



## Call Tracker plus ISDN and AAA Enhancements for the Cisco AS5300 and Cisco AS5800

---

This feature module describes the Call Tracker feature and a set of ISDN and AAA enhancements to expand the call handling and real-time monitoring capabilities of the Cisco AS5300 and Cisco AS5800 universal access servers. The Call Tracker feature captures detailed statistics on the status and progress of active calls and retains historical data for disconnected call sessions. Call Tracker collects session information such as call states and resources, traffic statistics, total bytes transmitted and received, user IP address, and disconnect reason. This data is maintained within the Call Tracker database tables, which are accessible through SNMP, CLI, or SYSLOG.

The ISDN enhancements provide additional call-handling functionality for incoming ISDN calls, including a timer for authentication responses, an override for ISDN cause codes, and a flag indicating support for B-channel busyouts. The AAA enhancements include authentication by dialed number identification service (DNIS), selective termination of call sessions, and expanded support for several RADIUS attributes.

This document includes the following sections:

- Feature Overview, page 2
- Supported Platforms, page 4
- Supported Standards, MIBs, and RFCs, page 5
- Prerequisites, page 5
- Configuration Tasks, page 6
- Configuration Examples, page 12
- Command Reference, page 16
- Debug Commands, page 60
- Glossary, page 65

# Feature Overview

This feature set provides additional call handling and monitoring functionality in the following areas:

- **Call Tracker**—A new subsystem for capturing detailed data on the progress and status of calls, from the time the network access server receives a setup request or allocates a channel, until a call is rejected, or terminated. This data is maintained within the Call Tracker database tables, which are accessible through SNMP, CLI, or SYSLOG. Session information for all active calls and calls in the setup state is stored in an active table, while records for disconnected calls are moved to a history table. Call Tracker is notified of applicable call events by related subsystems such as ISDN, PPP, CSM, Modem, EXEC, or TCP-Clear. SNMP traps are generated at the start of each call, when an entry is created in the active table, and at the end of each call, when an entry is created in the history table. Call Record SYSLOGs are available through configuration that will generate detailed information records for all call terminations. This information can be sent to SYSLOG servers for permanent storage and future analysis.
- **Modem service data**—The status and diagnostic data that is routinely collected from MICA modems is expanded to include new link statistics for active calls, such as the attempted transmit and receive rates, the maximum and minimum transmit and receive rates, and locally and remotely issued retrains and speedshift counters. This connection data is polled from the modem at user-defined intervals and passed to Call Tracker.
- **RADIUS attributes**—Define specific AAA elements in a user profile, which is stored on the RADIUS server. The RADIUS server sends the user profile to the network access server in the authentication Access-Accept packet. This feature set expands the functionality of authentication attribute 14 (Login-IP-Host) and accounting attribute 44 (Acct-Session-Id), and it introduces support for a new authentication attribute, attribute 76 (Prompt).
  - **Login-IP-Host**—IP address of the host to which the dial-in user is automatically connected at login. The user is connected directly to this host using the type of service indicated by the Login-Service attribute, such as TCP-Clear or Telnet. Cisco IOS functionality has been expanded to support up to three Login-IP-Host entries in RADIUS Access-Accept packets, which allows the network access server to attempt more than one host when trying to connect the user. The order in which the Login-IP-Host entries occur in the packet is the order in which a connection is attempted. The **ip tcp synwait-time** command determines how long the network access server waits before trying the next host in the list.
  - **Acct-Session-Id**—Unique integer linking accounting records for a call. Traditionally, the Acct-Session-Id is generated after authentication and is stored in the accounting record. To accommodate features such as DNIS authorization, you now have the option of including the Acct-Session-Id in Access-Request packets, so that authentication and accounting records can be linked.
  - **Prompt**—Indicates whether or not the network access server should echo user input to Access-Challenge responses. Previously, the only way to control the echoing of user input was by using the **radius-server challenge-noecho** command, which allowed you to suppress the echo behavior for all users. Support for the Prompt attribute allows you to control the behavior per individual user.
- **TCP-Clear connections**—The TCP system has been enhanced to provide additional connection information to Call Tracker, including:
  - The number and identity of hosts that were attempted before a connection was established, or the total of failed attempts if no connection was made.
  - The disconnect reason for active sessions, or the failure reason if the network access server failed to connect to a host before timing out.

- The source and destination endpoints for active sessions, consisting of the IP addresses and port numbers of the network access server and host.
- DSO Busyout—To busy out B channels on a PRI, the ISDN switch must support service messages. The new command, **isdn snmp busyout b-channel**, sets the MIB object, `cpmDS0BusyoutAllow`, indicating whether or not the switch supports service messages, thereby allowing the busyout of B channels. When the network access server receives an SNMP request for a busyout, it checks the value of this object. If the **no isdn snmp busyout b-channel** command is configured, the busyout request fails.
- DS0 status—MIB objects have been added to indicate the operational status of each DS0, the busyout state of DS0s, and whether service messages are supported by the ISDN switch, allowing a busyout. Specifically, the following objects are included in the CISCO-POP-MGMT-MIB:
  - `cpmDS0OperStatus`—DS0 status such as down, idle, setup, connected, or test.
  - `cpmDS0AdminStatus`—Busyout status of the DS0. This value can be set to up, busyout, or busyout immediate through SNMP or Cisco IOS software.
- DNIS preauthentication—Enables preauthentication at call setup based on the number dialed. The DNIS number is sent to the security server when a call is received. If authenticated by AAA, the call is accepted.
- Packet of disconnect (POD)—Terminates connections on the network access server when particular session attributes are identified. The POD client, residing on a UNIX workstation, sends disconnect packets to the POD server running on the network access server, by using session information obtained from AAA. The network access server terminates any inbound user session with one or more matching key attributes. It rejects requests that do not have the required fields or where an exact match is not found.
- ISDN guard timer—Implements a new managed timer for ISDN calls. Because response times for authentication requests can vary, for instance when using DNIS authentication, the guard timer allows you to control the handling of calls. The guard timer starts when the DNIS is sent to the RADIUS server. If the network access server does not receive a response from AAA before the guard timer expires, it accepts or rejects the call based on the configuration of the timer.
- ISDN cause code override—Overrides cause codes that are sent to ISDN applications. Cisco IOS software contains ISDN cause codes that handle specific functions such as modem availability and resource pooling. The ISDN cause code override feature is more general and overrides the specific ISDN cause codes.
- IP multicast heartbeat—Users of the multicast routing feature need a way to monitor the health of multicast delivery and be alerted when the delivery fails to meet certain parameters.

Although Multicast Routing Monitor (MRM) can be used for this purpose, it has two problems:

- It generates a SYSLOG message instead of an SNMP trap.
- It monitors the synthetic MRM test streams, but not a production multicast stream.

To meet user requirements, the following commands have been added or modified:

- **ipmulticast-heartbeat**: Keyword for the **snmp-server enable traps** command.
- **ip multicast heartbeat**: IP multicast command
- **debug ip mhbeat**: Debug command to monitor the action of the heartbeat trap

## Benefits

- Makes real-time monitoring of call activity easier and more comprehensive.
- Captures data for active and historical call sessions, allowing external applications to access the data using SNMP, CLI, or SYSLOG.
- Provides volume and usage statistics for call management decisions.
- Enables authentication using DNIS number, and provides a configurable timer for authentication responses.
- Allows the termination of PPP connections based on selected session parameters.
- IP Multicast Heartbeat allows the monitoring of the health of multicast delivery, and alerts when the delivery fails to meet certain parameters.

## Related Features and Technologies

The Cisco IOS Release 12.1 documentation set book titles provide information for:

- AAA, documented in the *Cisco IOS Security Configuration Guide*.
- ISDN, documented in the *Cisco IOS Dial Services Configuration Guide*.
- Modem management, documented in the *Cisco IOS Dial Services Configuration Guide*.
- SNMP, documented in *Cisco IOS Configuration Fundamentals Configuration Guide*.

## Related Documents

- *Cisco AS5300 Software Configuration Guide*
- *Cisco AS5800 OAM&P Guide*
- *Cisco IOS Dial Services Configuration Guide*, Cisco IOS Release 12.1
- *Cisco IOS Security Configuration Guide*, Cisco IOS Release 12.1
- *Cisco IOS Configuration Fundamentals Configuration Guide*, Cisco IOS Release 12.1

## Supported Platforms

- Cisco AS5300
- Cisco AS5800

# Supported Standards, MIBs, and RFCs

## Standards

No new or modified standards are supported by this feature.

## MIBs

- CISCO-CALL-TRACKER-MIB
- CISCO-CALL-TRACKER-MODEM-MIB
- CISCO-CALL-TRACKER-TCP-MIB
- CISCO-MODEM-MGMT-MIB
- CISCO-POP-MGMT-MIB
- CISCO-IPMROUTE-MIB

For descriptions of supported MIBs and how to use MIBs, see Cisco's MIB web site on CCO at: <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>.

## RFCs

RADIUS Extensions (draft-ietf-radius-ext04.txt)

# Prerequisites

Before configuring Call Tracker and its associated features, you must complete the following tasks on your network access server:

- Configure ISDN as described in *Cisco IOS Dial Services Configuration Guide*, Cisco IOS Release 12.1.
- Configure SNMP as described in *Cisco IOS Configuration Fundamentals Configuration Guide*, Cisco IOS Release 12.1.
- Configure AAA as described in *Cisco IOS Security Configuration Guide*, Cisco IOS Release 12.1.
- Define the characteristics of your RADIUS or TACACS+ security server as described in *Cisco IOS Security Configuration Guide*, Cisco IOS Release 12.1.

# Configuration Tasks

See the following sections for configuration tasks for Call Tracker and its related features. Each task in the list is identified as either optional or required.

- Configuring Call Tracker (Required)
- Verifying Call Tracker (Optional)
- Configuring Polling of Link Statistics (Optional)
- Configuring RADIUS Acct-Session-Id (Optional)
- Configuring RADIUS Login-IP-Host (Optional)
- Configuring RADIUS Prompt (Optional)
- Configuring AAA DNIS Authentication (Optional)
- Configuring AAA POD (Optional)
- Configuring ISDN PRI B-Channel Busyout (Optional)
- Configuring ISDN Cause Code Override (Optional)
- Configuring ISDN Guard Timer (Optional)
- Configuring IP Multicast Heartbeat (Optional)

## Configuring Call Tracker

To configure Call Tracker, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	Router# <b>config term</b>	Enters global configuration mode. You enter global configuration mode when the prompt changes to Router(config)#.
Step 2	Router(config)# <b>calltracker enable</b>	Enables Call Tracker.
Step 3	Router(config)# <b>calltracker history max-size</b> <i>number</i>	Sets the maximum call entries to store in the Call Tracker history table.
Step 4	Router(config)# <b>calltracker history retain-mins</b> <i>minutes</i>	Sets the number of minutes that calls are stored in the Call Tracker history table.
Step 5	Router(config)# <b>calltracker call-record</b> <terse verbose> [quiet]	Enables Call Tracker SYSLOG support for generating detailed Call Records.
Step 6	Router(config)# <b>snmp-server packetsize</b> <i>byte-count</i>	Sets the maximum packet size allowed for SNMP server requests and replies.
Step 7	Router(config)# <b>snmp-server queue-length</b> <i>length</i>	Sets the queue length for SNMP traps.
Step 8	Router(config)# <b>snmp-server enable traps calltracker</b>	Enables Call Tracker to send traps whenever a call starts or ends.
Step 9	Router(config)# <b>snmp-server host</b> <i>host community-string calltracker</i>	Specifies the name or Internet address of the host to send Call Tracker traps.

## Verifying Call Tracker

	Command	Purpose
Step 1	Router(config)# <b>show call calltracker summary</b>	Verifies the Call Tracker configuration and current status.

## Configuring Polling of Link Statistics

To poll modem-link statistics, perform the following tasks in global configuration mode:



### Note

The **modem link-info poll time** command consumes a significant amount of memory, approximately 500 bytes for each MICA modem call. Use this command only if you require the specific data that it collects; for instance, if you have enabled Call Tracker on your access server.

	Command	Purpose
Step 1	Router# <b>config term</b>	Enters global configuration mode.

	Command	Purpose
Step 2	Router(config)# <b>modem link-info poll time</b> <i>seconds</i>	Sets the polling interval at which link statistics for active calls are retrieved from the modem.

## Configuring RADIUS Acct-Session-Id

To configure AAA to include the Acct-Session-ID in Access-Request packets, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	Router# <b>config term</b>	Enters global configuration mode.
Step 2	Router(config)# <b>radius-server attribute 44 include-in-access-req</b>	Includes the Acct-Session-Id attribute in Access-Request packets.



**Note** The format of a user profile depends on the specific RADIUS server that you are using. The following two user profile examples are included to help illustrate Cisco IOS software functionality. These examples are not intended to demonstrate the actual configuration of your RADIUS server.

## Configuring RADIUS Login-IP-Host

To enable the network access server to attempt more than one login host when trying to connect a dial in user, you can enter as many as three Login-IP-Host entries in the user's profile on the RADIUS server. The following example shows that three Login-IP-Host instances have been configured for the user *joeuser*, and that TCP-Clear will be used for the connection:

```
joeuser      Password = xyz
             Service-Type = Login,
             Login-Service = TCP-Clear,
             Login-IP-Host = 10.0.0.0,
             Login-IP-Host = 10.2.2.2,
             Login-IP-Host = 10.255.255.255,
             Login-TCP-Port = 23
```

The order in which the hosts are entered is the order in which they are attempted. Use the **ip tcp synwait-time** command to set the seconds that the network access server waits before trying to connect to the next host on the list; the default is 30 seconds.



**Note** Your RADIUS server might permit more than three Login-IP-Host entries; however, the network access server supports only three hosts in Access-Accept packets.

## Configuring RADIUS Prompt

To control whether user responses to Access-Challenge packets are echoed to the screen, you can configure the Prompt attribute in the user profile on the RADIUS server. This attribute is included only in Access-Challenge packets. The following example shows the Prompt attribute set to No-Echo, which prevents the user's responses from echoing:

```
joeuser      Password = xyz
            Service-Type = Login,
            Login-Service = Telnet,
            Prompt = No-Echo,
            Login-IP-Host = 172.31.255.255
```

To allow user responses to echo, set the attribute to Echo. If the Prompt attribute is not included in the user profile, responses are echoed by default.

This attribute overrides the behavior of the **radius-server challenge-noecho** command configured on the access server. For example, if the access server is configured to suppress echoing, but the individual user profile allows echoing, then the user responses are echoed.



### Note

To use the Prompt attribute, your RADIUS server must be configured to support Access-Challenge packets.

## Configuring AAA DNIS Authentication

To configure DNIS authentication, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	Router# <b>config term</b>	Enters global configuration mode.
Step 2	Router(config)# <b>aaa preauth</b>	Enters AAA preauthentication mode.
Step 3	Router(config-preauth)# <b>group {radius   tacacs+   server-group}</b>	(Optional) Selects the security server to use for AAA preauthentication requests. The default is RADIUS.
Step 4	Router(config-preauth)# <b>dnis [password string]</b>	Enables preauthentication using DNIS and optionally specifies a password to use in Access-Request packets.

## Configuring AAA POD

To configure POD, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	Router# <b>config term</b>	Enters global configuration mode.
Step 2	Router(config)# <b>aaa accounting network default start-stop radius</b>	Enables AAA accounting records.

	Command	Purpose
Step 3	Router(config)# <b>aaa accounting delay-start</b>	Delays generation of the start accounting record until the Framed-IP-Address is assigned, allowing its use in the POD packet.
Step 4	Router(config)# <b>aaa pod server server-key string</b>	Enables POD reception.
Step 5	Router(config)# <b>radius-server host IP address non-standard</b>	Declares a RADIUS host that uses a vendor-proprietary version of RADIUS.

## Configuring ISDN PRI B-Channel Busyout

To allow the busyout of individual ISDN PRI B-channels, perform the following tasks in interface configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>interface serial controller:timeslot</b>	Enters interface configuration mode for a D-channel serial interface.
Step 2	Router(config-if)# <b>isdn snmp busyout b-channel</b>	Allows the busyout of individual PRI B-channels via SNMP.

## Configuring ISDN Cause Code Override

To configure ISDN cause code overrides, perform the following tasks in interface configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>interface serial controller:timeslot</b>	Enters interface configuration mode for a D-channel serial interface.
Step 2	Router(config-if)# <b>isdn disconnect-cause {cause-code-number   busy   not-available}</b>	Specifies an ISDN cause code to send to the switch.

## Configuring ISDN Guard Timer

To configure the ISDN guard timer, perform the following tasks in interface configuration mode:

	Command	Purpose
Step 1	Router(config)# <b>interface serial controller:timeslot</b>	Enters interface configuration mode for a D-channel serial interface.
Step 2	Router(config-if)# <b>isdn guard-timer msec</b>	Enables the guard timer and set the number of milliseconds that the access server waits for RADIUS to respond before rejecting or accepting (optional) a call.

## Configuring IP Multicast Heartbeat

To configure IP multicast heartbeat, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	Router# <b>config term</b>	Enters global configuration mode.
Step 2	Router(config)# <b>ip multicast-routing</b>	Enables IP multicast routing.
Step 3	Router(config)# <b>snmp-server host host traps community-string</b>	Specifies the recipient of an SNMP notification operation.
Step 4	Router(config)# <b>snmp-server enable traps ipmulticast</b>	Enables the router to send IP multicast traps.
Step 5	Router(config)# <b>ip multicast heartbeat interface group minimum window-size interval</b>	Enables the monitoring of the health of multicast delivery.

## Verifying

To verify that the Call Tracker feature is configured properly, perform these tasks:

- Enter the **show call calltracker summary** command. The output shows the number of active calls, calls moved to the history table at disconnect, and the history table size and timer.

```
Router# show call calltracker summary
```

```
Call Tracker Status:
Active Table:
- 7 call(s)
- 4473 bytes used (639 average, 639 maximum)
History Table:
- 50 of a maximum of 240 call(s) (20% full)
- 45157 bytes used (903 average, 921 maximum)
- 260000 minute(s) call retain time
API Front-end:
- event elements:512 total, 512 free, 0 in-use
- free event elements' low watermark:467
- events dropped due to unavailability of free elts:0
```



**Note** For a description of each output display field, see the **show call calltracker summary** command reference page.

- Enter the **debug calltracker** command. The output reports any configuration problems.

To verify that other features are configured correctly, enter the **show running-config** command.

# Configuration Examples

The following examples display the screen output using the **show running-config** command:

- Call Tracker
- IP Multicast Heartbeat
- Modem Polling Link Statistics
- RADIUS Acct-Session-Id
- DNIS Preauthentication
- POD Server Key
- ISDN B-Channel Busyout
- ISDN Cause Code Override and Guard Timer

## Call Tracker

```

!
calltracker enable
calltracker call-record terse
calltracker history max-size 50
calltracker history retain-mins 5000
!
snmp-server engineID local 0012345
snmp-server community public RW
snmp-server community private RW
snmp-server community wxyz123 view v1default RO
snmp-server trap-source FastEthernet0
snmp-server packet-size 17940
snmp-server queue-length 200
snmp-server location SanJose
snmp-server contact Bob
snmp-server enable traps snmp
snmp-server enable traps calltracker
snmp-server enable traps isdn call-information
snmp-server enable traps hsrp
snmp-server enable traps config
snmp-server enable traps entity
snmp-server enable traps envmon
snmp-server enable traps bgp
snmp-server enable traps ipmulticast-heartbeat
snmp-server enable traps rsvp
snmp-server enable traps frame-relay
snmp-server enable traps rtr
snmp-server enable traps syslog
snmp-server enable traps dlsw
snmp-server enable traps dial
snmp-server enable traps dsp card-status
snmp-server enable traps voice poor-qov
snmp-server host 10.255.255.255 wxyz123
snmp-server host 10.0.0.0 xxxyyy calltracker
!
radius-server host 172.16.0.0 auth-port 1645 acct-port 1646 non-standard
radius-server key xyz
!

```

## IP Multicast Heartbeat

```

!
ip multicast-routing
!
snmp-server host 224.1.0.1 traps public
snmp-server enable traps ipmulticast
ip multicast heartbeat ethernet0 224.1.1.1 1 1 10

```

## Modem Polling Link Statistics

```

!
clock timezone PDT -8
clock summer-time PDT recurring
calltracker enable
calltracker history retain-mins 10
modem link-info poll time 300
ip subnet-zero
no ip domain-lookup
ip host jjjxxx 192.168.255.255
ip host xxxyyy 172.31.255.255
ip domain-name cisco.com
!
isdn switch-type primary-5ess
chat-script dial ABORT ERROR ABORT BUSY ABORT "NO CARRIER" TIMEOUT 30 "" at OK "
mta receive maximum-recipients 0
partition flash 2 8 8
!

```

## RADIUS Acct-Session-Id

```

!
aaa new-model
aaa authentication login CONSOLE none
aaa authentication login RADIUS-LIST group radius local
aaa authentication ppp default local
aaa authentication ppp RADIUS-LIST group radius
aaa authorization exec RADIUS-LIST group radius if-authenticated
aaa authorization exec CONSOLE none
aaa authorization network RADIUS-LIST group radius if-authenticated
aaa accounting suppress null-username
aaa accounting delay-start
aaa accounting network default start-stop group radius
aaa configuration config-username pools-ISP-r2 password ascend
aaa nas port extended
enable secret 5 $ABCxyz
!
radius-server configure-nas
radius-server host 172.16.0.0 auth-port 1645 acct-port 1646 non-standard
radius-server timeout 15
radius-server attribute 6 on-for-login-auth
radius-server attribute 44 include-in-access-req
no radius-server attribute nas-port
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication
!

```

## DNIS Preauthentication

```

!
aaa new-model
aaa authentication login CONSOLE none
aaa authentication login RADIUS_LIST group radius
aaa authentication login TAC_PLUS group tacacs+ enable
aaa authentication login V.120 none
aaa authentication enable default enable group tacacs+
aaa authentication ppp RADIUS_LIST if-needed group radius
aaa authorization exec RADIUS_LIST group radius if-authenticated
aaa authorization exec V.120 none
aaa authorization network default group radius if-authenticated
aaa authorization network RADIUS_LIST if-authenticated group radius
aaa authorization network V.120 group radius if-authenticated
aaa accounting suppress null-username
aaa accounting exec default start-stop group radius
aaa accounting commands 0 default start-stop group radius
aaa accounting network default start-stop group radius
aaa accounting connection default start-stop group radius
aaa accounting system default start-stop group radius
aaa preauth
  dnis password Cisco-DNIS
aaa nas port extended
!
radius-server configure-nas
radius-server host 10.0.0.0 auth-port 1645 acct-port 1646 non-standard
radius-server host 10.255.255.255 auth-port 1645 acct-port 1646 non-standard
radius-server retransmit 2
radius-server deadtime 1
radius-server attribute nas-port format c
radius-server unique-ident 18
radius-server key MyKey
!

```

## POD Server Key

```

!
aaa new-model
aaa authentication ppp default radius
aaa accounting network default start-stop radius
aaa accounting delay-start
aaa pod server server-key xyz123
radius-server host 172.16.0.0 non-standard
radius-server key rad123
!

```

## ISDN B-Channel Busyout

```
!  
interface Serial0:23  
 ip address 172.16.0.0 192.168.0.0  
 no ip directed-broadcast  
 encapsulation ppp  
 no keepalive  
 dialer idle-timeout 400  
 dialer load-threshold 1 either  
 dialer-group 1  
 isdn switch-type primary-5ess  
 isdn incoming-voice modem  
 isdn snmp busyout b-channel  
 no fair-queue  
 no cdp enable  
!
```

## ISDN Cause Code Override and Guard Timer

```
!  
interface Serial0:23  
 no ip address  
 no ip directed-broadcast  
 encapsulation ppp  
 dialer rotary-group 0  
 isdn switch-type primary-5ess  
 isdn incoming-voice modem  
 isdn disconnect-cause 17  
 isdn guard-timer 3000 on-expiry accept  
 isdn calling-number 8005551234  
 no fair-queue  
 no cdp enable  
!
```

# Command Reference

This section documents new or modified commands. All other commands used with these features are documented in the Cisco IOS Release 12.1 command reference publications.

## New Commands

- **aaa pod server**
- **aaa preauth**
- **calltracker call-record**
- **calltracker enable**
- **calltracker history max-size**
- **calltracker history retain-mins**
- **dnis**
- **group**
- **ip multicast heartbeat**
- **isdn disconnect-cause**
- **isdn guard-timer**
- **isdn snmp busyout b-channel**
- **modem link-info poll time**
- **radius-server attribute 44 include-in-access-req**
- **radius-server challenge-noecho**
- **show call calltracker active**
- **show call calltracker handle**
- **show call calltracker history**
- **show call calltracker summary**
- **show modem calltracker**

## Modified Commands

- **snmp-server enable traps**
- **snmp-server host**

## aaa pod server

To enable inbound user sessions to be disconnected when specific session attributes are presented, use the **aaa pod server** global configuration command. Enter the **no** form of this command to disable this feature.

```
aaa pod server [port port-number] [auth-type {any | all | session-key}] server-key string
```

```
no aaa pod server
```

Syntax Description		
<b>port</b> <i>port-number</i>	(Optional) The network access server port to use for POD requests. If no port is specified, port 1700 is used.	
<b>auth-type</b>	(Optional) The type of authorization required for disconnecting sessions.	
<b>any</b>	Session that matches all attributes sent in the POD packet is disconnected. The POD packet can contain one or more of four key attributes (user-name, framed-IP-address, session-ID, and session-key).	
<b>all</b>	Only a session that matches all four key attributes is disconnected. <b>All</b> is the default.	
<b>session-key</b>	Session with a matching session-key attribute is disconnected. All other attributes are ignored.	
<b>server-key</b> <i>string</i>	The secret text string that is shared between the network access server and the client workstation. This secret string must be the same on both systems.	

**Defaults** The POD server function is disabled.

**Command Modes** Global configuration

Command History	Release	Modification
	12.1(3)T	This command was introduced.

**Usage Guidelines**

To disconnect a session, the values in one or more of the key fields in the POD request must match the values for a session on one of the network access server ports. Which values must match depends on the **auth-type** attribute defined in the command. If no auth-type is specified, all four values must match. If no match is found, all connections remain intact and an error response is returned. The key fields are as follows:

- User-Name
- Framed-IP-Address
- Session-Id
- Server-Key

**Examples**

The following example enables POD and sets the secret key to “ab9123.”

```
aaa pod server server-key ab9123
```

**Related Commands**

Command	Description
<b>debug aaa pod</b>	Displays debug messages for POD packets.
<b>aaa authentication</b>	Enables authentication.
<b>aaa accounting</b>	Enables accounting records.
<b>aaa accounting delay-start</b>	Delays generation of the start accounting record until the user IP address is established.
<b>radius-server host</b>	Identifies a RADIUS host.

# aaa preauth

To enter AAA preauthentication configuration mode, use the **aaa preauth** global configuration command. To disable preauthentication, use the **no** form of this command.

**aaa preauth**

**no aaa preauth**

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

**Defaults** Preauthentication is not enabled.

**Command Modes** Global configuration

Command History	Release	Modification
	12.1(2)T	This command was introduced.

**Usage Guidelines** To enter AAA preauthentication configuration mode, use the **aaa preauth** command. To configure preauthentication, use a combination of the **aaa preauth** commands: **group**, **clid**, **ctype**, **dnis**, and **dnis bypass**. You must configure the **group** command. You must also configure one or more of the **clid**, **ctype**, **dnis**, or **dnis bypass** commands.

In addition to using the preauthentication commands to configure preauthentication on the Cisco router, you must set up the preauthentication profiles on the RADIUS server.

You can use the **clid**, **ctype**, or **dnis** commands to define the list of the preauthentication elements. For each preauthentication element, you can also define options such as password (for all the elements, the default password is cisco). If you specify multiple elements, the preauthentication process will be performed on each element according to the order of the elements that you configure with the preauthentication commands. In this case, more than one RADIUS preauthentication profile is returned, but only the last preauthentication profile will be applied to the authentication and authorization later on, if applicable.

**Examples** The following example enables DNIS preauthentication using a RADIUS server and the password Ascend-DNIS:

```
aaa preauth
dnis password Ascend-DNIS
```

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>dnis</b>	Enables AAA preauthentication using DNIS.
	<b>group</b>	Selects the security server to use for AAA preauthentication.
	<b>isdn guard-timer</b>	Enables a timer for AAA server requests.

# calltracker call-record

To enable call record SYSLOG generation for the purpose of debugging, monitoring, or externally saving detailed call record information, use the **calltracker call-record** global configuration command. Use the **no** form of this command to disable call record SYSLOG generation.

**calltracker call-record** <terse | verbose> [quiet]

**no calltracker call-record** <terse | verbose> [quiet]

## Syntax Description

<b>terse</b>	Generates a brief set of call records containing a subset of the data stored within Call Tracker used primarily to manage calls.
<b>verbose</b>	Generates a complete set of call-records containing all of the data stored within Call Tracker used primarily to debug calls.
<b>quiet</b>	(Optional) Call Record will be sent only to configured SYSLOG server and not to console

## Defaults

Call Tracker call record logging is disabled.

## Command Modes

Global configuration

## Related Commands

Release	Modification
12.1(3)T	This command was introduced.

## Usage Guidelines

SYSLOG call records will be generated in the order of ten seconds of call termination. A small delay is needed to ensure that all subsystems finish reporting all appropriate information on call termination. Furthermore, the process of logging is considered a very low priority with respect to normal call processing and data routing. As such, logging all call records can be guaranteed if Call Tracker is properly configured. However, the delay from the time a call actually terminated can vary if the CPU is busy handling higher-priority processes.

Call Tracker records must be found within the History table for at least one minute after call termination for this capability to work. As such, one must ensure that Call Tracker history collection is not disabled with the **calltracker history** configuration options.

Because the call rates possible on a high-capacity access server can be rather large and the information provided by the call records is substantial, simply enabling normal SYSLOG call records can make the use of the console difficult. As such, by using the **quiet** option and having a SYSLOG server configured to capture the call records, the console can be freed from displaying any call records, yet still have the call records captured by a SYSLOG server.



**Related Commands**

<b>Command</b>	<b>Description</b>
<b>calltracker history max-size</b>	Sets the maximum calls saved in the history table.
<b>calltracker history retain-mins</b>	Sets the number of minutes to save calls in the history table.
<b>show call calltracker history</b>	Displays the detailed data stored within Call Tracker for terminated calls.
<b>show call calltracker summary</b>	Displays the number of calls in the active table and history table, and the values of the history table attributes.

# calltracker enable

To enable Call Tracker on the access server, use the **calltracker enable** global configuration command. To restore the default condition, use the **no** form of this command.

**calltracker enable**

**no calltracker enable**

---

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

---

**Defaults** Call Tracker is not enabled.

---

**Command Modes** Global configuration

---

Command History	Release	Modification
	12.1(3)T	This command was introduced.

---



---

**Usage Guidelines** To enable real-time call statistics from the MICA modem to Call Tracker, you must configure the **modem link-info poll time** command.

---

**Examples**

```
Router# config term
Router(config)# calltracker enable
Router(config)# calltracker history max-size number
Router(config)# calltracker history retain-mins minutes
Router(config)# calltracker call-record terse
Router(config)# snmp-server packetsize byte-count
Router(config)# snmp-server queue-length length
Router(config)# snmp-server enable traps calltracker
Router(config)# snmp-server host host community-string calltracker
```

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>dnis</b>	Enable Call Tracker SYSLOG support for generating detailed Call Records.
	<b>calltracker history max-size</b>	Sets the maximum calls saved in the history table.
	<b>calltracker history retain-mins</b>	Sets the number of minutes to save calls in the history table.
	<b>debug calltracker</b>	Displays debug messages tracing the Call Tracker processing flow.
	<b>modem link-info poll time</b>	Sets the interval at which active call statistics are polled from the MICA modem.
	<b>show call calltracker active</b>	Displays the detailed data stored within Call Tracker for active calls.
	<b>show call calltracker history</b>	Displays the detailed data stored within Call Tracker for terminated calls.
	<b>show call calltracker summary</b>	Displays the detailed data stored within Call Tracker for last call on specified modem.
	<b>snmp-server host</b>	Specifies the host to receive Call Tracker traps.

## calltracker history max-size

To set the maximum number of call entries stored in the Call Tracker history table, use the **calltracker history max-size** global configuration command. To restore the default value, use the **no** form of this command.

**calltracker history max-size** *number*

**no calltracker history max-size** *number*

### Syntax Description

<i>number</i>	The maximum call entries to store in the Call Tracker history table. The valid range is from 0 through 10 times the max DS0 supported on given platform. A value of 0 prevents any history from being saved.
---------------	--

### Defaults

The default maximum is dynamically calculated to be 1 times the maximum DS0 supported on given platform.

### Command Modes

Global configuration

### Command History

Release	Modification
12.1(3)T	This command was introduced.

### Usage Guidelines

Be careful when extending the history max-size, as this activity will cause Call Tracker to use more memory resources to store the additional call data. NAS memory consumption must be considered when increasing this parameter. The active call table is not affected by this command.

### Examples

The following example sets the history table size to 50 calls:

```
calltracker history max-size 50
```

### Related Commands

Command	Description
<b>calltracker history retain-mins</b>	Sets the number of minutes to save calls in the history table.
<b>show call calltracker history</b>	Displays the detailed data stored within Call Tracker for terminated calls.
<b>show call calltracker summary</b>	Displays the number of calls in the active table and history table, and the values of the history table attributes.

# calltracker history retain-mins

To set the number of minutes that call entries are stored in the Call Tracker history table, use the **calltracker history retain-mins** global configuration command. To restore the default value, use the **no** form of this command.

**calltracker history retain-mins** *minutes*

**no calltracker history retain-mins** *minutes*

## Syntax Description

*minutes*

The length of time to store calls in the Call Tracker history table. The valid range is from 0 through 26,000 minutes. A value of 0 prevents any history from being saved.

## Defaults

The default minutes is 5000.

## Command Modes

Global configuration

## Command History

### Release

12.1(3)T

### Modification

This command was introduced.

## Usage Guidelines

Active calls are not affected by this command. Entries in the active table are retained as long as the calls are connected.

## Examples

The following example sets the retain time for the history table to 5000 minutes:

```
calltracker history retain-mins 5000
```

## Related Commands

### Command

### Description

**calltracker history max-size**

Sets the maximum calls saved in the history table.

**show call calltracker history**

Displays the detailed data stored within Call Tracker for terminated calls.

**show call calltracker summary**

Displays the number of calls in the active table and history table, and the values of the history table attributes.

# dnis

To preauthenticate calls on the basis of the DNIS number, use the **dnis** AAA preauthentication configuration command. To remove the **dnis** command from your configuration, use the **no** form of this command.

**dnis** [**if-avail** | **required**] [**accept-stop**] [**password** *string*]

**no dnis** [**if-avail** | **required**] [**accept-stop**] [**password** *string*]

## Syntax Description

<b>if-avail</b>	(Optional) Implies that if the switch provides the data, RADIUS must be reachable and must accept the string in order for preauthentication to pass. If the switch does not provide the data, preauthentication passes.
<b>required</b>	(Optional) Implies that the switch must provide the associated data, that RADIUS must be reachable, and that RADIUS must accept the string in order for preauthentication to pass. If these three conditions are not met, preauthentication fails.
<b>accept-stop</b>	(Optional) Prevents subsequent preauthentication elements from being tried once preauthentication has succeeded for a call element.
<b>password</b> <i>string</i>	(Optional) Password to use in the Access-Request packet. The default is cisco.

## Defaults

The **if-avail** and **required** keywords are mutually exclusive. If the **if-avail** keyword is not configured, the preauthentication setting defaults to **required**.

The default password string is cisco.

## Command Modes

AAA preauthentication configuration

## Command History

Release	Modification
12.1(2)T	This command was introduced.

## Usage Guidelines

You may configure more than one of the AAA preauthentication commands (**clid**, **ctype**, **dnis**) to set conditions for preauthentication. The sequence of the command configuration decides the sequence of the preauthentication conditions. For example, if you configure **dnis**, then **clid**, then **ctype**, then this is the order of the conditions considered in the preauthentication process.

In addition to using the preauthentication commands to configure preauthentication on the Cisco router, you must set up the preauthentication profiles on the RADIUS server.

**Examples**

The following example enables DNIS preauthentication using a RADIUS server and the password Ascend-DNIS:

```
aaa preauth
dnis password Ascend-DNIS
```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>aaa preauth</b>	Enters AAA preauthentication mode.
<b>group</b>	Selects the security server to use for AAA preauthentication.
<b>isdn guard-timer</b>	Enables a timer for AAA server requests.

# group

To specify the AAA RADIUS server group to use for preauthentication, use the **group** AAA preauthentication configuration command. To remove the **group** command from your configuration, use the **no** form of this command.

```
group { radius | tacacs+ | server-group }
```

```
no group { radius | tacacs+ | server-group }
```

## Syntax Description

<b>group</b>	Enables authentication using the listed AAA server or server group. If no group is selected, <b>radius</b> is the default.
<b>radius</b>	Uses a RADIUS server for authentication.
<b>tacacs+</b>	Uses a TACACS+ server for authentication.
<i>server-group</i>	Name of the server group to use for authentication.

## Defaults

If this command is not configured, preauthentication is performed by a RADIUS server.

## Command Modes

AAA preauthentication configuration

## Command History

Release	Modification
12.1(2)T	This command was introduced.

## Usage Guidelines

You must configure a RADIUS server group with the **aaa group server radius** command in global configuration mode before using the **group** command in AAA preauthentication configuration mode.

You must configure the **group** command before you configure any other AAA preauthentication command (**clid**, **ctype**, **dnis**, or **dnis bypass**).

## Examples

The following example enables DNIS preauthentication using the abc123 server group and the password aaa-DNIS:

```
aaa preauth
group abc123
dnis password aaa-DNIS
```

## Related Commands

Command	Description
<b>aaa preauth</b>	Enters AAA preauthentication mode.
<b>dnis</b>	Enables AAA preauthentication using DNIS.

# ip multicast heartbeat

To monitor the health of multicast delivery and be alerted when the delivery fails to meet certain parameters, use the **ip multicast heartbeat** command in global configuration mode. To disable the heartbeat, use the **no** form of the command.

**ip multicast heartbeat** *group minimum window-size interval*

**no ip multicast heartbeat** *group minimum window-size interval*

Syntax Description	
<i>group</i>	A multicast group address (Class D address, between 224.0.0.0 and 239.255.255.255)
<i>minimum</i>	Number of packets to be received within a specified number of intervals ( <i>window-size</i> ).
<i>window-size</i>	Window size within which a specified number of intervals must receive a ( <i>minimum</i> ) specified number of packets.
<i>interval</i>	Number of seconds interval to receive packet. Value must be a multiple of 10.

**Defaults** The command is disabled by default.

**Command Modes** Global configuration

Command History	Release	Modification
	12.1(3)T	This command was introduced.

**Usage Guidelines**

The subject group is joined at the subject interface so multicast data for the subject group will be attracted toward the subject router.

The router monitors multicast packets destined to *group* at *interval* seconds. This is a binary decision. That is, the number of packets seen in this period is not as important as whether any packet for the group is seen or not.

If multicast packets were observed in less than *minimum* out of the last *window-size* intervals, an SNMP trap would be sent from this router to a network management station to indicate a loss of heartbeat exception. This trap will be defined in CISCO-IPMROUTE-MIB.my.

The value of *interval* must be a multiple of 10. In multicast distributed switching (MDS), statistics from VIP (in RSP) or LC (in GSR) are passed to the routing processor once every 10 seconds. Monitoring packets not in intervals of multiple of 10 seconds may lead to incorrect decisions.

This command does not create any multicast routing entries that is necessary for the monitoring of the heartbeat packets. These entries can be created by either the downstream members of the group, or with the **ip pim join-group** or **ip pim static-group** commands. If a multicast routing entry corresponding to a group address is expired due to lack of interest from the downstream members, the monitoring for the subject group would not work; that is, no more SNMP traps would be emitted.

**Examples**

The following is an example configuration of the **ip multicast heartbeat** command.

```
snmp-server enable traps ipmulticast-heartbeat
ip multicast heartbeat 224.0.1.53 1 1 10
```

In this example, multicast packets forwarded through this router to group address 224.0.1.53 will be monitored. If no packet for this group is received in a 10-second interval, an SNMP trap would be sent to a designated SNMP management station.

**Note**

This means it may take about 20 seconds of losing the multicast feed before the SNMP trap is sent.

**Related Commands**

Command	Description
<b>snmp-server enable traps</b>	Enables the router to send SNMP traps.
<b>debug ip mhbeat</b>	Monitors the action of the heartbeat trap.

# isdn disconnect-cause

To send a specific ISDN cause code to the switch, use the **isdn disconnect-cause** interface configuration command. To return to the default condition, use the **no** form of the command.

**isdn disconnect-cause** { *cause-code-number* | **busy** | **not-available** }

**no isdn disconnect-cause**

Syntax Description		
<i>cause-code-number</i>		Sends a cause code number (submitted as integer in the range of 1 through 127) to the switch.
<b>busy</b>		Sends the USER-BUSY code to the switch.
<b>not-available</b>		Sends the CHANNEL-NOT-AVAILABLE code to the switch.

**Defaults** The default condition is no cause code override. If the **isdn disconnect-cause** command is not configured, the default cause codes for the application are sent.

**Command Modes** Interface configuration

Command History	Release	Modification
	12.0(5)T	This command was introduced, and is a replacement for the <b>isdn modem-busy-cause</b> command.

**Usage Guidelines** The **isdn disconnect-cause** command overrides specific cause codes (such as modem availability and resource pooling) that are sent to the switch by ISDN applications. When the **isdn disconnect-cause** command is implemented, the configured cause codes are sent to the switch; otherwise, the default cause codes for the application are sent. ISDN protocol errors are still reflected in the cause codes and are not overridden.

**Examples** The following example sends the CHANNEL-NOT-AVAILABLE code to the ISDN switch:

```
interface serial0:20
isdn disconnect-cause not-available
```

Related Commands	Command	Description
	<b>isdn modem-busy cause</b>	Sends a specific cause code to the ISDN switch.

## isdn guard-timer

To enable a managed timer for authentication requests, use the **isdn guard-timer** interface configuration command. To reset the timer to its default value, use the **no** form of this command.

**isdn guard-timer** *msecs* [**on-expiry** {**accept** | **reject**}]

**no isdn guard-timer**

Syntax Description		
	<i>msecs</i>	Number of milliseconds that the network access server (NAS) waits for a response from the AAA security server. The valid range is from 1000 through 20,000.
	<b>on-expiry</b>	(Optional) Determines whether calls are accepted or rejected after the specified number of milliseconds has expired. If no expiry action is selected, calls are rejected.
	<b>accept</b>	Calls are accepted if the guard-timer expires before AAA responds.
	<b>reject</b>	Calls are rejected if the guard-timer expires before AAA responds.

**Defaults** The default timer value is eight (8) seconds and calls are rejected when the timer expires.

**Command Modes** Interface configuration

Command History	Release	Modification
	12.1(3)T	This command was introduced.

**Usage Guidelines** The guard-timer starts when the DNIS number is sent to AAA for authentication. When the timer expires, authentication ends and the call is accepted or rejected based on the configured expiry action.

**Examples** The following example sets the guard-timer to six (6) seconds and specifies that the call should be rejected if AAA does not respond within that interval:

```
interface serial 1/0/0:23
isdn guard-timer 6000 on-expiry reject
```

Related Commands	Command	Description
	<b>aaa preauth</b>	Enables authentication using DNIS numbers.

# isdn snmp busyout b-channel

To enable PRI B channels to be busied out via SNMP, use the **isdn snmp busyout b-channel** interface configuration command. To prevent B-channels from being busied out via SNMP, use the **no** form of this command.

**isdn snmp busyout b-channel**

**no isdn snmp busyout b-channel**

## Syntax Description

This command has no keywords or arguments.

## Defaults

The default value is TRUE; that is, setting busyout using SNMP is allowed.

## Command Modes

Interface configuration

## Command History

Release	Modification
12.1(3)T	This command was introduced.

## Usage Guidelines

To busy out B-channels on a PRI, the ISDN switch must support service messages. The **isdn snmp busyout b-channel** command sets the MIB object, cpmDSOBusyoutAllow, indicating whether or not the switch supports service messages, thereby allowing the busyout of B-channels. When the network access server receives an SNMP request for a busyout, it checks the value of this object. If the **no isdn snmp busyout b-channel** command is configured, the busyout request fails.

## Examples

The following example allows the busyout of B-channels for serial interface 0:23:

```
Router# conf t
Router(config)# interface serial 0:23
Router(config-if)# isdn snmp busyout b-channel
```

# modem link-info poll time

To set the polling interval at which link statistics are retrieved from the MICA modem, use the **modem link-info poll time** command. To return to the default condition, use the **no** form of this command.

**modem link-info poll time** *seconds*

**no modem link-info poll** *seconds*

<b>Syntax Description</b>	<i>seconds</i>	Number of seconds between polling intervals. The valid range is 10 to 65,535.
---------------------------	----------------	---

<b>Defaults</b>	Link statistics are not polled.
-----------------	---------------------------------

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.1(3)T	This command was introduced.

**Usage Guidelines** The **modem link-info poll time** command periodically polls active modem sessions to collect information such as attempted transmit and receive rates, maximum and minimum transmit and receive rates, and locally and remotely issued retrains and speedshift counters. This data is polled from MICA portware and passed unsolicited to Cisco IOS software.

Enabling the **modem link-info poll time** command disables the **modem poll time** command. Any **modem poll time** configuration is ignored because all modem events are sent to the access server unsolicited and no longer require polling by Cisco IOS software.



**Note** The **modem link-info poll time** command consumes a significant amount of memory, approximately 500 bytes for each MICA modem call. You should use this command only if you require the specific data that it collects; for instance, if you have enabled Call Tracker on your access server using the **calltracker call-record** command.

**Examples** The following example polls link statistics at 90 second intervals:

```
modem link-info poll time 300
```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>calltracker call-record</b>	Enables Call Tracker on the access server.
<b>show call calltracker active</b>	Displays the detailed data stored within Call Tracker for active calls.
<b>show call calltracker handle</b>	Displays the detailed data stored within Call Tracker for a specific call specified unique call handle identifier.
<b>show call calltracker history</b>	Displays the detailed data stored within Call Tracker for terminated calls.
<b>show modem calltracker</b>	Displays the detailed data stored within Call Tracker for the last call on the specified modem.

## radius-server attribute 44 include-in-access-req

To send the RADIUS Acct-Session-Id (attribute 44) in authentication Access-Request packets, use the **radius-server attribute 44 include-in-access-req** global configuration command. To return to the default condition, use the **no** form of this command.

**radius-server attribute 44 include-in-access-req**

**no radius-server attribute 44 include-in-access-req**

### Syntax Description

This command has no arguments or keywords.

### Defaults

The Acct-Session-Id is not included in Access-Request packets.

### Command Modes

Global configuration

### Command History

Release	Modification
12.1(3)T	This command was introduced.

### Usage Guidelines

Using the **radius-server attribute 44 include-in-access-req** command generates the Acct-Session\_Id prior to authentication, allowing call authentication records to be linked to their respective accounting records.

### Examples

The following example sends the Acct-Session-Id in authentication Access-Requests:

```
radius-server attribute 44 include-in-access-req
```

# radius-server challenge-noecho

To prevent user responses to Access-Challenge packets from displaying on the screen, use the **radius-server challenge-noecho** global configuration command. To return to the default condition, use the **no** form of this command.

**radius-server challenge-noecho**

**no radius-server challenge-noecho**

## Syntax Description

This command has no arguments or keywords.

## Defaults

All user responses to Access-Challenge packets are echoed to the screen.

## Command Modes

Global configuration

## Command History

Release	Modification
12.0(5)T	This command was introduced.

## Usage Guidelines

This command applies to all users. When the **radius-server challenge-noecho** command is configured, user responses to Access-Challenge packets are not displayed, unless the Prompt attribute in the user profile is set to *echo* on the RADIUS server. The Prompt attribute in a user profile overrides the **radius-server challenge-noecho** command, for the individual user. For more information, see “Configuring RADIUS Prompt” on page 9.

## Examples

The following example stops all user responses from displaying on the screen:

```
radius-server challenge-noecho
```

## show call calltracker active

To display all of the information stored within the Call Tracker active database for all active calls, use the **show call calltracker active** privileged EXEC command. This command allows you to display only calls for a single supported call category type, if desired.

```
show call calltracker active [category [isdn | modem | other | v110 | v120]]
```

### Syntax Description

<b>category [isdn   modem   other   v110   v120]</b>	(Optional) Displays Call Tracker data for a specific type of call. The default is to show all calls, regardless of type. By specifying the category, Call Tracker will only show calls whose records indicate that category.
--	--

### Defaults

The activity and configuration information is not displayed. The command **show call calltracker active** will show all calls, regardless of type, unless specified by the **category** option field.

### Command Modes

Privileged EXEC

### Command History

Release	Modification
12.1(3)T	This command was introduced.



**■** show call calltracker active

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>show call calltracker handle</b>	Displays the detailed data stored within Call Tracker for specific call having specified unique call handle identifier.
	<b>show call calltracker history</b>	Displays all the information stored within the Call Tracker history database table for the most recent disconnected calls.

# show call calltracker handle

To display all information stored within the Call Tracker active or history database table for a specified unique call handle identifier, use the **show call calltracker handle** privileged EXEC command.

**show call calltracker handle** *handle*

## Syntax Description

<i>handle</i>	Unique call identifier assigned by Call Tracker from the moment a DS0 B-Channel is requested. This identifier is a sequential number starting with handle 1.
---------------	--

## Defaults

The activity and configuration information is not displayed.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(3)T	This command was introduced.

## Usage Guidelines

Each call managed by Call Tracker is assigned a unique call handle. This handle is provided to users using SNMP, CLI, or SYSLOG for all forms of data transfers. Thus, it becomes easier to display the information desired for a given call, knowing its call handle, than manually searching through all Call Tracker database tables for latest updates.

## Examples

```
Router# show call calltracker handle 30
----- call handle=0000000030 -----
status=History, service=None, origin=Answer, category=Other
DS0 slot/cntr/chan=0/0/22, called=71071, calling=6669999
userid=(n/a), ip=0.0.0.0, mask=0.0.0.0
setup=10/16/1999 18:29:20, conn=0.00, phys=0.00, service=0.00, authen=0.00
init rx/tx b-rate=0/0, rx/tx chars=0/0
resource slot/port=(n/a)/(n/a), mp bundle=0, charged units=0, account id=0
duration(sec)=0.00, disc subsys=CSM, disc code=0x1A
disc text=Failed to find DSP resource
-----
```

## Related Commands

Command	Description
<b>show call calltracker active</b>	Displays all of the information stored within the Call Tracker active database for all active calls.
<b>show call calltracker history</b>	Displays all the information stored within the Call Tracker history database table for the most recent disconnected calls.

## show call calltracker history

To display all the information stored within the Call Tracker History Database Table for most recent disconnected calls, use the **show call calltracker history** command.

```
show call calltracker history [category [isdn | modem | other | v110 | v120]]
```

<b>Syntax Description</b>	<b>category</b> <isdn   modem   other   v110   v120>	(Optional) Displays Call Tracker data for a specific type of call. The default is to show all calls, regardless of type. By specifying the category, Call Tracker will only show calls whose records indicate that category.
---------------------------	--	--

<b>Defaults</b>	The activity and configuration information is not displayed. The command <b>show call calltracker history</b> will show all calls, regardless of type, unless specified by the <b>category</b> option field.
-----------------	--

<b>Command Modes</b>	Privileged EXEC
----------------------	-----------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.1(3)T	This command was introduced.





# show call calltracker summary

To display Call Tracker activity and configuration information such as the number of active calls and the history table attributes, use the **show call calltracker summary** privileged EXEC command.

## show call calltracker summary

### Syntax Description

This command has no keywords or arguments.

### Defaults

The activity and configuration information is not displayed.

### Command Modes

Privileged EXEC

### Command History

Release	Modification
12.1(3)T	This command was introduced.

### Examples

The following is sample output from the **show call calltracker summary** command:

```
Router# show call calltracker summary
Call Tracker Status:
  Active Table:
    - 7 call(s)
    - 4473 bytes used (639 average, 639 maximum)
  History Table:
    - 50 of a maximum of 240 call(s) (20% full)
    - 45157 bytes used (903 average, 921 maximum)
    - 260000 minute(s) call retain time
  API Front-end:
    - event elements:512 total, 512 free, 0 in-use
    - free event elements' low watermark:467
    - events dropped due to unavailability of free elts:0
```

Table 1 describes the fields shown in the **show call calltracker summary** display.

**Table 1** show call calltracker summary Command Field Descriptions

Field	Description
<b>Active Table:</b>	
call(s)	Number of active calls.
<i>n</i> bytes used ( <i>m</i> average, <i>o</i> maximum)	<i>n</i> =total memory used for all active calls <i>m</i> =average memory usage per call ( <i>n</i> /calls) <i>o</i> =highest single memory usage for a call

**Table 1** *show call calltracker summary Command Field Descriptions (continued)*

Field	Description
<b>History Table:</b>	
$x$ of a maximum of $n$ calls ( $o\%$ full)	Number of calls in the history table, the maximum allowed (as defined by the <b>calltracker history max-size</b> command), and the percentage of the history table that these calls consume.
$n$ bytes used ( $m$ average, $o$ maximum)	$n$ =total memory used for all active calls $m$ =average memory usage per call ( $n$ /calls) $o$ =highest single memory usage for a call
minute(s) call retain time	Number of minutes calls are retained in the history table. This parameter is configured using the <b>calltracker history retain-mins</b> command.
<b>API Front-end:</b>	
event elements	For Cisco internal use only.
free event elements' low watermark	For Cisco internal use only.
events dropped because of unavailability of free elements	For Cisco internal use only.

**Related Commands**

Command	Description
<b>show call calltracker active</b>	Displays all of the information stored within the Call Tracker active database for all active calls.
<b>show call calltracker history</b>	Displays all the information stored within the Call Tracker history database table for the most recent disconnected calls.

# show modem calltracker

To display all information stored within the Call Tracker active or history database for latest call assigned to specified modem, use the **show modem calltracker** privileged EXEC configuration command. This command allows you to display all Call Tracker data for a given modem when you do not have the call handle readily available and do not want to search the Call Tracker database.

```
show modem calltracker [slot/port]
```

<b>Syntax Description</b>	<i>slot/port</i>	(Optional) Specifies the location of a slot and modem port. Remember to include the forward slash (/) when entering this variable.
---------------------------	------------------	--

<b>Defaults</b>	The activity and configuration information is not displayed.
-----------------	--

<b>Command Modes</b>	Privileged EXEC
----------------------	-----------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.1(3)T	This command was introduced.



```
phase 4 info: 0x01834070808340708000
```

-----

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>show call calltracker active</b>	Displays all information stored within the Call Tracker active database for all active calls.
	<b>show call calltracker history</b>	Displays all the information stored within the Call Tracker history database table for the most recent disconnected calls.
	<b>show call calltracker handle</b>	Displays the detailed data stored within Call Tracker for specific call having specified unique call handle identifier.

## snmp-server enable traps

To enable the router to send SNMP traps, use the **snmp-server enable traps** global configuration command. To disable SNMP traps, use the **no** form of this command. The **calltracker** trap type was added for Cisco IOS Release 12.1(2)XH.

**snmp-server enable traps** [*trap-type*] [*trap-option*]

**no snmp-server enable traps** [*trap-type*] [*trap-option*]

### Syntax Description

*trap-type*

(Optional) Type of trap to enable. If no type is specified, all traps are sent (including the **envmon** and **repeater** traps). The trap type can be one of the following keywords:

- **bgp**—Sends Border Gateway Protocol (BGP) state change traps.
- **calltracker**—Sends Call Tracker traps.
- **config**—Sends configuration traps.
- **entity**—Sends Entity MIB modification traps.
- **envmon**—Sends Cisco enterprise-specific environmental monitor traps when an environmental threshold is exceeded. When the **envmon** keyword is used, you can specify a *trap-option* value.
- **frame-relay**—Sends Frame Relay traps.
- **ipmulticast**—Sends IP Multicast traps.
- **isdn**—Sends Integrated Services Digital Network (ISDN) traps. When the **isdn** keyword is used on Cisco 1600 series routers, you can specify a *trap-option* value.
- **repeater**—Sends Ethernet hub repeater traps. When the **repeater** keyword is selected, you can specify a *trap-option* value.
- **rtr**—Sends Response Time Reporter (RTR) traps.
- **snmp**—Sends Simple Network Management Protocol (SNMP) traps. When the **snmp** keyword is used, you can specify a *trap-option* value.
- **syslog**—Sends error message traps (Cisco SYSLOG MIB). You can specify the level of messages to be sent using the **logging history level** command.
- **voice**—Sends SNMP poor quality of voice traps, when used with the **qov** *trap-option* command.

---

*trap-option*

(Optional) When the **envmon** keyword is used, you can enable a specific environmental trap type, or accept all trap types from the environmental monitor system. If no option is specified, all environmental types are enabled. The option can be one or more of the following keywords:

- **voltage**
- **shutdown**
- **supply**
- **fan**
- **temperature.**

When the **isdn** keyword is used on Cisco 1600 series routers, you can specify the **call-information** keyword to enable an SNMP ISDN call information trap for the ISDN MIB subsystem, or you can specify the **isdnu-interface** keyword to enable an SNMP ISDN U interface trap for the ISDN U interface MIB subsystem.

When the **repeater** keyword is used, you can specify the repeater option. If no option is specified, all repeater types are enabled. The option can be one or more of the following keywords:

- **health**—Enables IETF Repeater Hub MIB (RFC 1516) health trap.
- **reset**—Enables IETF Repeater Hub MIB (RFC 1516) reset trap.

When the **snmp** keyword is used, you can specify the **authentication** option to enable SNMP Authentication Failure traps. (The **snmp-server enable traps snmp authentication** command replaces the **snmp-server trap-authentication** command.) If no option is specified, all SNMP traps are enabled.

When the **voice** keyword is used, you can enable poor quality of voice traps by using the **qov** option.

---



---

**Defaults**

This command is disabled by default. No traps are enabled.

If you enter this command with no keywords, the default is to enable all trap types.

Some trap types cannot be controlled with this command. These traps are either always enabled or enabled/disabled by some other means. For example, the linkUpDown messages are disabled by the **no snmp trap link-status** command.

---

**Command Modes**

Global configuration

**Command History**

Release	Modification
11.1	This command was introduced.
11.3(1) MA	The voice trap type was added.
12.1(3)T	The calltracker trap type was added.

**Usage Guidelines**

This command is useful for disabling traps that are generating a large amount of uninteresting or useless noise.

If you do not enter an **snmp-server enable traps** command, no traps controlled by this command are sent. In order to configure the router to send these SNMP traps, you must enter at least one **snmp-server enable traps** command. If you enter the command with no keywords, all trap types are enabled. If you enter the command with a keyword, only the trap type related to that keyword is enabled. To enable multiple types of traps, you must issue a separate **snmp-server enable traps** command for each trap type and option.

The **snmp-server enable traps** command is used in conjunction with the **snmp-server host** command. Use the **snmp-server host** command to specify which host or hosts receive SNMP traps. To send traps, you must configure at least one **snmp-server host** command.

For a host to receive a trap controlled by this command, both the **snmp-server enable traps** command and the **snmp-server host** command for that host must be enabled. If the trap type is not controlled by this command, only the appropriate **snmp-server host** command must be enabled.

The trap types used in this command all have an associated MIB object that allows them to be globally enabled or disabled. Not all of the trap types available in the **snmp-server host** command have notificationEnable MIB objects, so some of these cannot be controlled using the **snmp-server enable traps** command.

**Examples**

The following example enables the router to send Call Tracker traps:

```
configure terminal
snmp-server enable traps calltracker
```

The following example enables the router to send poor quality of voice traps:

```
configure terminal
snmp-server enable traps voice poor-qov
```

The following example enables the router to send all traps to the myhost.cisco.com host using the *public* community string:

```
snmp-server enable traps
snmp-server host myhost.cisco.com public
```

The following example enables the router to send Frame Relay and environmental monitor traps to the myhost.cisco.com host using the *public* community string:

```
snmp-server enable traps frame-relay
snmp-server enable traps envmon temperature
snmp-server host myhost.cisco.com public
```

The following example will not send traps to any host. The BGP traps are enabled for all hosts, but the only traps enabled to be sent to a host are ISDN traps.

```
snmp-server enable traps bgp
snmp-server host bob public isdn
```

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>snmp enable peer-trap poor-qov</b>	Generates poor quality of voice notification for applicable calls associated with VoIP dial peers.
	<b>snmp-server host</b>	Specifies the recipient of an SNMP notification operation.
	<b>snmp-server trap-source</b>	Specifies the interface (and hence the corresponding IP address) from where an SNMP trap should originate.
	<b>snmp trap illegal-address</b>	Issues an SNMP trap when a MAC address violation is detected on an Ethernet hub port.
	<b>snmp trap link-status</b>	Enables SNMP link trap generation.

## snmp-server host

To specify the recipient of an SNMP notification operation, use the **snmp-server host** global configuration command. To remove the specified host, use the **no** form of this command. The **calltracker** notification type was added for Cisco IOS Release 12.1(2)XH.

```
snmp-server host host [traps | informs] [version {1 | 2c}] community-string [udp-port port]
[notification-type]
```

```
no snmp-server host host [traps | informs]
```

Syntax Description	
<i>host</i>	Name or Internet address of the host.
<b>traps</b>	(Optional) Sends SNMP traps to this host. This is the default.
<b>informs</b>	(Optional) Sends SNMP informs to this host.
<b>version</b>	(Optional) Version of the Simple Network Management Protocol (SNMP) used to send the traps. <ul style="list-style-type: none"> <li>• <b>1</b>—SNMPv1. This option is not available with informs.</li> <li>• <b>2c</b> —SNMPv2C.</li> </ul>
<i>community-string</i>	Password-like community string sent with the notification operation.

<b>udp-port</b> <i>port</i>	UDP port of the host to use. The default is 162.
<i>notification-type</i>	<p>(Optional) Type of notification to be sent to the host. If no type is specified, all notifications are sent. The notification type can be one or more of the following keywords:</p> <ul style="list-style-type: none"> <li>• <b>bgp</b>—Sends Border Gateway Protocol (BGP) state change notifications.</li> <li>• <b>calltracker</b>—Sends Call Tracker notifications.</li> <li>• <b>config</b>—Sends configuration notifications.</li> <li>• <b>dspu</b>—Sends downstream physical unit (DSPU) notifications.</li> <li>• <b>entity</b>—Sends Entity MIB modification notifications.</li> <li>• <b>envmon</b>—Sends Cisco enterprise-specific environmental monitor notifications when an environmental threshold is exceeded.</li> <li>• <b>frame-relay</b>—Sends Frame Relay notifications.</li> <li>• <b>isdn</b>—Sends Integrated Services Digital Network (ISDN) notifications.</li> <li>• <b>llc2</b>—Sends Logical Link Control, type 2 (LLC2) notifications.</li> <li>• <b>rptr</b>—Sends standard repeater (hub) notifications.</li> <li>• <b>rsrb</b>—Sends remote source-route bridging (RSRB) notifications.</li> <li>• <b>rtr</b>—Sends response time reporter (RTR) notifications.</li> <li>• <b>sdlc</b>—Sends Synchronous Data Link Control (SDLC) notifications.</li> <li>• <b>sdllc</b>—Sends SDLC Logical Link Control (SDLLC) notifications.</li> <li>• <b>snmp</b>—Sends Simple Network Management Protocol (SNMP) notifications defined in RFC 1157.</li> <li>• <b>stun</b>—Sends serial tunnel (STUN) notifications.</li> <li>• <b>syslog</b>—Sends error message notifications (Cisco SYSLOG MIB). You can specify the level of messages to be sent with the <b>logging history level</b> command.</li> <li>• <b>tty</b>—Sends Cisco enterprise-specific notifications when a Transmission Control Protocol (TCP) connection closes.</li> <li>• <b>x25</b>—Sends X.25 event notifications.</li> </ul>

### Defaults

This command is disabled by default. No notifications are sent.

If you enter this command with no keywords, the default is to send all trap types to the host. No informs will be sent to this host.

If no **version** keyword is present, the default is version 1. If no **traps** or **informs** keyword is present, traps are enabled.

The **no snmp-server host** command with no keywords will disable traps, but not informs, to the host. In order to disable informs, use the **no snmp-server host informs** command.

### Command Modes

Global configuration

### Command History

Release	Modification
10.0	This command was introduced.
11.3(1) MA	The voice trap type was added.
12.1(3)T	The calltracker notification type was added.

### Usage Guidelines

SNMP notifications can be sent as traps or inform requests. Traps are unreliable because the receiver does not send acknowledgments when it receives traps. The sender cannot determine if the traps were received. However, an SNMP entity that receives an inform request acknowledges the message with an SNMP response PDU. If the sender never receives the response, the inform request can be sent again. Thus, informs are more likely to reach their intended destination.

However, informs consume more resources in the agent and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in memory until a response is received or the request times out. Although traps are sent only once, an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network.

If you do not enter an **snmp-server host** command, no notifications are sent. In order to configure the router to send SNMP notifications, you must enter at least one **snmp-server host** command. If you enter the command with no keywords, all trap types are enabled for the host. In order to enable multiple hosts, you must issue a separate **snmp-server host** command for each host. You can specify multiple notification types in the command for each host.

When multiple **snmp-server host** commands are given for the same host and kind of notification (trap or inform), each succeeding command overwrites the previous command. Only the last **snmp-server host** command will be in effect. For example, if you enter an **snmp-server host inform** command for a host and then enter another **snmp-server host inform** command for the same host, the second command will replace the first.

The **snmp-server host** command is used in conjunction with the **snmp-server enable** command. Use the **snmp-server enable** command to specify which SNMP notifications are sent globally. For a host to receive most notifications, at least one **snmp-server enable** command and the **snmp-server host** command for that host must be enabled.

However, some notification types cannot be controlled with the **snmp-server enable** command. For example, some notification types are always enabled. Other notification types are enabled by a different command. For example, the linkUpDown notifications are controlled by the **snmp trap link-status** command. These notification types do not require an **snmp-server enable** command.

A notification-type option availability depends on the router and Cisco IOS software features supported on the router. For example, the **envmon** notification-type is available only if the environmental monitor is part of the system.

**Examples**

The following example sends Call Tracker traps to the host address 172.30.2.160 using the community string public:

```
snmp-server enable traps calltracker
snmp-server host 172.30.2.160 public calltracker
```

The following example sends the SNMP and Cisco environmental monitor enterprise-specific traps to address 172.30.2.160:

```
snmp-server enable traps
snmp-server host 172.30.2.160 public snmp envmon
```

The following example sends all traps to the host myhost.cisco.com using the community string public:

```
snmp-server enable traps
snmp-server host myhost.cisco.com public
```

The following example does not send traps to any host. The BGP traps are enabled for all hosts, but only the ISDN traps are enabled to be sent to a host.

```
snmp-server enable traps bgp
snmp-server host bob public isdn
```

The following example enables the router to send all inform requests to the host myhost.cisco.com using the *public* community string:

```
snmp-server enable traps
snmp-server host myhost.cisco.com informs version 2c public
```

**Related Commands**

Command	Description
<b>show call calltracker active</b>	Displays all information stored within the Call Tracker active database for all active calls.
<b>snmp-server informs</b>	Specifies inform request options.
<b>snmp-server trap-source</b>	Specifies the interface (and hence the corresponding IP address) from where an SNMP trap should originate.
<b>snmp-server trap-timeout</b>	Defines how often to try resending trap messages on the retransmission queue.

# Debug Commands

This section describes new **debug** commands related to the Call Tracker and other features in this feature set. All other commands used with this feature are documented in the Cisco IOS Release 12.1 command reference publications.

- **debug aaa pod**
- **debug calltracker**
- **debug ip mhbeat**

# debug aaa pod

To display debug messages related to POD packets, use the **debug aaa pod** privileged EXEC command. To disable debugging output, use the **no** form of this command.

**debug aaa pod**

**no debug aaa pod**

## Syntax Description

This command has no keywords or arguments.

## Defaults

Debugging for POD packets is not enabled.

## Command History

Release	Modification
12.1(3)T	This command was introduced.

## Examples

The following example shows output from a successful POD request, when using the **show debug** command.

```
Router# debug aaa pod
AAA POD packet processing debugging is on
Router# show debug
General OS:
  AAA POD packet processing debugging is on
Router#
*Jul  9 16:04:32.271:POD:10.100.1.34 request queued
*Jul  9 16:04:32.271:POD:10.100.1.34 user  0.0.0.0 sessid 0x0 key 0xA5AFA004
*Jul  9 16:04:32.271:POD:      Line      User      IDB          Session Id Key
*Jul  9 16:04:32.271:POD:Skip Se0:21  meklund  0.0.0.0      0x0          0x0
*Jul  9 16:04:32.271:POD:KILL Se0:22  meklund  0.0.0.0      0x60000020 0xA5AFA004
*Jul  9 16:04:32.271:POD:Sending ACK to 10.100.1.34/1812
---
Interface Se0:22 was killed because the pod request contained a key of
0xA5AFA004 and pod was configured with the command

aaa pod server port 1812 auth-type any server-key mykey
```

## Related Commands

Command	Description
aaa pod server	Enables the POD feature.

# debug calltracker

To display debug messages tracing the Call Tracker processing flow, use the **debug calltracker** privileged EXEC command. To disable debugging output, use the **no** form of this command.

**debug calltracker**

**no debug calltracker**

---

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

---

**Defaults** Call Tracker debugging is not enabled.

---

Command History	Release	Modification
	12.1(3)T	This command was introduced.

---

**Examples**

The following example shows output for a call coming up, when using the **show debug** command:

```

Router# debug calltracker
Router# show debug
Call Tracker:
  Call Tracker's Registry API debugging is on
  Call Tracker's MIB API debugging is on
  Call Tracker's data base debugging is on
Router#
*Jan  1 17:33:00.195:
CT:find_actv_by_ct_hndl:node (hndl=416) not in active table
*Jan  1 17:33:00.195:Posting 19 event ../call-mgmt/calltrkr_api_frontend.c:801
CT:reading new event(s) from API front-end
*Jan  1 17:33:00.195:
CT:find_actv_by_ct_hndl:node (hndl=416) not in active table
CT:node (hndl=416) found in history table
CT:actv-->hist ok:node (hndl=416) is now (or already was) in history table
*Jan  1 17:33:00.319:ISDN Se0:23:calltrkr_disconnect hndl=416
*Jan  1 17:33:00.319:
Router#
CT:find_actv_by_ct_hndl:node (hndl=416) not in active table
*Jan  1 17:33:00.319:Posting 19 event ../call-mgmt/calltrkr_api_frontend.c:801
CT:reading new event(s) from API front-end
*Jan  1 17:33:00.319:
CT:find_actv_by_ct_hndl:node (hndl=416) not in active table
CT:node (hndl=416) found in history table
CT:actv-->hist ok:node (hndl=416) is now (or already was) in history table
*Jan  1 17:33:00.331:%LINK-3-UPDOWN:Interface Serial0:20, changed state to down
*Jan  1 17:33:00.331:Se0:20 PPP/CT/disconnect:zero handle detected, idb=0x61C25A84,
code=3
*Jan  1 17:33:02.183:%LINK-5-CHANGED:Interface Async8, changed state to reset
Router#
*Jan  1 17:33:02.183:As8 PPP/CT/disconnect:ct_hndl=416, idb=0x62235310
*Jan  1 17:33:02.183:
CT:find_actv_by_ct_hndl:node (hndl=416) not in active table
*Jan  1 17:33:02.183:Posting 19 event ../call-mgmt/calltrkr_api_frontend.c:801
CT:reading new event(s) from API front-end
*Jan  1 17:33:02.187:
CT:find_actv_by_ct_hndl:node (hndl=416) not in active table
CT:node (hndl=416) found in history table
CT:actv-->hist ok:node (hndl=416) is now (or already was) in history table
*Jan  1 17:33:03.191:%LINEPROTO-5-UPDOWN:Line protocol on Interface Async8, changed state
to down
Router#
*Jan  1 17:33:05.131:Posting 18 event ../call-mgmt/calltrkr_api_frontend.c:764
CT:reading new event(s) from API front-end
CT:calltrkr_dspbytecount():
  ct_hndl=416, tx/rx=426/647349
CT:node (hndl=416) found in history table
Router#
*Jan  1 17:33:07.183:%LINK-3-UPDOWN:Interface Async8, changed state to down

```

**Note**

The above output is an example only. The fields and type of information that are displayed in your output may vary when using the **debug calltracker** command.

**Related Commands**

Command	Description
<b>show call calltracker summary</b>	Displays Call Tracker activity and configuration information.

# debug ip mhbeat

To monitor the action of the heartbeat trap, use the **debug ip mhbeat** privileged EXEC command. To disable debugging output, use the **no** form of this command.

**debug ip mhbeat**

**no debug ip mhbeat**

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

**Defaults** Debugging is not enabled.

Command History	Release	Modification
	12.1(2)XH	This command was introduced.

## Examples

```
Router# debug ip mhbeat
IP multicast heartbeat debugging is on
Router# debug snmp packets
SNMP packet debugging is on

!
Router(config)# ip multicast heartbeat intervals-of 10
Dec 23 13:34:21.132: MHBEAT: ip multicast-heartbeat group 224.0.1.53 port 0
source 0.0.0.0 0.0.0.0 at-least 3 in 5 intervals-of 10 secondsd
Router#
Dec 23 13:34:23: %SYS-5-CONFIG_I: Configured from console by console
Dec 23 13:34:31.136: MHBEAT: timer ticked, t=1,i=1,c=0
Dec 23 13:34:41.136: MHBEAT: timer ticked, t=2,i=2,c=0
Dec 23 13:34:51.136: MHBEAT: timer ticked, t=3,i=3,c=0
Dec 23 13:35:01.136: MHBEAT: timer ticked, t=4,i=4,c=0
Dec 23 13:35:11.136: MHBEAT: timer ticked, t=5,i=0,c=0
Dec 23 13:35:21.135: Send SNMP Trap for missing heartbeat
Dec 23 13:35:21.135: SNMP: Queuing packet to 171.69.55.12
Dec 23 13:35:21.135: SNMP: V1 Trap, ent ciscoExperiment.2.3.1, addr 4.4.4.4, gentrap 6,
spectrap 1
ciscoIpMRouteHeartBeat.1.0 = 224.0.1.53
ciscoIpMRouteHeartBeat.2.0 = 0.0.0.0
ciscoIpMRouteHeartBeat.3.0 = 10
ciscoIpMRouteHeartBeat.4.0 = 5
ciscoIpMRouteHeartBeat.5.0 = 0
ciscoIpMRouteHeartBeat.6.0 = 3
```

Related Commands	Command	Description
	<b>ip multicast heartbeat</b>	Monitors the health of multicast delivery and alerts when the delivery fails to meet certain parameters.

# Glossary

**AAA**—The services of authentication, authorization, and accounting

**Access-Accept**—A response packet from the RADIUS server notifying the access server that the user is authenticated. This packet contains the user profile, which defines the specific AAA functions assigned to the user.

**Access-Challenge**—A response packet from the RADIUS server requesting that the user supply additional information before being authenticated.

**Access-Request**—A request packet sent to the RADIUS server by the access server requesting authentication of the user.

**accounting**—The process of recording what a user is doing.

**authentication**—The process of determining who a user is.

**authorization**—The process of determining what a user can do.

**B channel**—bearer channel. In ISDN, a full-duplex, 64-kbps channel used to send user data.

**CAS**—channel associated signaling. Call signaling that enables the access server to send or receive analog calls.

**cause codes**—(defined by ITU Series Q Recommendation 850) Code that indicates the reason for ISDN call failure or completion.

**DNIS**—Dialed Number Identification Service, also known as the called party number. The telephone number of the called party after translation occurs in the Public Switched Telephone Network. A given destination may have a different DNIS number based on how the call is placed (for example, 800 or direct dial).

**DS0**—Digital signal level 0. Framing specification used in transmitting digital signals over a single channel at 64-kbps on a T1 facility.

**ISDN**—Integrated Services Digital Network. Communication protocol offered by telephone companies that permits telephone networks to carry data, voice, and other source traffic.

**MIB**—Management Information Base. Database of network management information that is used and maintained by a network management protocol such as SNMP. The value of a MIB object can be changed or retrieved using SNMP commands, usually through a GUI network management system.

**NOC**—network operations center. Organization responsible for maintaining a network.

**POD**—Packet of disconnect. A process that allows a PPP session to be verified and then terminated by the network access server.

**PPP**—Point-to-Point Protocol. Successor to SLIP that provides router-to-router and host-to-network connections over synchronous and asynchronous circuits.

**PRI**—Primary Rate Interface. ISDN interface to primary rate access. Primary rate access consists of a single 64-kbps D channel plus 23 (T1) or 30 (E1) B channels for voice or data.

**RADIUS**—Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server containing all user authentication and network-service access information.

**SNMP**—Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security.

**switch**—Network device that filters, forwards, and floods frames based on the destination address of each frame. The switch operates at the data link layer of the OSI model.

**TACACS+**—Terminal Access Controller Access Control System Plus. Authentication protocol, developed by the DDN community, that provides remote access authentication and related services, such as event logging. User passwords are administered in a central database rather than in individual routers, providing an easily scalable network security solution.

**TCP-Clear**—A raw TCP dialup connection, not using the Telnet protocol. It allows a direct connection from the user's termination point on the network access server to the destination specified by the Login-IP-Host and Login-TCP-Port.

**Note**

---

For a list of other internetworking terms, see *Internetworking Terms and Acronyms*, available on the Documentation CD-ROM and Cisco Connection Online (CCO) at the following URL: <http://www.cisco.com/univercd/cc/td/doc/cisintwk/ita/index.htm>.

---