startup (test boolean) through write mib-data

- startup (test boolean), page 3
- startup (test existence), page 4
- startup (test threshold), page 5
- test (event trigger), page 7
- test snmp trap auth-framwork sec-violation, page 9
- test snmp trap bridge, page 10
- test snmp trap c6kxbar, page 11
- test snmp trap call-home, page 14
- test snmp trap config-copy, page 15
- test snmp trap dhcp bindings, page 17
- test snmp trap dhcp-snooping bindings, page 18
- test snmp trap dot1x, page 19
- test snmp trap entity-diag, page 20
- test snmp trap errdisable ifevent, page 22
- test snmp trap flex-links status, page 23
- test snmp trap fru-ctrl, page 24
- test snmp trap l2-control vlan, page 25
- test snmp trap l2tc, page 26
- test snmp trap mac-notification, page 27
- test snmp trap module-auto-shutdown, page 28
- test snmp trap port-security, page 29
- test snmp trap power-ethernet port-on-off, page 30
- test snmp trap snmp, page 31
- test snmp trap stack, page 33
- test snmp trap storm-control, page 34
- test snmp trap stpx, page 35
- test snmp trap syslog, page 36
- test snmp trap trustsec, page 38
- test snmp trap trustsec-interface, page 40
- test snmp trap trustsec-policy, page 41
- test snmp trap trustsec-server, page 42
- test snmp trap trustsec-sxp, page 43
- test snmp trap udld, page 45
- test snmp trap vswitch dual-active, page 46
- test snmp trap vswitch vsl, page 48
- test snmp trap vtp, page 49
- test snmp trap vtp pruning-change, page 51
- type (test existence), page 52
- url (bulk statistics), page 54
- value (test boolean), page 56
- value type, page 57
- wildcard (expression), page 59
- write mib-data, page 60
startup (test boolean)

To specify whether an event can be triggered for the Boolean trigger test, use the startup command in event trigger boolean configuration mode. To disable the configured settings, use the no form of this command.

startup

no startup

Syntax Description
This command has no arguments or keywords.

Command Default
The startup event is enabled when the Boolean trigger test is enabled.

Command Modes
Event trigger boolean configuration (config-event-trigger-boolean)

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.4(20)T</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(33)SRE</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRE.</td>
</tr>
<tr>
<td>12.2(50)SY</td>
<td>This command was integrated into Cisco IOS Release 12.2(50)SY.</td>
</tr>
</tbody>
</table>

Usage Guidelines
The startup command triggers an event when the conditions specified for the Boolean trigger test are met.

Examples
The following example shows how to specify startup for the Boolean trigger test:

```plaintext
Router(config)# snmp mib event trigger owner owner1 name EventA
Router(config-event-trigger)# test boolean
Router(config-event-trigger-boolean)# startup
Router(config-event-trigger-boolean)# end
```

Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>test</td>
<td>Enables a trigger test.</td>
</tr>
</tbody>
</table>
**startup (test existence)**

To specify whether an event can be triggered for the existence trigger test, use the `startup` command in event trigger existence configuration mode. To disable the configured settings, use the `no` form of this command.

```
startup {present|absent}
no startup {present|absent}
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>present</td>
<td>Triggers the present startup test when the existence trigger conditions are met.</td>
</tr>
<tr>
<td>absent</td>
<td>Triggers the absent startup test when the existence trigger conditions are met.</td>
</tr>
</tbody>
</table>

**Command Default**

By default, both present and absent startup tests are triggered.

**Command Modes**

Event trigger existence configuration (config-event-trigger-existence)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.4(20)T</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(33)SRE</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRE.</td>
</tr>
<tr>
<td>12.2(50)SY</td>
<td>This command was integrated into Cisco IOS Release 12.2(50)SY.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `startup` command triggers an event when the conditions specified for the existence trigger test are met.

**Examples**

The following example shows how to specify startup for the existence trigger test:

```
Router(config)# snmp mib event trigger owner owner1 name EventA
Router(config-event-trigger)# test existence
Router(config-event-trigger-existence)# startup
Router(config-event-trigger-existence)# end
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>test</td>
<td>Enables a trigger test.</td>
</tr>
</tbody>
</table>
startup (test threshold)

To specify whether an event can be triggered for the threshold trigger test, use the `startup` command in event trigger threshold configuration mode. To disable the configured settings, use the `no` form of this command.

```
startup {rising|falling|rise-or-falling}
no startup
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rising</td>
<td>Specifies the rising threshold value to check against the set value during startup when the trigger type is threshold.</td>
</tr>
<tr>
<td>falling</td>
<td>Specifies the falling threshold value to check against the set value during startup when the trigger type is threshold.</td>
</tr>
<tr>
<td>rise-or-falling</td>
<td>Specifies the rising or falling threshold value to check against the set value during startup when the trigger type is threshold. This is the default value.</td>
</tr>
</tbody>
</table>

**Command Default**
The rising or falling threshold value is checked against the set value during startup when the trigger type is threshold.

**Command Modes**
Event trigger threshold configuration (config-event-trigger-threshold)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.4(20)T</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(33)SRE</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRE.</td>
</tr>
<tr>
<td>12.2(50)SY</td>
<td>This command was integrated into Cisco IOS Release 12.2(50)SY.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
The `startup` command starts an event when conditions for the threshold trigger test are met.

**Examples**
The following example shows how to specify startup for the threshold trigger test:

```
Router(config)# snmp mib event trigger owner owner1 name EventA
Router(config)# test threshold
Router(config-event-trigger-threshold)# startup rising
Router(config-event-trigger-threshold)# end
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>test</td>
<td>Enables a trigger test.</td>
</tr>
</tbody>
</table>
test (event trigger)

To specify the type of test to perform during an event trigger, use the test command in event trigger configuration mode. To disable the trigger test configuration settings, use the no form of this command.

```
test {existence| boolean| threshold}
no test {existence| boolean| threshold}
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>existence</td>
<td>Enables the existence trigger test.</td>
</tr>
<tr>
<td>boolean</td>
<td>Enables the Boolean trigger test. Boolean test is the default trigger test performed during event triggers.</td>
</tr>
<tr>
<td>threshold</td>
<td>Enables the threshold trigger test.</td>
</tr>
</tbody>
</table>

**Command Default**
The Boolean trigger test is enabled by default.

**Command Modes**
Event trigger configuration (config-event-trigger)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.4(20)T</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(33)SRE</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRE.</td>
</tr>
<tr>
<td>12.2(50)SY</td>
<td>This command was integrated into Cisco IOS Release 12.2(50)SY.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
The trigger table in the Event MIB has supplementary tables for additional objects that are configured based on the type of test performed for the trigger. For each trigger entry type such as existence, threshold, or Boolean, the corresponding tables (existence, threshold, and Boolean tables) are populated with the information required to perform the test. You can set event triggers based on existence, threshold, and Boolean trigger types.

The existence trigger tests are performed based on the following parameters:

- Absent
- Present
- Changed

The Boolean tests are comparison tests that are performed based on one of the following parameters:
• Unequal
• Equal
• Less
• Less Or Equal
• Greater
• Greater Or Equal

The threshold tests are performed based on the following parameters:
• Rising
• Falling
• Rising or Falling

Examples

The following example shows how to enable the existence trigger test:

Router(config)# snmp mib event trigger owner owner1 name triggerA
Router(config-event-trigger)# test existence
Router(config-event-trigger-existence)#

The following example shows how to enable the Boolean trigger test:

Router(config)# snmp mib event trigger owner owner1 name EventA
Router(config-event-trigger)# test boolean
Router(config-event-trigger-boolean)#

The following example shows how to enable the threshold trigger test:

Router(config)# snmp mib event trigger owner owner1 name triggerA
Router(config-event-trigger)# test threshold
Router(config-event-trigger-threshold)#

Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>comparison</td>
<td>Specifies the type of Boolean comparison to be performed.</td>
</tr>
<tr>
<td>event owner</td>
<td>Specifies the event owner for an event trigger according to the trigger type and status of the trigger.</td>
</tr>
<tr>
<td>object list</td>
<td>Configures a list of objects during an event.</td>
</tr>
<tr>
<td>startup</td>
<td>Specifies whether an event can be triggered for the Boolean, existence, or threshold trigger test.</td>
</tr>
<tr>
<td>value</td>
<td>Sets a value for the Boolean trigger test.</td>
</tr>
</tbody>
</table>
test snmp trap auth-framework sec-violation

To test CISCO-AUTH-FRAMEWORK-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the test snmp trap auth-framework sec-violation command in privileged EXEC mode.

Syntax Description
This command has no keywords or arguments.

Command Default
This command has no default setting.

Command Modes
Privileged EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SX1</td>
<td>This command was introduced on the Supervisor Engine 720.</td>
</tr>
</tbody>
</table>

Examples

This example shows the output of the SNMP camSecurityViolationNotif trap when it is not configured:

Router# test snmp trap auth-framework sec-violation
cafSecurityViolationNotif was disabled.
Router#

This example shows the output of the SNMP camSecurityViolationNotif trap when it is configured:

Router# test snmp trap auth-framework sec-violation
cafSecurityViolationNotif was sent.
Router#
test snmp trap bridge

To test BRIDGE-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the **test snmp trap bridge** command in privileged EXEC mode.

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>newroot</td>
<td>Tests SNMP newRoot notifications.</td>
</tr>
<tr>
<td>topologychange</td>
<td>Tests SNMP topologyChange notifications.</td>
</tr>
</tbody>
</table>

**Command Default**

This command has no default setting.

**Command Modes**

Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of **test snmp trap bridge newroot** when snmp-server enable traps bridge newroot is not configured:

```
Router# test
   snmp trap bridge newroot
newRoot notification is disabled.
Router#
```

This example shows the output of **test snmp trap bridge newroot** when snmp-server enable traps bridge newroot is configured:

```
Router# test
   snmp trap bridge newroot
newRoot notification was sent.
Router#
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp-server enable traps bridge</td>
<td>Enables the SNMP BRIDGE-MIB notifications.</td>
</tr>
</tbody>
</table>
test snmp trap c6kxbar

To test CISCO-CAT6K-CROSSBAR-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the test snmp trap c6kxbar command in privileged EXEC mode.

test snmp trap c6kxbar {flowctrl-bus| intbus-crcvrd| intbus-crcexcd| swbus| tm-channel-above| tm-channel-below| tm-swbus-above| tm-swbus-below}

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowctrl-bus</td>
<td>Tests SNMP cc6kxbarFlowCtrlBusThrExcdNotif notifications.</td>
</tr>
<tr>
<td>intbus-crcvrd</td>
<td>Tests SNMP cc6kxbarIntBusCRCErrRcvrdNotif notifications.</td>
</tr>
<tr>
<td>intbus-crcexcd</td>
<td>Tests SNMP cc6kxbarIntBusCRCErrExcdNotif notifications.</td>
</tr>
<tr>
<td>swbus</td>
<td>Tests SNMP cc6kxbarSwBusStatusChangeNotif notifications.</td>
</tr>
<tr>
<td>tm-channel-above</td>
<td>Tests cc6kxbarTMChUtilAboveNotif notifications.</td>
</tr>
<tr>
<td>tm-channel-below</td>
<td>Tests cc6kxbarTMChUtilBelowNotif notifications.</td>
</tr>
<tr>
<td>tm-swbus-above</td>
<td>Tests cc6kxbarTMSwBusUtilAboveNotif notifications.</td>
</tr>
<tr>
<td>tm-swbus-below</td>
<td>Tests cc6kxbarTMSwBusUtilBelowNotif notifications.</td>
</tr>
</tbody>
</table>

Command Default

This command has no default setting.

Command Modes

Privileged EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720.</td>
</tr>
<tr>
<td>12.2(33)SXI5</td>
<td>Added tm-channel-above, tm-channel-below and tm-swbus-above, tm-channel-below keywords.</td>
</tr>
</tbody>
</table>
Usage Guidelines

- The **flowctrl-bus** keyword is supported on the Supervisor Engine 32 only.
- The **tm-channel-above** and **tm-channel-below** keywords are not supported on Supervisor Engine 32.

Examples

This example shows the output of the SNMP cc6kxbarFlowCtrlBusThrExcdNotif notification when it is not configured:

Router# test
    snmp trap c6kxbar flowctrl-bus
cc6kxbarFlowCtrlBusThrExcdNotif notification is disabled.
Router#

This example shows the output of the SNMP cc6kxbarFlowCtrlBusThrExcdNotif notification when it is configured:

Router# test
    snmp trap c6kxbar flowctrl-bus
cc6kxbarFlowCtrlBusThrExcdNotif notification was sent.
Router#

This example shows the output of the SNMP cc6kxbarIntBusCRCErrExcdNotif notification when it is not configured:

Router# test
    snmp trap c6kxbar intbus-crcexcd
cc6kxbarIntBusCRCErrExcdNotif notification is disabled.
Router#

This example shows the output of the SNMP cc6kxbarIntBusCRCErrExcdNotif notification when it is configured:

Router# test
    snmp trap c6kxbar intbus-crcexcd
cc6kxbarIntBusCRCErrExcdNotif notification was sent.
Router#

This example shows the output of the SNMP cc6kxbarIntBusCRCErrRcvrdNotif notification when it is not configured:

Router# test
    snmp trap c6kxbar intbus-crcvrd
cc6kxbarIntBusCRCErrRcvrdNotif notification is disabled.
Router#

This example shows the output of the SNMP cc6kxbarIntBusCRCErrRcvrdNotif notification when it is configured:

Router# test
    snmp trap c6kxbar intbus-crcvrd
cc6kxbarIntBusCRCErrRcvrdNotif notification was sent.
Router#

This example shows the output of the SNMP cc6kxbarSwBusStatusChangeNotif notification when it is not configured:

Router# test
    snmp trap c6kxbar swbus
cc6kxbarSwBusStatusChangeNotif notification is disabled.
Router#

This example shows the output of the SNMP cc6kxbarSwBusStatusChangeNotif notification when it is configured:

Router# test
    snmp trap c6kxbar swbus
cc6kxbarSwBusStatusChangeNotif notification was sent.
Router#
This example shows the output of the SNMP cc6kxbarTMChUtilAboveNotif notification when it is not configured:

```
Router# test snmp trap c6kxbar tm-channel-above
cc6kxbarTMChUtilAboveNotif notification is disabled.
Router#
```

This example shows the output of the SNMP cc6kxbarTMChUtilAboveNotif notification when it is configured:

```
Router# test snmp trap c6kxbar tm-channel-above
cc6kxbarTMChUtilAboveNotif notification was sent.
Router#
```

This example shows the output of the SNMP cc6kxbarTMChUtilBelowNotif notification when it is not configured:

```
Router# test snmp trap c6kxbar tm-channel-below
cc6kxbarTMChUtilBelowNotif notification is disabled.
Router#
```

This example shows the output of the SNMP cc6kxbarTMChUtilBelowNotif notification when it is configured:

```
Router# test snmp trap c6kxbar tm-channel-below
cc6kxbarTMChUtilBelowNotif notification was sent.
Router#
```

This example shows the output of the SNMP cc6kxbarTMSwBusUtilAboveNotif notification when it is not configured:

```
Router# test snmp trap c6kxbar tm-swbus-above
cc6kxbarTMSwBusUtilAboveNotif notification is disabled.
Router#
```

This example shows the output of the SNMP cc6kxbarTMSwBusUtilAboveNotif notification when it is configured:

```
Router# test snmp trap c6kxbar tm-swbus-above
cc6kxbarTMSwBusUtilAboveNotif notification was sent.
Router#
```

This example shows the output of the SNMP cc6kxbarTMSwBusUtilBelowNotif notification when it is not configured:

```
Router# test snmp trap c6kxbar tm-swbus-below
cc6kxbarTMSwBusUtilBelowNotif notification is disabled.
Router#
```

This example shows the output of the SNMP cc6kxbarTMSwBusUtilBelowNotif notification when it is configured:

```
Router# test snmp trap c6kxbar tm-swbus-below
cc6kxbarTMSwBusUtilBelowNotif notification was sent.
Router#
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp-server enable traps c6kxbar</td>
<td>Enables the SNMP c6kxbar notification traps.</td>
</tr>
</tbody>
</table>
test snmp trap call-home

To test CISCO-CALLHOME-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the test snmp trap call-home command in privileged EXEC mode.

test snmp trap call-home {message-send-fail| server-fail}

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message-send-fail</td>
<td>Tests SNMP ccmSmtpMsgSendFailNotif notifications.</td>
</tr>
<tr>
<td>server-fail</td>
<td>Tests SNMP ccmSmtpServerFailNotif notifications.</td>
</tr>
</tbody>
</table>

**Command Default**

This command has no default setting.

**Command Modes**

Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the SNMP ccmSmtpMsgSendFailNotif notification when it is not configured:

Router# test
    snmp trap call-home message-send-fail
ccmSmtpMsgSendFailNotif notification is disabled.
Router#

This example shows the output of the SNMP ccmSmtpMsgSendFailNotif notification when it is configured:

Router# test
    snmp trap call-home message-send-fail
ccmSmtpMsgSendFailNotif notification was sent.
Router#

This example shows the output of the SNMP ccmSmtpServerFailNotif notification when it is not configured:

Router# test
    snmp trap call-home server-fail
ccmSmtpServerFailNotif notification is disabled.
Router#

This example shows the output of the SNMP ccmSmtpServerFailNotif notification when it is configured:

Router# test
    snmp trap call-home server-fail
ccmSmtpServerFailNotif notification was sent.
Router#
test snmp trap config-copy

To verify the reception of config-copy notifications by the Network Management System (NMS) or the Simple Network Management Protocol (SNMP) manager in a simulated scenario, use the **test snmp trap config-copy** command in privileged EXEC mode.

**Syntax Description**

This command has no arguments or keywords.

**Command Modes**

Privileged EXEC (#)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(33)SRE</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRE.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The Config-Copy MIB facilitates the copying of SNMP agent configuration files to the startup configuration or the local Cisco IOS file system, and vice versa. The config-copy notifications are sent to the NMS or the SNMP manager to indicate the successful completion of config-copy operation to or from the SNMP agent.

**Examples**

The following example shows how to simulate the verification of config-copy traps:

```
Router# test snmp trap config-copy
Generating CONFIG-COPY-MIB trap
00:20:44: SNMP: Queuing packet to 10.2.14.2
00:20:44: SNMP: V2 Trap, reqid 2, errstat 0, erridx 0
sysUpTime.0 = 124470
snmpTrapOID.0 = ccCopyMIBTraps.1
ccCopyTable.1.5.2 = 10.10.10.10
ccCopyTable.1.6.2 =
ccCopyTable.1.10.2 = 3
ccCopyTable.1.11.2 = 124470
ccCopyTable.1.12.2 = 124470
Router#
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug snmp packet</td>
<td>Displays information about every SNMP packet sent or received by the router.</td>
</tr>
<tr>
<td>snmp-server enable traps</td>
<td>Enables all SNMP notification types that are available on your system.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>snmp-server host</td>
<td>Specifies the recipient of an SNMP notification operation.</td>
</tr>
</tbody>
</table>
**test snmp trap dhcp bindings**

To test the cdsBindingsNotification trap, use the `test snmp trap dhcp bindings` EXEC command.

`test snmp trap dhcp bindings`

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

This command has no default settings.

**Command Modes**

Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>Support for this command was introduced on the Catalyst 6500 series switch.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows how to test the cdsBindingsNotification traps:

```
Router# test snmp trap dhcp bindings
cdsBindingsNotification notification is disabled.
```
To test the `cdsBindingsNotification` trap, use the `test snmp trap dhcp-snooping bindings` privileged EXEC command.

```
Router# test snmp trap dhcp-snooping bindings
cdsBindingsNotification notification is disabled.
```
test snmp trap dot1x

To test CISCO-PAE-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the `test snmp trap dot1x` command in privileged EXEC mode.

```
test snmp trap dot1x {auth-fail-vlan| guest-vlan| no-auth-fail-vlan| no-guest-vlan}
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth-fail-vlan</td>
<td>Tests SNMP cpaeAuthFailVlanNotif notifications.</td>
</tr>
<tr>
<td>guest-vlan</td>
<td>Tests SNMP cpaeGuestVlanNotif notifications.</td>
</tr>
<tr>
<td>no-auth-fail-vlan</td>
<td>Tests SNMP cpaeNoAuthFailedVlanNotif notifications.</td>
</tr>
<tr>
<td>no-guest-vlan</td>
<td>Tests SNMP cpaeNoGuestVlanNotif notifications.</td>
</tr>
</tbody>
</table>

**Command Default**

This command has no default setting.

**Command Modes**

Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the SNMP cpaeAuthFailVlanNotif notification when it is not configured:

```
Router# test snmp trap dot1x auth-fail-vlan
cpaeAuthFailVlanNotif notification was disabled.
Router#
```

This example shows the output of the SNMP cpaeAuthFailVlanNotif notification when it is configured:

```
Router# test snmp trap dot1x auth-fail-vlan
cpaeAuthFailVlanNotif notification was sent.
Router#
```
test snmp trap entity-diag

To test CISCO-ENTITY-DIAG-MIB Simple Network Management Protocol (SNMP) notifications (traps
and informs), use the test snmp trap c6kxbarc command in privileged EXEC mode.

test snmp trap entity-diag {boot-up-fail| hm-test-recover| hm-thresh-reached| scheduled-test-fail}

Syntax Description

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot-up-fail</td>
<td>Tests SNMP ceDiagBootUpFailedNotif notifications.</td>
</tr>
<tr>
<td>hm-test-recover</td>
<td>Tests SNMP ceDiagHMTestRecoverNotif notifications.</td>
</tr>
<tr>
<td>hm-thresh-reached</td>
<td>Tests SNMP ceDiagHMThresholdReachedNotif notifications.</td>
</tr>
<tr>
<td>scheduled-test-fail</td>
<td>Tests SNMP ceDiagScheduledTestFailedNotif notifications.</td>
</tr>
</tbody>
</table>

Command Default

This command has no default setting.

Command Modes

Privileged EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SX</td>
<td>This command was introduced on the Supervisor Engine 720.</td>
</tr>
</tbody>
</table>

Examples

This example shows the output of the SNMP ceDiagBootupFailedNotif notification when it is not configured:

Router# test
    snmp trap entity-diag boot-up-fail
ceDiagBootupFailedNotif notification is disabled.
Router#

This example shows the output of the SNMP ceDiagBootupFailedNotif notification when it is configured:

Router# test
    snmp trap entity-diag boot-up-fail
ceDiagBootupFailedNotif notification was sent.
Router#

This example shows the output of the SNMP ceDiagHMTestRecoverNotif notification when it is not configured:

Router# test
    snmp trap dot1x hm-test-recover
ceDiagHMTestRecoverNotif notification is disabled.
Router#
This example shows the output of the SNMP ceDiagHMTestRecoverNotif notification when it is configured:

Router# test
snmp trap dot1x hm-test-recover
ceDiagHMTestRecoverNotif notification was sent.
Router#

This example shows the output of the SNMP ceDiagHMThresholdReachedNotif notification when it is not configured:

Router# test snmp trap entity-diag hm-thresh-reached
ceDiagHMThresholdReachedNotif notification is disabled.
Router#

This example shows the output of the SNMP ceDiagHMThresholdReachedNotif notification when it is configured:

Router# test snmp trap entity-diag hm-thresh-reached
ceDiagHMThresholdReachedNotif notification was sent.
Router#

This example shows the output of the SNMP ceDiagScheduledTestFailedNotif notification when it is not configured:

Router# test snmp trap entity-diag scheduled-test-fail
ceDiagHMThresholdReachedNotif notification is disabled.
Router#

This example shows the output of the SNMP ceDiagScheduledTestFailedNotif notification when it is configured:

Router# test snmp trap entity-diag scheduled-test-fail
ceDiagHMThresholdReachedNotif notification was sent.
Router#
test snmp trap errdisable ifevent

To test CISCO-ERR-DISABLE-MIB cErrDisableInterfaceEventRev1 Simple Network Management Protocol (SNMP) traps and informs, use the `test snmp trap errdisable ifevent` command in privileged EXEC mode.

**test snmp trap errdisable ifevent**

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

**Command Default**
This command has no default settings.

**Command Modes**
Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI4</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of `test snmp trap errdisable ifevent` when snmp-server enable traps errdisable is not configured:

```
Router# test
snmp trap errdisable ifevent
cErrDisableInterfaceEventRev1 notification is disabled.
Router#
```

This example shows the output of `test snmp trap errdisable ifevent` when snmp-server enable traps errdisable is configured:

```
Router# test
  snmp trap errdisable ifevent
cErrDisableInterfaceEventRev1 notification was sent.
Router#
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp-server enable traps errdisable</td>
<td>Enables SNMP errdisable notifications.</td>
</tr>
</tbody>
</table>
test snmp trap flex-links status

To test CISCO-FLEX-LINKS-MIB cflIfStatusChangeNotif traps Simple Network Management Protocol (SNMP) traps and informs, use the `test snmp trap flex-links status` command in privileged EXEC mode.

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

**Command Default**
This command has no default settings.

**Command Modes**
Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the SNMP cflIfStatusChangeNotif trap when it is not configured:

```
Router# test snmp trap flex-links status
cflIfStatusChangeNotif notification is disabled.
Router#
```

This example shows the output of the SNMP cflIfStatusChangeNotif trap when it is configured:

```
Router# test snmp trap flex-links status
  cflIfStatusChangeNotif notification was sent.
  Router#
```
test snmp trap fru-ctrl

To test CISCO-ENTITY-FRU-CONTROL-MIB traps Simple Network Management Protocol (SNMP) traps and informs, use the **test snmp trap fru-ctrl** command in privileged EXEC mode.

```
test snmp trap fru-ctrl {insert| module-status| power-status| ps-out-change| remove}
```

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>insert</td>
<td>Tests SNMP cefcFRUIInserted notifications.</td>
</tr>
<tr>
<td>module-status</td>
<td>Tests SNMP cefcModuleStatusChange notifications.</td>
</tr>
<tr>
<td>power-status</td>
<td>Tests SNMP cefcPowerStatusChange notifications.</td>
</tr>
<tr>
<td>ps-out-change</td>
<td>Tests SNMP cefcPowerSupplyOutputChange notifications.</td>
</tr>
<tr>
<td>remove</td>
<td>Tests SNMP cefcFRURemoved notifications.</td>
</tr>
</tbody>
</table>

### Command Default

This command has no default settings.

### Command Modes

Privileged EXEC mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

### Examples

This example shows the output of the test SNMP cefcFRUIInserted trap when it is not configured:

```
Router# test
    snmp trap fru-ctrl insert
cefcFRUInserted notification is disabled.
Router#
```

This example shows the output of the SNMP cefcFRUIInserted trap when it is configured:

```
Router# test
    snmp trap fru-ctrl insert
cefcFRUInserted notification was sent.
Router#
```
test snmp trap l2-control vlan

To test CISCO-ENTITY-FRU-CONTROL-MIB clcVLANMacLimitNotif traps Simple Network Management Protocol (SNMP) traps and informs, use the test snmp trap l2-control vlan command in privileged EXEC mode.

Syntax Description
This command has no keywords or arguments.

Command Default
This command has no default settings.

Command Modes
Privileged EXEC mode

Command History
Release Modification
12.2(33)SXI This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.

Examples
This example shows the output of the clcVLANMacLimitNotif trap when it is not configured:

Router# test
    snmp trap l2-control vlan
    clcVLANMacLimitNotif notification is disabled.
Router#

This example shows the output of the SNMP clcVLANMacLimitNotif trap when it is configured:

Router# test
    snmp trap l2-control vlan
    clcVLANMacLimitNotif notification was sent.
Router#
test snmp trap l2tc

To test CISCO-L2-TUNNEL-CONFIG-MIB traps Simple Network Management Protocol (SNMP) traps and informs, use the test snmp trap l2tc command in privileged EXEC mode.

test snmp trap l2tc {drop|shutdown|sys-threshold}

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drop</td>
<td>Tests SNMP cltcTunnelDropThresholdExceeded notifications.</td>
</tr>
<tr>
<td>shutdown</td>
<td>Tests SNMP cltcTunnelShutdwonThresholdExceeded notifications.</td>
</tr>
<tr>
<td>sys-threshold</td>
<td>Tests SNMP cltcTunnelSysDropThresholdExceeded notifications.</td>
</tr>
</tbody>
</table>

**Command Default**

This command has no default settings.

**Command Modes**

Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SX1</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the cltcTunnelDropThresholdExceeded trap when it is not configured:

```
Router# test
    snmp trap l2tc drop
cltcTunnelDropThresholdExceeded notification is disabled.
Router#
```

This example shows the output of the SNMP cltcTunnelDropThresholdExceeded trap when it is configured:

```
Router# test
    snmp trap l2tc drop
cltcTunnelDropThresholdExceeded notification was sent.
Router#```
test snmp trap mac-notification

To test CISCO-MAC-NOTIFICATION-MIB traps Simple Network Management Protocol (SNMP) traps and informs, use the **test snmp trap mac-notification** command in privileged EXEC mode.

**test snmp trap mac-notification** \{change|move|threshold\}

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>change</td>
<td>Tests SNMP cmnMacChangeNotification notifications.</td>
</tr>
<tr>
<td>move</td>
<td>Tests SNMP cmnMacMoveNotification notifications.</td>
</tr>
<tr>
<td>threshold</td>
<td>Tests SNMP cmnMacThresholdExceedNotification notifications.</td>
</tr>
</tbody>
</table>

**Command Default**

This command has no default settings.

**Command Modes**

Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the SNMP cmnMacChangeNotification trap when it is not configured:

```
Router# test
snmp trap mac-notification change
cmnMacChangeNotification notification is disabled.
Router#
```

This example shows the output of the SNMP cmnMacChangeNotification trap when it is configured:

```
Router# test
snmp trap mac-notification change
cmnMacChangeNotification notification was sent.
Router#
```
test snmp trap module-auto-shutdown

To test CISCO-MODULE-AUTO-SHUTDOWN-MIB traps Simple Network Management Protocol (SNMP) traps and informs, use the `test snmp trap module-auto-shutdown` command in privileged EXEC mode.

```
test snmp trap module-auto-shutdown {auto-shutdown| sys-action}
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>auto-shutdown</th>
<th>Tests SNMP cmasModuleAutoShutdown notifications.</th>
</tr>
</thead>
<tbody>
<tr>
<td>sys-action</td>
<td>Tests SNMP cmasModuleSysActionNotif notifications.</td>
<td></td>
</tr>
</tbody>
</table>

**Command Default**

This command has no default settings.

**Command Modes**

Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the SNMP cmasModuleAutoShutdown trap when it is not configured:

```
Router# test
    snmp trap module-auto-shutdown auto-shutdown
cmasModuleAutoShutdown notification is disabled.
Router#
```

This example shows the output of the SNMP cmasModuleAutoShutdown trap when it is configured:

```
Router# test
    snmp trap module-auto-shutdown auto-shutdown
cmasModuleAutoShutdown notification is sent.
Router#
```
test snmp trap port-security

To test CISCO-PORT-SECURITY-MIB traps Simple Network Management Protocol (SNMP) traps and informs, use the test snmp trap port-security command in privileged EXEC mode.

test snmp trap port-security {ifvlan-mac| mac}

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ifvlan-mac</td>
<td>Tests SNMP cpsIfVlanSecureMacAddrViolation notifications.</td>
</tr>
<tr>
<td>mac</td>
<td>Tests SNMP cpsSecureMacAddrViolation notifications.</td>
</tr>
</tbody>
</table>

Command Default

This command has no default settings.

Command Modes

Privileged EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

Examples

This example shows the output of the SNMP cpsIfVlanSecureMacAddrViolation trap when it is not configured:

Router# test
   snmp trap port-security ifvlan-mac
   cpsIfVlanSecureMacAddrViolation notification is disabled.
Router#

This example shows the output of the SNMP cpsIfVlanSecureMacAddrViolation trap when it is configured:

Router# test
   snmp trap port-security ifvlan-mac
   cpsIfVlanSecureMacAddrViolation notification was sent.
Router#
test snmp trap power-ethernet port-on-off

To test POWER-ETHERNET-MIB traps Simple Network Management Protocol (SNMP) traps and informs, use the test snmp trap power-ethernet command in privileged EXEC mode.

test snmp trap power-ethernet port-on-off

Syntax Description
This command has no keywords or arguments.

Command Default
This command has no default settings.

Command Modes
Privileged EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

Examples

This example shows the output of the SNMP pethPsePortOnOffNotification trap when it is not configured:

```
Router# test
snmp trap power-ethernet port-on-off
pethPsePortOnOffNotification notification is disabled.
Router#
```

This example shows the output of the SNMP pethPsePortOnOffNotification trap when it is configured:

```
Router# test
snmp trap power-ethernet port-on-off
pethPsePortOnOffNotification notification was sent.
Router#
```
test snmp trap snmp

To verify the reception of Simple Network Management Protocol (SNMP) notifications by the Network Management System (NMS) or the SNMP manager in a simulated scenario, use the test snmp trap snmp command in privileged EXEC mode.

test snmp trap snmp {authentication| coldstart| linkup| linkdown| warmstart}

Syntax Description

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>authentication</strong></td>
<td>Verifies the generation and reception of the SNMP authentication failure notification by the SNMP manager. The authentication failure trap indicates that the SNMP agent has received a protocol message from the SNMP manager that is not properly authenticated.</td>
</tr>
<tr>
<td><strong>coldstart</strong></td>
<td>Verifies the generation and reception of the SNMP coldStart notifications by the SNMP manager. A coldStart trap indicates that the SNMP agent is reinitializing and its configuration may have changed.</td>
</tr>
<tr>
<td><strong>linkup</strong></td>
<td>Verifies the generation and reception of the SNMP linkUp notifications by the SNMP manager. A linkUp trap indicates if a communication link represented in the agent's configuration is activated.</td>
</tr>
<tr>
<td><strong>linkdown</strong></td>
<td>Verifies the generation and reception of the SNMP linkDown notifications by the SNMP manager. A linkDown trap indicates if a communication link represented in the agent’s configuration fails.</td>
</tr>
<tr>
<td><strong>warmstart</strong></td>
<td>Verifies the generation and reception of the SNMP warmStart notifications by the SNMP manager. A warmStart trap indicates that the SNMP agent is reinitializing and its configuration is not modified.</td>
</tr>
</tbody>
</table>

Command Modes

Privileged EXEC (#)

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(33)SRE</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRE.</td>
</tr>
</tbody>
</table>
Usage Guidelines
SNMP traps or notifications provide information about improper user authentication, restarts, closing of a connection, loss of connection to a neighbor router, or other significant events to the NMS.

Before testing the SNMP traps, configure the SNMP manager for the device and enable SNMP traps.

Examples
The following example shows how to simulate the verification of the authentication failure trap:

```
Router# test snmp trap snmp authentication
Generating Authentication failure trap
Sep 12 08:37:49.935: SNMP: Queuing packet to 10.4.9.2
Sep 12 08:37:49.935: SNMP: V1 Trap, ent snmpTraps, addr 192.168.0.1, gentrap 4
lsystem.5.0 = 10.10.10.10
  ciscoMgmt.412.1.1.2.0 = 10.10.10.10
Sep 12 08:38:55.995: SNMP: Packet sent via UDP to 10.4.9.2
Sep 12 08:38:56.263: SNMP: Packet sent via UDP to 10.4.9.2
```

Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug snmp packet</td>
<td>Displays information about every SNMP packet sent or received by the router.</td>
</tr>
<tr>
<td><strong>snmp-server enable traps</strong></td>
<td>Enables all SNMP notification types that are available on your system.</td>
</tr>
<tr>
<td><strong>snmp-server host</strong></td>
<td>Specifies the recipient of an SNMP notification operation.</td>
</tr>
</tbody>
</table>
test snmp trap stack

To test CISCO-STACK-MIB traps Simple Network Management Protocol (SNMP) traps and informs, use the test snmp trap stack command in privileged EXEC mode.

```
test snmp trap stack {chassis-off| chassis-on| module-down| module-up}
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chassis-off</td>
<td>Test SNMP chassisAlarmOff notifications.</td>
</tr>
<tr>
<td>chassis-on</td>
<td>Tests SNMP chassisAlarmOn notifications.</td>
</tr>
<tr>
<td>module-down</td>
<td>Tests SNMP moduleDown notifications.</td>
</tr>
<tr>
<td>module-up</td>
<td>Tests SNMP moduleUp notifications.</td>
</tr>
</tbody>
</table>

**Command Default**

This command has no default settings.

**Command Modes**

Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the SNMP chassisAlarmOff trap when it is not configured:

```
Router# test snmp trap stack chassis-off
chassisAlarmOff notification is disabled.
Router#
```

This example shows the output of the SNMP chassisAlarmOff trap when it is configured:

```
Router# test snmp trap stack chassis-off
chassisAlarmOff notification was sent.
Router#
```
test snmp trap storm-control

To test the Simple Network Management Protocol (SNMP) CISCO-PORT-STORM-CONTROL-MIB traps, use the **test snmp trap storm-control** command in privileged EXEC mode.

```
test snmp trap storm-control event-rev1
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event-rev1</td>
<td>Tests the cpscEventRev1 trap.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command Modes</th>
<th>Privileged EXEC (#)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Command History</th>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12.2(33)SXJ</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usage Guidelines</th>
<th>SNMP traps or notifications provide information about storm-control events.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Examples</th>
<th>The following example shows how to test the SNMP CISCO-PORT-STORM-CONTROL-MIB trap:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Router# test snmp trap storm-control event-rev1 cpscEventRev1 notification was sent. Router#</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>snmp-server enable traps storm-control</td>
<td>Enables SNMP storm-control trap notifications.</td>
</tr>
<tr>
<td></td>
<td>snmp-server host</td>
<td>Specifies the recipient of an SNMP notification operation.</td>
</tr>
</tbody>
</table>
test snmp trap stpx

To test CISCO-STP-EXTENSIONS-MIB traps Simple Network Management Protocol (SNMP) traps and informs, use the `test snmp trap stpx` command in privileged EXEC mode.

```
Router# test snmp trap stpx inconsistency
stpxInconsistencyUpdate notification is disabled.
Router#
```

This example shows the output of the SNMP stpxInconsistencyUpdate trap when it is not configured:

```
Router# test
snmp trap stpx inconsistency
stpxInconsistencyUpdate notification was sent.
Router#
```

This example shows the output of the SNMP stpxInconsistencyUpdate trap when it is configured:

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inconsistency</td>
<td>Tests SNMP stpxInconsistencyUpdate notifications.</td>
</tr>
<tr>
<td>loop-inconsistency</td>
<td>Tests SNMP stpxLoopInconsistencyUpdate notifications.</td>
</tr>
<tr>
<td>root-inconsistency</td>
<td>Tests SNMP stpxRootInconsistencyUpdate notifications.</td>
</tr>
</tbody>
</table>

**Command Default**

This command has no default settings.

**Command Modes**

Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the SNMP stpxInconsistencyUpdate trap when it is not configured:

```
Router# test snmp trap stpx inconsistency
stpxInconsistencyUpdate notification is disabled.
Router#
```

This example shows the output of the SNMP stpxInconsistencyUpdate trap when it is configured:

```
Router# test snmp trap stpx inconsistency
stpxInconsistencyUpdate notification was sent.
Router#
```
test snmp trap syslog

To verify the reception of the system logging message Simple Network Management Protocol (SNMP) notifications by the SNMP manager in a simulated scenario, use the test snmp trap syslog command in privileged EXEC mode.

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

**Command Modes**
Privileged EXEC (#)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(33)SRE</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRE.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
System logging messages are status notification messages that are generated by the routing device during operation. These messages are typically logged to a destination such as the terminal screen, or to a remote syslog host.

**Examples**
The following example shows how to replicate a syslog trap and its reception by the NMS:

```
Router# test snmp trap syslog
Generating SYSLOG-MIB Trap
00:07:25: SNMP: Queuing packet to 10.4.9.2
00:07:25: SNMP: V1 Trap, ent ciscoSyslogMIB.2, addr 192.16.12.8, gentrap 6, spectra
clogHistoryEntry.2.1 = TEST
clogHistoryEntry.3.1 = 5
clogHistoryEntry.4.1 = 1.3.6.1.4.1.9.9.10.1
clogHistoryEntry.5.1 = Syslog test trap
clogHistoryEntry.6.1 = 44596
00:07:25: SNMP: Queuing packet to 10.4.9.2
00:07:25: SNMP: V2 Trap, reqid 4, errstat 0, erridx 0
sysUpTime.0 = 44596
snmpTrapOID.0 = ciscoSyslogMIB.2.0.1
clogHistoryEntry.2.1 = TEST
clogHistoryEntry.3.1 = 5
clogHistoryEntry.4.1 = 1.3.6.1.4.1.9.9.10.1
clogHistoryEntry.5.1 = Syslog test trap
clogHistoryEntry.6.1 = 44596
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug snmp packet</td>
<td>Displays information about every SNMP packet sent or received by the router.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td><code>snmp-server enable traps</code></td>
<td>Enables all SNMP notification types that are available on your system.</td>
</tr>
<tr>
<td><code>snmp-server host</code></td>
<td>Specifies the recipient of an SNMP notification operation.</td>
</tr>
</tbody>
</table>
**test snmp trap trustsec**

To test CISCO-TRUSTSEC-MIB Simple Network Management Protocol (SNMP) notification (traps and informs), use the `test snmp trap trustsec` command in privileged EXEC mode.

```
test snmp trap trustsec {authz-file-error| cache-file-error| keystore-file-error| keystore-sync-fail| random-number-fail| src-entropy-fail}
```

**Syntax Description**

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>keystore-sync-fail</td>
<td>Tests SNMP ctsSwKeystoreSyncFailNotif notifications.</td>
</tr>
<tr>
<td>random-number-fail</td>
<td>Tests SNMP ctsSapRandomNumberFailNotif notifications.</td>
</tr>
</tbody>
</table>

**Command Modes**

Privileged EXEC (#).

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1(1)SY</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the test SNMP ctsAuthzCacheFileErrNotif trap when it is not configured:

```
Device# test snmp trap trustsec authz-file-error
cctsAuthzCacheFileErrNotif notification is disabled.
```

This example shows the output of the test SNMP ctsAuthzCacheFileErrNotif trap when it is configured:

```
Device# test snmp trap trustsec authz-file-error
cctsAuthzCacheFileErrNotif notification was sent.
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>snmp-server enable traps trustsec</code></td>
<td>Enables SNMP trustsec notification traps and informs.</td>
</tr>
</tbody>
</table>
test snmp trap trustsec-interface

To test CISCO-TRUSTSEC-INTERFACE-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the **test snmp trap trustsec-interface** command in privileged EXEC mode.

```
  test snmp trap trustsec-interface {authc-fail| authz-fail| sap-fail| supplicant-fail| unauthorized}
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authc-fail</td>
</tr>
<tr>
<td>authz-fail</td>
</tr>
<tr>
<td>sap-fail</td>
</tr>
<tr>
<td>supplicant-fail</td>
</tr>
<tr>
<td>unauthorized</td>
</tr>
</tbody>
</table>

**Command Modes**

Privileged EXEC (#)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1(1)SY</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the test SNMP ctsIfAuthenticationFailNotif trap when it is not configured:

```
Device# test snmp trap trustsec-interface authc-fail
ctsIfAuthenticationFailNotif notification is disabled.
```

This example shows the output of the test SNMP ctsIfAuthenticationFailNotif trap when it is configured:

```
Device# test snmp trap trustsec-interface authc-fail
ctsIfAuthenticationFailNotif notification was sent.
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp-server enable traps trustsec-interface</td>
<td>Enables SNMP trustsec-interface notification traps and informs.</td>
</tr>
</tbody>
</table>
test snmp trap trustsec-policy

To test CISCO-TRUSTSEC-POLICY-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the **test snmp trap trustsec-policy** command in privileged EXEC mode.

**test snmp trap trustsec-policy {authz-sgacl-fail| peer-policy-updated}**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>authz-sgacl-fail</th>
<th>Tests SNMP ctspAuthorizationSgaclFailNotif notifications.</th>
</tr>
</thead>
<tbody>
<tr>
<td>peer-policy-updated</td>
<td>Tests SNMP ctspPeerPolicyUpdatedNotif notifications.</td>
<td></td>
</tr>
</tbody>
</table>

**Command Modes**

Privileged EXEC (#)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1(1)SY</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the test SNMP ctspAuthorizationSgaclFailNotif trap when it is not configured:

```
Device# test snmp trap trustsec-policy authz-sgacl-fail
ctspAuthorizationSgaclFailNotif notification is disabled.
```

This example shows the output of the test SNMP ctspAuthorizationSgaclFailNotif trap when it is configured:

```
Device# test snmp trap trustsec-policy authz-sgacl-fail
ctspAuthorizationSgaclFailNotif notification was sent.
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp-server enable traps trustsec-policy</td>
<td>Enables SNMP trustsec-policy notification traps and informs.</td>
</tr>
</tbody>
</table>
test snmp trap trustsec-server

To test CISCO-TRUSTSEC-SERVER-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the `test snmp trap trustsec-server` command in privileged EXEC mode.

```
test snmp trap trustsec-server {provision-secret|radius-server}
```

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>provision-secret</td>
<td>Tests SNMP ctsvNoProvisionSecretNotif notifications.</td>
</tr>
<tr>
<td>radius-server</td>
<td>Tests SNMP ctsvNoRadiusServerNotif notifications.</td>
</tr>
</tbody>
</table>

### Command Modes

Privileged EXEC (#)

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1(1)SY</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Examples

This example shows the output of the test SNMP ctsvNoProvisionSecretNotif trap when it is not configured:

```
Device# test snmp trap trustsec-server provision-secret
ctsNoProvisionSecretNotif notification is disabled.
```

This example shows the output of the test SNMP ctsvNoProvisionSecretNotif trap when it is configured:

```
Device# test snmp trap trustsec-sxp-server provision-secret
ctsNoProvisionSecretNotif notification was sent.
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp-server enable traps trustsec-server</td>
<td>Enables SNMP trustsec-server notification traps and informs.</td>
</tr>
</tbody>
</table>
test snmp trap trustsec-sxp

To test CISCO-TRUSTSEC-SXP-MIB Simple Network Management Protocol (SNMP) notification (traps and informs), use the test snmp trap trustsec-sxp command in privileged EXEC mode.

test snmp trap trustsec-sxp {binding-conflict| binding-err| binding-expn-fail| conn-config-err| conn-down| conn-srcaddr-err| conn-up| msg-parse-err| oper-nodeid-change}

Syntax Description

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>binding-conflict</td>
<td>Tests SNMP ctsxSxpBindingConflictNotif notifications.</td>
</tr>
<tr>
<td>binding-err</td>
<td>Tests SNMP ctsxSxpBindingErrNotif notifications.</td>
</tr>
<tr>
<td>binding-expn-fail</td>
<td>Tests SNMP ctsxSxpBindingExpnFailNotif notifications.</td>
</tr>
<tr>
<td>conn-config-err</td>
<td>Tests SNMP ctsxSxpConnConfigErrNotif notifications.</td>
</tr>
<tr>
<td>conn-down</td>
<td>Tests SNMP ctsxSxpConnDownNotif notifications.</td>
</tr>
<tr>
<td>conn-srcaddr-err</td>
<td>Tests SNMP ctsxSxpConnSourceAddrErrNotif notifications.</td>
</tr>
<tr>
<td>conn-up</td>
<td>Tests SNMP ctsxSxpConnUpNotif notifications.</td>
</tr>
<tr>
<td>oper-nodeid-change</td>
<td>Tests SNMP ctsxSxpOperNodeIdChangeNotif notifications.</td>
</tr>
</tbody>
</table>

Command Modes

Privileged EXEC (#)

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1(1)SY</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Examples

This example shows the output of the test SNMP ctsxSxpBindingConflictNotif trap when it is not configured:

Device# test snmp trap trustsec-sxp binding-conflict
ctsxSxpBindingConflictNotif notification is disabled.
This example shows the output of the test SNMP ctsxSxpBindingConflictNotif trap when it is configured:

```
Device# test snmp trap trustsec-sxp binding-conflict
ctsxSxpBindingConflictNotif notification was sent.
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>snmp-server enable traps trustsec-sxp</code></td>
<td>Enables SNMP trustsec-sxp notification traps and informs.</td>
</tr>
</tbody>
</table>
test snmp trap udld

To test CISCO-UDLDP-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the `test snmp trap udld` command in privileged EXEC mode.

`test snmp trap udld` {link-fail-rpt | status-change}

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>link-fail-rpt</td>
<td>Tests SNMP cudldFastHelloLinkFailRptNotification notifications.</td>
</tr>
<tr>
<td>status-change</td>
<td>Tests SNMP cudldFastHelloStatusChangeNotification notifications.</td>
</tr>
</tbody>
</table>

**Command Default**

This command has no default setting.

**Command Modes**

Privileged EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI4</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows the output of the SNMP cudldFastHelloLinkFailRptNotification notification when it is not configured:

```
Router# test
    snmp trap udld link-fail-rpt
cudldFastHelloLinkFailRptNotification notification is disabled.
Router#
```

This example shows the output of the SNMP cudldFastHelloLinkFailRptNotification notification when it is configured:

```
Router# test
    snmp trap udld link-fail-rpt
cudldFastHelloLinkFailRptNotification notification was sent.
Router#
```
**test snmp trap vswitch dual-active**

To test whether the CISCO-VIRTUAL-SWITCH-MIB Simple Network Management Protocol (SNMP) notification (trap) can be generated in the dual-active state, use the `test snmp trap vswitch dual-active` command in privileged EXEC mode.

```
Device(config)# test snmp trap vswitch dual-active
```

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

The CISCO-VIRTUAL-SWITCH-MIB SNMP notification is not sent.

**Command Modes**

Privileged EXEC (#)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1(1)SY</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `snmp-server enable traps vswitch dual-active` command enables the dual-active state change notification. When the VSS changes state to dual-active, the SNMP agent sends the `cvsDualActiveDetectionNotif` notification.

Enable the `snmp-server enable traps vswitch dual-active` command before running the `test snmp trap vswitch dual-active` command.

**Examples**

The following is sample output from the `test snmp trap vswitch dual-active` command when the SNMP `cvsDualActiveDetectionNotif` notification is enabled:

```
Device(config)# snmp-server enable traps vswitch dual-active
Device(config)# exit
Device# test snmp trap vswitch dual-active

cvsDualActiveDetectionNotif notification was sent.
```

The following is sample output from the `test snmp trap vswitch dual-active` command when the SNMP `cvsDualActiveDetectionNotif` notification is disabled:

```
Device(config)# no snmp-server enable traps vswitch dual-active
Device(config)# exit
Device# test snmp trap vswitch dual-active

cvsDualActiveDetectionNotif notification is disabled.
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>snmp-server enable traps vswitch dual-active</code></td>
<td>Enables the CISCO-VIRTUAL-SWITCH-MIB SNMP cvsDualActiveDetectionNotif notification.</td>
</tr>
<tr>
<td><code>test snmp trap vswitch vsl</code></td>
<td>Tests the CISCO-VIRTUAL-SWITCH-MIB SNMP notification (trap and inform).</td>
</tr>
</tbody>
</table>
**test snmp trap vsswitch vsl**

To test CISCO-VIRTUAL-SWITCH-MIB Simple Network Management Protocol (SNMP) notifications (traps and informs), use the test **snmp trap vsswitch vsI** command in privileged EXEC mode.

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

```
Command Default
```
This command has no default setting.

```
Command Modes
```
Privileged EXEC mode

```
Command History
```
<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720.</td>
</tr>
</tbody>
</table>

```
Examples
```
This example shows the output of the SNMP cvsVSLConnectionChangeNotif notification when it is not enabled:

```
Router# test
snmp trap vsswitch vsI
```
cvsVSLConnectionChangeNotif notification is disabled.
Router#

This example shows the output of the SNMP cvsVSLConnectionChangeNotif notification when it is enabled:

```
Router# test
snmp trap vsswitch vsI
```
cvsVSLConnectionChangeNotif notification was sent.
Router#
test snmp trap vtp

To test CISCO-VTP-MIB traps Simple Network Management Protocol (SNMP) traps and informs, use the test snmp trap vtp command in privileged EXEC mode.

test snmp trap vtp {digest-error| mode-change| port-status| pruning-change| rev-error| server-disable| v1-detected| version-change| vlan-create| vlan-delete}

Syntax Description

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>digest-error</td>
<td>Tests SNMP vtpConfigDigestError notifications.</td>
</tr>
<tr>
<td>mode-change</td>
<td>Tests SNMP vtpLocalModeChange notifications.</td>
</tr>
<tr>
<td>port-status</td>
<td>Tests SNMP vlanTrunkPortDynamicStatusChange notifications.</td>
</tr>
<tr>
<td>pruning-change</td>
<td>Tests SNMP vtpPruningStateOperChange notifications.</td>
</tr>
<tr>
<td>rev-error</td>
<td>Tests SNMP vtpConfigRevNumberError notifications.</td>
</tr>
<tr>
<td>server-disable</td>
<td>Tests SNMP vtpServerDisabled notifications.</td>
</tr>
<tr>
<td>v1-detected</td>
<td>Tests SNMP vtpVersionOneDeviceDetected notifications.</td>
</tr>
<tr>
<td>version-change</td>
<td>Tests SNMP vtpVersionInUseChanged notifications.</td>
</tr>
<tr>
<td>vlan-create</td>
<td>Tests SNMP vtpVlanCreated notifications.</td>
</tr>
<tr>
<td>vlan-delete</td>
<td>Tests SNMP vtpVlanDeleted notifications.</td>
</tr>
</tbody>
</table>

Command Default

This command has no default settings.

Command Modes

Privileged EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI</td>
<td>This command was introduced on the Supervisor Engine 720 and Supervisor Engine 32.</td>
</tr>
</tbody>
</table>
This example shows the output of the SNMP vtpConfigDigestError trap when it is not configured:

Router# test
    snmp trap vtp digest-error
   vtpConfigDigestError notification is disabled.
Router#

This example shows the output of the SNMP vtpConfigDigestError trap when it is configured:

Router# test
    snmp trap vtp digest-error
   vtpConfigDigestError notification was sent.
Router#
**test snmp trap vtp pruning-change**

To test the vtpPruningStateOperChange trap, use the `test snmp trap vtp pruning-change` EXEC command.

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

### Command Default
This command has no default settings.

### Command Modes
EXEC mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.2(33)SXI4</td>
<td>Support for this command was introduced on the Catalyst 6500 series.</td>
</tr>
</tbody>
</table>

### Examples
This example shows that testing the vtpPruningStateOperChange cannot occur without first enabling SNMP VTP traps:

```
Router# test snmp trap vtp pruning-change
vtpPruningStateOperChange notification is disabled.
```

This example shows how to test the vtpPruningStateOperChange:

```
Router# test snmp trap vtp pruning-change
vtpPruningStateOperChange notification is sent.
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp-server enable traps vtp</td>
<td>Enables SNMP VTP traps.</td>
</tr>
</tbody>
</table>
**type (test existence)**

To specify the type of existence trigger test to perform, use the `type` command in event trigger existence configuration mode. To disable the specified trigger test type, use the `no` form of this command.

```plaintext
type {present|absent|changed}
no type {present|absent|changed}
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>present</td>
<td>Specifies whether the trigger conditions for the existence test are present.</td>
</tr>
<tr>
<td>absent</td>
<td>Specifies whether the trigger conditions for the existence test are absent.</td>
</tr>
<tr>
<td>changed</td>
<td>Specifies whether the trigger conditions for the existence test are changed.</td>
</tr>
</tbody>
</table>

**Command Default**
By default, both present and absent tests are performed.

**Command Modes**
Event trigger existence configuration (config-event-trigger-existence)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.4(20)T</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(33)SRE</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRE.</td>
</tr>
<tr>
<td>12.2(50)SY</td>
<td>This command was integrated into Cisco IOS Release 12.2(50)SY.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
The existence trigger tests are performed based on the following parameters:

- Absent
- Present
- Changed

When the test type is not specified, both present and absent tests are performed.
The following example shows how to specify the existence trigger test as present:

```text
Router(config)#snmp mib event trigger owner owner1 name triggerA
Router(config-event-trigger)# test existence
Router(config-event-trigger-existence)# type present
Router(config-event-trigger-existence)# end
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>test</td>
<td>Enables a trigger test.</td>
</tr>
</tbody>
</table>
url (bulk statistics)

To specify the host to which bulk statistics files should be transferred, use the `url` command in Bulk Statistics Transfer configuration mode. To remove a previously configured destination host, use the `no` form of this command.

```
url {primary secondary} url
no url {primary secondary}
```

**Syntax Description**

<table>
<thead>
<tr>
<th>primary</th>
<th>Specifies the URL to be used first for bulk statistics transfer attempts.</th>
</tr>
</thead>
<tbody>
<tr>
<td>secondary</td>
<td>Specifies the URL to be used for bulk statistics transfer attempts if the transfer to the primary URL is not successful.</td>
</tr>
</tbody>
</table>

```
url
```

Destination URL address for the bulk statistics file transfer. Use FTP, RCP, or TFTP. The Cisco IOS File System (IFS) syntax for these URLs is as follows:

- **ftp**: `[[[[username][:password]]@]]location[/directory]/filename`
- **rcp**: `[[[[username]]@]]location[/directory]/filename`
- **tftp**: `[[[location]]/directory]/filename`

The `location` argument is typically an IP address.

**Command Default**

No host is specified.

**Command Modes**

Bulk Statistics Transfer configuration (config-bulk-tr)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.0(24)S</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.3(2)T</td>
<td>This command was integrated into Cisco IOS Release 12.3(2)T.</td>
</tr>
<tr>
<td>12.2(25)S</td>
<td>This command was integrated into Cisco IOS Release 12.2(25)S.</td>
</tr>
<tr>
<td>12.2(33)SRA</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRA.</td>
</tr>
<tr>
<td>12.2(33)SXH</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SXH.</td>
</tr>
</tbody>
</table>
This command was integrated into Cisco IOS Release 12.2(33)SB.

This command was integrated into Cisco IOS Release XE 2.1.

Usage Guidelines

For bulk statistics transfer retry attempts, a single retry consists of an attempt to send first to the primary URL, and then to the secondary URL.

Examples

In the following example, an FTP server is used as the primary destination for the bulk statistics file. If a transfer to that address fails, an attempt is made to send the file to the TFTP server at 192.168.10.5. No retry command is specified, which means that only one attempt to each destination will be made.

```
Router(config)# snmp mib bulkstat transfer ifMibTesting
Router(config-bulk-tr)# schema carMibTesting1
Router(config-bulk-tr)# schema carMibTesting2
Router(config-bulk-tr)# format bulkBinary
Router(config-bulk-tr)# transfer-interval 60
Router(config-bulk-tr)# buffer-size 10000
Router(config-bulk-tr)# url primary ftp://user2:pswd@192.168.10.5/functionality/
Router(config-bulk-tr)# url secondary tftp://user2@192.168.10.8/tftpboot/
Router(config-bulk-tr)# buffer-size 2500000
Router(config-bulk-tr)# enable
Router(config-bulk-tr)# exit
```

Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>retry (bulk statistics)</td>
<td>Configures the number of retries that should be attempted for sending bulk statistics files.</td>
</tr>
<tr>
<td>snmp mib bulkstat transfer</td>
<td>Names a bulk statistics transfer configuration and enters Bulk Statistics Transfer configuration mode.</td>
</tr>
</tbody>
</table>
value (test boolean)

To set a value for the Boolean trigger test, use the `value` command in event trigger boolean configuration mode. To disable the configured settings, use the `no` form of this command.

```
value integer-value
no value
```

**Syntax Description**

| integer-value | Numerical value to set for the Boolean test. The default is 0. |

**Command Default**

The Boolean trigger test value is set to 0.

**Command Modes**

Event trigger boolean configuration (config-event-trigger-boolean)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.4(20)T</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(33)SRE</td>
<td>This command was integrated into Cisco IOS Release 12.2(33)SRE.</td>
</tr>
<tr>
<td>12.2(50)SY</td>
<td>This command was integrated into Cisco IOS Release 12.2(50)SY.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `value` command specifies the value to be set for the Boolean trigger test.

**Examples**

The following example shows how to set a value for the Boolean trigger test:

```
Router(config)# snmp mib event trigger owner owner1 name triggerA
Router(config-event-trigger)# test boolean
Router(config-event-trigger-boolean)# value 10
Router(config-event-trigger-boolean)# end
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>test</td>
<td>Enables a trigger test.</td>
</tr>
</tbody>
</table>
value type

To specify the type of bulkstat expression to use during object sampling, use the `value type` command in Bulkstat expression configuration mode. To disable the specified value type, use the `no` form of this command.

```
value type [counter32|unsigned32|timeticks|integer32[ipaddress|octetstring|objectid|counter64]
no value type
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>counter32</td>
<td>Specifies a counter32 value. Counter32 specifies a value that represents a count. The value ranges from 0 to 4,29,49,67,295.</td>
</tr>
<tr>
<td>unsigned32</td>
<td>Specifies an unsigned integer value. Unsigned32 specifies a value that includes only non-negative integers. The value ranges from 0 to 4294967295.</td>
</tr>
<tr>
<td>timeticks</td>
<td>Specifies a value based on timeticks. Timeticks represents a non-negative integer value that specifies the elapsed time between two events, in units of hundredth of a second. When objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1), the description of the object type identifies this reference period.</td>
</tr>
<tr>
<td>integer32</td>
<td>Specifies an integer32 value. The Integer32 represents 32-bit signed integer values for the Simple Network Management Protocol (SNMP). The value range includes both negative and positive numbers.</td>
</tr>
<tr>
<td>ipaddress</td>
<td>Specifies a value based on the IP address. The IP address is a string of four octets. The IP address value type is generally an IPv4 address. This value is encoded as four bytes in the network byte order.</td>
</tr>
<tr>
<td>octetstring</td>
<td>Specifies a value based on octetstring. The octetstring specifies octets of binary or textual information. The octet string length ranges from 0 to 65535 octets.</td>
</tr>
<tr>
<td>objectid</td>
<td>Specifies a value based on the object identifier of an object. Each object type in a MIB is identified by an object identifier value assigned by the administrator. The object identifier identifies the value type that has an assigned object identifier value.</td>
</tr>
<tr>
<td>counter64</td>
<td>Specifies a counter64 value that represents a count. However, the counter64 value ranges from 0 to 18446744073709551615. This value type is used when a 32-bit counter rollover occurs in less than an hour.</td>
</tr>
</tbody>
</table>
By default, the value type is not configured.

expression configuration (config-expression)

Bulkstat data set expression configuration (config-bs-ds-expr)

This command was introduced.12.4(20)T

This command was integrated into Cisco IOS Release 12.2(33)SRE.12.2(33)SRE

This command was integrated into Cisco IOS Release 12.2(50)SY.12.2(50)SY

This command was integrated into Cisco IOS Release 15.3(1)S.15.3(1)S

This command was integrated into Cisco IOS Release XE 3.8S.Cisco IOS Release XE 3.8S

The value type command specifies a value for expression evaluation.

The following example shows how to specify the counter32 value type:

Device> enable
Device# configure terminal
Device(config)# snmp mib expression owner owner1 name ExpressionA
Device(config-expression)# value type counter32

The following example shows how to specify the counter32 value type for Bulkstat expression data set:

Device> enable
Device# configure terminal
Device(config)# bulkstat data interface-util type expression
Device(config-bs-ds-expr)# expression 100*$1+$2
Device(config-bs-ds-expr)# value type counter32

Command

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp mib expression owner</td>
<td>Specifies the owner for an expression.</td>
</tr>
<tr>
<td>bulkstat data</td>
<td>Configures Bulkstat data set for expression type.</td>
</tr>
</tbody>
</table>
**wildcard (expression)**

To specify whether an object used for evaluating an expression is to be wildcarded during an event configuration, use the `wildcard` command in expression configuration mode. To remove the wildcard object identifier, use the `no` form of this command.

```plaintext
wildcard
no wildcard
```

**Syntax Description**

This command has no arguments or keywords.

**Command Default**

This command is enabled by default.

**Command Modes**

Expression configuration (config-expression)

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.4(20)T</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>12.2(50)SY</td>
<td>This command was integrated into Cisco IOS Release 12.2(50)SY.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `wildcard` command allows you to apply a single expression to multiple instances of the same MIB object. When you specify this choice and provide a partial object identifier, the application obtains the object values and discovers the instances of the object. By default, the objects are identified based on instances and are not wildcarded.

**Examples**

The following example shows how to specify the wildcard object identifier by using the `wildcard` command:

```plaintext
Router(config)# snmp mib expression owner owner1 name expression1
Router(config-expression)# object 2
Router(config-expression-object)# wildcard
Router(config-expression-object)# end
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object id</td>
<td>Specifies the object identifier of an object associated with an event.</td>
</tr>
<tr>
<td>snmp mib expression owner</td>
<td>Specifies the owner of an expression.</td>
</tr>
</tbody>
</table>
write mib-data

To save MIB data to system memory (NVRAM) for MIB Data Persistence, use the `write mib-data` command in EXEC mode.

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.0(1)M</td>
<td>This command was introduced in a release earlier than Cisco IOS Release 15.0(1)M.</td>
</tr>
<tr>
<td>12.2(33)SRC</td>
<td>This command was integrated into a release earlier than Cisco IOS Release 12.2(33)SRC.</td>
</tr>
<tr>
<td>12.2(33)SXI</td>
<td>This command was integrated into a release earlier than Cisco IOS Release 12.2(33)SXI.</td>
</tr>
<tr>
<td></td>
<td>This command was implemented on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
</tbody>
</table>

Usage Guidelines

The MIB Data Persistence feature allows the SNMP data of a MIB to be persistent across reloads; that is, the values of certain MIB objects are retained even if your networking device reboots.

To determine which MIBs support "MIB Persistence" in your release, use the `snmp mib persist` command in global configuration mode.

Any modified MIB data must be written to NVRAM memory using the `write mib-data` command. If the `write mib-data` command is not used, modified MIB data is not saved automatically, even if MIB Persistence is enabled. Executing the `write mib-data` command saves only the current MIB data; if the MIB object values are changed, you should reenter the `write mib-data` command to ensure that those values are persistent across reboots.

Examples

The following example shows the enabling of event MIB persistence, circuit MIB persistence, and saving the changes to set object values for these MIBs to NVRAM:

```
Router# configure terminal
Router(config)# snmp mib persist circuit
Router(config)# snmp mib persist event
Router(config)# end
Router# write mib-data
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>snmp mib persist</code></td>
<td>Enables MIB data persistence.</td>
</tr>
</tbody>
</table>
startup (test boolean) through write mib-data