DTMF and RTP Relay

In this section:
- SIP DTMF Trigger Detection
- DTMF Interworking
- DTMF Interworking Without DSP
- SIP RTP Relay
- Audio Stream Examples Using DTMF
- KPML DTMF Support
- Call Flows

SIP DTMF Trigger Detection

The SIP Dual-Tone Multi-Frequency (DTMF) trigger detection and notification functionality enables the SBC Core to look for specific DTMF trigger patterns across the packet network, and to notify an external SIP entity when such patterns are detected. If a mid-call trigger is configured for a call, it is activated as soon as the call is connected. When the SBC detects DTMF input matching the pattern criteria of the trigger, it sends the DTMF input in a SIP INFO message to the SIP call peer with content type application/DTMF.

The SBC Core supports the following methods of relaying DTMF digits for transcoded calls:
- In-band—Leaves the DTMF tones in-band as encoded audio.
- Out-of-band—Carries DTMF in the signaling protocol (SIP or H.323).
- RFC 2833—Encodes DTMF into RTP using a format and Payload Type (PT) distinct from the audio encoding. The SBC supports up to five DTMF payloads bound to different clock rates if the SIP Trunk Group services flag honorSdpClockRate is enabled (Refer to SIP Trunk Group - Services - CLI or Trunk Group - SIP Trunk Group (EMA) page for configuration details.).

![Warning](https://example.com)

**DSP resources are used when interworking between different codecs. When using in-band to out-of-band or in-band to DTMF digit (RFC 2833) interworking (both directions), DSP resources are required.**

![Warning](https://example.com)

**Pass-through calls are supported when receiving digits through RFC 2833 or out-of-band (SIP INFO).**

DTMF Interworking

The SBC also supports interworking between different DTMF Relay methods. For packet-to-packet calls both signaling and media (bearer) traffic flow through the SBC (whether or not media transcoding is also employed). For signaling traffic, the SBC inherently provides address and port mapping between the addresses used within the carrier's own network and those “visible” to the peering partner.

The SBC includes two seeded DTMF trigger profiles:
- Ingress—Applied to the calling (ingress leg) side.
- Egress—Applied to the called (egress leg) side.

To enable this feature, configure and enable the SBC DTMF trigger for the appropriate leg of the call. Changes to configuration parameters do not affect existing calls.

To configure via CLI, use following syntax:

```bash
% set profiles dtmfTrigger <egress | ingress> interdigitTimeout <timeout_value> longdigitDuration <longdigitDuration> pattern <trigger_pattern> state <enabled | disabled>
```

Below is a CLI example of enabling DTMF interworking for PSP "profile1" with transcoding.
Refer to DTMF Trigger - CLI, Packet Service Profile - CLI or Dtmf Trigger Profile - Dtmf Trigger, Packet Service Profile - Flags, Packet To Packet Control - Condition In Addition To No Common Codec (EMA) for configuration details.

**DTMF Interworking Without DSP**

The SBC supports interworking between RFC2833/RFC4733 and out-of-band DTMF methods without using DSPs. Below is a CLI example of enabling DTMF interworking for PSP "profile1" without transcoding.

```
% set profiles media packetServiceProfile profile1 flags digitDetectSendEnabled enable
% set profiles media packetServiceProfile profile1 flags interworkDtmfWithoutTranscoding enable
```

To use DTMF interworking without DSP resources, the Packet Service Profile flag **digitDetectSendEnabled** must be enabled.

Refer to Packet Service Profile - CLI or Packet Service Profile - Flags (EMA) for configuration details.

**SIP RTP Relay**

To preserve the privacy and security model for media flows as well, the SBC implements an RTP Relay. In RTP Relay, the call endpoints send media packets to an address/port on the SBC. As the media packets flow through the SBC, the source and destination addresses are changed such that the SBC is itself always one of the endpoints in each call leg. Thus, the carrier's internal addresses are not exposed to the peering partner, and inbound media flows are directed only to a limited set of well-defined SBC addresses for security filtering purposes.

Figure below depicts the two SIP IP "pipes" used by the SBC:

- The **SIP Signaling Channel** exchanges call signaling packets with a SIP Application Server (AS).
- The **IP Media Stream** exchanges call data with a SIP Media Server (MS), IAD or SIP phone.

*Figure 1: SIP IP Pipe Example Using SBC 5000 Series*
The SIP Application Server manages the SIP MS/IAD/phone through the IP cloud, shown by the dotted line in Figure 1. This interaction is logically removed from the SIP SBC software and hence is not annotated in the following examples.

Audio call data may be encoded according to any of the algorithms below for delivery through the IP Media Stream:

- G.711
- G.711 with Silence Suppression
- Cloned encoding algorithms
- Isolating the tones and designating them explicitly within the IP Media Stream
- Isolating the tones, then designating and delivering them explicitly on the SIP Signaling Channel instead of the IP Media Stream

The IP Media Stream uses the Realtime Transport Protocol (RTP) to deliver call data for all audio compression algorithms.

### Audio Stream Examples Using DTMF

**Table 1: Audio Stream Examples With DTMF**

<table>
<thead>
<tr>
<th>Audio Stream</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711</td>
<td>G.711 audio packet stream containing uncompressed DTMF information encoded in IP Media Stream along with all other audio data.</td>
</tr>
<tr>
<td>G.711/RFC2833</td>
<td>G.711 audio packet stream containing DTMF information that is placed into the RTP stream using the RFC2833 mechanism. In this case, the underlying DTMF audio also remains embedded in the audio RTP packets.</td>
</tr>
<tr>
<td>G.711/RFC2833 (with DTMF Digits Removed)</td>
<td>G.711 audio packet stream containing DTMF that is placed into the RTP stream using the RFC2833 mechanism, but further manipulates the RTP stream by removing these tones from the audio RTP packets.</td>
</tr>
<tr>
<td>G.711 (with DTMF Digits Removed)</td>
<td>G.711 audio packet stream with all DTMF removed. The DTMF is delivered through DTMF INFO messages that are conveyed on the SIP Signaling Channel. The RFC2833 mechanism is not present in this delivery scheme.</td>
</tr>
<tr>
<td>G.72X</td>
<td>Audio compression packet stream containing compressed DTMF information that is encoded in the IP Media Stream along with all other (compressed) audio data.</td>
</tr>
<tr>
<td>G.72X/RFC2833</td>
<td>Audio compression packet stream containing DTMF information that is placed into the RTP stream using the RFC2833 mechanism. In this case, the underlying DTMF audio also remains embedded in the audio RTP packets.</td>
</tr>
<tr>
<td>G.72X/RFC2833 (with DTMF Digits Removed)</td>
<td>Audio compression packet stream containing DTMF that is placed into the RTP stream using the RFC2833 mechanism, as above, but further manipulates the RTP stream by removing these tones from the audio RTP packets.</td>
</tr>
<tr>
<td>G.72X/ (with DTMF Digits Removed)</td>
<td>Audio compression packet stream that has all DTMF removed. The DTMF is delivered through DTMF INFO messages that are conveyed on the SIP Signaling Channel. The RFC2833 mechanism is not present in this delivery scheme.</td>
</tr>
<tr>
<td>SIP Call Signaling Packets without DTMF INFO Messages</td>
<td>Exchange signaling information for SIP calls. In this case, these packets do not contain any DTMF information, and are conveyed on the SIP Signaling Channel.</td>
</tr>
<tr>
<td>SIP Call Signaling Packets with DTMF INFO Messages</td>
<td>Exchange all signaling information for SIP calls, including DTMF information. DTMF digits are delivered via SIP INFO messages. These packets are conveyed on the SIP Signaling Channel.</td>
</tr>
</tbody>
</table>

### KPML DTMF Support
The Key Press Markup Language (KPML) functionality enables the SBC to look for DTMF trigger patterns and to notify an external SIP entry when such patterns are detected. When there is a SUBSCRIBE with a KPML event, the SBC SIP Front End (SIPFE) matches the KPML dialog event parameters and also determines which SIP Signaling Group is hosting this SUBSCRIBE event. The SUBSCRIBE event is then sent to the correct SIP Signaling Group and the XML/KPML document is parsed. This action gives the SIP Signaling Group enough information to send a digit map pattern to the Node Resource Manager Agent (NRMA). If the digit pattern matches, the NRMA send a message back to SIP Signaling Group about the matched digit pattern. Once the digit pattern is matched, the SIP Signaling Group is able to send SIP NOTIFY message to the awaiting SIP client.

There are two types of subscriptions that the KPML recognizes:

- **One-Shot Subscription**: The one-shot subscription occurs once and terminated after a pattern match occurs and a report is sent in a NOTIFY message. The "Subscription-State" in the NOTIFY message is set to "terminated".

- **Persistent Subscription**: The persistent subscription remains active even after the pattern match occurs. There are two sub-types:
  - **Continuous Notify**: The User Interface (UI) sends a NOTIFY message whenever a user input matches a subscribed pattern.
  - **Single Notify**: The UI sends a NOTIFY message whenever a user input matches the first subscribed pattern. There are no NOTIFY messages until the SBC receives a new subscription request on the subscription dialog.

**Note:**
The stream reversal is not supported for H.323-SIP calls.

## Call Flows

### One-Shot Subscription Call Flow

**Figure 2: One-Shot Subscription Call Flow**

Where:

F1: KPML Subscribe Message from Application Server to SIPSG.
F2: 200 OK message from SIPSG to Application Server.
F3: Digit Detect message from SIPSG to NRMA.
F5: Digit Detect notify message (pending) from NRMA to SIPSG.
F6: Notify message from SIPSG to Application Server.
F7: 200 OK message from Application Server to SIPSG.
F8: Mid call Digit Detect Notify message from NRMA to SIPSG.
F9: Notify message from SIPSG to Application Server.
F10: Notify OK message from Application Server to SIPSG.

**Persistent Request Call Flow**

**Figure 3:** Persistent Request Call Flow

Where:
F1: KPML Subscribe Message from Application Server to SIPSG.
F2: 200 OK message from SIPSG to Application Server.
F3: Digit Detect message from SIPSG to NRMA.
F5: Digit Detect notify message (pending) from NRMA to SIPSG.
F6: Notify message from SIPSG to Application Server.
F7: 200 OK message from Application Server to SIPSG.
F8: Digit Detect Notify message (if match found) from NRMA to SIPSG.
F9: Notify message from SIPSG to Application Server.
F10: Digit Detect Notify message (if match found) from NRMA to SIPSG.
F11: 200 OK Notify message from Application Server to SIPSG.
F12: Notify message from SIPSG to Application Server.
F13: Notify message from Application Server to SIPSG.

**Outward Call Termination Call Flow**

*Figure 4: Outward Call Termination Call Flow*

**Inward Call Termination Call Flow**

*Figure 5: Inward Call Termination Call Flow*

**Subscription Expiration Call Flow**
Reverse Stream Call Flow

When a KPML subscription is received with "stream=reverse" attribute, the SBC detects digits in the reverse direction with respect to the dialog details present in Event header of the SUBSCRIBE message.

⚠️ The KPML supports GW-GW calls. However, KPML does not support GW-GW for Reverse Stream calls.

Figure 6: Subscription Expiration Call Flow

Figure 7: Reverse Stream Call Flow
The SBC supports both the subscriptions (normal and reverse) to detect digits in both the directions of the call.