

# Configure DTMF Relay on CUBE

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Conventions](#)

[Background Information](#)

[Supported DTMF-Relay Methods for CUBE](#)

[Supported In-band DTMF Relay Methods](#)

[Supported Out-Of-Band DTMF Relay Methods](#)

[Support for In-band Audio DTMF through G711](#)

[Supported DTMF-Relay Methods for H323](#)

[H.245 Alphanumeric](#)

[H.245 Signal](#)

[Named Telephony Events \(NTE\) - RFC2833](#)

[Cisco Proprietary RTP](#)

[Supported DTMF-Relay Methods for SIP](#)

[NTE - RFC2833](#)

[Unsolicited NOTIFY \(UN\)](#)

[Key Press Markup Language \(KPML\)](#)

[Information \(INFO\)](#)

[Configure DTMF-Relay on CUBE](#)

[Configure DTMF Relay for H323](#)

[Configure DTMF Relay for SIP](#)

[Configure DTMF Relay Digit-Drop](#)

[Validate and Troubleshoot DTMF Relay](#)

[Validate OOB DTMF Relay for H323](#)

[H.245 Alphanumeric Capability Advertisement](#)

[H.245 Alphanumeric Transmission Example](#)

[H.245 Signal Capability Advertisement](#)

[H.245 Signal Transmission Example](#)

[Confirm In-Band DTMF Relay for H323](#)

[RFC2833 Capability Support Advertisement](#)

[Validate OOB DTMF Relay for SIP](#)

[Unsolicited NOTIFY \(UN\) Advertisement Example](#)

[Unsolicited NOTIFY \(UN\) Transmission Example](#)

[Key Press Markup Language \(KPML\) Advertisement Example](#)

[KPML Transmission Example](#)

[DTMF Interworking](#)

[When Does CUBE Require Transcoding Resources for DTMF](#)

[DTMF Interworking Between Inband G711 to RFC2833](#)

[Other DTMF Interworking Options](#)

[When are MTP Resources Required by CUCM](#)

[MTP Devices Supported by CUCM](#)

[Software MTP \(CiscoIP Voice Media Streaming Application\)](#)

[Software MTP \(Based on CiscoIOS\)](#)

[Hardware MTP \(PVDM2, Cisco NM-HDV2 and NM-HD-1V/2V/2VE\)](#)

[Hardware MTP \(Cisco 2900 and 3900 Series Routers with PVDM3\)](#)

[When to Use Software or Hardware MTP](#)  
[CUCM Media Resource Group \(MRG\) and Media Resource Group List \(MRGL\) Considerations for MTP](#)  
[SCCP MTP Messages](#)  
[DTMF Relay between CUCM and CUBE](#)  
[CUCM SIP Trunk to CUBE](#)  
[CUCM H323 trunk to CUBE](#)  
[CUBE Dynamic/Asymmetric Payloads](#)  
[Symmetric Payloads Example](#)  
[DTMF Relay Negotiation](#)  
[DTMF Relay Transmission](#)  
[Asymmetric Payloads Example](#)  
[DTMF Relay Negotiation](#)  
[DTMF Relay Transmission](#)  
[Which DTMF Relay Method to Use](#)  
[Preferred DTMF Relay Methods for H.323](#)  
[Preferred DTMF Relay Methods for SIP](#)  
[Related Information](#)

## Introduction

This document describes the process to configure Dual-Tone Multi-Frequency (DTMF) relay for Cisco Unified Border Element (CUBE) Enterprise.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics.

- Basic knowledge of DTMF tones
- Basic knowledge of how to configure and use Cisco® IOS voice (such as dial-peers)
- Basic knowledge of how to configure and use CUBE
- Basic knowledge of the signaling used by the SIP and H323 protocols
- Basic knowledge of how to debug VoIP protocols like H323 and SIP

### Components Used

The information in this document is based on these software and hardware versions.

- Cisco Unified Border Element that runs on Cisco IOS.
- Cisco Unified Communications Manager 7.x or later.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

### Conventions

Refer to [Cisco Technical Tips Conventions](#) for information on document conventions.

## Background Information

This document also provides information and commands on how to configure, verify, and troubleshoot DTMF relay for the different VoIP gateway protocols supported by CUBE.

## Supported DTMF-Relay Methods for CUBE

CUBE supports a broad variety of DTMF relay methods for both In-band and Out-Of-Band (OOB) for the H.323 and Session Initiation Protocol (SIP) signaling protocols.

### Supported In-band DTMF Relay Methods

- In-band audio DTMF through G711
- RFC2833

### Supported Out-Of-Band DTMF Relay Methods

- H.245 Alphanumeric
- H.245 Signal
- SIP Unsolicited NOTIFY
- SIP KPML
- SIP INFO

## Support for In-band Audio DTMF through G711

Voice In-band audio or G711 DTMF refers to the transport of audible tones over the voice audio stream, without any additional involvement of the signaling protocol or the DSP for their transmission other than to setup the call normally and pass the audio end to end and use the G711Ulaw/Alaw codec. This means that the CUBE/Cisco IOS only passes the audio of the tones that come from one end to the other as if it is normal voice audio. The important measure to take for this method is to ensure that the calls are established and use the G711Ulaw/Alaw codec specifically because to use a codec that would compress the audio (any other codec than G711) distorts the DTMF tones and is likely render them unrecognizable to the receiving end. This is because the compression algorithm utilized by high compression codecs was designed to recognize and predict human voice and not DTMF tones.

In-band audio/G711 DTMF is supported with any VoIP signaling protocol and only requires the G711 codec to be enforced for the calls end-to-end. One must also keep in mind that the any transcoding treatment from a low-bit-rate (LBR) codec to G711 most likely distorts the tones as well.

---

**Note:** It is common for some confusion to arise when you discuss this DTMF relay method because the term In-band is used to refer to the transport of DTMF within the RTP stream called as Named Telephony Event (NTE/RFC2833) and when it is In-band audio tones. It is always important to clarify the actual method required/supported to apply the proper configuration and use the right approach to troubleshoot.

---

## Supported DTMF-Relay Methods for H323

### H.245 Alphanumeric

DTMF digits are separated from the voice stream and sent through the H.245 signaling channel OOB instead of sent through the RTP channel. The tones are transported in H.245 User Input Indication messages. The

H.245 signaling channel is a reliable channel and the packets that transport the DTMF tones are guaranteed to be delivered. All systems that are H.323 Version 2-compliant are required to support the dtmf-relay h245-alphanumeric command. However, support of the dtmf-relay h245-signal command is optional.

## **H.245 Signal**

OOB method which is similar to H.245 alphanumeric, allows passage of the tone duration information, thereby it addresses a potential problem with the alphanumeric method when interworking with other vendor's systems.

## **Named Telephony Events (NTE) - RFC2833**

This method transports DTMF tones in separate RTP packets according to section 3 of RFC 2833. RFC 2833 defines formats of NTE RTP packets used to transport DTMF digits, hook flash, and other telephony events between two peer endpoints. With the NTE method, the endpoints perform per-call negotiation of the DTMF relay parameters to determine the payload type value for the NTE RTP packets and supported NTE digit events. As a result, DTMF tones are communicated via RTP packets with a payload type value different from the values negotiated for other media packets; which provides a reliable method to transport the digits and avoid them not being recognized when they get compressed via the codec used to encode the voice, video or fax traffic.

RFC2833/NTE DTMF relay is considered an In-band method because the digits are transported within the RTP audio traffic itself without any involvement of the GW signaling protocol.

It is important to point out that the RFC2833/NTE method must not be confused with the voice In-band audio or G711 RTP stream since the later is just the audible tones that are passed as normal audio without any relay signaling method being aware or involved in the process. It means that they are just plain audio tones being passed end-to-end using the G711Ulaw/Alaw codec.

Some other facts about NTE with H323:

- H.323 supports RFC2833 as of V4
- Cisco IOS always advertises its 2833 support in TCS
- CUCM only supports NTE through an H.323 ICT.

## **Cisco Proprietary RTP**

With this method DTMF tones are sent in the same RTP channel as voice data. However, the DTMF tones are encoded differently from the voice samples, and are identified as payload type 121, which enables the receiver to identify them as DTMF tones. This method is not supported by CUCM and its use has been discontinued.

## **Supported DTMF-Relay Methods for SIP**

### **NTE - RFC2833**

In-band RFC2833 NTE payload types and attributes are negotiated between the two ends at call setup that use the Session Description Protocol (SDP) within the body section of the SIP message.

### **Unsolicited NOTIFY (UN)**

With this method, the digits are sent OOB as SIP NOTIFY messages within the payload of the message

body.

## Key Press Markup Language (KPML)

Based on [RFC4730](#), digits are transported OOB using XML within Subscribe/NOTIFY messages. It is mostly used for SIP endpoints registered to CUCM or CME, but also with ITSPs.

## Information (INFO)

Digits are relayed as OOB SIP INFO messages between the ends. This method does not require any configuration and is accepted and related by CUBE automatically.

---

**Note:** SIP INFO is not supported by Unified CM.

---

---

**Note:** When both the UN and NTE methods are negotiated, Cisco IOS always chooses UN over NTE to avoid double tones and In-band 2833 NTE packet is suppressed. Also, for CUCM, UN is used only when no other option is available. Likewise, if both KPML and UN are present, Cisco Call Manager (CCM) chooses KPML over UN.

---

## Configure DTMF-Relay on CUBE

By default, DTMF relay is disabled for both H323 and SIP dial-peers (except for SIP INFO); it is mandatory to configure the DTMF relay method to be used end-to-end on both the inbound and outbound dial-peers for each call leg.

## Configure DTMF Relay for H323

```
<#root>
```

```
Router(config)#dial-peer voice 1 voip
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
```

You can configure more than one method per dial-peer, depending on the requirements of the terminating ends.

```
<#root>
```

```
Router(config-dial-peer)#
```

```
dtmf-relay rtp-nte
```

```
?
  cisco-rtp          Cisco Proprietary RTP
```

digit-drop	Digits to be passed out-of-band and in-band digits dropped
h245-alphanumeric	DTMF Relay via H245 Alphanumeric IE
h245-signal	DTMF Relay via H245 Signal IE

## Configure DTMF Relay for SIP

<#root>

```
Router(config)#dial-peer voice 1 voip
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 SUBSCRIBE/NOTIFY
sip-NOTIFY	DTMF Relay via SIP NOTIFY messages

You can configure more than one method per dial-peer, depending on the requirements of the terminating ends.

<#root>

```
Router(config-dial-peer)#
```

**dtmf-relay rtp-nte**

?

cisco-rtp	Cisco Proprietary RTP
digit-drop	Digits to be passed out-of-band and in-band digits dropped
h245-alphanumeric	DTMF Relay via H245 Alphanumeric IE
h245-signal	DTMF Relay via H245 Signal IE
sip-kpml	DTMF Relay via KPML over SIP SUBSCRIBE/NOTIFY
sip-NOTIFY	DTMF Relay via SIP NOTIFY messages

---

**Note:** Add the **session protocol sip** command under the dial-peer for the SIP dtmf-relay options to become available.

---

## Configure DTMF Relay Digit-Drop

In order to avoid duplicate digits by relaying the same DTMF digits through in-band and out-of band methods to the outgoing leg for calls interworking from an in-band (RTP-NTE specifically) to an out-of band method, configure the **dtmf-relay rtp-nte digit-drop** command on the inbound dial-peer and the desired out-of-band method on the outgoing dial-peer. Otherwise, the same digit is sent in OOB as well as

in-band and gets interpreted as duplicate digits by the receiving end.

When the digit-drop option is configured in the inbound leg, CUBE suppresses NTE packets and only relay digits that use the OOB method configured on the outbound leg.

As shown in this image, the digit-drop option is available only when interworking between these DTMF relay methods.

	Inbound-leg	Outbound-leg
H323	rtp-nte (RFC2833)	h245-alphanumeric , h245-signal
SIP	rtp-nte (RFC2833)	sip-notify

For example, configure the **dtmf-relay rtp-nte digit-drop** command on the inbound dial-peer for a SIP leg sending digits through RFC2833, and then on the outbound H.323 side configure either dtmf-relay h245-alphanumeric or dtmf-relay h245-signal; this must result in CUBE suppressing the NTE packets and send out only the OOB H245 events instead.

For more information see [DTMF Relay Digit Drop](#).

## Validate and Troubleshoot DTMF Relay

### Validate OOB DTMF Relay for H323

#### H.245 Alphanumeric Capability Advertisement

In order to validate whether an endpoint is advertising the H.245 alphanumeric capability, look for this line inside the H.245 Terminal Capability Set (TCS) message using **debug h245 asn1**.

```
capability receiveUserInputCapability : basicString : NULL
```

#### H.245 Alphanumeric Transmission Example

Here is an example of an endpoint transmitting the digit 1 using the H245 alphanumeric method using **debug h245 asn1**.

```
<#root>
```

```
000510: Sep 28 19:02:02.716: H245 MSC OUTGOING PDU ::=  
  value MultimediaSystemControlMessage ::= indication : userInput :
```

```
alphanumeric
```

```
:
```

```
"1"
```

## H.245 Signal Capability Advertisement

In order to confirm whether an endpoint is advertising H.245 signal capability, look for this line inside the H.245 Terminal Capability Set (TCS) message that uses debug h245 asn1.

```
capability receiveUserInputCapability : dtmf : NULL
```

## H.245 Signal Transmission Example

This is an example of an endpoint transmitting the digit 1 with duration of 100 msec using the H245 signal method. There are two messages, the first message indicates the digit being dialed with a duration of 4s. However, the second signal (signalUpdate) updates the digit duration value to 100msec instead.

```
<#root>
```

```
000555: Sep 28 19:12:05.364: H245 MSC OUTGOING PDU ::=
      value MultimediaSystemControlMessage ::= indication : userInput :
```

```
signal
```

```
:
  {
    signalType "1"
    duration 4000
  }
```

```
000558: Sep 28 19:12:05.368: H245 MSC OUTGOING PDU ::=
      value MultimediaSystemControlMessage ::= indication : userInput :
```

```
signalUpdate
```

```
:
  {
```

```
duration 100
```

```
    rtp
    {
      logicalChannelNumber 2
    }
  }
```

## Confirm In-Band DTMF Relay for H323

Endpoints that have H.323 V5 can indicate that they support RFC2833 via a capability message within the TerminalCapabilitySet (TCS) message.

### RFC2833 Capability Support Advertisement

In order to confirm whether an endpoint is advertizing RFC2833 capability, look for this structure inside the H.245 TCS message that uses debug h245 asn1 (in the example payload-type 101 is being advertised for the



events from 0 to 16).

```
<#root>
capabilityTableEntryNumber 34
    capability receiveRTPAudioTelephonyEventCapability :
    {

dynamicRTPPayloadType 101
    audioTelephoneEvent
    "0-16"
    }
```

## Validate OOB DTMF Relay for SIP

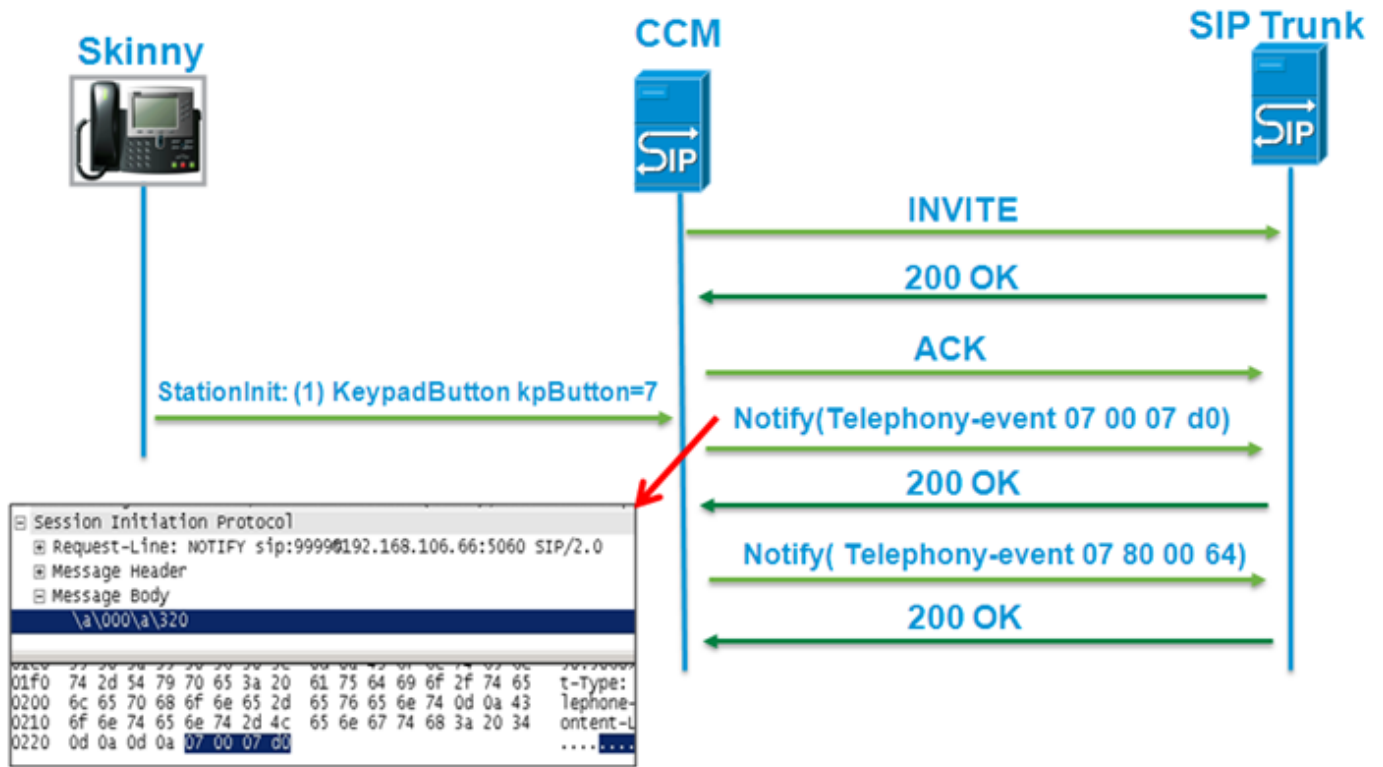
### Unsolicited NOTIFY (UN) Advertisement Example

In order to confirm whether an endpoint is advertising the Unsolicited NOTIFY (UN) capability, look for this line inside the INVITE message and/or response messages to the INVITE using debug ccsip messages.

```
<#root>
INVITE sip:9999@192.168.106.66:5060 SIP/2.0
    Call-Info: <sip:192.168.106.50:5060>;method="
NOTIFY ;Event=telephone-event;Duration=2000
“
```

### Unsolicited NOTIFY (UN) Transmission Example

The UN method transmits the digits as binary data inside the NOTIFY message; so you cannot see what digit is being transported by using debug ccsip messages. You can either need a packet capture (PCAP) or have to run **debug ccsip all** command to see the digit within the binary data outputs.



Example of how the same digit 7 dialed would look like when running `debug ccsip all` command.

<#root>

```
001738: Oct  9 15:37:24.577: //-1/xxxxxxxxxxxx/SIP/Msg/sipDisplayBinaryData&colon;
Sending: Binary Message Body
```

```
001739: Oct  9 15:37:24.577: Content-Type: audio/telephone-event
```

```
07
```

```
00 07 D0
```

```
001756: Oct  9 15:37:24.577: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
```

```
NOTIFY sip:9999@192.168.106.66:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.106.50:5060;branch=z9hG4bK10E8E5C
From: <sip:2010@192.168.105.189>;tag=557BFE8-9EE
To: <sip:9999@192.168.106.66>;tag=cuecebad539
Call-ID: 87C4CAE-115E11E2-8184AAE4-EF882E8F@192.168.253.1
CSeq: 106 NOTIFY
Event: telephone-event
Subscription-State: active
Contact: <sip:192.168.106.50:5060>
Content-Type: audio/telephone-event
Content-Length: 4
```

```
001763: Oct  9 15:37:24.593: //0/000000000000/SIP/Msg/ccsipDisplayMsg:
```

```
Received:
SIP/2.0 200 Ok
Via: SIP/2.0/UDP 192.168.106.50:5060;branch=z9hG4bK10E8E5C
To: <sip:9999@192.168.106.66>;tag=cuecebad539
From: <sip:2010@192.168.105.189>;tag=557BFE8-9EE
Call-ID: 87C4CAE-115E11E2-8184AAE4-EF882E8F@192.168.253.1
CSeq: 106 NOTIFY
Content-Length: 0
```

Allow-Events: refer  
Allow-Events: telephone-event  
Allow-Events: message-summary

## Key Press Markup Language (KPML) Advertisement Example

The KPML capability is listed within the Allow-Events SIP header. For KPML digit transmissions, the transmitting endpoint needs to first send a subscription to the KPML service; SUBSCRIBE message requesting the capability is transmitted; followed by a NOTIFY message from the receiving end marking the subscription-state for the KPML events as active.

Initial INVITE advertising the capability.

<#root>

```
INVITE sip:95554445001@192.168.105.25:5060 SIP/2.0
Allow-Events:
kpml, telephone-event
```

The terminating end requests subscription to the KMPL events.

<#root>

SUBSCRIBE

```
    sip:2010@192.168.106.50:5060 SIP/2.0
Event:
kpml
```

Content-Type: application/

kpml-request+xml

The originating end responds with a NOTIFY setting the state to active.

<#root>

NOTIFY

```
    sip:192.168.105.25:5060 SIP/2.0
Event: kpml
Subscription-State:
active
```

## KPML Transmission Example

After the subscription has taken place, the endpoints can transmit the digits using NOTIFY messages with KPML events through XML. Example of digit 1 being transmitted.

```
<#root>
NOTIFY
  sip:192.168.105.25:5060 SIP/2.0
Event:
  kpml

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

## DTMF Interworking

CUBE supports around 30 different types of DTMF interworking. It is able to interwork and transcode between different relay methods based on the **dtmf-relay** command configured within the matched inbound and outbound dial-peers for the call.

Refer to the [DTMF Interoperability Table](#) section of the [CUBE Configuration Guide](#) for details on DTMF Interworking Support.

## When Does CUBE Require Transcoding Resources for DTMF

CUBE requires transcoding resources registered locally in these scenarios

- Interworking between RFC2833 and Voice In-band
- Interworking between an OOB method and RFC2833 for flow-around calls

CUBE is able to interwork between all other DTMF relay methods with flow-through calls without the need of a transcoder.

## DTMF Interworking Between Inband G711 to RFC2833

CUBE is able to interwork between Inband G711 DTMF (raw audio tones) to RFC2833. However, these requirements need to be met

- The codec used must be G711 end-to-end. This is a restriction because if a LBR codec was to be used, then the tones would get distorted due to the compression loss.
- Transcoding resources must be available and registered with the CUBE accordingly. This because the CUBE needs to allocate a transcoding resource (more specifically: a DSP resource) to the media RTP stream to inject or listen for tones within the audio stream.
- The dial-peer for the inband-tones leg must not have any DTMF relay command configured. This

requirement is no longer needed for IOS XE 16.12.x or later. CUBE can dynamically allocate a Transcoder even if the inbound/ITSP dial-peer has dtmf-relay rtp-nte configured. The decision of allocating a Transcoder can depend on the SDP negotiation between the peer devices.

- The dial-peer for the RFC2833 leg must have dtmf-relay rtp-nte configured.
- Do not enable digit-drop on any of the dial-peers involved with the call.

## Other DTMF Interworking Options

There is also an additional set of interworking commands that could be required on specific call scenarios; which can be configured globally or at dial-peer level.

```
<#root>
```

```
dtmf-interworking {
```

```
  rtp-nte
```

```
  |
```

```
  standard
```

```
  |
```

```
  system
```

```
}
```

```
rtp-nte
```

Enables a delay between the dtmf-digit begin and dtmf-digit end events of RTP NTE packets.

**Standard**

Generates RTP NTE packets that are RFC 4733 compliant.

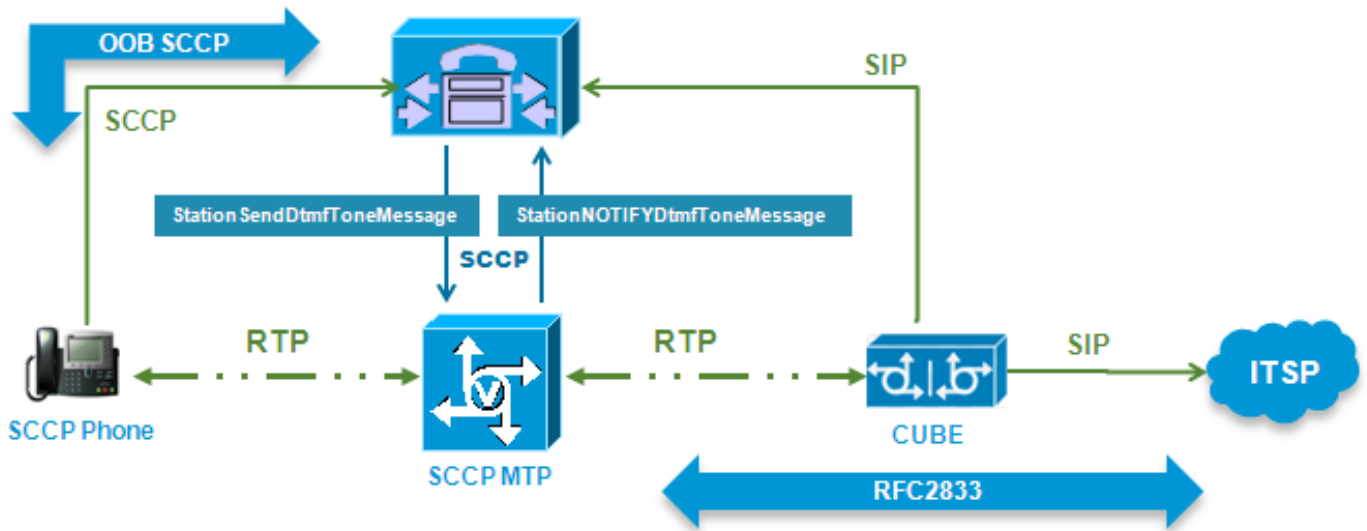
**System**

Specifies the default global DTMF interworking configuration. This keyword is available only in dial peer

## When are MTP Resources Required by CUCM

MTP resource becomes necessary when CUCM needs to interwork different DTMF methods between two devices, one of them using the RFC2833 method specifically and the other an OOB method. In this scenario, the CUCM needs to allocate the necessary resources to transmit and/or detect the in-band tones due to the DTMF relay mismatch between the two ends.

The role of the MTP is to monitor the RTP traffic and detect NTE events from the RFC2833 leg or to inject the NTE events into the RTP stream if requested by the CUCM. If the MTP detects inbound NTE events from the endpoint that only support RFC2833 then it sends a **SCCP StationNOTIFYDtmfToneMessage** to the CUCM and informs it of the tone that was detected in the stream. The CUCM in turn sends the same digit and uses the signaling protocol (OOB) to the other end. If the CUCM receives an OOB DTMF signal from the OOB DTMF endpoint then it sends a **SCCP StationSendDtmfToneMessage** to the MTP so that the MTP can inject the requested tone into the RTP stream in the form of NTE events.



## MTP Devices Supported by CUCM

### Software MTP (Cisco IP Voice Media Streaming Application)

Software MTP is a device that is implemented by enabling the Cisco IP Voice Media Streaming Application on a CUCM server. When the installed application is configured as an MTP application, it registers with a CUCM node and informs CUCM of how many MTP resources it supports. A software MTP device supports only G.711 streams. CUCM's default settings allow it to handle up to 48 calls as per per software MTP. For details on how to modify the service parameters, refer to the appropriate version of the [Cisco Unified Communications Manager Administration Guide](#).

### Software MTP (Based on Cisco IOS)

This MTP allows configuration of any of these codecs, however only one can be configured at a given time G.711 mu-law and a-law, G.729a, G.729, G.729ab, G.729b, and passthrough. Some of these are not pertinent to a CUCM implementation.

Router configurations permit up to 1,000 individual streams, which support 500 transcoded sessions which generates 10 Mbytes of traffic. The Cisco ISR G2s and ASR routers can support significantly higher numbers than this.

This MTP consumes CPU cycles to operate. Make a note of the number of sessions enabled as it could impact the CPU's performance and trigger high CPU utilization.

### Hardware MTP (PVDM2, Cisco NM-HDV2 and NM-HD-1V/2V/2VE)

This hardware uses the PVDM-2 modules for providing DSPs.

### Hardware MTP (Cisco 2900 and 3900 Series Routers with PVDM3)

These routers use the PVDM3 DSPs natively on the motherboards or PVDM2 with an adaptor on the motherboard or on service modules.

---

**Note:** You cannot configure G.729 or G.729b when configuring hardware MTP resources in Cisco IOS. However, Unified CM can use hardware transcoding resources as MTPs if all other MTP

---

---

resources are exhausted or otherwise unavailable.

---

## When to Use Software or Hardware MTP

The type of MTP to deploy in your network depends on specific codec parameters supported by the endpoints, gateways and trunks in the call flow

- The Codec flavors to be used
- The Codec packet size to be used (packetization)
- T.38 faxing usage (requires Codec Pass-Through support)

Based on these parameters you can safely choose and deploy the correct resources required by your network.

As shown in the table, the different features supported by different MTP and transcoder types

Type	Same Codecs	Different Codecs	Different Packetization	Codec Pass-through	Notes
<b>CUCM SW MTP</b>	Yes	No	Yes	No	G711 Alaw-Ulaw transcoding and repacketization
<b>Cisco IOS HW MTP</b>	Yes	No	No	Yes	Support for any codec (and same flavor) as long as the same packetization. No transcoding.
<b>Cisco IOS SW MTP</b>	Yes	No	No	Yes	Support any codec (and same flavor) as long as the same packetization. No transcoding.
<b>Cisco IOS Regular Xcoder</b>	Yes	Yes	Yes	Yes	As long as at least one side is G711u/G711a, it supports any repacketization and transcoding.
<b>Cisco IOS Universal Xcoder</b>	Yes	Yes	Yes	Yes	Support in any codec, packetization and transcoding.

For more information about MTP configuration in CUCM please refer to the [Media Termination Point Configuration Example](#) .

## CUCM Media Resource Group (MRG) and Media Resource Group List (MRGL) Considerations for MTP

When creating and assigning media resources to media resource groups (MRG) and media resource group lists (MRGL), take some additional points to consider to avoid over-subscription of the best resources for specific call flows and prioritize them accordingly. CUCM is unable to pick the best device to use, when it selects a media resource for a call, from a given list of MTPs and transcoders if they have the same priority or order. Instead, it chooses the first device that supports the requested capabilities. So even if the call is using G711 on both legs, if the first device it finds is a transcoder then it allocates it as a MTP for the call and not look for a MTP resource further down the list.

Another similar behavior occurs when you have both universal and regular transcoders. The CUCM could use the regular transcoders first on a call where one of the legs was G711, and then fail when a call gets transferred to a destination that uses a non-G711 codec, because the CUCM is not going to release the current transcoder and get another one when the call is transferred.

The best design practice to get around this behavior is to assign all MTP-only devices in a single MRG, then the universal transcoders to another MRG and the regular transcoders to a third MRG; and then prioritize them in that same order within the MRGL. Now, this design cannot work for every topology and must be reviewed on a case-by-base basis.

## **SCCP MTP Messages**

These SCCP messages are exchanged between the CUCM and MTP resources for DTMF handling.

- StationCapabilitiesRes
- StationUpdateCapabilities
- StationSubscribeDtmfPayloadReq
- StationSubscribeDTMFPayloadErrv
- StationSubscribeDtmfPayloadRes
- StationUnsubscribeDtmfPayloadErr
- StationNOTIFYDtmfToneMessage
- StationSendDtmfToneMessage
- StationUnsubscribeDtmfPayloadReq
- StationUnsubscribeDtmfPayloadRes

## **DTMF Relay between CUCM and CUBE**

### **CUCM SIP Trunk to CUBE**

CUBE supports KPML, NTE, or Unsolicited Notify as the DTMF mechanism, depending on its configuration. Because there can be a mix of endpoints in the system, multiple methods can be configured on the CUBE simultaneously in order to minimize MTP requirements.

On CUBE, configure both sip-kpml and rtp-nte as DTMF relay methods under SIP dial peers. This configuration enables DTMF exchange with all types of endpoints, including those that support only NTE and those that support only OOB methods, without the need for MTP resources. With this configuration, the gateway negotiates both NTE and KPML with CUCM. If NTE is not supported by the Unified CM endpoint, then KPML is used for DTMF exchange. If both methods are negotiated successfully, then the gateway relies on NTE to receive digits and does not subscribe to KPML.

CUBE also has the ability to use Unsolicited Notify (UN) method for DTMF. The UN method sends a SIP Notify message with a body that contains text describing the DTMF tone. This method is also supported on Unified CM and can be used if sip-kpml is not available. Configure sip-notify as the DTMF relay method. Note that this method is Cisco proprietary.



CUBEs configured for only NTE relay, or that due to some interworking limitation, can only provide NTE and required MTP resources to be allocated on the CUCM side when communicating with endpoints that do not support NTE.

You can find More information on CUCM [SIP Trunk MTP requirements](#)

## CUCM H323 trunk to CUBE

CUCM dynamically chooses the DTMF transport method for H323 trunks; so there are no configurable options to choose one over the other. If you want to force a specific DTMF relay method, then you can do so from the CUBE dial-peer configuration for this trunk.

Even when H323 CUBEs support NTE, the NTE option must not be used because it is not supported on CUCM for H.323 gateways/trunks at this time; so CUCM does not advertise this capability at the moment H245 media capabilities are exchanged. The CUCM's preferred option is H.245 Signal.

MTP resources are required to establish calls to an H.323 CUBE if the other endpoint does not have signaling capability in common with CUCM. For example, a Cisco Unified IP Phone 7960 that runs the SIP stack supports only NTEs, so an MTP is needed with an H.323 trunk so H245 Alphanumeric can be used on the H323 leg.

## CUBE Dynamic/Asymmetric Payloads

As of Cisco IOS version 15.1(1)T (CUBE 1.4) support for Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP to SIP Calls was introduced.

This feature allows the CUBE to handle the interworking of: dynamic payload types for audio/video codecs, NSE and DTMF; which up to this point was limited because the Cisco IOS would reserve a static range and only allow the same payload types to be negotiated on both call-legs and reject the call with a 488 error response for mismatching audio/video /NSE codecs (or fallback to voice inband G711 DTMF) for mismatching NTE payloads. Therefore, the feature allows the CUBE to un-reserve or free payload types dynamically for the interworking with SIP providers or third-party devices that use a different range of payload types to another leg that would not support them or that requires a different mapping specifically.

A call leg on CUBE is considered to be symmetric or asymmetric based on the payload type value exchanged through SDP during the offer and answer with the endpoint.

- A symmetric endpoint accepts and sends the same payload type for NTE events or a specific codec for a call leg.
- An asymmetric endpoint can accept and send different payload types for NTE events or a specific codec for a call leg.

This command is available to specify the usage of asymmetric payloads; the command can be applied globally under the **voice service voip** enter **sip** config mode or at dial-peer level using the **voice-class sip** CLI

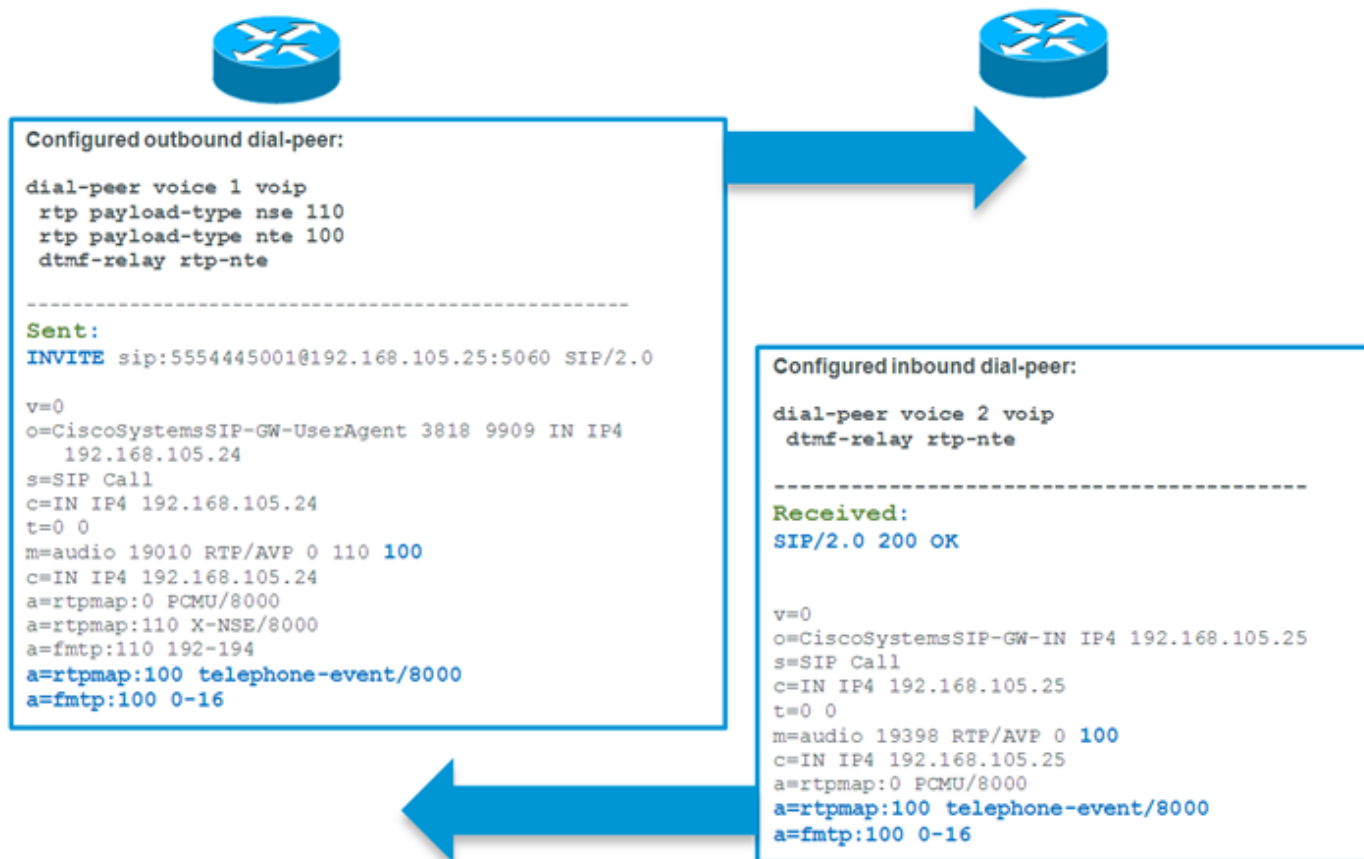
```
asymmetric payload {dtmf | dynamic-codecs | full | system}
```

For more information about Dynamic/Asymmetric payloads please navigate to [Dynamic payload type interworking for DTMF and codec packets for SIP to SIP calls](#)

## Symmetric Payloads Example

Here is an example of how the SDP would look like for a symmetric payload negotiation and the output from the **debug voip rtp session named event** while DTMF tones are being transmitted. Please note that the configuration used to force the Cisco IOS can use a different payload type for NTE events that use the **rtp payload-type nte** command.

## DTMF Relay Negotiation



## DTMF Relay Transmission



```
s=DSP d=VoIP payload 0x64 ssrc 0x6AFC8F9C sequence 0x3F9F timestamp 0x1FEC6DD4
Pt:100 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x6AFC8F9C sequence 0x3FA0 timestamp 0x1FEC6DD4
Pt:100 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x6AFC8F9C sequence 0x3FA1 timestamp 0x1FEC6DD4
Pt:100 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x6AFC8F9C sequence 0x3FA2 timestamp 0x1FEC6DD4
Pt:100 Evt:1 Pkt:04 01 90 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x6AFC8F9C sequence 0x3FA3 timestamp 0x1FEC6DD4
Pt:100 Evt:1 Pkt:84 03 20 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x6AFC8F9C sequence 0x3FA4 timestamp 0x1FEC6DD4
Pt:100 Evt:1 Pkt:84 03 20 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x6AFC8F9C sequence 0x3FA5 timestamp 0x1FEC6DD4
Pt:100 Evt:1 Pkt:84 03 20 <Snd>>>
```



```
s=DSP d=VoIP payload 0x64 ssrc 0x7F26919 sequence 0x449F timestamp 0x9C3C18BD
Pt:100 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x7F26919 sequence 0x44A0 timestamp 0x9C3C18BD
Pt:100 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x7F26919 sequence 0x44A1 timestamp 0x9C3C18BD
Pt:100 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x7F26919 sequence 0x44A2 timestamp 0x9C3C18BD
Pt:100 Evt:1 Pkt:04 01 90 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x7F26919 sequence 0x44A3 timestamp 0x9C3C18BD
Pt:100 Evt:1 Pkt:84 03 20 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x7F26919 sequence 0x44A4 timestamp 0x9C3C18BD
Pt:100 Evt:1 Pkt:84 03 20 <Snd>>>
s=DSP d=VoIP payload 0x64 ssrc 0x7F26919 sequence 0x44A5 timestamp 0x9C3C18BD
Pt:100 Evt:1 Pkt:84 03 20 <Snd>>>
```



## Asymmetric Payloads Example

Here is an example of how the SDP would look like for an asymmetric payload negotiation and the output from the **debug voip rtp session named event** command while DTMF tones are being transmitted. Please note the configuration used to force the Cisco IOS to use a different payload type for NTE events and uses the **rtp payload-type nte** commands and the **voice-class sip asymmetric payload dtmf** CLI.

## DTMF Relay Negotiation



**Configured outbound dial-peer:**

```
dial-peer voice 1 voip
rtp payload-type nse 110
rtp payload-type nte 100
dtmf-relay rtp-nte
```

**Sent:**

```
INVITE sip:5554445001@192.168.105.25:5060 SIP/2.0
```

```
v=0
o=CiscoSystemsSIP-GW-IN IP4 192.168.105.24
s=SIP Call
c=IN IP4 192.168.105.24
t=0 0
m=audio 19162 RTP/AVP 0 110 100
c=IN IP4 192.168.105.24
a=rtpmap:0 PCMU/8000
a=rtpmap:110 X-NSE/8000
a=fmtp:110 192-194
a=rtpmap:100 telephone-event/8000
a=fmtp:100 0-16
```

**Configured inbound dial-peer:**

```
dial-peer voice 2 voip
no modem passthrough
rtp payload-type nte 107
dtmf-relay rtp-nte
voice-class sip asymmetric payload dtmf
```

**Received:**

```
SIP/2.0 200 OK
```

```
v=0
o=CiscoSystemsSIP-GW-IN IP4 192.168.105.25
s=SIP Call
c=IN IP4 192.168.105.25
t=0 0
m=audio 19452 RTP/AVP 0 107
c=IN IP4 192.168.105.25
a=rtpmap:0 PCMU/8000
a=rtpmap:107 telephone-event/8000
a=fmtp:107 0-16
```

## DTMF Relay Transmission



```
s=DSP d=VoIP payload 0x6B ssrc 0x6AFC9FAB sequence 0x9F46 timestamp 0xE4B93524
Pt:107 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x6AFC9FAB sequence 0x9F47 timestamp 0xE4B93524
Pt:107 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x6AFC9FAB sequence 0x9F48 timestamp 0xE4B93524
Pt:107 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x6AFC9FAB sequence 0x9F49 timestamp 0xE4B93524
Pt:107 Evt:1 Pkt:04 01 90 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x6AFC9FAB sequence 0x9F4A timestamp 0xE4B93524
Pt:107 Evt:1 Pkt:84 03 20 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x6AFC9FAB sequence 0x9F4B timestamp 0xE4B93524
Pt:107 Evt:1 Pkt:84 03 20 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x6AFC9FAB sequence 0x9F4C timestamp 0xE4B93524
Pt:107 Evt:1 Pkt:84 03 20 <Snd>>>
```

```
s=DSP d=VoIP payload 0x6B ssrc 0x1906919 sequence 0x9F46 timestamp 0x9C36BAFD
Pt:100 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x1906919 sequence 0x9F47 timestamp 0x9C36BAFD
Pt:100 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x1906919 sequence 0x9F48 timestamp 0x9C36BAFD
Pt:100 Evt:1 Pkt:04 00 00 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x1906919 sequence 0x9F49 timestamp 0x9C36BAFD
Pt:100 Evt:1 Pkt:04 01 90 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x1906919 sequence 0x9F4A timestamp 0x9C36BAFD
Pt:100 Evt:1 Pkt:84 03 20 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x1906919 sequence 0x9F4B timestamp 0x9C36BAFD
Pt:100 Evt:1 Pkt:84 03 20 <Snd>>>
s=DSP d=VoIP payload 0x6B ssrc 0x1906919 sequence 0x9F4C timestamp 0x9C36BAFD
Pt:100 Evt:1 Pkt:84 03 20 <Snd>>>
```

# Which DTMF Relay Method to Use

When you choose the DTMF-relay to use, you need to take into consideration these variables.

- Devices and Platforms involved.
- VoIP protocols involved.
- Media path and supported Codecs.
- Supported or preferred DTMF relay methods.

## Preferred DTMF Relay Methods for H.323

The preferred method for H323 would be to use OOB through H.245 alphanumeric or signal in almost all scenarios. You can also use RFC2833 as long as CUCM is not involved.

## Preferred DTMF Relay Methods for SIP

- **SIP trunks to service providers** - whenever there is a SIP trunk to a SIP provider involved or interaction with 3<sup>rd</sup> party SIP devices or IVR systems then in-band through RFC2833 is preferred.
- **SIP trunk to CUCM or CME** - enable both RFC2833 and KPML.
- **SIP trunk to CUE** - the default method for CUE is UN but you can also configure it to use NTE; which is also the best option if the call comes from a SIP provider to the CUE system.

## Related Information

[Universal Voice Transcoding Support for IP-to-IP Gateways](#)

[DTMF Conversion](#)

[Unified Border Element Transcoding Configuration Example](#)

[Using Cisco Unified Communications Manager to Configure Transcoding and Media Termination Point](#)

[Configuring DTMF Relay Digit-Drop on a Cisco Unified Border Element](#)

[SIP Trunk MTP Requirements](#)

[SIP INFO Method for DTMF Tone Generation](#)

[H.323 Trunks with Media Termination Points](#)

[CUBE 9.0 Local Transcoding Interface \(LTI\)](#)