debugging	Displays the state of each debugging option.
show ip dns name-list	Displays a particular DNS name list or all configured name lists.
show logging	Displays the contents of logging buffers.

debug ip dns view

To enable debugging output for Domain Name System (DNS) view events, use the **debug ip dns view** command in privileged EXEC mode. To disable debugging output for a DNS view, use the **no** form of this command.

debug ip dns view

no debug ip dns view

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging output is disabled for DNS views.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(9)T	This command was introduced.

Usage Guidelines

This command enables the writing of DNS view event messages to system message logging (syslog) output. A DNS view event can be any of the following:

- The addition or removal of a DNS view definition.
- The addition or removal of a DNS forwarding name server setting for a DNS view.
- The addition or removal of a DNS resolver setting for a DNS view.
- The enabling or disabling of logging of a syslog message each time a DNS view is used.

To display which debugging options are enabled (DNS name list, DNS view, or DNS view list), use the **show debugging** command. To show the syslog history statistics and buffer contents, use the **show logging** command.

Examples

The following sample output from the **debug ip dns view** command shows the default DNS view being configured:

```
Router# debug ip dns view

DNS View debugging is on
.
.
.
Router# show debugging

DNS View debugging is on
.
.
```

```
.
Router# show logging

.
.
.
DNS_VIEW: creating view view1
DNS_VIEW: Clearing logging in view default
DNS_VIEW: Setting domain lookup in view default
DNS_VIEW: Setting domain name to cisco.com in view default
DNS_VIEW: Setting  domain list example1.com in view default
DNS_VIEW: Setting  domain list example1.com example2.com in view default
DNS_VIEW: Setting  domain list example1.com example2.com example3.com in view default
DNS_VIEW: Setting domain multicast to 192.0.2.10 in view default
DNS_VIEW: Setting domain lookup in view default
DNS_VIEW: Setting domain timeout to 7 in view default
DNS_VIEW: Setting domain retry to 7 in view default
DNS_VIEW: Setting  domain name-server 192.0.2.204 192.0.2.205 in view default
DNS_VIEW: Setting  domain name-server 192.0.2.204 192.0.2.205 192.0.2.206 in view default
DNS_VIEW: Setting domain name-server interface FastEthernet0/1 in view default
DNS_VIEW: Setting domain round-robin to 4 in view default
DNS_VIEW: Setting dns forwarding in view default
DNS_VIEW: Setting  dns forwarder 192.0.2.11 in view default
DNS_VIEW: Setting  dns forwarder 192.0.2.11 192.0.2.12 in view default
DNS_VIEW: Setting  dns forwarder 192.0.2.11 192.0.2.12 192.0.2.13 in view default
```

Related Commands

Command	Description
ip dns view	Enters DNS view configuration mode for the specified DNS view so that the logging setting, forwarding parameters, and resolving parameters can be configured for the view.
show debugging	Displays the state of each debugging option.
show logging	Displays the contents of logging buffers.

debug ip dns view-list

To enable debugging output for Domain Name System (DNS) view list events, use the **debug ip dns view-list** command in privileged EXEC mode. To disable debugging output for a DNS view list, use the **no** form of this command.

debug ip dns view-list

no debug ip dns view-list

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging output is disabled for DNS view lists.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(9)T	This command was introduced.

Usage Guidelines

This command enables the writing of DNS view list event messages to system message logging (syslog) output. A DNS view list event can be any of the following:

- The addition or removal of a DNS view list definition. To add or remove a DNS view list definition, use the **ip dns view-list** command.
- The addition or removal of a DNS view list member (a DNS view and the relative order in which it is to be checked in the view list) to or from a DNS view list. To add or remove a DNS view list member, use the **view** command.
- The setting or clearing of a DNS view list assignment as the default view list (using the **ip dns server view-group** command) or to an interface (using the **ip dns view-group** command).

To show which debugging options are enabled (DNS name list, DNS view, or DNS view list), use the **show debugging** command. To show the syslog history statistics and buffer contents, use the **show logging** command.

Examples

The following sample output from the **debug ip dns vies-list** command shows the addition of the DNS view list definition named userlist5. Next, five DNS views are added as members of the DNS view list.

```
Router# debug ip dns view-list
```

```
DNS View-list debugging is on
.
.
.
Router# show debugging
```

```
DNS View-list debugging is on
.
.
.
Router# show logging

*May 16 23:31:17.491: DNS_VIEWLIST: creating view-list userlist5
*May 16 23:31:17.711: DNS_VIEWLIST: adding member user1 vrf vpn101 order 10 to view-list
userlist5
*May 16 23:31:18.583: DNS_VIEWLIST: adding member user2 vrf vpn102 order 20 to view-list
userlist5
*May 16 23:31:19.851: DNS_VIEWLIST: adding member user3 vrf vpn103 order 30 to view-list
userlist5
*May 16 23:31:21.007: DNS_VIEWLIST: adding member user4 vrf vpn204 order 45 to view-list
userlist5
*May 16 23:31:22.199: DNS_VIEWLIST: adding member default order 60 to view-list userlist5
```

Related Commands

Command	Description
ip dns server view-group	Specifies the DNS view list to use to determine which DNS view to use handle incoming queries that arrive on an interface not configured with a DNS view list.
ip dns view-group	Specifies the DNS view list to use to determine which DNS view to use to handle incoming DNS queries that arrive on a specific interface.
ip dns view-list	Enters DNS view list configuration mode so that DNS views can be added to or removed from the ordered list of DNS views.
show debugging	Displays the state of each debugging option.
show logging	Displays the contents of logging buffers.
view	Enters DNS view list member configuration mode so that usage restrictions can be configured for the view list member.

debug ip domain

To enable Domain Name System (DNS) debugging and view DNS debugging information, use the **debug ip domain** command in privileged EXEC mode. To disable DNS debugging, use the **no** form of this command.

debug ip domain

no debug ip domain

Syntax Description

This command has no arguments or keywords.

Note Use the **debug ip domain** command form to enable DNS debugging and view basic DNS debugging information. To view more DNS debugging options such as DNS server response debugging and so on, use the question mark (?) online help function.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.13S	This command was introduced.
15.4(3)S	This command was integrated into Cisco IOS Release 15.4(3)S.

Examples

The following is sample output from the **debug ip domain** command:

```
Device> enable
Device# debug ip domain

Domain Name System debugging is on
Device#
*Jul 18 09:16:19.546: DNS: Incoming UDP query (id#8168)
*Jul 18 09:16:19.547: DNS: Type 1 DNS query (id#8168) for host 'abc.google.com' from
209.165.200.230(27106)
*Jul 18 09:16:19.547: DNS: Servicing request using view default
*Jul 18 09:16:19.547: search_nametype_index: abc.google.com
*Jul 18 09:16:19.547: search_nametype_index: found abc.google.com for abc.google.com
*Jul 18 09:16:19.547: search_nametype_index: abc.google.com
*Jul 18 09:16:19.547: search_nametype_index: found abc.google.com for abc.google.com
*Jul 18 09:16:19.547: search_nametype_index: google.com
*Jul 18 09:16:19.547: search_nametype_index: com
*Jul 18 09:16:19.547: search_nametype_index: abc.google.com
*Jul 18 09:16:19.547: search_nametype_index: found abc.google.com for abc.google.com
*Jul 18 09:16:19.547: DNS: Reply to client 209.165.200.230/27106 query A
*Jul 18 09:16:19.547: DNS: Finished processing query (id#8168) in 0.001 secs
*Jul 18 09:16:19.547: DNS: Sending response to 209.165.200.230/27106, len 48
```

Related Commands

Command	Description
debug ip domain replies	Enables DNS server response debugging and displays debugging information for DNS server responses to clients.
ip dns server	Enables the DNS server on a device.
ip dns server view-group	Specifies the default DNS server view list for a device.

debug ip domain replies

To enable debugging for Domain Name System (DNS) server responses to clients and view debugging information for DNS server responses to clients, use the **debug ip domain replies** command in privileged EXEC mode. To disable DNS server response debugging, use the **no** form of this command.

debug ip domain replies [**detail**]

no debug ip domain replies [**detail**]

Syntax Description

detail	(Optional) Displays detailed debugging information for DNS server responses to clients.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
Cisco IOS XE Release 3.13S	This command was introduced.
15.4(3)S	This command was integrated into Cisco IOS Release 15.4(3)S.

Examples

The following is sample output from the **debug ip domain replies** command:

```
Device> enable
Device# debug ip domain replies

Domain Name System Reply debugging is on


*Jul 18 09:17:22.868: DNS: Finished processing query (id#34422) in 0.000 secs
*Jul 18 09:17:23.663: DNS: Finished processing query (id#51171) in 0.000 secs
*Jul 18 09:17:23.665: DNS: Finished processing query (id#46198) in 0.000 secs
```

Examples

Sample Output for Detailed DNS Response Debugging

```
Device> enable
Device# debug ip domain replies detail

Domain Name System Reply debugging is on (detailed)


*Jul 18 09:17:58.635: DNS: Send reply from internal information:
*Jul 18 09:17:58.635: DOM: id=47025, response, opcode=0, aa=0, tc=0, rd=1, ra=1
*Jul 18 09:17:58.635:      rcode=0, qdcount=1, ancount=1, nscount=0, arcount=0
*Jul 18 09:17:58.635:      query name is abc.google.com, qtype=1, class=1
*Jul 18 09:17:58.635: Answer section:
*Jul 18 09:17:58.635:    Name='abc.google.com'
```

```
*Jul 18 09:17:58.635:    RR type=1, class=1, ttl=10, data length=4
*Jul 18 09:17:58.635:      IP=12.12.12.12
*Jul 18 09:17:58.635: Authority section:
*Jul 18 09:17:58.635: Additional record section:
*Jul 18 09:17:58.635: DNS: Finished processing query (id#47025) in 0.001 secs


*Jul 18 09:17:58.637: DNS: Send reply from internal information:
*Jul 18 09:17:58.637: DOM: id=25881, response, opcode=0, aa=0, tc=0, rd=1, ra=1
*Jul 18 09:17:58.637:      rcode=0, qdcount=1, ancount=1, nscount=0, arcount=0
*Jul 18 09:17:58.637:      query name is abc.google.com, qtype=1, class=1
*Jul 18 09:17:58.637: Answer section:
*Jul 18 09:17:58.637:    Name='abc.google.com'
*Jul 18 09:17:58.637:    RR type=1, class=1, ttl=10, data length=4
*Jul 18 09:17:58.637:      IP=12.12.12.12
*Jul 18 09:17:58.637: Authority section:
*Jul 18 09:17:58.637: Additional record section:
*Jul 18 09:17:58.637: DNS: Finished processing query (id#25881) in 0.001 secs


*Jul 18 09:17:58.638: DNS: Send reply from internal information:
*Jul 18 09:17:58.638: DOM: id=41387, response, opcode=0, aa=0, tc=0, rd=1, ra=1
*Jul 18 09:17:58.638:      rcode=0, qdcount=1, ancount=1, nscount=0, arcount=0
*Jul 18 09:17:58.638:      query name is abc.google.com, qtype=1, class=1
*Jul 18 09:17:58.638: Answer section:
*Jul 18 09:17:58.638:    Name='abc.google.com'
*Jul 18 09:17:58.638:    RR type=1, class=1, ttl=10, data length=4
*Jul 18 09:17:58.638:      IP=12.12.12.12
*Jul 18 09:17:58.638: Authority section:
*Jul 18 09:17:58.638: Additional record section:
*Jul 18 09:17:58.638: DNS: Finished processing query (id#41387) in 0.000 secs
```

Related Commands

Command	Description
debug ip domain	Enables DNS debugging and displays DNS debugging information.
ip dns server	Enables the DNS server on a device.
ip dns server view-group	Specifies the default DNS server view list for a device.

debug ip drp

To display Director Response P rotocol (DRP) information, use the **debug ip drp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip drp

no debug ip drp

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

The **debug ip drp** command is used to debug the director response agent used by the Distributed Director product. The Distributed Director can be used to dynamically respond to Domain Name System (DNS) queries with the IP address of the "best" host based on various criteria.

Examples

The following is sample output from the **debug ip drp**command. This example shows the packet origination, the IP address that information is routed to, and the route metrics that were returned.

```
Router# debug ip drp
DRP: received v1 packet from 172.69.232.8, via Ethernet0
DRP: RTQUERY for 172.69.58.94 returned internal=0, external=0
```

The table below describes the significant fields shown in the display.

Table 17: debug ip drp Field Descriptions

Field	Description
DRP: received v1 packet from 172.69.232.8, via Ethernet0	Router received a version 1 DRP packet from the IP address shown, via the interface shown.
DRP: RTQUERY for 172.69.58.94	DRP packet contained two Route Query requests. The first request was for the distance to the IP address 171.69.113.50.
internal	If nonzero, the metric for the internal distance of the route that the router uses to send packets in the direction of the client. The internal distance is the distance within the autonomous system of the router.
external	If nonzero, the metric for the Border Gateway Protocol (BGP) or external distance used to send packets to the client. The external distance is the distance outside the autonomous system of the router.

debug ip dvmrp

Note The **debug ip dvmrp**command is not available in 12.2(33)SRB, 15.0(1)M, and later 12.2SR, 15.0M, and T releases.

To display information on Distance Vector Multiprotocol Routing Protocol (DVMRP) packets received and sent, use the **debug ip dvmrp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip dvmrp [**detail** [*access-list*] [**in**| **out**]]

no debug ip dvmrp [**detail** [*access-list*] [**in**| **out**]]

Syntax Description

detail	(Optional) Enables a more detailed level of output and displays packet contents.
access-list	(Optional) Causes the **debug ip dvmrp** command to restrict output to one access list.
in	(Optional) Causes the **debug ip dvmrp** command to output packets received in DVMRP reports.
out	(Optional) Causes the **debug ip dvmrp** command to output packets sent in DVMRP reports.

Command Modes

Privileged EXEC

Usage Guidelines

Use the debug ip dvmrp detail command with care. This command generates a substantial amount of output and can interrupt other activity on the router when it is invoked.

Examples

The following is sample output from the **debug ip dvmrp** command:

```
Router# debug ip dvmrp
DVMRP: Received Report on Ethernet0 from 172.19.244.10
DVMRP: Received Report on Ethernet0 from 172.19.244.11
DVMRP: Building Report for Ethernet0 224.0.0.4
DVMRP: Send Report on Ethernet0 to 224.0.0.4
DVMRP: Sending IGMP Reports for known groups on Ethernet0
DVMRP: Received Report on Ethernet0 from 172.19.244.10
DVMRP: Received Report on Tunnel0 from 192.168.199.254
DVMRP: Received Report on Tunnel0 from 192.168.199.254
DVMRP: Received Report on Tunnel0 from 192.168.199.254
DVMRP: Received Report on Tunnel0 from 192.168.199.254
DVMRP: Received Report on Tunnel0 from 192.168.199.254
DVMRP: Received Report on Tunnel0 from 192.168.199.254
DVMRP: Building Report for Tunnel0 224.0.0.4
```

```
DVMRP: Send Report on Tunnel0 to 192.168.199.254
DVMRP: Send Report on Tunnel0 to 192.168.199.254
DVMRP: Send Report on Tunnel0 to 192.168.199.254
DVMRP: Send Report on Tunnel0 to 192.168.199.254
DVMRP: Radix tree walk suspension
DVMRP: Send Report on Tunnel0 to 192.168.199.254
```

The following lines show that the router received DVMRP routing information and placed it in the mroute table:

```
DVMRP: Received Report on Ethernet0 from 172.19.244.10
DVMRP: Received Report on Ethernet0 from 172.19.244.11
```

The following lines show that the router is creating a report to send to another DVMRP router:

```
DVMRP: Building Report for Ethernet0 224.0.0.4
DVMRP: Send Report on Ethernet0 to 224.0.0.4
```

The table below provides a list of internet multicast addresses supported for host IP implementations.

Table 18: Internet Multicast Addresses

Address	Description	RFC
224.0.0.0	Base address (reserved)	RFC 1112
224.0.0.1	All systems on this subnet	RFC 1112
224.0.0.2	All routers on this subnet	
224.0.0.3	Unassigned	
224.0.0.4	DVMRP routers	RFC 1075
224.0.0.5	OSPFIGP all routers	RFC 1583

The following lines show that a protocol update report has been sent to all known multicast groups. Hosts use Internet Group Management Protocol (IGMP) reports to communicate with routers and to request to join a multicast group. In this case, the router is sending an IGMP report for every known group to the host, which is running mrouted. The host then responds as though the router were a host on the LAN segment that wants to receive multicast packets for the group.

```
DVMRP: Sending IGMP Reports for known groups on Ethernet0
```

The following is sample output from the **debug ip dvmrp detail** command:

```
Router# debug ip dvmrp detail

DVMRP: Sending IGMP Reports for known groups on Ethernet0
DVMRP: Advertise group 224.2.224.2 on Ethernet0
DVMRP: Advertise group 224.2.193.34 on Ethernet0
DVMRP: Advertise group 224.2.231.6 on Ethernet0
DVMRP: Received Report on Tunnel0 from 192.168.199.254
DVMRP: Origin 150.166.53.0/24, metric 13, distance 0
DVMRP: Origin 150.166.54.0/24, metric 13, distance 0
DVMRP: Origin 150.166.55.0/24, metric 13, distance 0
DVMRP: Origin 150.166.56.0/24, metric 13, distance 0
DVMRP: Origin 150.166.92.0/24, metric 12, distance 0
DVMRP: Origin 150.166.100.0/24, metric 12, distance 0
DVMRP: Origin 150.166.101.0/24, metric 12, distance 0
DVMRP: Origin 150.166.142.0/24, metric 8, distance 0
```

```
DVMRP: Origin 150.166.200.0/24, metric 12, distance 0
DVMRP: Origin 150.166.237.0/24, metric 12, distance 0
DVMRP: Origin 150.203.5.0/24, metric 8, distance 0
```

The following lines show that this group is available to the DVMRP router. The mrouted process on the host will forward the source and multicast information for this group through the DVMRP cloud to other members.

```
DVMRP: Advertise group 224.2.224.2 on Ethernet0
```

The following lines show the DVMRP route information:

```
DVMRP: Origin 150.166.53.0/24, metric 13, distance 0
DVMRP: Origin 150.166.54.0/24, metric 13, distance 0
```

The metric is the number of hops the route has covered, and the distance is the administrative distance.

debug ip eigrp

To display information on Enhanced Interior Gateway Routing Protocol (EIGRP) protocol packets, use the **debug ip eigrp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip eigrp [**vrf** *vrf-name*]

no debug ip eigrp [**vrf** *vrf-name*]

Syntax Description

vrf *vrf-name*	(Optional) Restricts output to a specific VRF.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(21)S	This command was modified. The **vrf** *vrf-name* keyword and argument were added.

Usage Guidelines

This command helps you analyze the packets that are sent and received on an interface. Because the **debug ip eigrp** command generates a substantial amount of output, only use it when traffic on the network is light.

Examples

The following is sample output from the **debug ip eigrp** command:

```
Router# debug ip eigrp
IP-EIGRP: Processing incoming UPDATE packet
IP-EIGRP: Ext 192.168.3.0 255.255.255.0 M 386560 - 256000 130560 SM 360960 - 256000 104960
IP-EIGRP: Ext 192.168.0.0 255.255.255.0 M 386560 - 256000 130560 SM 360960 - 256000 104960
IP-EIGRP: Ext 192.168.3.0 255.255.255.0 M 386560 - 256000 130560 SM 360960 - 256000 104960
IP-EIGRP: 172.69.43.0 255.255.255.0, - do advertise out Ethernet0/1
IP-EIGRP: Ext 172.69.43.0 255.255.255.0 metric 371200 - 256000 115200
IP-EIGRP: 192.135.246.0 255.255.255.0, - do advertise out Ethernet0/1
IP-EIGRP: Ext 192.135.246.0 255.255.255.0 metric 46310656 - 45714176 596480
IP-EIGRP: 172.69.40.0 255.255.255.0, - do advertise out Ethernet0/1
IP-EIGRP: Ext 172.69.40.0 255.255.255.0 metric 2272256 - 1657856 614400
IP-EIGRP: 192.135.245.0 255.255.255.0, - do advertise out Ethernet0/1
IP-EIGRP: Ext 192.135.245.0 255.255.255.0 metric 40622080 - 40000000 622080
IP-EIGRP: 192.135.244.0 255.255.255.0, - do advertise out Ethernet0/1
```

The table below describes the significant fields shown in the display.

Table 19: debug ip eigrp Field Descriptions

Field	Description
IP-EIGRP:	Indicates that this is an IP EIGRP message.

Field	Description
Ext	Indicates that the following address is an external destination rather than an internal destination, which would be labeled as Int.
M	Displays the computed metric, which includes the value in the SM field and the cost between this router and the neighbor. The first number is the composite metric. The next two numbers are the inverse bandwidth and the delay, respectively.
SM	Displays the metric as reported by the neighbor.

The following example shows how to turn on debugging output for a specific VRF in an EIGRP instance:

```
Router# debug ip eigrp vrf red
EIGRP-IPv4 Route Event debugging is on
```

Related Commands

Command	Description
vrf definition	Defines a virtual routing and forwarding instance.

debug ip eigrp notifications

To display Enhanced Interior Gateway Routing Protocol (EIGRP) events and notifications in the console of the router, use the **debug ip eigrp notifications** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip eigrp notifications

no debug ip eigrp notifications

Syntax Description

This command has no arguments or keywords.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
15.0(1)M	This command was integrated into Cisco IOS Release 15.0(1)M.

Usage Guidelines

The output of the debug ip eigrp notifications command displays EIGRP events and notifications.

Examples

The following example output shows that the NSF-aware router has received the restart notification. The NSF-aware router will now wait for end of transmission (EOT) to be sent from the restarting neighbor (NSF-capable).

```
Router# debug ip eigrp notifications
*Oct  4 11:39:18.092:EIGRP:NSF:AS2. Rec RS update from 135.100.10.1,
00:00:00. Wait for EOT.
*Oct  4 11:39:18.092:%DUAL-5-NBRCHANGE:IP-EIGRP(0) 2:Neighbor
135.100.10.1 (POS3/0) is up:peer NSF restarted
```

debug ip error

To display IP errors, use the **debug ip error** command in privileged EXEC mode. To disable debugging errors, use the **no** form of this command.

debug ip error *access-list-number* **[detail] [dump]**

no debug ip error

Syntax Description

access-list-number	(Optional) The IP access list number that you can specify. If the datagram is not permitted by that access list, the related debugging output (or IP error) is suppressed. Standard, extended, and expanded access lists are supported. The range of standard and extended access lists is from 1 to 199. The range of expanded access lists is from 1300 to 2699.
detail	(Optional) Displays detailed IP error debugging information.
dump	(Hidden) Displays IP error debugging information along with raw packet data in hexadecimal and ASCII forms. This keyword can be enabled with individual access lists and also with the **detail** keyword. **Note** The **dump** keyword is not fully supported and should be used only in collaboration with Cisco Technical Support. See the caution notes below, in the usage guidelines, for more specific information.

Command Default

No default behavior or values.

Command Modes

Privileged EXEC

Usage Guidelines

This command is used for IP error debugging. The output displays IP errors which are locally detected by this router.

Caution Enabling this command will generate output only if IP errors occur. However, if the router starts to receive many packets that contain errors, substantial output may be generated and severely affect system performance. This command should be used with caution in production networks. It should only be enabled when traffic on the IP network is low, so other activity on the system is not adversely affected. Enabling the **detail** and **dump** keywords use the highest level of system resources of the available configuration options for this command, so a high level of caution should be applied when enabling either of these keywords.

Caution The **dump** keyword is not fully supported and should be used only in collaboration with Cisco Technical Support. Because of the risk of using significant CPU utilization, the dump keyword is hidden from the user and cannot be seen using the "?" prompt. The length of the displayed packet information may exceed the actual packet length and include additional padding bytes that do not belong to the IP packet. Also note that the beginning of a packet may start at different locations in the dump output depending on the specific router, interface type, and packet header processing that may have occurred before the output is displayed.

Examples

The following is sample output from the **debug ip error**command:

```
Router# debug ip error

IP packet errors debugging is on
04:04:45:IP:s=10.8.8.1 (Ethernet0/1), d=10.1.1.1, len 28, dispose ip.hopcount
```

The IP error in the above output was caused when the router attempted to forward a packet with a time-to-live (TTL) value of 0. The "ip.hopcount" traffic counter is incremented when a packet is dropped because of an error. This error is also displayed in the output of the **show ip traffic** command by the "bad hop count" traffic counter.

The table below describes the significant fields shown in the display.

Table 20: debug ip error Field Descriptions

Field	Description
IP:s=10.8.8.1 (Ethernet0/1)	The packet source IP address and interface.
d=10.1.1.1, len 28	The packet destination IP address and prefix length.
dispose ip.hopcount	This traffic counter increments when an IP packet is dropped because of an error.

The following is sample output from the **debug ip error**command enabled with the **detail** keyword:

```
Router# debug ip error detail
IP packet errors debugging is on (detailed)

1d08h:IP:s=10.0.19.100 (Ethernet0/1), d=10.1.1.1, len 28, dispose udp.noport

1d08h: UDP src=41921, dst=33434
```

1d08h:IP:s=10.0.19.100 (Ethernet0/1), d=10.2.2.2, len 28, dispose ip.hopcount

```
1d08h:    UDP src=33691, dst=33434
```

The detailed output includes layer 4 information in addition to the standard output. The IP error in the above output was caused when the router received a UDP packet when no application was listening to the UDP port. The "udp.noport" traffic counter is incremented when the router drops a UDP packet because of this error. This error is also displayed in the output of the **show ip traffic** command by the "no port" traffic counter under "UDP statistics."

The table below describes the significant fields shown in the display.

Table 21: debug ip error detail Field Descriptions

Field	Description
IP:s=10.0.19.100 (Ethernet0/1)	The IP packet source IP address and interface.
d=10.1.1.1, len 28	The IP packet destination and prefix length.
dispose udp.noport	The traffic counter that is incremented when a UDP packet is dropped because of this error.

The following is sample output from the **debug ip error**command enabled with the **detail** and **dump** keywords:

```
Router# debug ip error detail dump
IP packet errors debugging is on (detailed) (dump)
1d08h:IP:s=10.0.19.100 (Ethernet0/1), d=10.1.1.1, len 28, dispose udp.noport
1d08h:    UDP src=37936, dst=33434
03D72360:                      0001 42AD4242            ..B-BB
03D72370:0002FCA5 DC390800 4500001C 30130000  ..|%\9..E...0...
03D72380:01116159 0A001364 0A010101 9430829A  ..aY...d.....0..
03D72390:0008C0AD                             ..@-
1d08h:IP:s=10.0.19.100 (Ethernet0/1), d=10.2.2.2, len 28, dispose ip.hopcount
1d08h:    UDP src=41352, dst=33434
03C01600:                      0001 42AD4242            ..B-BB
03C01610:0002FCA5 DC390800 4500001C 302A0000  ..|%\9..E...0*..
03C01620:01116040 0A001364 0A020202 A188829A  ..`@...d....!...
03C01630:0008B253                             ..2S
```

Note The **dump** keyword is not fully supported and should be used only in collaboration with Cisco Technical Support. See the caution in the usage guidelines section of this command reference page for more specific information.

The output from the **debug ip error** command, when the **dump** keyword is enabled, provides raw packet data in hexadecimal and ASCII forms. This addtional output is displayed in addition to the standard output. The dump keyword can be used with all of the available configuration options of this command.

The table below describes the significant fields shown in the display.

Table 22: debug ip error detail dump Field Descriptions

Field	Description
IP:s=10.0.19.100 (Ethernet0/1)	The IP packet source IP address and interface.

Field	Description
d=10.1.1.1, len 28	The IP packet destination and prefix length.
dispose udp.noport	The traffic counter that is incremented when a UDP packet is dropped because of this error.

Related Commands

Command	Description
show ip traffic	Displays statistics about IP traffic.

debug ip flow cache

To enable debugging output for NetFlow cache, use the **debug ip flow cache** command in user EXEC or privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip flow cache

no debug ip flow cache

Syntax Description

This command has no arguments or keywords.

Command Default

Debugging output for NetFlow data export is disabled.

Command Modes

User EXEC Privileged EXEC

Command History

Release	Modification
12.0(1)	This command was introduced.
12.3(1)	Debugging output for NetFlow v9 data export was added.
12.3(7)T	Debugging output for NetFlow for IPv6 was added.
12.2(30)S	This command was integrated into Cisco IOS Release 12.2(30)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Examples

The following is sample output from the **debug ip flow export** command:

```
Router# debug ip flow cache
IP Flow cache allocation debugging is on
Router# show ipv6 flow
IP packet size distribution (0 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
IP Flow Switching Cache, 0 bytes
  0 active, 0 inactive, 0 added
  0 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
SrcAddress                                  InpIf    DstAddress
                OutIf    Prot SrcPrt DstPrt Packets
c7200-vxr-2#
```

```
000037: 01:56:26: IPFLOW: Allocating Sub-Flow cache, without hash flags.
000038: 01:56:26: IPFLOW: Sub-Flow table enabled.
000039: 01:56:26: IPFLOW: Sub-Flow numbers are:
    24 sub-flows per chunk, 0 hashflag len,
    1 chunks allocated, 12 max chunks,
    24 allocated records, 24 free records, 960 bytes allocated
000040: 01:56:26: IPFLOW: Sub-Flow cache removed
```

Related Commands

Command	Description
export destination	Enables the exporting of information from NetFlow aggregation caches.
ip flow-aggregation cache	Enables NetFlow aggregation cache schemes.
ip flow-export	Enables the exporting of information in NetFlow cache entries.
ipv6 flow-aggregation cache	Enables NetFlow aggregation cache schemes for IPv6 configurations.
ipv6 flow export	Enables the exporting of information in NetFlow cache entries for IPv6 NetFlow configurations.
show ip cache flow aggregation	Displays the NetFlow aggregation cache configuration.
show ip flow export	Display the statistics for NetFlow data export.

debug ip flow export

To enable debugging output for NetFlow data export, use the **debug ip flow export** command in user EXEC or privileged EXEC mode. To disable debugging output for NetFlow data export, use the **no** form of this command.

debug ip flow export

no debug ip flow export

Syntax Description

This command has no keywords or arguments.

Command Default

Debugging output for NetFlow data export is disabled.

Command Modes

User EXEC Privileged EXEC

Command History

Release	Modification
12.0(1)	This command was introduced.
12.3(1)	Debugging output for NetFlow v9 data export was added.
12.3(7)T	This command was modified so that NetFlow v9 data is collected for both IPv4 and IPv6.
12.2(18)S	This command was integrated into Cisco IOS Release 12.2(18)S.
12.2(30)S	This command was integrated into Cisco IOS Release 12.2(30)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF	This command was integrated into Cisco IOS Release 12.2(18)SXF.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Examples

The following is sample output from the **debug ip flow export** command:

```
Router# debug ip flow export
IP Flow export mechanism debugging is on
*Mar 6 22:56:21.627:IPFLOW:Sending export pak to 2001::FFFE/64 port 9999
*Mar 6 22:56:21.627:IPFLOW:Error sending export packet:Adjacency failure
```

Related Commands

Command	Description
export destination	Enables the exporting of information from NetFlow aggregation caches.
ipv6 flow-aggregation cache	Enables NetFlow aggregation cache schemes for IPv6.
ipv6 flow-export	Enables the exporting of information in NetFlow cache entries.
show ip cache flow aggregation	Displays the NetFlow accounting aggregation cache statistics.
show ip flow export	Displays the statistics for NetFlow data export.
show ipv6 flow export	Displays the statistics for NetFlow data export for IPv6.

debug ip ftp

To activate the debugging option to track the transactions submitted during an FTP session, use the **debug ip ftp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip ftp

no debug ip ftp

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Usage Guidelines

The **debug ip ftp** command is useful for debugging problems associated with FTP.

While configuring the ftp password, only encryption types 0 and 7 are allowed. Other encryption types will invoke an "Invalid encryption type" error.

If encryption type 7 has been chosen, the cli will check if the supplied password is encrypted (encrypted by Cisco proprietary algorithm). If the supplied password is found to be Cisco-encrypted, it will be configured. Otherwise the error "Invalid encrypted password" will be shown. The option 7 expects a Cisco-encrypted password to be supplied in the cli.

While configuring the ftp password, if encryption type 0 has been chosen, the cli will encrypt the password as long as the "service password-encryption" is enabled.

Examples

The following is an example of the **debug ip ftp**command:

```
Router# debug ip ftp
FTP transactions debugging is on
```

The following is sample output from the **debug ip ftp** command:

```
FTP: 220 ProFTPD 1.2.0pre8 Server (DFW Nostrum FTP Server) [defiant.dfw.nostrum.com]
Dec 27 22:12:09.133: FTP: ---> USER router
Dec 27 22:12:09.133: FTP: 331 Password required for router.
Dec 27 22:12:09.137: FTP: ---> PASS WQHK5JY2
Dec 27 22:12:09.153: FTP: 230 Anonymous access granted, restrictions apply.
Dec 27 22:12:09.153: FTP: ---> TYPE I
Dec 27 22:12:09.157: FTP: 200 Type set to I.
Dec 27 22:12:09.157: FTP: ---> PASV
.
.
.
.
.
.
.
.
.
.
.
.
.
```

```
Dec 27 22:12:09.173: FTP: ---> QUIT
Dec 27 22:12:09.181: FTP: 221 Goodbye.
```