Configuring SIP DTMF Features

First Published: October 24, 2001
Last Updated: April 18, 2010

This chapter describes the following SIP features that support dual-tone multifrequency (DTMF) signaling:

- RFC 2833 DTMF Media Termination Point (MTP) Passthrough
- DTMF Events Through SIP Signaling
- DTMF Relay for SIP Calls Using Named Telephone Events
- SIP INFO Method for DTMF Tone Generation
- SIP NOTIFY-Based Out-of-Band DTMF Relay Support
- SIP KPML-Based Out-of-Band DTMF Relay Support
- SIP Support for Asymmetric SDP

Feature History for the RFC 2833 DTMF MTP Passthrough

Release	Modification
12.4(11)T	This feature was introduced.

Feature History for DTMF Events Through SIP Signaling

Release	Modification
12.2(11)T	This feature was introduced.

Feature History for DTMF Relay for SIP Calls Using NTE

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(2)XB1	This feature was implemented on an additional platform.
12.2(8)T	This feature was integrated into this release.
12.2(11)T	This feature was implemented on additional platforms.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Feature History for SIP INFO Method for DTMF Tone Generation

Release	Modification
12.2(11)T	This feature was introduced.

Feature History for SIP NOTIFY-Based Out-of-Band DTMF Relay Support

Release	Modification
12.3(4)T	This feature was introduced.

Feature History for SIP KPML-Based Out-of-Band DTMF Relay Support

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

Feature History for the SIP Support for Asymmetric SDP

Release	Modification
12.4(15)T	This feature was introduced.

Finding Feature Information

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

Contents

Restrictions for SIP DTMF

RFC 2833 DTMF MTP Passthrough Feature

- The RFC 2833 DTMF MTP Passthrough feature adds support for passing Dual-Tone Multifrequency (DTMF) tones transparently between Session Initiation Protocol (SIP) endpoints that require either transcoding or use of the RSVP Agent feature. If the T38 Fax Relay feature is also configured on this IP network, configure the voice gateways to use a payload type other than PT97 or PT98 for fax relay negotiation, or depending on whether the SIP endpoints support different payload types, configure Cisco Unified CME to use a payload type other than PT97 or PT98 for DTMF.

DTMF Events Through SIP Signaling Feature

- The DTMF Events Through SIP Signaling feature adds support for sending telephone-event notifications via SIP NOTIFY messages from a SIP gateway. The events for which notifications are sent out are DTMF events from the local Plain Old Telephone Service (POTS) interface on the gateway. Notifications are not sent for DTMF events received in the Real-Time Transport Protocol (RTP) stream from the recipient user agent.

DTMF Relay for SIP Calls Using NTEs Feature

- The SIP NTE DTMF relay feature is available only for SIP calls on Cisco VoIP gateways. The SIP NTE DTMF relay feature supports only hookflash relay and does not support hookflash generation for advanced features such as call waiting and conferencing.

SIP INFO Method for DTMF Tone Generation Feature

- Minimum signal duration is 100 ms. If a request is received with a duration less than 100 ms, the minimum duration of 100 ms is used by default.
- Maximum signal duration is 5000 ms. If a request is received with a duration longer than 5000 ms, the maximum duration of 5000 ms is used by default.
- If no duration parameter is included in a request, the gateway defaults to a signal duration of 250 ms.

SIP NOTIFY-Based Out-of-Band DTMF Relay Support Feature

- To support Skinny Client Control Protocol (SCCP) IP phones, originating and terminating SIP gateways can use NOTIFY-based out-of-band DTMF relay. NOTIFY-based out-of-band DTMF relay is a Cisco proprietary function.
- You can configure support only on a SIP VoIP dial peer.

SIP KPML-Based Out-of-Band DTMF Relay Support Feature

- For incoming dial peers, if you configure multiple DTMF negotiation methods, the first value you configure takes precedence, then the second, and then the third.
- For incoming dial peers, the first out-of-band negotiation method takes precedence over other DTMF negotiation methods, except when the **dtmf-relay rtp-nte** command has precedence; in this case, the **dtmf-relay sip-kpml** command takes precedence over other out-of-band negotiation methods.
- For incoming dial peers, if both the **dtmf-relay rtp-nte** and **dtmf-relay sip-kpml** commands and notification mechanisms are enabled and negotiated, the gateway relies on RFC 2833 notification to receive digits and a SUBSCRIBE for KPML is not initiated.
- SIP KPML support complies to the IEFT draft “draft-ietf-sipping-kpml-04.txt” with the following limitations:
  - The SIP gateway always initiates SUBSCRIBE in the context of an established INVITE dialog. The gateway supports receiving SUBSCRIBE in the context of an established INVITE dialog, as well as out-of-call context requests with a leg parameter in the Event header. If the request code does not match an existing INVITE dialog, the gateway sends a NOTIFY with KPML status-code 481 and sets Subscription-State to terminated.
  - The gateway does not support the Globally Routable User Agent (GRUU) requirement. The Contact header in the INVITE/200 OK message is generated locally from the gateway’s contact information.
  - The gateway always initiates persistent subscriptions, but the gateway receives and processes persistent and one-shot subscriptions.

  - The gateway supports only single-digit reporting. There is no need for inter-digit timer support. The only regular expressions supported are those which match a single digit. For example:

    <regex>x</regex>—Matches any digit 0 through 9

    <regex>1</regex>—Matches digit 1

    <regex>[x#*ABCD]</regex>—Matches to any digit 0 through 9, # (the pound sign), * (an asterisk), or A, B, C, or D

    <regex>[24]</regex>—Matches digits 2 or 4

    <regex>[2-9]</regex>—Matches on any digit 2 through 9

    <regex>[^2-9]</regex>—Matches digits 0 or 1
  - The gateway does not support long key presses, which are detected and reported as a single digit press.
  - Digit suppression is not supported (pre tag for suppressing inband digits).
  - Individual stream selection is not supported. A SUBSCRIBE request for KPML applies to all audio streams in the dialog (stream element and reverse are not supported).
- You can configure support only on a SIP VoIP dial peer.
- In Cisco Unified Border Element (Cisco UBE), RTP-NTE to RTP-NTE DTMF interworking is not supported when you use High Density Voice Network Module (NM-HDV) for transcoding.

Prerequisites for SIP DTMF

DTMF Relay for SIP Calls Using NTEs Feature

- Ensure that you have a working VoIP network using SIP on Cisco gateways.

Information About SIP DTMF

To configure SIP DTMF support, you should understand the following concepts:

RFC 2833 DTMF MTP Passthrough

The RFC 2833 DTMF Media Termination Point (MTP) Passthrough feature passes DTMF tones transparently between Session Initiation Protocol (SIP) endpoints that require either transcoding or use of the RSVP Agent feature. (An RSVP agent is a Cisco IOS-based Resource Reservation Protocol [RSVP] proxy server that registers with the call manager—Cisco Unified CallManager or Cisco Unified CallManager Express—as a media-termination point or a transcoder device.)

The MTP or transcoding module on a gateway detects RFC 2833 (DTMF) packets from an IP endpoint. You can configure whether it should do either or both of the following:

- Generate and send an out-of-band signal event to the call manager
- Pass the packets through to the other IP endpoint (default)

You can configure this instruction from the call manager, from the gateway, or both. The gateway can itself contain a call manager with an MTP or transcoder, or it can connect to another gateway that contains a call manager with an MTP or transcoder. Configuration on the call manager takes precedence over configuration on the gateway.

DTMF Events Through SIP Signaling

The DTMF Events Through SIP Signaling feature provides the following benefits:

- Provides DTMF event notification for SIP messages.
- Provides the capability of receiving hookflash event notification through the SIP NOTIFY method.
- Enables third-party call control, or other signaling mechanisms, to provide enhanced services, such as calling card and messaging services.
- Allows the user to communicate with the application outside of the media connection.

The DTMF Events Through SIP Signaling feature allows telephone event notifications to be sent through SIP NOTIFY messages, using the SIP SUBSCRIBE/NOTIFY method as defined in the Internet Engineering Task Force (IETF) draft, *SIP-Specific Event Notification.* The feature also supports sending DTMF notifications based on the IETF draft, *draft-mahy-sip-signaled-digits-01.txt, Signaled Telephony Events in the Session Initiation Protocol (SIP).*

DTMF Dialing

DTMF dialing consists of simultaneous voice-band tones generated when a button is pressed on a telephone. The use of DTMF signaling for this feature enables support for advanced telephony services. Currently there are a number of application servers and service creation platforms that do not support media connections. To provide value-added services to the network, these servers and platforms need to be aware of signaling events from a specific participant in the call. Once the server or platform is aware of the DTMF events that are being signaled, it can use third-party call control, or other signaling mechanisms, to provide enhanced services. Examples of the types of services and platforms that are supported by this feature are various voice web browser services, Centrex switches or business service platforms, calling card services, and unified message servers. All of these applications require a method for the user to communicate with the application outside of the media connection. The DTMF Events Through SIP Signaling feature provides this signaling capability.

This feature is related to the SIP INFO Method for DTMF Tone Generation feature, which adds support for out-of-band DTMF tone generation using the SIP INFO method. Together the two features provide a mechanism to both send and receive DTMF digits along the signaling path.

NOTIFY Messages

The SIP event notification mechanism uses NOTIFY messages to signal when certain telephony events take place. In order to send DTMF signals through NOTIFY messages, the gateway notifies the subscriber when DTMF digits are signaled by the originator. The notification contains a message body with a SIP response status line.

The following sample message shows a NOTIFY message from the Notifier letting the Subscriber know that the subscription is completed. The combination of the From, To, and Call-ID headers identifies the call leg. The Events header specifies the event type being signaled, and the Content-Type specifies the Internet media type. The Content-Length header indicates the number of octets in the message body.

```
NOTIFY sip:subscriber@example1.com SIP/2.0
Via: SIP/2.0/UDP example2.com:5060
From: Notifier <sip:notifier@example2.com>;tag=5678-EFGH
To: Subscriber <sip:subscriber@example1.com>;tag=1234-ABCD
Call-ID: 12345@example2.com
CSeq: 104 NOTIFY
Contact: Notifier <sip:notifier@example2.com>
Events: telephone-event;rate=1000
Content-Type: audio/telephone-event
Content-Length: 4
```

DTMF Relay for SIP Calls Using NTEs

Feature benefits include the following:

- Reliable DTMF digit relay between Cisco VoIP gateways when low-bandwidth codecs are used
- Ability to communicate with SIP phone software that uses NTE packets to indicate DTMF digits

The SIP NTE DTMF relay feature is used for the following applications:

- Reliable DTMF Relay, page 6
- SIP IP Phone Support, page 7

Note This feature is implemented only for SIP calls on VoIP gateways.

Reliable DTMF Relay

The SIP NTE DTMF relay feature provides reliable digit relay between Cisco VoIP gateways when a low-bandwidth codec is used. Using NTE to relay DTMF tones provides a standardized means of transporting DTMF tones in Real-Time Transport Protocol (RTP) packets according to section 3 of RFC 2833, *RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*, developed by the Internet Engineering Task Force (IETF) Audio/Video Transport (AVT) working group. RFC 2833 defines formats of NTE RTP packets used to transport DTMF digits, hookflash, and other telephony events between two peer endpoints.

DTMF tones are generated when a button on a touch-tone phone is pressed. When the tone is generated, it is compressed, transported to the other party, and decompressed. If a low-bandwidth codec, such as a G.729 or G.723 is used without a DTMF relay method, the tone may be distorted during compression and decompression.

With the SIP NTE DTMF relay feature, the endpoints perform per-call negotiation of the DTMF relay method. They also negotiate to determine the payload type value for the NTE RTP packets.

In a SIP call, the gateway forms a Session Description Protocol (SDP) message that indicates the following:

- If NTE will be used
- Which events will be sent using NTE
- NTE payload type value

The SIP NTE DTMF relay feature can relay hookflash events in the RTP stream using NTP packets.

Note The SIP NTE DTMF relay feature does not support hookflash generation for advanced features such as call waiting and conferencing.

SIP IP Phone Support

The SIP NTE DTMF relay feature adds SIP phone support. When SIP IP phones are running software that does not have the capability to generate DTMF tones, the phones use NTE packets to indicate DTMF digits. With the SIP NTE DTMF relay feature, Cisco VoIP gateways can communicate with SIP phones that use NTE packets to indicate DTMF digits. The Cisco VoIP gateways can relay the digits to other endpoints.

SIP INFO Method for DTMF Tone Generation

This section describes the SIP INFO Method for DTMF Tone Generation feature, which uses the SIP INFO method to generate dual-tone multifrequency (DTMF) tones on the telephony call leg. SIP methods or request message types, request a specific action be taken by another user agent or proxy server. The SIP INFO message is sent along the signaling path of the call. With the feature, upon receipt of a SIP INFO message with DTMF relay content, the gateway generates the specified DTMF tone on the telephony end of the call.

The SIP INFO Method for DTMF Tone Generation feature is always enabled, and is invoked when a SIP INFO message is received with DTMF relay content. This feature is related to the SIP NOTIFY-Basec Out-of-Band DTMF Relay Support feature, which provides the ability for an application to be notified about DTMF events using SIP NOTIFY messages. Together, the two features provide a mechanism to both send and receive DTMF digits along the signaling path.

Note For information on sending DTMF event notification using SIP NOTIFY messages, see the "DTMF Events Through SIP Signaling" section on page 5.

SIP INFO Messages

The SIP INFO method is used by a user agent to send call signaling information to another user agent with which it has an established media session. The following example shows a SIP INFO message with DTMF content:

```
INFO sip:2143302100@172.17.2.33 SIP/2.0
Via: SIP/2.0/UDP 172.80.2.100:5060
From:  <sip:9724401003@172.80.2.100>;tag=43
To:  <sip:2143302100@172.17.2.33>;tag=9753.0207
Call-ID: 984072_15401962@172.80.2.100
CSeq: 25634 INFO
Supported: 100rel
```

```
Supported: timer
Content-Length: 26
Content-Type: application/dtmf-relay

Signal= 1
Duration= 160
```

This sample message shows a SIP INFO message received by the gateway with specifics about the DTMF tone to be generated. The combination of the From, To, and Call-ID headers identifies the call leg. The signal and duration headers specify the digit, in this case 1, and duration, 160 milliseconds in the example, for DTMF tone play.

SIP NOTIFY-Based Out-of-Band DTMF Relay

SCCP IP phones do not support in-band DTMF digits; they are capable of sending only out-of-band DTMF digits. To support SCCP devices, originating and terminating SIP gateways can use Cisco proprietary NOTIFY-based out-of-band DTMF relay. In addition, NOTIFY-based out-of-band DTMF relay can also be used by analog phones attached to analog voice ports (FXS) on the router.

NOTIFY-based out-of-band DTMF relay sends messages bidirectionally between the originating and terminating gateways for a DTMF event during a call. If multiple DTMF relay mechanisms are enabled on a SIP dial peer and are negotiated successfully, NOTIFY-based out-of-band DTMF relay takes precedence.

The originating gateway sends an Invite message with a SIP Call-Info header to indicate the use of NOTIFY-based out-of-band DTMF relay. The terminating gateway acknowledges the message with an 18x or 200 Response message, also using the Call-Info header. The Call-Info header for NOTIFY-based out-of-band relay appears as follows:

```
Call-Info: <sip: address>; method="NOTIFY;Event=telephone-event;Duration=msec"
```

Note Duration is the interval between NOTIFY messages sent for a single digit and is set by means of the **notify telephone-event** command.

First, the NOTIFY-based out-of-band DTMF relay mechanism is negotiated by the SIP Invite and 18x/200 Response messages. Then, when a DTMF event occurs, the gateway sends a SIP NOTIFY message for that event. In response, the gateway expects to receive a 200 OK message.

The NOTIFY-based out-of-band DTMF relay mechanism is similar to the DTMF message format described in RFC 2833. NOTIFY-based out-of-band DTMF relay consists of 4 bytes in a binary encoded format. The message format is shown in Figure 74; field descriptions are listed in Table 43.

Figure 74 Message Format of NOTIFY-Based Out-of-Band DTMF Relay

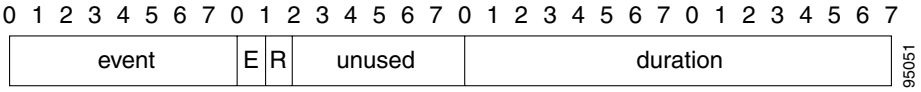


Table 43 Fields in NOTIFY-based out-of-band DTMF relay Message

Field	Description
event	The DTMF event that is between 0-9, A, B, C, D, #, * and flash.
E	E signifies the end bit. If E is set to a value of 1, the NOTIFY message contains the end of the DTMF event. Thus, the duration parameter in this final NOTIFY message measures the complete duration of the event.
R	Reserved.
unused	In RFC 2833, unused corresponds to the volume field, but is not used in NOTIFY-based out-of-band DTMF relay.
duration	Duration of this DTMF event, in milliseconds.

Sending NOTIFY Messages

As soon as the DTMF event is recognized, the gateway sends out an initial NOTIFY message for this event with the duration negotiated in the Invite's Call-Info header. For the initial NOTIFY message, the end bit is set to zero. Afterward, one of the following actions can occur:

- If the duration of the DTMF event is less than the negotiated duration, the originating gateway sends an end NOTIFY message for this event with the duration field containing the exact duration of the event and the end bit set to 1.
- If the duration of the DTMF event is greater than the negotiated duration, the originating gateway sends another NOTIFY message for this event after the initial timer runs out. The updated NOTIFY message has a duration of twice the negotiated duration. The end bit is set to 0 because the event is not yet over. If the event lasts beyond the duration specified in the first updated NOTIFY message, another updated NOTIFY message is sent with three times the negotiated duration.
- If the duration of the DTMF event is exactly the negotiated duration, either of the above two actions occurs, depending on whether the end of the DTMF event occurred before or after the timer ran out.

For example, if the negotiated duration is 600 ms, as soon as a DTMF event occurs, the initial NOTIFY message is sent with duration as 600 ms. Then a timer starts for this duration.

- If the DTMF event lasts only 300 ms, the timer stops and an end NOTIFY message is sent with the duration as 300 ms.
- If the DTMF event lasts longer than 600 ms (1000 ms), when the timer expires an updated NOTIFY message is sent with the duration as 1200 ms and the timer restarts. When the DTMF event ends, an end NOTIFY message is sent with the duration set to 1000 ms.

Every DTMF event corresponds to at least two NOTIFYs: an initial NOTIFY message and an end NOTIFY message. There might also be some update NOTIFYs involved, if the total duration of the event is greater than the negotiated max-duration interval. Because DTMF events generally last for less than 1000 ms, setting the duration using **notify telephone-event** command to more than 1000 ms reduces the total number of NOTIFY messages sent. The default value of **notify telephone-event** command is 2000 ms.

Receiving NOTIFY Messages

Once a NOTIFY message is received by the terminating gateway, the DTMF tone plays and a timer is set for the value in the duration field. Afterward, one of the following actions can occur:

- If an end NOTIFY message for a DTMF event is received, the tone stops.

- If an update is received, the timer is updated according to the duration field.
- If an update or end NOTIFY message is not received before the timer expires, the tone stops and all subsequent NOTIFY messages for the same DTMF event or DTMF digit are ignored until an end NOTIFY message is received.
- If a NOTIFY message for a different DTMF event is received before an end NOTIFY message for the current DTMF event is received (which is an unlikely case), the current tone stops and the new tone plays. This is an unlikely case because for every DTMF event there needs to be an end NOTIFY message, and unless this is successfully sent and a 200 OK is received, the gateway cannot send other NOTIFY messages.

Note In-band tones are not passed while NOTIFY-based out-of-band DTMF relay is used as the DTMF relay method.

Two commands allow you to enable or disable NOTIFY-based out-of-band DTMF relay on a dial peer. The functionality is advertised to the other end using Invite messages if it is enabled by the commands, and must be configured on both the originating and terminating SIP gateways. A third command allows you to verify DTMF relay status.

- **dtmf-relay (VoIP)**
- **notify telephone-event**
- **show sip-ua status**

SIP KPML-Based Out-of-Band DTMF Relay

KPML support is required on SIP gateways for non-conferencing calls, and for interoperability between SIP products and SIP phones. If you configure KPML on the dial peer, the gateway sends INVITE messages with “kpml” in the Allow-Events header. Currently, all configured DTMF methods are recognized and sent in the outgoing INVITE. If you configure rtp-nte (RFC 2833), sip-notify, and sip-kpml, the outgoing INVITE contains a call-info header, an Allow-Events header with KPML, and an sdp with rtp-nte payload.

DTMF negotiation is performed based on the matching inbound dial-peer configuration. The gateway negotiates to either just cisco-rtp, just rtp-nte, rtp-nte + kpml, just kpml, or just sip-notify. If you configure more than one out-of-band DTMF method, preference goes from highest to lowest in the order they were configured. Whichever DTMF negotiation method you configure first takes precedence.

A gateway negotiates both rtp-nte and KPML if both are supported and advertised in the incoming INVITE. However, in this case, the gateway relies on the rtp-nte DTMF method to receive digits and a SUBSCRIBE for KPML is not initiated, however the gateway still accepts SUBSCRIBEs for KPML. This prevents double-digit reporting problems at the gateway.

The following example shows the INVITE and SUBSCRIBE sequence for KPML.

```
Sent:
INVITE sip:8888@172.18.193.250:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKC1ECC
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>
Date: Fri, 01 Mar 2002 00:15:59 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
Supported: 100rel,timer,resource-priority,replaces
Min-SE:  1800
Cisco-Guid: 1424162198-736104918-2148455531-3036263926
User-Agent: Cisco-SIPGateway/IOS-12.x
```

```
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO,
REGISTER
CSeq: 101 INVITE
Max-Forwards: 70
Timestamp: 1014941759
Contact: <sip:172.18.193.251:5060>
Expires: 180
Allow-Events: kpml, telephone-event
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 221

v=0
o=CiscoSystemsSIP-GW-UserAgent 1438 8538 IN IP4 172.18.193.251
s=SIP Call
c=IN IP4 172.18.193.251
t=0 0
m=audio 17576 RTP/AVP 0 19
c=IN IP4 172.18.193.251
a=rtpmap:0 PCMU/8000
a=rtpmap:19 CN/8000
a=ptime:20

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKC1ECC
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Date: Fri, 01 Mar 2002 01:02:34 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
Timestamp: 1014941759
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO,
REGISTER
Require: 100rel
RSeq: 3482
Allow-Events: kpml, telephone-event
Contact: <sip:8888@172.18.193.250:5060>
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 221

v=0
o=CiscoSystemsSIP-GW-UserAgent 9384 6237 IN IP4 172.18.193.250
s=SIP Call
c=IN IP4 172.18.193.250
t=0 0
m=audio 17468 RTP/AVP 0 19
c=IN IP4 172.18.193.250
a=rtpmap:0 PCMU/8000
a=rtpmap:19 CN/8000
a=ptime:20

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKC1ECC
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Date: Fri, 01 Mar 2002 01:02:38 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
Timestamp: 1014941759
```

```
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO,
REGISTER
Allow-Events: kpml, telephone-event
Contact: <sip:8888@172.18.193.250:5060>
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 221

v=0
o=CiscoSystemsSIP-GW-UserAgent 9384 6237 IN IP4 172.18.193.250
s=SIP Call
c=IN IP4 172.18.193.250
t=0 0
m=audio 17468 RTP/AVP 0 19
c=IN IP4 172.18.193.250
a=rtpmap:0 PCMU/8000
a=rtpmap:19 CN/8000

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
ACK sip:8888@172.18.193.250:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKEB8B
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Date: Fri, 01 Mar 2002 00:16:00 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
Max-Forwards: 70
CSeq: 101 ACK
Allow-Events: kpml, telephone-event
Content-Length: 0

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
SUBSCRIBE sip:8888@172.18.193.250:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKFF36
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 103 SUBSCRIBE
Max-Forwards: 70
Date: Fri, 01 Mar 2002 00:16:15 GMT
User-Agent: Cisco-SIPGateway/IOS-12.x
Event: kpml
Expires: 7200
Contact: <sip:172.18.193.251:5060>
Content-Type: application/kpml-request+xml
Content-Length: 327

<?xml version="1.0" encoding="UTF-8"?><kpml-request
xmlns="urn:ietf:params:xml:ns:kpml-request"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:kpml-request kpml-request.xsd"
version="1.0"><pattern persist="persist"><regex
tag="dtmf">[x*#ABCD]</regex></pattern></kpml-request>

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SUBSCRIBE sip:172.18.193.251:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.250:5060;branch=z9hG4bK5FE3
From: <sip:8888@172.18.193.250>;tag=39497C-2EA
To: <sip:172.18.193.251>;tag=EA330-F6
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
```

```
CSeq: 101 SUBSCRIBE
Max-Forwards: 70
Date: Fri, 01 Mar 2002 01:02:46 GMT
User-Agent: Cisco-SIPGateway/IOS-12.x
Event: kpml
Expires: 7200
Contact: <sip:172.18.193.250:5060>
Content-Type: application/kpml-request+xml
Content-Length: 327

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKFF36
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Date: Fri, 01 Mar 2002 01:02:51 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 103 SUBSCRIBE
Content-Length: 0
Contact: <sip:172.18.193.250:5060>
Expires: 7200

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.250:5060;branch=z9hG4bK5FE3
From: <sip:8888@172.18.193.250>;tag=39497C-2EA
To: <sip:172.18.193.251>;tag=EA330-F6
Date: Fri, 01 Mar 2002 00:16:24 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 101 SUBSCRIBE
Content-Length: 0
Contact: <sip:172.18.193.251:5060>
Expires: 7200

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
NOTIFY sip:172.18.193.250:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bK101EA4
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 104 NOTIFY
Max-Forwards: 70
Date: Fri, 01 Mar 2002 00:16:24 GMT
User-Agent: Cisco-SIPGateway/IOS-12.x
Event: kpml
Subscription-State: active
Contact: <sip:172.18.193.251:5060>
Content-Length: 0

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
NOTIFY sip:172.18.193.251:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.250:5060;branch=z9hG4bK6111
From: <sip:8888@172.18.193.250>;tag=39497C-2EA
To: <sip:172.18.193.251>;tag=EA330-F6
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 102 NOTIFY
Max-Forwards: 70
Date: Fri, 01 Mar 2002 01:02:51 GMT
User-Agent: Cisco-SIPGateway/IOS-12.x
Event: kpml
```

```
Subscription-State: active
Contact: <sip:172.18.193.250:5060>
Content-Length: 0

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.250:5060;branch=z9hG4bK6111
From: <sip:8888@172.18.193.250>;tag=39497C-2EA
To: <sip:172.18.193.251>;tag=EA330-F6
Date: Fri, 01 Mar 2002 00:16:32 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 102 NOTIFY
Content-Length: 0

//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
NOTIFY sip:172.18.193.250:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bK1117DE
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 105 NOTIFY
Max-Forwards: 70
Date: Fri, 01 Mar 2002 00:37:33 GMT
User-Agent: Cisco-SIPGateway/IOS-12.x
Event: kpml
Subscription-State: active
Contact: <sip:172.18.193.251:5060>
Content-Type: application/kpml-response+xml
Content-Length: 113

<?xml version="1.0" encoding="UTF-8"?><kpml-response version="1.0" code="200" text="OK"
digits="1" tag="dtmf"/>

/-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bK1117DE
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Date: Fri, 01 Mar 2002 01:24:08 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 105 NOTIFY
Content-Length: 0

...
```

SIP Support for Asymmetric SDP

The SIP Support for Asymmetric SDP feature allows a SIP gateway to receive an offer or answer with a different payload than it prefers, and treat the payloads as asymmetric, as long as both the payloads are not in use. The gateway interprets the payload that it receives and generates RTP packets for DTMF events and dynamic codecs. The gateway expects to receive the DTMF events and dynamic codec payload value it originally sent in the request/response message. In this way, the gateway advertises its preferred payload in the answer to the offer that it receives.

For delayed media cases, if the gateway receives an INVITE message with no SDP, the corresponding response or offer carries the preferred payload in the SDP. The gateway expects that the remote end will use the payload offered in RTP packets that it receives. The gateway uses the payload type received in the answer in all the RTP packets that it generates, as long as this payload is not in use.

The following sections describe the behavior of a Universal Agent Client (UAC) and Universal Agent Server (UAS) gateway for asymmetric payload handling:

- UAC Behavior for Asymmetric DTMF Payloads, page 15
- UAS Behavior for Asymmetric DTMF Payloads, page 15
- UAC Behavior for Asymmetric Dynamic Codec Payloads, page 16
- UAS Behavior for Asymmetric Dynamic Codec Payloads, page 17

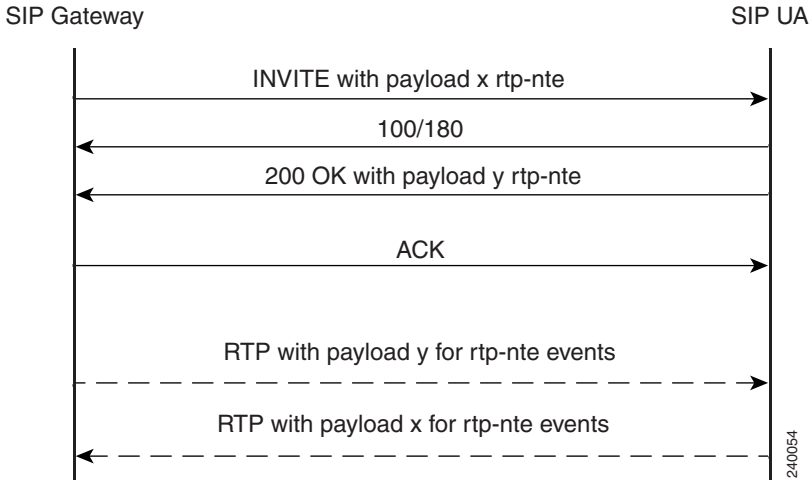
UAC Behavior for Asymmetric DTMF Payloads

When a UAC gateway originates an INVITE message, it advertises the preferred DTMF relay payload and method. However, when the gateway receives an 18x/2xx response with a different DTMF payload, the gateway first checks if the associated DTMF relay method can successfully negotiate with it. If negotiation is successful, the gateway checks if the new payload is available for use. If it is available, the gateway uses the received payload in all RTP (RFC 2833) events packets it generates. At the same time, the gateway expects the remote end to send packets with the preferred payload that the gateway advertised in its original request.

If the gateway is unable to use the new payload that the remote end requests in the answer, then DTMF relay negotiation fails for the requested DTMF relay method.

Figure 75 summarizes the call behavior for a Universal Access Controller (UAC) when using asymmetric DTMF payloads.

Figure 75 UAC Behavior for Asymmetric DTMF Payloads



UAS Behavior for Asymmetric DTMF Payloads

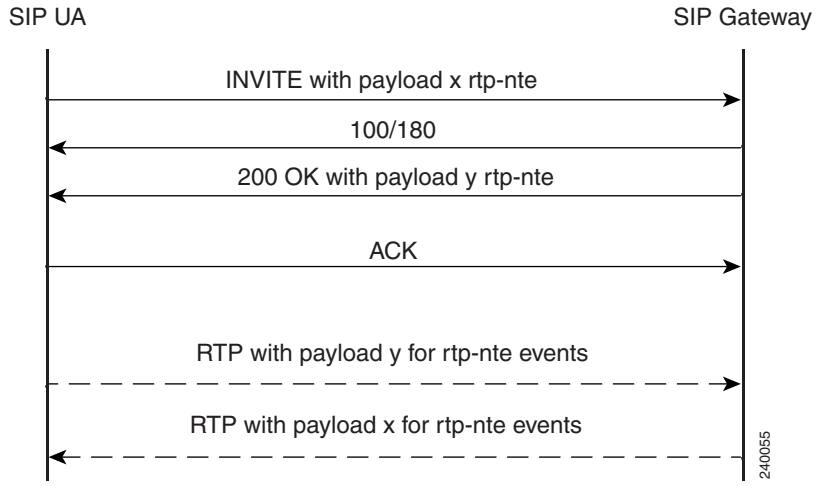
When a gateway receives an INVITE message that has a preferred DTMF relay method and payload, and the requested DTMF relay is negotiated successfully, the gateway checks if the requested DTMF payload is available for use. If it is available, the gateway checks if you configured a preferred payload at the matching dial peer.

If you did not configure a preferred payload, the gateway uses its default value for that particular DTMF relay method. If the preferred payload is different from the one in the original request, the payload in the original request is used in all RTP (RFC 2833) events packets that the gateway generates. The gateway responds with the preferred payload in the SDP in the response to the original request. The gateway expects the remote end to send packets with the preferred payload that the gateway advertised in its response.

When the gateway receives a delayed media request, the gateway offers the preferred DTMF relay and payload in its response. If the remote end responds with the same DTMF relay and different payload, this new payload is used to generate RTP RFC 2833 events packets. The gateway expects the remote end to send packets with the preferred payload that the gateway advertised in its response.

Figure 76 summarizes the call behavior for a Universal Access Server (UAS) when using asymmetric DTMF payloads.

Figure 76 UAS Behavior for Asymmetric DTMF Payloads



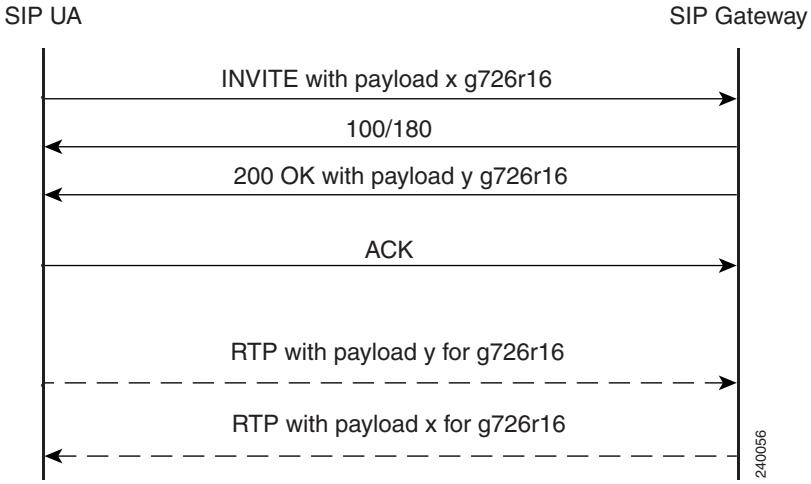
UAC Behavior for Asymmetric Dynamic Codec Payloads

When the gateway originates an INVITE message, the gateway advertises the preferred dynamic codec. However, when the gateway receive an 18x/2xx response with a different dynamic payload, the gateway first checks whether the new payload, in the answer SDP message, is being used for the same dynamic codec that was advertised. If the new payload is the same codec, and if the dynamic payload is available for use, the received payload is used in all RTP dynamic codec packets that are generated by the gateway. At the same time, the gateway expects the remote end to send packets with the preferred payload that the gateway advertised in its original request.

If the gateway is unable to use the new payload that the remote end requests in the answer, then codec negotiation for that particular dynamic codec fails.

Figure 77 summarizes the call behavior for a UAC when using asymmetric dynamic codec payloads.

Figure 77 UAC Behavior for Asymmetric Dynamic Codec Payloads



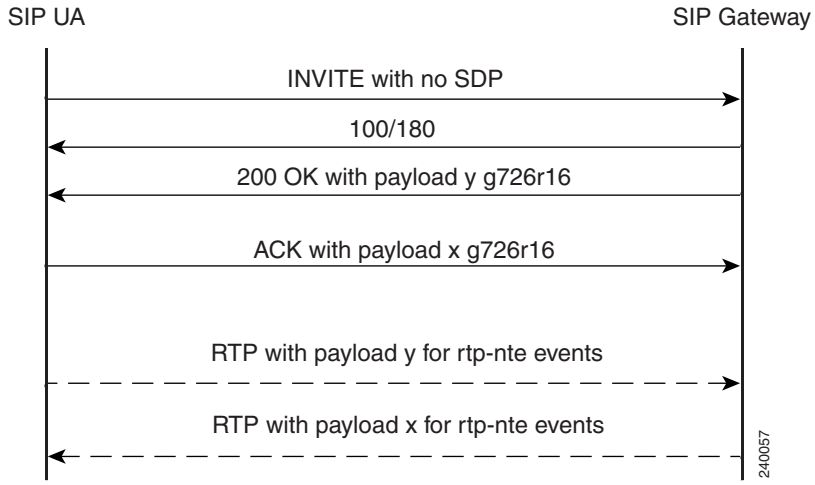
UAS Behavior for Asymmetric Dynamic Codec Payloads

When a gateway receives an INVITE message which has a dynamic codec and payload, and the requested dynamic codec and payload is negotiated successfully, the gateway uses the same payload in its response to the INVITE message. The gateway cannot generate a different dynamic payload in a response message.

Beginning with Cisco IOS Release 12.4(15)T, when a gateway receives a delayed media request, you can configure the gateway to offer the preferred dynamic codec and payload in its response. If the remote end responds with the same dynamic codec and a different payload, this new payload is used to generate the codec RTP packets. The gateway expects the remote end to send packets with the preferred payload that the gateway advertised in its response.

Figure 78 summarizes the call behavior for UAS when using asymmetric dynamic codec payloads.

Figure 78 UAS Behavior for Asymmetric Dynamic Codec Payloads



How to Configure SIP DTMF Features

This section contains the following procedures:

Note

- Before you perform a procedure, familiarize yourself with the following information:
  - "Restrictions for SIP DTMF" section on page 2
  - "Prerequisites for SIP DTMF" section on page 4
- For help with a procedure, see the verification and troubleshooting sections listed above.

Configuring Passthrough on a Gateway that Connects to an MTP or Transcoder Gateway

To configure RFC 2833 DTMF MTP Passthrough on a gateway that does not itself contain an MTP or transcoder but connects to another gateway that does, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **dtmf-relay rtp-nte**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`dial-peer voice`** *`tag`* **`voip`** **Example:** `Router(config)# dial-peer voice 103 voip`	Enters dial-peer configuration mode for the specified VoIP dial peer.
Step 4	**`dtmf-relay rtp-nte`** **Example:** `Router(config-dial-peer-voice)# dtmf-relay rtp-nte`	Specifies that the gateway should relay DTMF tones between telephony interfaces and an IP network by using RTP with the Named Telephone Event (NTE) payload type.
Step 5	**`exit`** **Example:** `Router(config-dial-peer-voice)# exit`	Exits the current configuration mode.

Note If you plan to employ both this feature and Cisco Fax Relay—both of which use the same default payload type (96 or 97)—in the same RTP session, you may need to configure the gateway that negotiates fax relay to use a different payload type.

To configure a gateway to use a different payload type, use the **rtp payload-type** command as in the following example:

```
Router(config-dial-peer)# rtp payload-type nte 105
```

Configuring DTMF Events Through SIP Signaling

To configure the DTMF Events Through SIP Signaling feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **timers notify** *number*
5. **retry notify** *number*

6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**timers notify** *number* **Example:** `Router(config-sip-ua)# timers notify 100`	Sets the amount of time that the user agent waits before retransmitting the Notify message. The argument is as follows: • *number*—Time, in ms, to wait before retransmitting. Range: 100 to 1000. Default: 500.
Step 5	**retry notify** *number* **Example:** `Router(config-sip-ua)# retry notify 6`	Sets the number of times that the Notify message is retransmitted to the user agent that initiated the transfer or Refer request. The argument is as follows: • *number*—Number of retries. Range: 1 to 10. Default: 10.
Step 6	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring DTMF Relay for SIP Calls Using NTEs

This section includes the following procedures:

- Configure DTMF Relay for SIP Calls Using NTEs, page 20
- Configuring Passthrough on a Gateway that Connects to an MTP or Transcoder Gateway, page 18

Configure DTMF Relay for SIP Calls Using NTEs

To configure the DTMF Relay for SIP Calls Using NTEs feature, perform the following steps.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **dial-peer voice** *number* **voip**
4. **session protocol sipv2**
5. **dtmf-relay rtp-nte**
6. **rtp payload-type nte** *number* **comfort-noise** [**13** | **19**]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *number* **voip** **Example:** `Router(config)# dial-peer voice 10 voip`	Enters dial-peer VoIP configuration mode for the specified dial peer.
Step 4	**session protocol sipv2** **Example:** `Router(config-dial-peer)# session protocol sipv2`	Specifies a session protocol for calls between local and remote routers using the packet network. The keyword is as follows: • **sipv2**—Dial peer uses the IETF SIP. Use this keyword with the SIP option.
Step 5	**dtmf-relay rtp-nte** **Example:** `Router(config-dial-peer)# dtmf-relay rtp-nte`	Specifies how an H.323 or SIP gateway relays DTMF tones between telephone interfaces and an IP network. The keyword is as follows: • **rtp-nte**—Forwards tones by using RTP with the NTE payload type.

	Command or Action	Purpose
Step 6	**rtp payload-type nte** *number* **comfort-noise** [**13** | **19**] **Example:** `Router(config-dial-peer)# rtp payload-type nte 100 comfort-noise 13`	Identifies the payload type of a RTP packet. Keywords and arguments are as follows: • **nte** *number*—Named telephone event (NTE). Range: 96 to 127. Default: 101. • **comfort-noise**—RTP payload type of comfort noise. If you are connected to a gateway that complies with the *RTP Payload for Comfort Noise July 2001* draft, use 13. If you are connected to an older Cisco gateway that uses DSPware before version 3.4.32, use 19.
Step 7	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring SIP INFO Method for DTMF Tone Generation

You cannot configure, enable, or disable this feature. You can display SIP statistics, including SIP INFO method statistics, by using the **show sip-ua statistics** and **show sip-ua status** commands in privileged EXEC mode. See the following fields for SIP INFO method statistics:

- OkInfo 0/0, under SIP Response Statistics, Success, displays the number of successful responses to an INFO request.
- Info 0/0, under SIP Total Traffic Statistics, displays the number of INFO messages received and sent by the gateway.

Note

- To see sample output of these **show** commands, see the "Configuring Passthrough on a Gateway that Connects to an MTP or Transcoder Gateway" section on page 18.
- To reset the counters for the **sip-ua statistics** command, use the **clear sip-ua statistics** command.

Configuring SIP NOTIFY-Based Out-of-Band DTMF Relay

To configure the SIP NOTIFY-Basec Out-of-Band DTMF Relay feature, perform the following steps.

Note Cisco proprietary NOTIFY-based out-of-band DTMF relay adds support for devices that do not support in-band DTMF. This configuration must be done on both originating and terminating gateways. With this configuration, DTMF tones are forwarded by using SIP NOTIFY messages in SIP Invites or 18x or 200 Response messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **dial-peer voice** *tag* **voip**
4. **dtmf-relay sip-notify**
5. **exit**
6. **sip-ua**
7. **notify telephone-event max-duration** *time*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enters dial-peer configuration mode for the designated dial peer.
Step 4	**dtmf-relay sip-notify** **Example:** `Router(config-dial-peer)# dtmf-relay sip-notify`	Forwards DTMF tones using SIP NOTIFY messages.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.
Step 6	**sip-ua** **Example:** `Router (config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 7	**notify telephone-event max-duration** *time* **Example:** `Router(config-sip-ua)# notify telephone-event max-duration 2000`	Sets the maximum time interval allowed between two consecutive NOTIFY messages for a single DTMF event. Keyword and argument are as follows: • **max-duration** *time*—Time, in ms, between consecutive NOTIFY messages for a single DTMF event. Range: 500 to 3000. Default: 2000.
Step 8	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP KPML-Based Out-of-Band DTMF Relay

To configure the SIP KPML Out-of-Band DTMF Relay feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **dtmf-relay sip-kpml**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enters dial-peer configuration mode for the designated dial peer.
Step 4	**dtmf-relay sip-kpml** **Example:** `Router(config-dial-peer)# dtmf-relay sip-kpml`	Forwards DTMF tones using SIP KPML messages.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Verifying SIP DTMF Support

To verify SIP DTMF support, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show running-config**
2. **show sip-ua retry**

3. **show sip-ua statistics**
4. **show sip-ua status**
5. **show sip-ua timers**
6. **show voip rtp connections**
7. **show sip-ua calls**

DETAILED STEPS

Step 1 **show running-config**

Use this command to show dial-peer configurations.

The following sample output shows that the **dtmf-relay sip-notify** command is configured in dial peer 123:

```
Router# show running-config

.
.
.
dial-peer voice 123 voip
 destination-pattern [12]...
 monitor probe icmp-ping
 session protocol sipv2
 session target ipv4:10.8.17.42
 dtmf-relay sip-notify
```

The following sample output shows that DTMF relay and NTE are configured on the dial peer.

```
Router# show running-config

!
dial-peer voice 1000 pots
 destination-pattern 4961234
 port 1/0/0
!
dial-peer voice 2000 voip
 application session
 destination-pattern 4965678
 session protocol sipv2
 session target ipv4:192.0.2.34
 dtmf-relay rtp-nte
! RTP payload type value = 101 (default)
!
dial-peer voice 3000 voip
 application session
 destination-pattern 2021010101
 session protocol sipv2
 session target ipv4:192.0.2.34
 dtmf-relay rtp-nte
 rtp payload-type nte 110
! RTP payload type value = 110 (user assigned)
!
```

Step 2 **show sip-ua retry**

Use this command to display SIP retry statistics.

```
Router# show sip-ua retry

SIP UA Retry Values
```

```
invite retry count = 6 response retry count = 1
bye retry count = 1 cancel retry count = 1
prack retry count = 10 comet retry count = 10
reliable 1xx count = 6 notify retry count = 10
```

Step 3 **show sip-ua statistics**

Use this command to display response, traffic, and retry SIP statistics.

Tip To reset counters for the **show sip-ua statistics** display, use the **clear sip-ua statistics** command.

```
Router# show sip-ua statistics

SIP Response Statistics (Inbound/Outbound)
Informational:
Trying 4/2, Ringing 2/1,
Forwarded 0/0, Queued 0/0,
SessionProgress 0/0
Success:
OkInvite 1/2, OkBye 0/1,
OkCancel 1/0, OkOptions 0/0,
OkPrack 2/0, OkPreconditionMet 0/0,
OkNotify 1/0, 202Accepted 0/1
Redirection (Inbound only):
MultipleChoice 0, MovedPermanently 0,
MovedTemporarily 0, SeeOther 0,
UseProxy 0, AlternateService 0
Client Error:
BadRequest 0/0, Unauthorized 0/0,
PaymentRequired 0/0, Forbidden 0/0,
NotFound 0/0, MethodNotAllowed 0/0,
NotAcceptable 0/0, ProxyAuthReqd 0/0,
ReqTimeout 0/0, Conflict 0/0, Gone 0/0,
LengthRequired 0/0, ReqEntityTooLarge 0/0,
ReqURITooLarge 0/0, UnsupportedMediaType 0/0,
BadExtension 0/0, TempNotAvailable 0/0,
CallLegNonExistent 0/0, LoopDetected 0/0,
TooManyHops 0/0, AddrIncomplete 0/0,
Ambiguous 0/0, BusyHere 0/0
RequestCancel 1/0, NotAcceptableMedia 0/0
Server Error:
InternalError 0/1, NotImplemented 0/0,
BadGateway 0/0, ServiceUnavail 0/0,
GatewayTimeout 0/0, BadSipVer 0/0,
PreCondFailure 0/0
Global Failure:
BusyEverywhere 0/0, Decline 0/0,
NotExistAnywhere 0/0, NotAcceptable 0/0
SIP Total Traffic Statistics (Inbound/Outbound) /* Traffic Statistics
Invite 3/2, Ack 3/2, Bye 1/0,
Cancel 0/1, Options 0/0,
Prack 0/2, Comet 0/0,
Notify 0/1, Refer 1/0
Retry Statistics          /* Retry Statistics
Invite 0, Bye 0, Cancel 0, Response 0,
Prack 0, Comet 0, Reliable1xx 0, Notify 0
```

Following is sample output verifying configuration of the SIP INFO Method for DTMF Tone Generation feature

```
Router# show sip-ua statistics
```

```
SIP Response Statistics (Inbound/Outbound)
Informational:
Trying 1/1, Ringing 0/0,
Forwarded 0/0, Queued 0/0,
SessionProgress 0/1
Success:
OkInvite 0/1, OkBye 1/0,
OkCancel 0/0, OkOptions 0/0,
OkPrack 0/0, OkPreconditionMet 0/0
OkSubscibe 0/0,OkNotify 0/0,
OkInfo 0/0, 202Accepted 0/0
Redirection (Inbound only):
MultipleChoice 0, MovedPermanently 0,
MovedTemporarily 0, SeeOther 0,
UseProxy 0, AlternateService 0
Client Error:
BadRequest 0/0, Unauthorized 0/0,
PaymentRequired 0/0, Forbidden 0/0,
NotFound 0/0, MethodNotAllowed 0/0,
NotAcceptable 0/0, ProxyAuthReqd 0/0,
ReqTimeout 0/0, Conflict 0/0, Gone 0/0,
LengthRequired 0/0, ReqEntityTooLarge 0/0,
ReqURITooLarge 0/0, UnsupportedMediaType 0/0,
BadExtension 0/0, TempNotAvailable 0/0,
CallLegNonExistent 0/0, LoopDetected 0/0,
TooManyHops 0/0, AddrIncomplete 0/0,
Ambiguous 0/0, BusyHere 0/0,
BadEvent 0/0
Server Error:
InternalError 0/0, NotImplemented 0/0,
BadGateway 0/0, ServiceUnavail 0/0,
GatewayTimeout 0/0, BadSipVer 0/0
Global Failure:
BusyEverywhere 0/0, Decline 0/0,
NotExistAnywhere 0/0, NotAcceptable 0/0
SIP Total Traffic Statistics (Inbound/Outbound)
    Invite 0/0, Ack 0/0, Bye 0/0,
    Cancel 0/0, Options 0/0,
    Prack 0/0, Comet 0/0,
    Subscribe 0/0, Notify 0/0,
    Refer 0/0, Info 0/0
Retry Statistics
Invite 0, Bye 0, Cancel 0, Response 0, Notify 0
```

Step 4 **show sip-ua status**

Use this command to display status for the SIP user agent.

```
Router# show sip-ua status

SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 2 (rfc 2782)
SDP application configuration:
 Version line (v=) required
 Owner line (o=) required
 Session name line (s=) required
 Timespec line (t=) required
 Media supported: audio image
```

```
 Network types supported: IN
 Address types supported: IP4
 Transport types supported: RTP/AVP udptl
```

The following sample output shows that the time interval between consecutive NOTIFY messages for a telephone event is the default of 2000 ms:

```
Router# show sip-ua status

SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP early-media for 180 responses with SDP: ENABLED
SIP max-forwards : 6
SIP DNS SRV version: 2 (rfc 2782)
NAT Settings for the SIP-UA
Role in SDP: NONE
Check media source packets: DISABLED
Maximum duration for a telephone-event in NOTIFYs: 2000 ms
SIP support for ISDN SUSPEND/RESUME: ENABLED
Redirection (3xx) message handling: ENABLED

 SDP application configuration:
 Version line (v=) required
 Owner line (o=) required
 Timespec line (t=) required
 Media supported: audio image
 Network types supported: IN
 Address types supported: IP4
 Transport types supported: RTP/AVP udptl
```

The following sample output shows configuration of the SIP INFO Method for DTMF Tone Generation feature.

```
Router# show sip-ua status

SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 2 (rfc 2782)
SDP application configuration:
 Version line (v=) required
 Owner line (o=) required
 Session name line (s=) required
 Timespec line (t=) required
 Media supported: audio image
 Network types supported: IN
 Address types supported: IP4
 Transport types supported: RTP/AVP udptl
```

Step 5 **show sip-ua timers**

Use this command to display the current settings for SIP user-agent timers.

```
Router# show sip-ua timers

SIP UA Timer Values (millisecs)
trying 500, expires 300000, connect 500, disconnect 500
comet 500, prack 500, rel1xx 500, notify 500
```

Step 6 **show voip rtp connections**

Use this command to show local and remote Calling ID and IP address and port information.

Step 7 **show sip-ua calls**

Use this command to ensure the DTMF method is SIP-KPML.

The following sample output shows that the DTMF method is SIP-KPML.

```
router# show sip-ua calls

SIP UAC CALL INFO

Call 1
SIP Call ID                : 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
   State of the call       : STATE_ACTIVE (7)
   Substate of the call    : SUBSTATE_NONE (0)
   Calling Number          :
   Called Number           : 8888
   Bit Flags               : 0xD44018 0x100 0x0
   CC Call ID              : 6
   Source IP Address (Sig ): 192.0.2.1
   Destn SIP Req Addr:Port : 192.0.2.2:5060
   Destn SIP Resp Addr:Port: 192.0.2.3:5060
   Destination Name        : 192.0.2.4.250
   Number of Media Streams : 1
   Number of Active Streams: 1
   RTP Fork Object         : 0x0
   Media Mode              : flow-through
   Media Stream 1
     State of the stream      : STREAM_ACTIVE
     Stream Call ID           : 6
     Stream Type              : voice-only (0)
     Negotiated Codec         : g711ulaw (160 bytes)
    Codec Payload Type       : 0
     Negotiated Dtmf-relay    : sip-kpml
     Dtmf-relay Payload Type  : 0
     Media Source IP Addr:Port: 192.0.2.5:17576
     Media Dest IP Addr:Port  : 192.0.2.6:17468
     Orig Media Dest IP Addr:Port : 0.0.0.0:0

   Number of SIP User Agent Client(UAC) calls: 1

SIP UAS CALL INFO

   Number of SIP User Agent Server(UAS) calls: 0
```

Configuring SIP Support for SDP

This section describes the procedures for configuring and associating the SIP Support for Asymmetric SDP feature. These procedures include the following:

- How to Configure a SIP Support for Asymmetric SDP Globally for a Gateway, page 30
- How to Configure SIP Support for Asymmetric SDP on a Dial Peer, page 31

How to Configure a SIP Support for Asymmetric SDP Globally for a Gateway

To configure an asymmetric payload globally on a SIP network, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service {pots | voatm | vofr | voip}**
4. **sip**
5. **asymmetric payload [dtmf | dynamic-codecs | full]**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service {pots | voatm | vofr | voip}** **Example:** `Router(config)# voice service voip`	Enters voice-service voip configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-ser)# sip`	Enters SIP parameters mode.
Step 5	**asymmetric payload [dtmf | dynamic-codecs | full]** **Example:** `Router(conf-ser-sip)# asymmetric payload dtmf`	Enables the gateway to send and receive DTMF and dynamic codec RTP packets with different payloads. In this example, the asymmetric payload is configured for DTMF.
Step 6	**exit** **Example:** `Router (conf-ser-sip)# exit`	Exits the current mode.

How to Configure SIP Support for Asymmetric SDP on a Dial Peer

To configure an asymmetric payload on a dial peer, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* {**pots** | **vofr** | **voip**}
4. **voice-class sip**
5. **asymmetric payload** [**dtmf** | **dynamic-codecs** | **full**]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* {**pots** | **vofr** | **voip**} **Example:** `Router(config)# dialpeer voice 111 voip`	Enters dial-peer voip configuration mode.
Step 4	**voice-class sip** **Example:** `Router(conf-dial-peer)# voice-class sip`	Enters SIP parameters mode.
Step 5	**asymmetric payload** [**dtmf** | **dynamic-codecs** | **full**] **Example:** `Router(conf-dial-peer)# asymmetric payload dynamic-codecs`	Enables the gateway to send and receive DTMF and dynamic codec RTP packets with different payloads.
Step 6	**exit** **Example:** `Router (conf-ser-sip)# exit`	Exits the current mode.

Troubleshooting Tips

Note For general troubleshooting tips and a list of important **debug** commands, see the "Basic Troubleshooting Procedures" section on page 1.

- To enable debugging for RTP named event packets, use the **debug voip rtp** command.
- To enable KPML debugs, use the **debug kpml** command.
- To enable SIP debugs, use the **debug ccsip** command.
- Collect debugs while the call is being established and during digit presses.
- If an established call is not sending digits though KPML, use the **show sip-ua calls** command to ensure SIP-KPML is included in the negotiation process.

Configuration Examples for SIP DTMF Features

This section provides the following configuration examples:

DTMF Relay for SIP Calls Using NTEs: Examples

DTMF Relay using RTP-NTE

The following is an example of DTMF relay using RTP-NTE:

```
Router(config)# dial-peer voice 62 voip
Router(config-dial-peer)# session protocol sipv2
Router(config-dial-peer)# dtmf-relay rtp-nte
```

RTP Using Payload Type NTE

The following is an example of RTP Using Payload Type NTE with the default value of 101:

```
Router(config)# dial-peer voice 62 voip
Router(config-dial-peer)# rtp payload-type nte 101
```

SIP NOTIFY-Based Out-of-Band DTMF Relay: Example

```
Current configuration : 3394 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
memory-size iomem 15
```

```
ip subnet-zero
!
no ip domain lookup
!
voice service voip
 redirect ip2ip
sip
 redirect contact order best-match

ip dhcp pool vespa
 network 192.0.2.0 255.255.255.0
 option 150 ip 192.0.2.2
 default-router 192.0.2.3
!
voice call carrier capacity active
!
voice class codec 1
 codec preference 2 g711ulaw
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
interface Ethernet0/0
 ip address 192.0.2.4 255.255.0.0
 half-duplex
!
interface FastEthernet0/0
 ip address 192.0.2.5 255.255.255.0
 speed auto
 no cdp enable
 h323-gateway voip interface
 h323-gateway voip id vespa2 ipaddr 192.0.2.6
!
router rip
 network 192.0.2.0
 network 209.165.201.0
!
ip default-gateway 192.0.2.9
ip classless
ip route 0.0.0.0 0.0.0.0 192.0.2.10
no ip http server
ip pim bidir-enable
!
tftp-server flash:SEPDEFAULT.cnf
tftp-server flash:P005B302.bin
call fallback active
!
call application global default.new
call rsvp-sync
!
voice-port 1/0
!
voice-port 1/1
!
mgcp profile default
!
dial-peer voice 1 pots
 destination-pattern 5100
 port 1/0
!
dial-peer voice 2 pots
```

```
 destination-pattern 9998
 port 1/1
!
dial-peer voice 123 voip
 destination-pattern [12]...
 session protocol sipv2
 session target ipv4:10.8.17.42
 dtmf-relay sip-notify
!
gateway
!
sip-ua
 retry invite 3
 retry register 3
 timers register 150
 registrar dns:myhost3.example.com expires 3600
 registrar ipv4:192.0.2.11 expires 3600 secondary
!
telephony-service
 max-dn 10
 max-conferences 4
!
ephone-dn 1
number 4001
!
ephone-dn 2
number 4002
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
login
line vty 5 15
 login
!
no scheduler allocate
end
```

SIP KPML-Based Out-of-Band DTMF Relay: Example

```
router(config-dial-peer)# dtmf
router(config-dial-peer)# dtmf-relay ?
  cisco-rtp          Cisco Proprietary RTP
  h245-alphanumeric  DTMF Relay via H245 Alphanumeric IE
  h245-signal        DTMF Relay via H245 Signal IE
  rtp-nte            RTP Named Telephone Event RFC 2833
  sip-kpml           DTMF Relay via KPML over SIP SUBCRIBE/NOTIFY
  sip-notify         DTMF Relay via SIP NOTIFY messages

router(config-dial-peer)# dtmf-relay sip-kpml
router(config-dial-peer)# end

%SYS-5-CONFIG_I: Configured from console by console
router#sh run

Building configuration...Current configuration : 2430 bytes
!
version 12.4
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
```

```
no service password-encryption
!
hostname mahoney
!
boot-start-marker
boot-end-marker
!
logging buffered 5000000 debugging
!
no aaa new-model
!
resource policy
!
clock timezone EST 0
ip cef
ip name-server 192.0.2.21
ip name-server 192.0.2.22
!
voice-card 0
!
voice service voip
 sip
  min-se  90
  registrar server
!
voice class codec 1
 codec preference 1 g711ulaw
 codec preference 2 g729r8
 codec preference 3 g729br8
 codec preference 4 g711alaw
 codec preference 5 g726r16
 codec preference 6 g726r24
 codec preference 7 g726r32
 codec preference 8 g723ar53
 codec preference 9 g723ar63
!
!
voice register pool  1
 id ip 192.0.2.168 mask 0.0.0.0
 dtmf-relay rtp-nte
!
!
interface FastEthernet0/0
 ip address 192.0.2.1 255.255.255.0
 no ip proxy-arp
no ip mroute-cache
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip default-gateway 192.0.2.200
ip route 0.0.0.0 0.0.0.0 192.0.2.1
ip route 0.0.0.0 0.0.0.0 192.0.2.225
!
ip http server
!
control-plane
!
voice-port 2/0
```

```
!
voice-port 2/1
!
voice-port 2/2
.
.
.
voice-port 2/22
!
voice-port 2/23
!
!
dial-peer voice 1 pots
 destination-pattern 8888
 port 2/1
!
dial-peer voice 9999 voip
 destination-pattern 9999
 session protocol sipv2
 session target ipv4:192.0.2.228
 dtmf-relay sip-kpml
 codec g711ulaw
!
dial-peer voice 5555555 voip
 destination-pattern 5555555
 session protocol sipv2
 session target ipv4:192.0.2.230
 codec g711ulaw
!
dial-peer voice 36 voip
 destination-pattern 36601
session protocol sipv2
 session target ipv4:192.0.2.235
 codec g711ulaw
!
dial-peer voice 444 voip
 destination-pattern 444
 session protocol sipv2
 session target ipv4:192.0.2.140
 codec g711ulaw
!
dial-peer voice 333 voip
 destination-pattern 333
 session protocol sipv2
 session target ipv4:192.0.2.200
!
!
sip-ua
 retry invite 3
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
!
end
```

RFC 2833 DTMF MTP Passthrough: Example

The following example shows a sample configuration of the RFC 2833 DTMF MTP Passthrough feature on a SIP gateway.

```
dial-peer voice 1000 voip
 destination-pattern .T
 session protocol sipv2
 session target ipv4:10.120.70.10
 incoming called-number .T
 dtmf-relay rtp-nte
!
sip-ua
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
!
!
end
```

SIP Support for Asymmetric SDP: Example

This section contains the following configuration examples:

- Configuring SIP Support for Asymmetric SDP Globally on a Gateway: Example, page 37
- Configuring SIP Support for Asymmetric SDP on a Dial Peer: Example, page 37

Configuring SIP Support for Asymmetric SDP Globally on a Gateway: Example

The following example shows how to configure asymmetric SDP globally on a gateway:

```
gateway> enable
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# voice service voip
gateway(conf-voi-serv)# sip
gateway(conf-serv-sip)# asymmetric payload dtmf
```

Configuring SIP Support for Asymmetric SDP on a Dial Peer: Example

The following examples shows how to configure asymmetric SDP on a dial peer:

```
gateway> enable
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# dial-peer voice 111 voip
gateway(conf-dial-peer)# voice-class sip
gateway(conf-dial-peer)# asymmetric payload dtmf
```

Additional References

The following sections provide references related to the SIP DTMF features.

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
SIP commands	*Cisco IOS Voice Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	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

RFCs

RFC	Title
RFC 2833	*RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html