



Bidirectional Forwarding Detection MIB

The Bidirectional Forwarding Detection (BFD) MIB, Version 2 feature enables Simple Network Management Protocol (SNMP) agent support in Cisco IOS software for BFD management, as implemented in the Bidirectional Forwarding Detection Management Information Base (draft-ietf-bfd-mib-02.txt). The SNMP agent code operating with the BFD MIB enables a standardized, SNMP-based approach to be used in managing the BFD features in Cisco IOS software. The BFD MIB feature introduces the CISCO-IETF-BFD-MIB. The BFD MIB is also VPN-aware, which allows SNMP to differentiate incoming packets from different VPNs.

- [Finding Feature Information, page 1](#)
- [Restrictions for the Bidirectional Forwarding Detection MIB, page 1](#)
- [Information About the Bidirectional Forwarding Detection MIB, page 2](#)
- [How to Configure the Bidirectional Forwarding Detection MIB, page 7](#)
- [Configuration Examples for the Bidirectional Forwarding Detection MIB, page 9](#)
- [Additional References, page 12](#)
- [Feature Information for the Bidirectional Forwarding Detection MIB, page 13](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for the Bidirectional Forwarding Detection MIB

The following restrictions apply to the BFD MIB for Cisco IOS releases:

- This MIB supports read-only (RO) permission for MIB objects, except for `ciscoBfdSessNotificationsEnable`, which has read-write access to enable or disable BFD traps via SNMP set commands.

- The BFD Session Mapping Table (ciscoBfdSessMapTable) maps the complex indexing of the BFD sessions to the flat BFDIndex used in the ciscoBfdSessionTable.
- BFD does not support 64-bit counters. The session performance table (ciscoBfdSessionPerfTable) collects BFD performance counts on a per session basis. This table augments the ciscoBfdSessionTable.
- The VRF-Aware functionality of BFD MIB is not supported with IPv6 addresses.

Information About the Bidirectional Forwarding Detection MIB

BFD MIB Cisco Implementation

The BFD MIB is based on the Internet Engineering Task Force (IETF) draft MIB entitled draft-ietf-bfd-mib-02.txt which includes objects describing features that support BFD.

Slight differences between the IETF draft MIB and the implementation of the BFD capabilities within Cisco IOS software require some minor translations between the BFD MIB and the internal data structures of Cisco IOS software. These translations are made by the SNMP agent code that is installed and operating on various hosts within the network. This SNMP agent code, running in the background as a low priority process, provides a management interface to Cisco IOS software.

The SNMP objects defined in the BFD MIB can be displayed by any standard SNMP utility. All BFD MIB objects are based on the IETF draft MIB; thus, no specific Cisco SNMP application is required to support the functions and operations pertaining to the BFD MIB.

Capabilities Supported by the BFD MIB

The following functionality is supported in the BFD MIB:

- The ability to generate and queue notification messages that signal changes in the operational status of BFD sessions.
- The ability to make the BFD MIB VPN aware.
- Extensions to existing SNMP commands that provide the ability to enable, disable, and configure notification messages for BFD sessions.
- The ability to specify the name or the IP address of a network management station (NMS) in the operating environment to which notification messages are to be sent.
- The ability to write notification configurations into nonvolatile memory.

Notification Generation Events

When BFD notifications are enabled with the **snmp-serverenabletrapsbfd** command with the **session-up** and **session-down** keywords, notification messages relating to specific events within Cisco IOS software are generated and sent to a specified NMS in the network.

For example, a bfdSessUp notification is sent to an NMS when BFD is configured.

Conversely, a bfdSessDown notification is generated and sent to an NMS when BFD is disabled.

Benefits of Bidirectional Forwarding Detection MIB

The BFD MIB provides the following benefits:

- Provides a standards-based SNMP interface for retrieving information about BFD.
- Forwards notification messages to a designated NMS for evaluation or action by network administrators.

Features and Technologies Related to BFD MIB

The BFD MIB feature is used in conjunction with the following features and technologies:

- Standards-based SNMP network management application
- BFD

Supported Objects in the BFD MIB

BFD General Variables (scalars)

The following parameters apply globally to the router's BFD process:

- `ciscoBfdAdminStatus` is The global administrative status of BFD in this router. The value `enabled` denotes that the BFD Process is active on at least one interface; `disabled` means it is not enabled on any interface.
- `ciscoBfdVersionNumber` is the current default version number of the BFD protocol.
- `ciscoBfdSessNotificationsEnable` enables the emission of `ciscoBfdSessUp` and `ciscoBfdSessDown` notifications when set to `true` (1); otherwise these notifications are not emitted.

BFD Session Table

The BFD Session Table specifies BFD session specific information and contains the following entries:

- `ciscoBfdSessTable` describes the BFD sessions.
- `ciscoBfdSessEntry` describes BFD session.
- `ciscoBfdSessIndex` contains an index used to represent a unique BFD session on this device. This is an Index and it does not show up in the MIB walk as an object.
- `ciscoBfdSessApplicationId` contains an index used to indicate a local application which owns or maintains this BFD session. This application ID provides a convenient way to segregate sessions by the applications that maintain them. The value corresponds to the `ClientID` in the output of the `showbfdclient` command.
- `ciscoBfdSessDiscriminator` specifies the local discriminator for this BFD session, used to uniquely identify it.
- `ciscoBfdSessRemoteDiscr` specifies the session discriminator chosen by the remote system for this BFD session.

- `ciscoBfdSessUdpPort` specifies the UDP Port for BFD. The default value is the well-known value for this port.
- `ciscoBfdSessState` specifies the perceived state of the BFD session. Valid values are `adminDown` (1), `down` (2), `init` (3), and `up` (4).
- `ciscoBfdSessRemoteHeardFlag` specifies status of BFD packet reception from the remote system. The flag is set to true (1) if the local system is actively receiving BFD packets from the remote system. The flag is set to false (0) if the local system has not received BFD packets recently (within the detection time) or if the local system is attempting to tear down the BFD session. This object is applicable only if the session is running at version 0. If the session is running version 1, that value will return false.
- `ciscoBfdSessDiag` displays a diagnostic code specifying the local system's reason for the last transition of the session from up (1) to some other state. This object is accessible only for notifications and will not display in the MIB walk for the `ciscoBfdSessTable`. The codes are:
 - `BfdInterval`—The delay in microseconds.
 - `BfdDiag`—A diagnostic code:
 - `noDiagnostic`(0)
 - `controlDetectionTimeExpired`(1)
 - `echoFunctionFailed`(2)
 - `neighborSignaledSessionDown`(3)
 - `forwardingPlaneReset`(4)
 - `pathDown`(5)
 - `concatenatedPathDown`(6)
 - `administrativelyDown`(7)
 - `reverseConcatenatedPathDown` (8)
- `ciscoBfdSessOperMode` specifies the current operating mode of the BFD session. The supported values are:
 - `asyncModeWEchoFun` (1),
 - `asynchModeWOEchoFun` (2),
- `ciscoBfdSessDemandModeDesiredFlag` indicates the local system's desire to use demand mode. It is set to true (1) if the local system wishes to use demand mode or false (0) if not. Demand Mode is not supported and therefore will always return a value of 0.
- `ciscoBfdSessEchoFuncModeDesiredFlag` indicates that the local system's desire to use echo mode. It is set to true (1) if the local system wishes to use Echo mode or false (0) if not.
- `ciscoBfdSessControlPlanIndepFlag` indicates if the local system's can function through a disruption of the control plane. It is set to true (1) if the local system BFD implementation is independent of the control plane. Otherwise, the value is set to false (0). This value will always return a value of 0.
- `ciscoBfdSessAddrType` specifies the IP address of the interface associated with this BFD session. Only values `unknown` (0), `ipv4` (1) or `ipv6` (2) are supported. A value of `unknown` (0) is allowed only when

the outgoing interface is of type point-to-point, or when the BFD session is not associated with a specific interface.

- `ciscoBfdSessAddr` specifies the IP address of the interface associated with this BFD session. The value is set to zero when BFD session is not associated with a specific interface.
- `ciscoBfdSessDesiredMinTxInterval` specifies the minimum interval, in microseconds, that the local system would like to use when transmitting BFD control packets.
- `ciscoBfdSessReqMinRxInterval` specifies the minimum interval, in microseconds, between received BFD control packets the local system can support.
- `ciscoBfdSessReqMinEchoRxInterval` specifies the minimum interval, in microseconds, between received BFD Echo packets that this system can support. If echo mode is disabled for the configured interface for the session, this object will return value of 0.
- `ciscoBfdSessDetectMult` specifies the detect time multiplier.
- `ciscoBfdSessStorType` indicates the storage type for this object. The storage type for this entry is a read-only implementation that is always volatile.
- `ciscoBfdSessRowStatus` This object is a read-only implementation that is always active.
- `ciscoBfdSessAuthPresFlag` indicates the local system's desire to use Authentication. It is set to true (1) if the local system wishes the session to be authenticated or false (0) if not. Authentication is not supported and this object will always return a value of 0.
- `ciscoBfdSessAuthenticationType` specifies the authentication type used for this BFD session. This field is valid only when the authentication present bit is set. This object is not valid in BFD in Cisco IOS.

BFD Session Performance Table

`ciscoBfdSessPerfTable` specifies BFD session performance counters and augments the `ciscoBfdSessionTable`. This table contains the following entries:

- `ciscoBfdSessPerfEntry` includes an entry created by a BFD-enabled node for every BFD session. `ciscoBfdCounterDiscontinuityTime` is used to indicate potential discontinuity for all counter objects in this table.
- `ciscoBfdSessPerfPktIn` specifies the total number of BFD messages received for this BFD session.
- `ciscoBfdSessPerfPktOut` specifies the total number of BFD messages sent for this BFD session.
- `ciscoBfdSessUpTime` specifies the value of `sysUpTime` on the most recent occasion at which the session came up. If no such up event exists, the value is zero.
- `ciscoBfdSessPerfLastSessDownTime` specifies the value of `sysUpTime` on the most recent occasion at which the last time communication was lost with the neighbor. If no such down event exists, the value is zero.
- `ciscoBfdSessPerfLastCommLostDiag` specifies the BFD diag code for the last time communication was lost with the neighbor. This object is not supported.
- `ciscoBfdSessPerfSessUpCount` specifies the number of times this session has gone into the up state since the router last rebooted.
- `ciscoBfdSessPerfDiscTime` indicates the value of `sysUpTime` on the most recent occasion at which any one or more of the session counters suffered a discontinuity. The relevant counters are the specific

instances associated with this BFD session of any Counter32 object contained in the BfdSessPerfTable. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then the value is zero. This object is not supported.

- `ciscoBfdSessPerfPktInHC` represents the total number of BFD messages received for this BFD session. It must be equal to the least significant 32 bits of `ciscoBfdSessPerfPktIn` if `ciscoBfdSessPerfPktInHC` is supported according to the rules spelled out in RFC2863.
- `ciscoBfdSessPerfPktOutHC` represents the total number of BFD messages transmitted for this BFD session. It must be equal to the least significant 32 bits of `ciscoBfdSessPerfPktIn` if `ciscoBfdSessPerfPktOutHC` is supported according to the rules spelled out in RFC2863.

BFD Session Mapping Table

The BFD Session Mapping Table maps the complex indexing of the BFD sessions to the flat BFDIndex used in the `ciscoBfdSessionTable`. If the value of the `ciscoBfdSessAddr` (an OID) has more than 111 sub-identifiers, then OIDs of column instances in this table have more than 128 sub-identifiers and cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3. The BFD Session Mapping table contains the following entries:

- `ciscoBfdSessMapEntry` describes BFD session that is mapped to this index. If the value of the `mplsInSegmentMapLabelPtrIndex` (an OID) has more than 111 sub-identifiers, then OIDs of column instances in this table have more than 128 sub-identifiers and cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3.
- `ciscoBfdSessMapBfdIndex` specifies the BfdIndex referred to by the indexes of this row. In essence, a mapping is provided between these indexes and the `ciscoBfdSessTable`. This is Index and does not show up in the MIB walk as an object.

See the MIB Walk for BFD MIB: Example in the configuration example section for an example of the mapping.

BFD Notifications

Notification contains the following entries. The range mode for this notification is not supported. Therefore, only a single notification is sent for one of the `ciscoBfdSessTable` entries representing this session.

- `ciscoBfdSessUp` generates a notification when the `ciscoBfdSessState` object for one or more entries in `ciscoBfdSessTable` is about to enter the up (4) state from some other state. The value of `ciscoBfdSessDiag` is set equal to `noDiagnostic(0)`.
- `ciscoBfdSessDown` generates a notification when the `ciscoBfdSessState` object for one or more entries in `ciscoBfdSessTable` is about to enter the down (2) or adminDown (1) states from some other state. The values of `ciscoBfdSessDiag` returns the Diag code providing the reason for this new state (that is, `pathDown (5)` or `administrativelyDown (7)`).

How to Configure the Bidirectional Forwarding Detection MIB

Enabling the SNMP Agent for BFD MIB Notifications

The SNMP agent for the BFD MIB is disabled by default. To enable the SNMP agent for BFD MIB notifications, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show running-config | includesnmp**
3. **configure terminal**
4. **snmp-server community** *string* [**view***view-name*] [**ro** | **rw**] [**ipv6nacl**] [*access-list-number*]
5. **snmp-server enable traps bfd** [*session-up*] [*session-down*]
6. **exit**
7. **write memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router# enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show running-config includesnmp Example: Router# show running-config include snmp	Displays the running configuration to determine if an SNMP agent is already running. <ul style="list-style-type: none"> • If no SNMP information is displayed, go to Step 4 . If any SNMP information is displayed, you can modify the information or change it as needed.
Step 3	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 4	snmp-server community <i>string</i> [view <i>view-name</i>] [ro rw] [ipv6nacl] [<i>access-list-number</i>] Example: Router(config)# snmp-server community comaccess ro 4	Enables the community string. <ul style="list-style-type: none"> • The example enables snmp with community string comaccess and read-only access.

	Command or Action	Purpose
Step 5	snmp-server enable traps bfd [session-up] [session-down] Example: <pre>Router(config)# snmp-server enable traps bfd</pre>	Enables a router to send SNMP notifications or informs to an SNMP host. Note This command is optional. After SNMP is enabled, all MIBs are available for the user to query.
Step 6	exit Example: <pre>Router(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.
Step 7	write memory Example: <pre>Router# write memory</pre>	Writes the modified configuration to NVRAM, permanently saving the settings.

Verifying the Status of the SNMP Agent

To verify that the SNMP agent has been enabled on a host network device, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show running-config | includesnmp**
3. **show bfd neighbors detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router# enable</pre>	Enables privileged EXEC mode <ul style="list-style-type: none"> • Enter your password if prompted
Step 2	show running-config includesnmp Example: <pre>Router# show running-config include snmp</pre>	Displays the running configuration on the target device and is used to examine the output for displayed SNMP information.

	Command or Action	Purpose
Step 3	show bfd neighbors detail Example: Router# show bfd neighbors detail	Displays BFD protocol parameters and timers for each neighbor.

Example

The following example displays the running configuration on the target device and its SNMP information.

```
Router# show running-config | include snmp
.
.
.
snmp-server community public rw
snmp-server community private ro
```

Any **snmp-server** statement that appears in the output and takes the form shown here verifies that SNMP has been enabled on that device.

Configuration Examples for the Bidirectional Forwarding Detection MIB

Enabling the SNMP Agent to Enable BFD Notifications Example

The following example shows how to enable an SNMP agent on a host network device:

```
Router# configure terminal
Router(config)# snmp-server community
privatero
```

The following example shows how to allow read-only access to all BFD MIB objects relating to members of access list 4 that specify the comaccess community string. No other SNMP agents will have access to any BFD MIB objects.

```
Router(config)# snmp-server community comaccess ro 4
```

The following example shows how to enable a router to send BFD-related SNMP notifications or informs to an SNMP host.

```
Router(config)# snmp-server enable traps bfd
```

Viewing BFD Sessions Example

The following example shows the output of the **show bfd neighbors** command, which displays BFD sessions and timers for each neighbor.

```
Router# show bfd neighbors
```

```

NeighAddr          LD/RD    RH/RS    State    Int
10.0.0.2           7/7     Up       Up       Et1/2.2
10.1.0.2           6/6     Up       Up       Et1/2.5
DDDD::1            1/1     Up       Up       Et1/3

```

Router# **show bfd neighbors detail**

```

NeighAddr          LD/RD    RH/RS    State    Int
10.0.0.2           9/8     Up       Up       Gi3/8.1
Session state is UP and using echo function with 50 ms interval.
OurAddr: 10.0.0.1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 5
Received MinRxInt: 1000000, Received Multiplier: 5
Holddown (hits): 0(0), Hello (hits): 1000(350)
Rx Count: 352, Rx Interval (ms) min/max/avg: 1/1000/874 last: 464 ms ago
Tx Count: 351, Tx Interval (ms) min/max/avg: 756/1000/876 last: 524 ms ago
Elapsed time watermarks: 0 0 (last: 0)
Registered protocols: CEF OSPF
Uptime: 00:05:07
Last packet: Version: 1 - Diagnostic: 0
State bit: Up - Demand bit: 0
Poll bit: 0 - Final bit: 0
Multiplier: 5 - Length: 24
My Discr.: 8 - Your Discr.: 9
Min tx interval: 1000000 - Min rx interval: 1000000
Min Echo interval: 50000
NeighAddr          LD/RD    RH/RS    State    Int
10.1.0.2           6/6     Up       Up       Gi3/8.2
Session state is UP and using echo function with 50 ms interval.
OurAddr: 10.1.0.1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 5
Received MinRxInt: 1000000, Received Multiplier: 5
Holddown (hits): 0(0), Hello (hits): 1000(352)
Rx Count: 352, Rx Interval (ms) min/max/avg: 1/1000/880 last: 248 ms ago
Tx Count: 354, Tx Interval (ms) min/max/avg: 1/1000/875 last: 244 ms ago
Elapsed time watermarks: 0 0 (last: 0)
Registered protocols: CEF OSPF
Uptime: 00:05:09
Last packet: Version: 1 - Diagnostic: 0
State bit: Up - Demand bit: 0
Poll bit: 0 - Final bit: 0
Multiplier: 5 - Length: 24
My Discr.: 6 - Your Discr.: 6
Min tx interval: 1000000 - Min rx interval: 1000000
Min Echo interval: 50000

```

MIB Walk for BFD MIB Example

This example shows sample output from a MIB walk of the BFD MIB:

ciscoBfdSessMapTable

```

ciscoBfdSessMapBfdIndex.1.7.1.4.10.1.0.1 = 65543
ciscoBfdSessMapBfdIndex.3.1.2.16.221.221.0.0.0.0.0.0.0.0.0.0.0.0.2 = 196609
ciscoBfdSessMapBfdIndex.4.6.1.4.40.4.0.1 = 262150
The MapTable index includes the following information about BFD sessions and clients:
Index example: 1.7.1.4.10.1.0.1
1 - Client id
7 - Local discriminator
1 - IP address type (1 - IPv4, 2- IPv6)
4 - Length of next string (4 for IPv4 addresses or 16 for IPv6 addresses)
10.1.0.1 - IP address of the BFD session

```

ciscoBfdSessTable

```

ciscoBfdSessApplicationId.65543 = 1
ciscoBfdSessApplicationId.196609 = 3
ciscoBfdSessApplicationId.262150 = 4
ciscoBfdSessDiscriminator.65543 = 7
ciscoBfdSessDiscriminator.196609 = 1
ciscoBfdSessDiscriminator.262150 = 6
ciscoBfdSessRemoteDiscr.65543 = 7
ciscoBfdSessRemoteDiscr.196609 = 1
ciscoBfdSessRemoteDiscr.262150 = 6
ciscoBfdSessUdpPort.65543 = 3785
ciscoBfdSessUdpPort.196609 = 3784
ciscoBfdSessUdpPort.262150 = 3785
ciscoBfdSessState.65543 = up
ciscoBfdSessState.196609 = up
ciscoBfdSessState.262150 = up
ciscoBfdSessRemoteHeardFlag.65543 = false
ciscoBfdSessRemoteHeardFlag.196609 = false
ciscoBfdSessRemoteHeardFlag.262150 = false
ciscoBfdSessOperMode.65543 = asyncModeWEchoFun
ciscoBfdSessOperMode.196609 = asynchModeWOEchoFun
ciscoBfdSessOperMode.262150 = asyncModeWEchoFun
ciscoBfdSessDemandModeDesiredFlag.65543 = false
ciscoBfdSessDemandModeDesiredFlag.196609 = false
ciscoBfdSessDemandModeDesiredFlag.262150 = false
ciscoBfdSessEchoFuncModeDesiredFlag.65543 = true
ciscoBfdSessEchoFuncModeDesiredFlag.196609 = false
ciscoBfdSessEchoFuncModeDesiredFlag.262150 = true
ciscoBfdSessControlPlanIndepFlag.65543 = false
ciscoBfdSessControlPlanIndepFlag.196609 = false
ciscoBfdSessControlPlanIndepFlag.262150 = false
ciscoBfdSessAddrType.65543 = ipv4
ciscoBfdSessAddrType.196609 = ipv6
ciscoBfdSessAddrType.262150 = ipv4
ciscoBfdSessAddr.65543 = 28:01:00:01
ciscoBfdSessAddr.196609 = DD:DD:00:00:00:00:00:00:00:00:00:00:00:00:00:02
ciscoBfdSessAddr.262150 = 10:04:00:01
ciscoBfdSessDesiredMinTxInterval.65543 = 1000000
ciscoBfdSessDesiredMinTxInterval.196609 = 50000
ciscoBfdSessDesiredMinTxInterval.262150 = 1000000
ciscoBfdSessReqMinRxInterval.65543 = 1000000
ciscoBfdSessReqMinRxInterval.196609 = 50000
ciscoBfdSessReqMinRxInterval.262150 = 1000000
ciscoBfdSessReqMinEchoRxInterval.65543 = 50000
ciscoBfdSessReqMinEchoRxInterval.196609 = 0
ciscoBfdSessReqMinEchoRxInterval.262150 = 50000
ciscoBfdSessDetectMult.65543 = 5
ciscoBfdSessDetectMult.196609 = 5
ciscoBfdSessDetectMult.262150 = 5
ciscoBfdSessStorType.65543 = volatile
ciscoBfdSessStorType.196609 = volatile
ciscoBfdSessStorType.262150 = volatile
ciscoBfdSessRowStatus.65543 = active
ciscoBfdSessRowStatus.196609 = active
ciscoBfdSessRowStatus.262150 = active
ciscoBfdSessAuthPresFlag.65543 = false
ciscoBfdSessAuthPresFlag.196609 = false
ciscoBfdSessAuthPresFlag.262150 = false
ciscoBfdSessAuthenticationType.65543 = 0
ciscoBfdSessAuthenticationType.196609 = 0
ciscoBfdSessAuthenticationType.262150 = 0

```

ciscoBfdSessPerfTable

```

ciscoBfdSessPerfPktIn.65543 = 246
ciscoBfdSessPerfPktIn.196609 = 5159
ciscoBfdSessPerfPktIn.262150 = 290
ciscoBfdSessPerfPktOut.65543 = 247
ciscoBfdSessPerfPktOut.196609 = 5416

```

```

ciscoBfdSessPerfPktOut.262150 = 291
ciscoBfdSessUpTime.65543 = 43376
ciscoBfdSessUpTime.196609 = 39781
ciscoBfdSessUpTime.262150 = 39736
ciscoBfdSessPerfLastSessDownTime.65543 = 0
ciscoBfdSessPerfLastSessDownTime.196609 = 0
ciscoBfdSessPerfLastSessDownTime.262150 = 0
ciscoBfdSessPerfLastCommLostDiag.65543 = 0
ciscoBfdSessPerfLastCommLostDiag.196609 = 0
ciscoBfdSessPerfLastCommLostDiag.262150 = 0
ciscoBfdSessPerfSessUpCount.65543 = 1
ciscoBfdSessPerfSessUpCount.196609 = 1
ciscoBfdSessPerfSessUpCount.262150 = 1
ciscoBfdSessPerfDiscTime.65543 = 0
ciscoBfdSessPerfDiscTime.196609 = 0
ciscoBfdSessPerfDiscTime.262150 = 0
ciscoBfdSessPerfPktInHC.65543 = 247
ciscoBfdSessPerfPktInHC.196609 = 5179
ciscoBfdSessPerfPktInHC.262150 = 291
ciscoBfdSessPerfPktOutHC.65543 = 248
ciscoBfdSessPerfPktOutHC.196609 = 5440
ciscoBfdSessPerfPktOutHC.262150 = 292

```

Additional References

Related Documents

Related Topic	Document Title
BFD	<i>IP Routing Bidirectional Forwarding Detection Configuration Guide</i>
Configuring SNMP support for a VPN	<i>SNMP Support over VPNs—Context-Based Access</i>

Standards and RFCs

Standard/RFC	Title
draft-ietf-bfd-mib-03	Bidirectional Forwarding Detection MIB
RFC 2026	The Internet Standards Process

MIBs

MIB	MIBs Link
BFD MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for the Bidirectional Forwarding Detection MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for the Bidirectional Forwarding Detection MIB

Feature Name	Releases	Feature Information
Bidirectional Forwarding Detection MIB, Version 2	12.2(33)SRE 15.1(1)SG 15.1(1)SY	<p>The Bidirectional Forwarding Detection MIB feature enables the SNMP agent support in Cisco IOS software for BFD management, as implemented in the CISCO-IETF-BFD-MIB.</p> <p>The following commands were introduced or modified:</p> <ul style="list-style-type: none"> • snmp-server enable traps bfd • snmp-server host

