Oracle® Communications Session Border Controller

Transcoding Essentials Guide
Release S-CX6.3.7M3

September 2014

Notices

Contents

Oracle Legal Notices

Preface

About this guide

Your Documentation Supplement

The Transcoding Essentials Guide Release Version S-CX6.3.7 M3 is intended to augment the body of documentation published for the S-CX6.3.0 release and the Oracle Enterprise Session Border Controller documentation set. This document covers the new features and functions available with the installation of a Transcoding NIU. In addition, it offers summaries of ACLI diagnostic commands and new alarms.

Audience

This guide is written for network administrators and architects.

Related Documentation

The following table lists the members that comprise the documentation set for this release:

Document Name	Document Description
Acme Packet 4500 System Hardware Installation Guide (400-0101-00)	Contains information about the components and installation of the Acme Packet 4500 system.
Acme Packet 3800 Hardware Installation Guide (400-0118-00)	Contains information about the components and installation of the Acme Packet 3800 system.
Release Notes	Contains information about the current documentation set release, including new features and management changes.
ACLI Configuration Guide	Contains information about the administration and software configuration of the SBC.
ACLI Reference Guide	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Maintenance and Troubleshooting Guide	Contains information about logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.
MIB Reference Guide	Contains information about Management Information Base (MIBs), Acme Packet's enterprise MIBs, general trap information, including specific details

Document Name	Document Description
	about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects.
Accounting Guide	Contains information about accounting support, including details about RADIUS accounting.
HDR Resource Guide	Contains information about the Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Administrative Security Essentials	Contains information about support for its Administrative Security license.

Revision History

Date	Revision Number	Description
March 1, 2013	Revision 1.00	Initial Release
May 09, 2013	Revision 1.01	Added the following OIDs to the section “Acme Packet System Management MIB (ap-smgmt.mib)”: • apSysXCodeEVRCCapacity 1.3.6.1.4.1.9148.3.2.1.1.39 • apSysXCodeEVRCBCapacity 1.3.6.1.4.1.9148.3.2.1.1.40 Also added when the apSysXCode traps would get triggered.
August 9, 2013	Revision 2.00	The release of this document corresponds with release of S-CX6.3.7M2 software. • Increased jitter buffer capacity to 500ms • Increased packet reordering window to 500ms
September 17, 2014	Revision 3.00	The release of this document corresponds with release of S-CX6.3.7M3 software. • The release notes section that follows has been edited to list updates per the 3 M releases of this software.

Release Notes

Minimum requirements for Version S-CX6.3.7 and E-CX6.3.7

- Transcoding NIU
- Bootloader - 06/21/2011
- A new bootloader is required in order to support booting for a .tar file
- The tarball includes the Digital Signal Processor (DSP) firmware
- Display the current bootloader using: **show version boot**
- Net-Net 4250 is not supported

CX6.3.7M3 Release Features

- S-CX630M5p6 release base

- DSP firmware update
- C2.0.2 SPL Engine

CX6.3.7M2 Release Features

- S-CX630M5 release base
- DSP firmware update
- C2.0.2 SPL Engine
- Capacity of the adaptive jitter buffer increased to 500ms to improve call quality in adverse network conditions
- Capacity of the packet re-ordering window increased to 500ms to improve call quality in adverse network conditions

CX6.3.7M1 Release Features

- S-CX630M2p5 release base
- DSP firmware update
- C2.0.2 SPL Engine
- SDP insertion for initial INVITE and following reINVITEs
- Fax Transcoding Improvements
- Digest Authentication
- AMR-WB Realm Stats

DTMF interworking

- SIP-INFO to/from RFC2833 on transcoded and non-transcoded calls
- SIP-INFO to/from in-band G.711 on transcoded calls
- RFC2833 to/from in-band G.711 on transcoded calls
- RFC2833 to/from SIP-KPML internetworking NOT supported on transcoded calls

QoS Reporting

- QoS for non-transcoded calls is supported using the FPGA on the NIU only for IPv4 calls
- QoS for non-transcoded IPv6 calls is not supported.
- QoS for transcoded calls using DSP is NOT supported (future)

Release Caveats

- SIP protocol support only
- H323/IWF is NOT supported for transcoded or non-transcoded sessions in this release
- Border Gateway functionality is NOT supported

T.38 Fax Transcoding

- T.38 Fax transcoding available for G711 only @ 10ms, 20ms, 30ms ptime
- Fax codec policy based on D7.0 fax transcoding policy

Packet Trace

- Media packet tracing for transcoded calls is NOT supported
- Non-transcoded calls are not affected
- Signaling packet trace is supported for transcoded and non-transcoded calls

Lawful Intercept

- Lawful intercept for media that is transcoded is NOT supported

- Lawful intercept for non-transcoded media flows is supported
- Signaling lawful intercept is supported for transcoded and non-transcoded calls

Media interface speed

- 1 Gbps media interface support only
- Link redundancy NOT supported
- **show media utilization** NOT supported

Call Recording

- Call recording for transcoded sessions is NOT supported

Hide Media Update

- Not applicable/supported for transcoded sessions

AMR and AMR-WB codecs

- Supported with license (in increments of 25)
- AMR and AMR-WB each require a separate license
- All modes (rates) are supported
- Octet-aligned and bandwidth efficient modes are supported
- CRC, robust-sorting and interleaving disabled

Simultaneous Support of Encryption and Transcoding

The transcoding NIU is not equipped with the acceleration hardware required for supporting IPsec and SRTP encryption. This acceleration hardware is supported on other NIU models for the Acme Packet 3820 and Acme Packet 4500. When configured with the transcoding NIU and S-Cx6.3.0:

- IPsec w/transcoding is not supported
- SRTP w/transcoding is not supported
- TLS for signaling IS supported with transcoding

CAM Capacities

Refer to the following table for Oracle Enterprise Session Border Controller Version CX6.3.7 M3 CAM capacities:

Platform	Denied	Trusted	Media	Untrusted	Dynamic Trusted	NAT	ARP
Acme Packet 4500	32000	8000	64000	2000	250000	126976	4096
Acme Packet 3820	16000	2000	32000	1000	125000	61440	4096

DSP Firmware Upgrade Boot Time

The Oracle Enterprise Session Border Controller requires an extended boot time when updating the Digital Signal Processor (DSP) firmware on the Transcoding Network Interface Unit (NIU). When the Oracle Enterprise Session Border Controller boots with a Transcoding capable software image, it checks that the hardware is running the latest version of the DSP firmware. If it is running the latest version of DSP firmware, the boot process proceeds normally. If it is not running the latest version of the DSP firmware, the system must update the hardware; this may require several extra minutes for the boot. All subsequent boots skip the DSP firmware step (until a new software image is loaded that includes newer DSP firmware).

1

Transcoding

Introduction

Transcoding is the ability to convert between media streams that are based upon disparate codecs. The Oracle Enterprise Session Border Controller supports IP-to-IP transcoding for SIP sessions, and can connect two voice streams that use different coding algorithms with one another.

This ability allows providers to:

- Handle the complexity of network connections and the range of media codecs with great flexibility
- Optimize bandwidth availability by enforcing the use of different compression codecs
- Normalize traffic in the core network to a single codec
- Enact interconnection agreements between peer VoIP networks to use approved codecs

By providing transcoding capabilities at the network edge rather than employing core network resources for the same functions, the Oracle Enterprise Session Border Controller provides cost savings. It also provides a greater degree of flexibility and control over the codec(s) used in providers' networks and the network with which they interconnect.

In addition, placing the transcoding function in the Oracle Enterprise Session Border Controller and at the network edge means that transcoding can be performed on the ingress and egress of the network. The Oracle Enterprise Session Border Controller transcodes media flows between networks that use incompatible codecs, and avoids back-hauling traffic to a centralized location, alleviating the need for multimedia resource function processors (MRFPs) and media gateways (MGWs) to support large numbers of codecs. This maximizes channel density usage for the MRFPs and MGWs so that they can reserve them for their own specialized functions.

Feature Support

Note that the M1 version does not support the following capabilities:

- Border Gateway Function (BGF)
- Secure Real-Time Transport Protocol (SRTP)
- Multimedia Internet KEYing (MIKEY)

Hardware Support for the Acme Packet 4500 and Acme Packet 3820 Transcoding NIU

A Transcoding Network Interface Unit (NIU) provides the DSP resources that enable the Oracle Enterprise Session Border Controller's transcoding feature. You can install one to twelve transcoding modules in a Transcoding NIU to provide increasing transcoding capacity.

Transcoding License

The transcoding NIU requires a special license for transcoding to and from AMR and AMR-WB audio codecs. Licenses are purchased in bins of 25. AMR and AMR-WB each require a separate license.

Transcoding Capacity

Transcoding capacity depends on the following:

- Codecs used for transcoding
- Number of transcoding modules installed on the Transcoding NIU; you can have between one and twelve Transcoding Modules installed in a Oracle Enterprise Session Border Controller
- Capacity scales linearly with each extra transcoding module installed

Codec Policy Definition

Codec policies describe how to manipulate SDP messages as they cross the Oracle Enterprise Session Border Controller. The Oracle Enterprise Session Border Controller bases its decision to transcode a call on codec policy configuration and the SDP. Each codec policy specifies a set of rules to be used for determining what codecs are retained, removed, and how they are ordered within SDP.

Syntax

The following parameters are used to create a codec policy. Their syntax is described inline.

allow-codecs

allow-codecs—The allow-codecs parameter configures the codecs that are allowed and/or removed from the SDP. A blank list allows nothing, ***** allows all codecs, **none** removes all codecs, the **:no** designation blocks the specific codec or class of media, and the **:force** designation is used to remove all non-forced codecs.

The **allow-codecs** parameter is configured in the following way:

- <codec>**:no**—blocks the specific codec
- *****—allow all codecs.
- <codec>**:force—**If any forced codec is present in an SDP offer, all non-forced codecs are stripped from the m-line.
- **audio:no**—audio m= line is disabled
- **video:no**—video m= line is disabled

For example, if you configure PCMU in the allow-codecs parameter, the PCMU codec, received in an SDP message is allowed on to the next step of transcoding processing, and all other codecs are removed.

The order of precedence is for removing codecs according to codec policy is:

1. <codec>**:no**—Overrides all other allow-codecs parameter actions.
2. **audio:no** / **video:no**. An allow-codecs line like “allow-codecs PCMU audio:no” disables the PCMU m= line because audio:no has a higher precedence than the specific codec.
3. <codec>**:force**
4. <codec> Specific codec name and those codecs configured in the add-codecs-on-egress list.
5. ***** has the lowest precedence of all flags. For example "**allow-codecs * PCMU:no**" allows all codecs except PCMU.

order-codecs

order-codecs—The order-codecs parameter is used to re-order the codecs in the m= line as the SDP is passed on to the next step. This parameter overwrites the order modified by the **add-codecs-on-egress** command, when relevant. The following is valid syntax for this parameter:

- <blank>—Do not re-order codecs
- *—You can add a <codec> before or after the **`*`** which means to place all unnamed codecs before or after (the position of the *) the named codec. For example:
- <codec> *—Puts the named codec at the front of the codec list.
- * <codec>—Puts the named codec at the end of the codec list.
- <codec1 > * <codec2>—Puts <codec1> first, <codec2> last, and all other unspecified codecs in between them.
- <codec>—When the * is not specified, it is assumed to be at the end.

Any codec name is allowed in the **`order-codecs`** parameter, even those not defined or not transcodable. An * tells the order-codecs parameter where to place unspecified codecs with respect to the named codecs. Refer to the examples below.

- <blank>—do not reorder m= line
- PCMU *—Place PCMU codec first, all others follow
- * PCMU—Place PCMU codec last, all others proceed PCMU
- G729 * PCMU—Place G729 codec first, PCMU codec last, all others remain in between
- PCMU—If * is not specified, it is assumed to be at the (PCMU *).

Add on Egress

`add-codecs-on-egress`—This parameter adds a codec to the SDP's m= line only when the codec policy is referenced from an egress realm (except in one 2833 scenario). Codecs entered for this parameter are added to the front of the m= line. Signaling codecs are added to the end of the m= line.

Transcoding can only occur if this parameter is configured. There is a special case for 2833 support where the add-codecs-on-egress parameter is configured for an ingress realm.

Packetization Time

`packetization-time`—This parameter specifies a media packetization time in ms to use within the realm referencing this codec policy. Packetization time You must also enable the force ptime parameter to enable transrating in conjunction with configuring the packetization time.

Transcoding Configuration

The Oracle Enterprise Session Border Controller performs transcoding functions— allowing the entities with incompatible codecs to communicate with each other— between two call legs. The two endpoints can be located in one or two realms or networks. The Oracle Enterprise Session Border Controller decides to transcode a call by evaluating messages in the SDP offer-answer transaction with respect to system configuration. An SDP offer can be in a SIP message such as an INVITE or a reINVITE, and contains information about the codecs the offerer would like to use. The answerer answers the SDP offer with its own set of supported codecs. reINVITEs are treated as new negotiations, with respect to the actual SDP offerer and answerer. The Oracle Enterprise Session Border Controller can manipulate an SDP message by reordering codec preference, and by adding and deleting codecs.

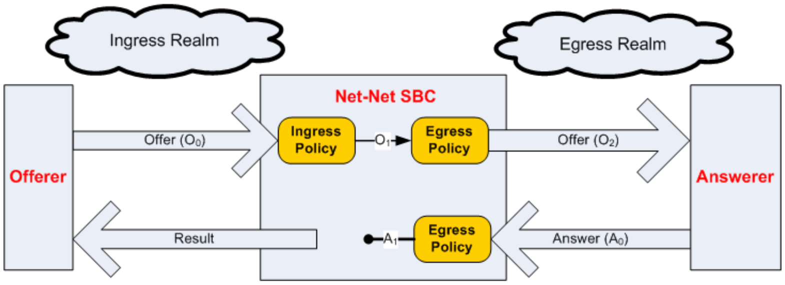
Transcoding Processing Overview

Transcoding processing is viewed in terms of the ingress and egress realms. The ingress realm is where the SDP offer is received by the Oracle Enterprise Session Border Controller. The egress realm is where the SDP offer is sent, and where the SDP answer is expected to be received from (i.e., the answerer's realm). A call is defined as transcodable if an egress or ingress policy exists for the session and if the session is not subject to media release, as specified in the realm configuration.

To understand the details of transcoding processing, refer to the following diagram. An SDP offer, O0, is received by the Oracle Enterprise Session Border Controller in the ingress realm. The ingress codec policy is applied here and the SDP offer is transformed into O1. O1 is then passed to and processed by the egress codec policy. This SDP message is then forwarded to the answerer as O2. The answerer replies with A0 to the Oracle Enterprise Session Border Controller, which is subjected to the egress codec policy again and transformed into A1.

When policy dictates not to transcode the call, the Result SDP sent back to the offerer is based on the common codecs shared between A1 and O1. The Oracle Enterprise Session Border Controller first constructs the list of codecs that are present in both in O1 and A1. Then, the Oracle Enterprise Session Border Controller maintains the codec order from A1 for the Result as it is sent to the offerer.

When policy dictates to transcode the call, the top transcodable codec in O1 is used in the ingress realm and the top non-signaling codec in A1 is used in the egress realm.



Defining Codec Policies

The following definitions are required for understanding transcoding processing:

DTMFable Codecs—Uncompressed codecs that are capable of properly transmitting a DTMF waveform. The only codecs designated as DTMFable are PCMU and PCMA.

FAXable Codecs—Uncompressed codecs that are capable of properly transmitting a T.30 waveform. The only codecs designated as FAXable are PCMU and PCMA.

Signaling Codecs—Non-audio codecs that are interleaved into a media stream but cannot be used on their own. The only codecs designated as Signaling Codecs by the Oracle Enterprise Session Border Controller are telephone-event and CN (comfort noise).

Disabling an m= line—This is in reference to an m= line in an SDP message. It means setting the m= line's port to 0 (RFC 3264). Enabling an m= line means it has a non-zero port. The m= line's mode attribute (sendrecv/inactive/rtcponly, etc) is not considered.

A codec policy is created by configuring the following information:

- Which Codecs are allowed and which are denied in a realm.
- Which Codecs should be added to the SDP m= lines for an egress realm.
- The preferred order of codecs to indicate in an SDP m= line.
- The packetization time which should be enforced within a realm.

Ingress Policy

Incoming SDP is first subject to the ingress codec policy. If no codec policy is specified in the realm config for the ingress realm, or the m= lines in the SDP offer are disabled (by a 0 port number), the SDP is transformed to O1 unchanged.

The ingress codec policy first removes all un-allowed codecs, as configured in the **`allow-codecs`** parameter (<codec>:no). For each un-allowed codec removed, its rtpmap and fmtp attributes are also removed from SDP. Next, the remaining codecs are ordered with the **`order-codecs`** parameter. Ordering is when the codec policy rearranges the codecs in the SDP m= line. This is useful to suggest the codec preferences to impose within the egress realm. O1 is then processed by the egress codec policy after a realm is chosen as the destination.

In practical terms, the ingress policy can be used for filtering high-bandwidth codecs from the access realm. It can also be used for creating a suggested, prioritized list of codecs to use in the ingress realm.

Egress Policy

The Oracle Enterprise Session Border Controller applies egress codec policy to the SDP that has already been processed by ingress policy. The egress policy is applied before the SDP exits the system into the egress realm. If no egress codec policy is defined, or the SDP's m= lines are disabled (with a 0 port), the SDP is passed untouched from the ingress policy into the egress network.

The egress codec policy first removes all un-allowed codecs in the **`allow-codecs`** parameter (<codec>:no). Codecs on the **`add-codecs-on-egress`** list are not removed from the egress policy regardless of the how the **`allow-codecs`** parameter is configured. If the result does not contain any non-signaling codecs, the ptime attribute is removed from the SDP. Codecs not present in O1 that are configured in the **`add-codecs-on-egress`** parameter are added to the SDP, only if O1 contains one or more transcodable codecs.

Note: Transcoding can only occur for a call if you have configured the add-codecs-on-egress parameter in an egress codec policy.

If codecs with dynamic payload types (those between 96 and 127, inclusive) are added to the SDP, the lowest unused payload number in this range is used for the added codec.

The following rules are also applied for egress policy processing:

If O1 contains at least one transcodable codec, the codecs listed in the Egress policy are added to the SDP.

- telephone-event, as configured in add-codecs-on-egress will only be added if O1 contains at least one DTMFable codec.
- T.38, as configured in add-codecs-on-egress will only be added if there is no T.38 and there is at least one FAXable codec (G711Fall Back (FB)) in O1. T.38 will then be added as a new m= image line to the end of SDP.

  If G711FB is not allowed in the egress policy, the Oracle Enterprise Session Border Controller disables the m= line with the FAXable codec. Otherwise if G711FB is allowed, pass it through the regular offer processing allowing/adding only FAXable codecs.
- G711FB, as configured in add-codecs-on-egress will only be added if there is no G711FB and there is T.38 in O1. G711FB will then be added as a new m= audio line to the end of SDP.

  If T.38 is not allowed in the egress policy, the Oracle Enterprise Session Border Controller disables the m= image line. Otherwise if T.38 is allowed, pass it through the regular offer processing.

If the result of the egress policy does not contain any non-signaling codecs, audio or video, the m= line is disabled, by setting the port number to 0.

The m= line is next ordered according to rules for the order-codecs parameter.

Finally, all attributes, a= lines, ptime attribute, and all other unrecognized attributes are maintained from O1. Likewise, appropriate attributes for codecs added by the add on egress parameter are added to SDP. Finally, rtpmap and fmtp parameters are retained for codecs not removed from the original offer. The result of all this is O2, as shown in the overview diagram.

In practical terms, codec policies can be used to normalize codecs and ptime in the core realm where the network conditions are clearly defined.

Codec policies can also be used to force the most bandwidth-conserving codecs anywhere in the network.

Post Processing

If any errors are encountered during the Ingress and Egress policy application, or other violations of RFC3264 occur, the call is rejected. If O2 does not contain any enabled m= lines at the conclusion of the initial call setup, the call is rejected. If O2 does not contain any enabled m= lines at the conclusion of a reINVITE, the reINVITE is rejected and the call reverts back to its previous state.

Answer Processing and Examples

Unoffered Codec Reordering

According to RFC 3264, the answerer can add codecs that were not offered to the Answer. The answerer may add new codecs as a means of advertising capabilities. RFC 3264 stipulates that these unoffered codecs must not be used.

The RFC does not dictate where in the m= line these codecs can appear and it is valid that they may appear as the most preferred codecs.

In order to simplify the answer processing, the Oracle Enterprise Session Border Controller moves all unoffered codecs in A0 to the back of the SDP answer before any other answer processing is applied.

Non-transcoded Call

The decision to transcode is based on the top non-signaling codec in A1. If the top A1 codec is present in O1, the call proceeds, non-transcoded. This is the rule for non-signaling codecs (i.e., not RFC 2833 nor FAX).

Transcoded Call

The following two conditions must then be met to transcode the call's non-signaling media:

- The top A1 codec is not present in the O1 m= line
- The top A1 codec was added by the egress policy

If these rule are met, the Oracle Enterprise Session Border Controller will transcode between the top A1 codec and the top transcodable, non-signaling O1 codec.

Voice Transcoding

The following examples use the ingress and egress codec policies listed at the top of each scenario. The examples use changing SDP offers and answers, which contribute to unique results, per example. The effects of the SDP offers and answers are explained in each example.

Voice Scenario 1

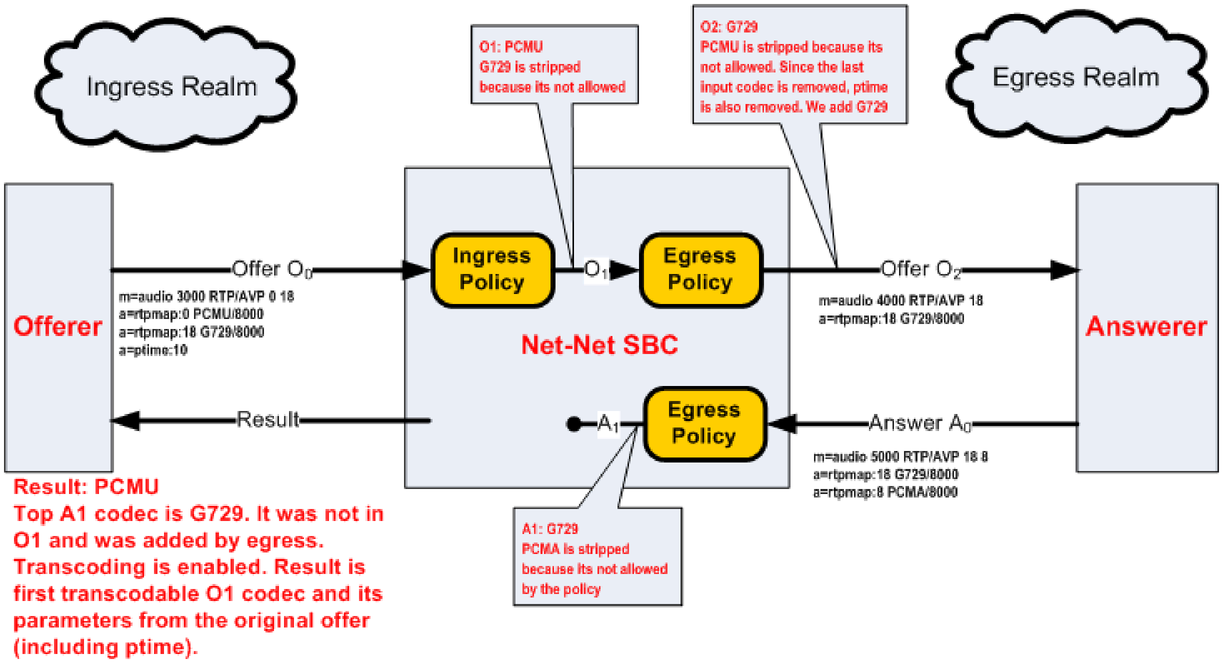
The following ingress and egress policies are used for scenario 1.

Ingress Policy		Egress Policy	
allow-codecs	PCMU GSM	allow-codecs	G729 GSM G722
add-codecs-on-egress	PCMU	add-codecs-on-egress	G729
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

Note: The codec in the ingress policy's add-codecs-on-egress parameter has no effect in the following examples. Its presence would have an effect if a reINVITE was initiated from egress realm, effectively reversing the roles of the codec policies.

1. In the following diagram, PCMU and G729 are offered. Ingress policy removes G729 and allows PCMU. The egress policy adds G729 and strips PCMU from offered SDP and forwards it on to the answerer (ptime is also removed because the last codec was removed).

   The SDP answer agreed to use G729 and adds PCMA. The egress policy then strips PCMA from the SDP answer. At this point, the top codec in A1, G729 is checked against O1. Since G729 is not present in O1, it is transcoded to PCMU.



O1: PCMU
G729 is stripped because its not allowed

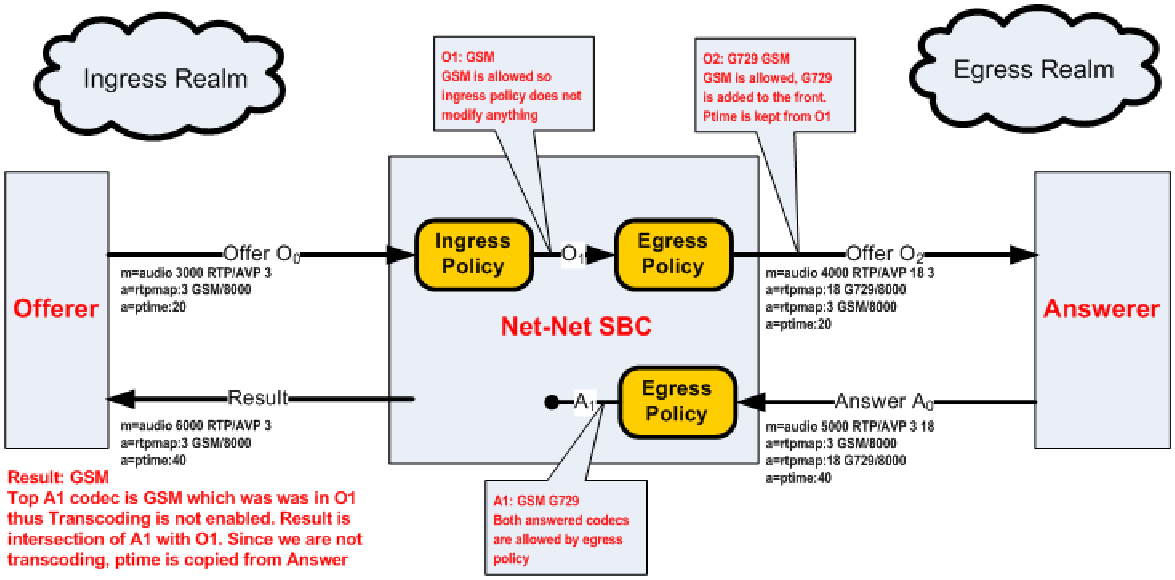
O2: G729
PCMU is stripped because its not allowed. Since the last input codec is removed, ptime is also removed. We add G729

A1: G729
PCMA is stripped because its not allowed by the policy

Result: PCMU
Top A1 codec is G729. It was not in O1 and was added by egress. Transcoding is enabled. Result is first transcodable O1 codec and its parameters from the original offer (including ptime).

2. In the following diagram, GSM is in the original SDP offer. It is then passed through to O1. Egress policy adds G729 and retains ptime from GSM and sends this to the answerer as O2.

   The SDP answer agrees to use G729 and GSM, but prioritizes GSM. The egress policy allows both codecs through, unchanged. Because A1 and O1 both have GSM, it is used for the non-transcoded call. Ptime is copied from A0 to the result.



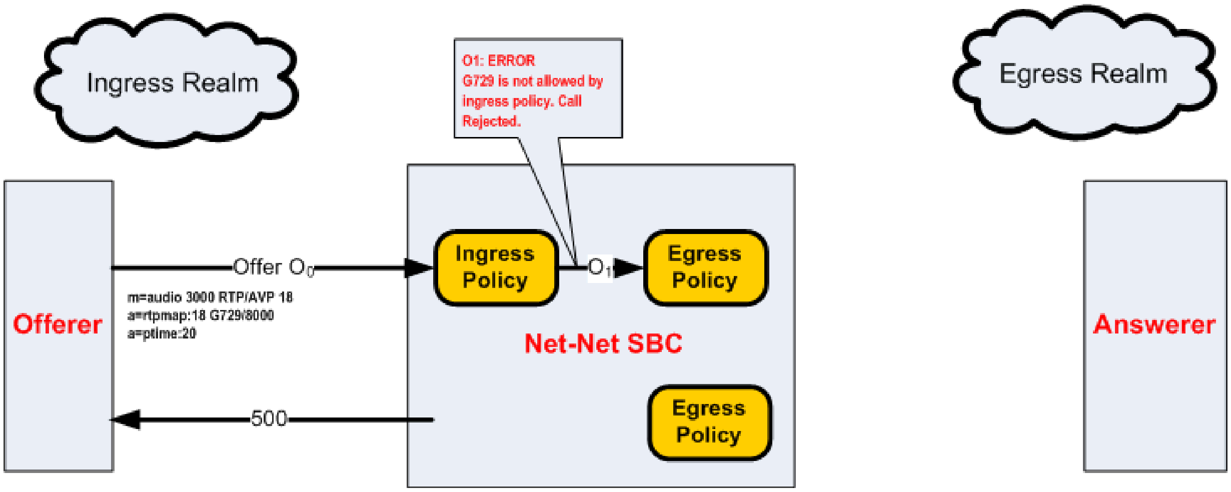
O1: GSM
GSM is allowed so ingress policy does not modify anything

O2: G729 GSM
GSM is allowed, G729 is added to the front. Ptime is kept from O1

A1: GSM G729
Both answered codecs are allowed by egress policy

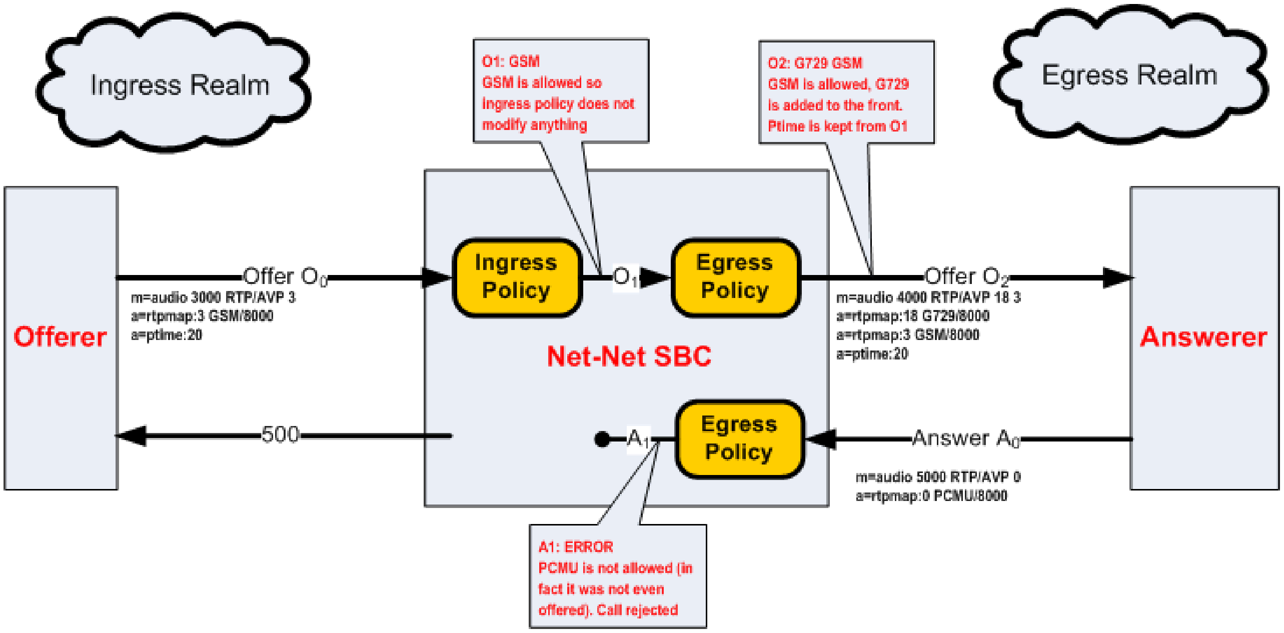
Result: GSM
Top A1 codec is GSM which was was in O1 thus Transcoding is not enabled. Result is intersection of A1 with O1. Since we are not transcoding, ptime is copied from Answer

3. In the following diagram, G729 in the original SDP offer. Because once G729 is removed, no non-signaling are left in O1, thus the call is rejected.



4. In the following diagram, GSM is in the original SDP offer. It is then passed through to O1. Egress policy adds G729 and retains ptime from O1 and sends this to the answerer as O2.

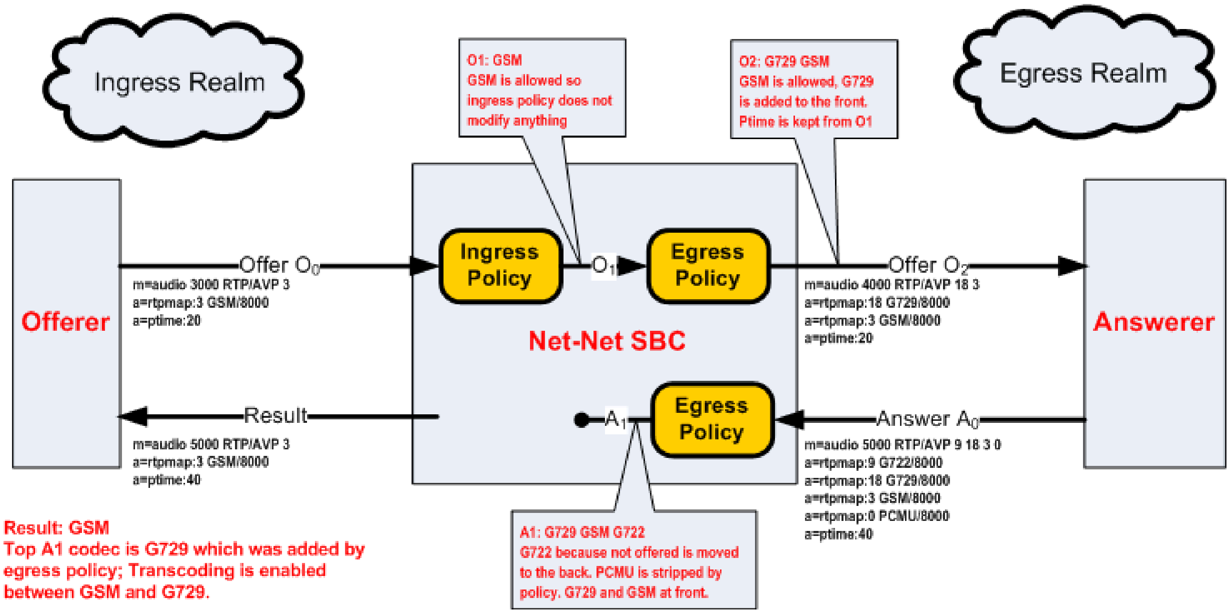
   The SDP answer states that the answerer wants to use PCMU. This is a violation of the RFC3264. Therefore the call is rejected.



   In this example, when the negotiation fails, the Oracle Enterprise Session Border Controller sends a 500 message to the offerer and a BYE message to the answerer.

5. In the following diagram, GSM is in the original SDP offer. It is then passed through to O1. Egress policy adds G729 and retains ptime from O1 and sends this to the answerer as O2.

   The SDP answer replies with G722 G729 GSM and PCMU. PCMU is stripped by policy, G722 is moved to the back of the answer because it was not offered. The top A1 codec was not in O1, and was added by egress policy, therefore the call is transcoded between GSM and G729.



```
m=audio 3000 RTP/AVP 3
a=rtpmap:3 GSM/8000
a=ptime:20
```

```
m=audio 4000 RTP/AVP 18 3
a=rtpmap:18 G729/8000
a=rtpmap:3 GSM/8000
a=ptime:20
```

```
m=audio 5000 RTP/AVP 9 18 3 0
a=rtpmap:9 G722/8000
a=rtpmap:18 G729/8000
a=rtpmap:3 GSM/8000
a=rtpmap:0 PCMU/8000
a=ptime:40
```

```
m=audio 5000 RTP/AVP 3
a=rtpmap:3 GSM/8000
a=ptime:40
```

O1: GSM
GSM is allowed so ingress policy does not modify anything

O2: G729 GSM
GSM is allowed, G729 is added to the front. Ptime is kept from O1

A1: G729 GSM G722
G722 because not offered is moved to the back. PCMU is stripped by policy. G729 and GSM at front.

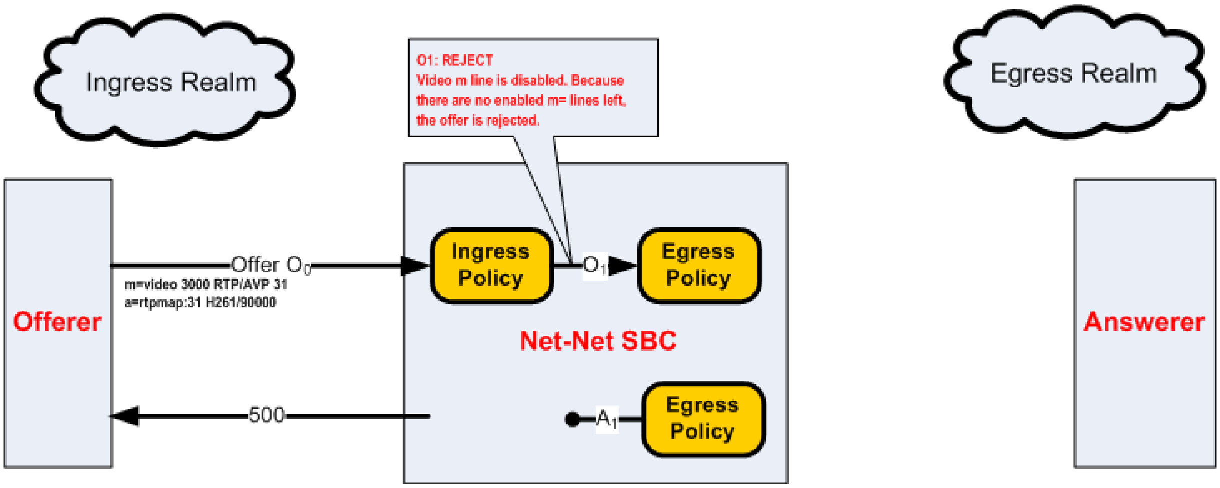
Result: GSM
Top A1 codec is G729 which was added by egress policy; Transcoding is enabled between GSM and G729.

Voice Scenario 2

The following ingress and egress policies are used for scenario 2.

Ingress Policy		Egress Policy	
allow-codecs	Video:no PCMU:force * PCMA:force	allow-codecs	* PCMA:no
add-codecs-on-egress		add-codecs-on-egress	iLBC G726-16
order-codecs		order-codecs	G726-16 * PCMU
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

1. In the following diagram, a video m= line is offered. The ingress policy disables the video m= lines. With no enabled m= lines left, the call is rejected.



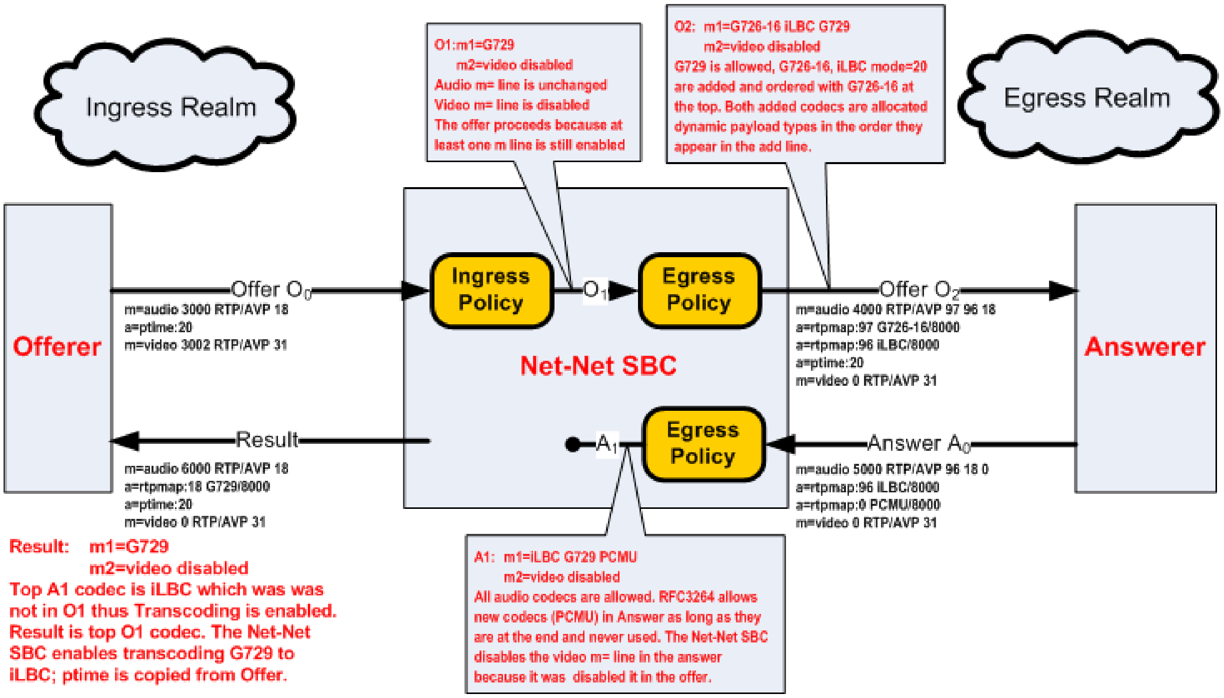
```
m=video 3000 RTP/AVP 31
a=rtpmap:31 H261/90000
```

O1: REJECT
Video m line is disabled. Because there are no enabled m= lines left, the offer is rejected.

2. In the following diagram, G729 and video are offered to the Oracle Enterprise Session Border Controller. Ingress policy allows G729 and disables the video m= line. The egress policy adds iLBC and G726-16, and then orders the codecs according to the order-codecs parameter. The ptime is maintained between O0 and O2. Both added

codecs are allocated dynamic payload types in the order they appear in their m= line. A disabled Video m= line is passed on to the answerer.

The SDP answer agreed to use iLBC, G729, and adds PCMU, and reorders them as stated. The disabled video m= line is maintained. At this point, the top codec in A1, iLBC is used and transcoded with the top codec in O1, G729.



O1:m1=G729
m2=video disabled
Audio m= line is unchanged
Video m= line is disabled
The offer proceeds because at least one m line is still enabled

O2: m1=G726-16 iLBC G729
m2=video disabled
G729 is allowed, G726-16, iLBC mode=20 are added and ordered with G726-16 at the top. Both added codecs are allocated dynamic payload types in the order they appear in the add line.

```
m=audio 3000 RTP/AVP 18
a=ptime:20
m=video 3002 RTP/AVP 31
```

```
m=audio 4000 RTP/AVP 97 96 18
a=rtpmap:97 G726-16/8000
a=rtpmap:96 iLBC/8000
a=ptime:20
m=video 0 RTP/AVP 31
```

```
m=audio 6000 RTP/AVP 18
a=rtpmap:18 G729/8000
a=ptime:20
m=video 0 RTP/AVP 31
```

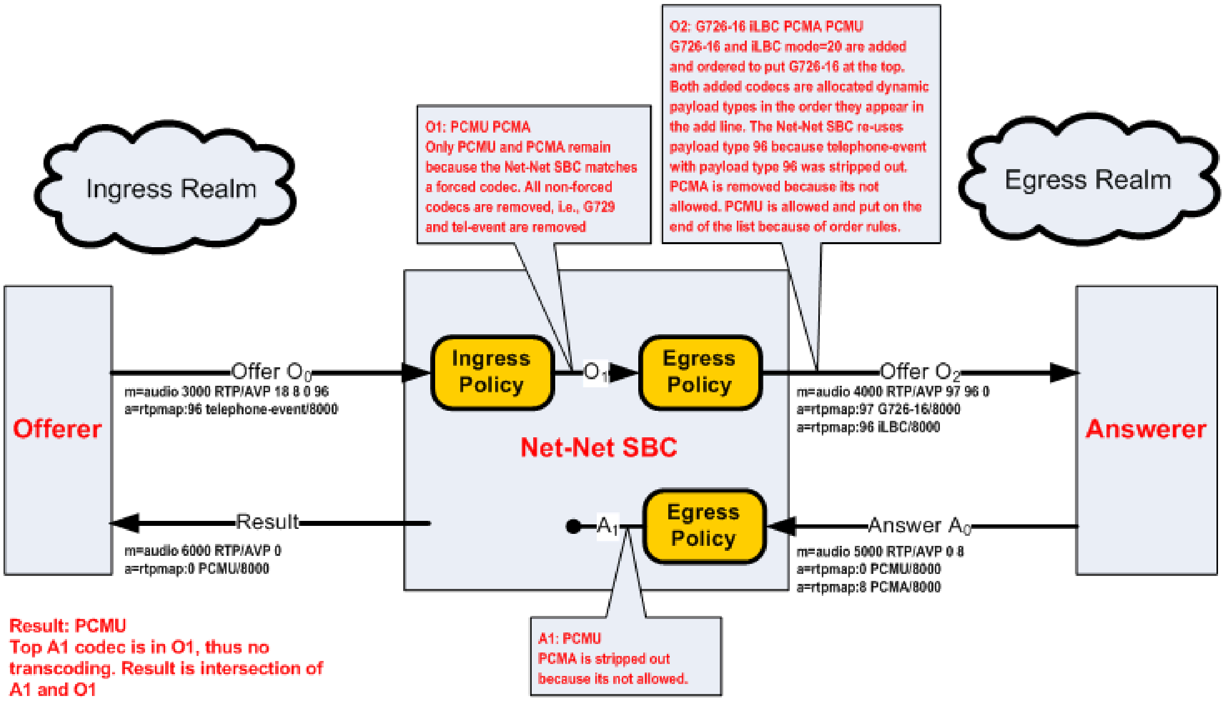
```
m=audio 5000 RTP/AVP 96 18 0
a=rtpmap:96 iLBC/8000
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 31
```

Result: m1=G729
m2=video disabled
Top A1 codec is iLBC which was was not in O1 thus Transcoding is enabled. Result is top O1 codec. The Net-Net SBC enables transcoding G729 to iLBC; ptime is copied from Offer.

A1: m1=iLBC G729 PCMU
m2=video disabled
All audio codecs are allowed. RFC3264 allows new codecs (PCMU) in Answer as long as they are at the end and never used. The Net-Net SBC disables the video m= line in the answer because it was disabled it in the offer.

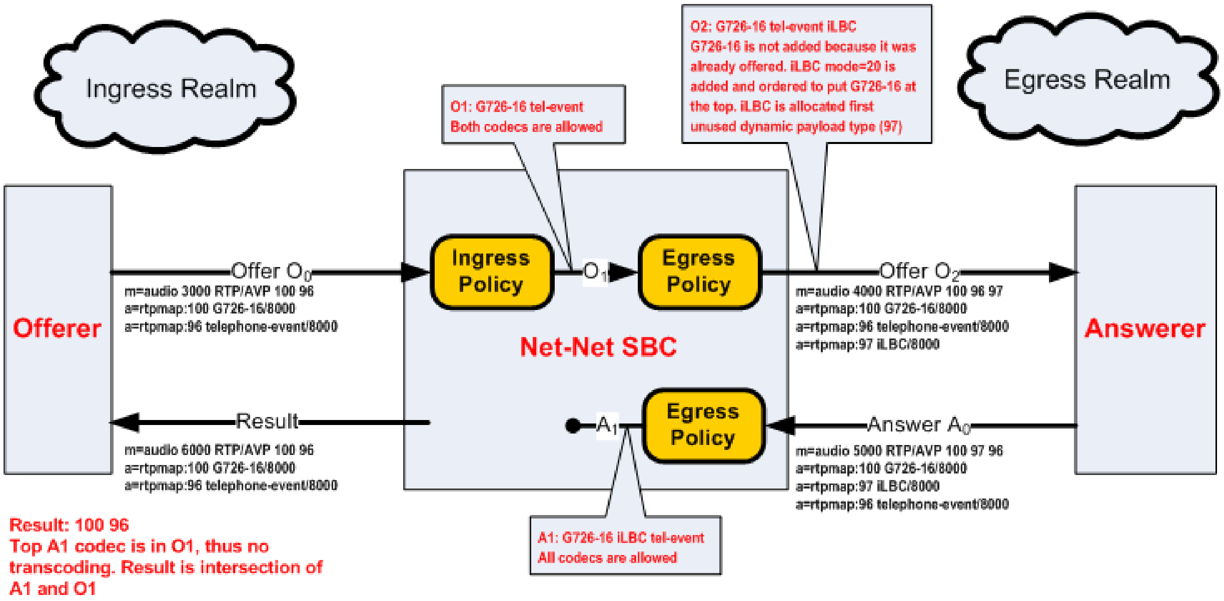
3. In the following diagram, G729 and video are offered to the Oracle Enterprise Session Border Controller. Ingress policy allows G729 and disables the video m= line. The egress policy adds iLBC and G726-16, and then orders the codecs according to the order-codecs parameter. The ptime is maintained between O0 and O2. Both added codecs are allocated dynamic payload types in the order they appear in their m= line.

   The SDP answer only wants to use PCMU and PCMA. The egress policy removes PCMA and passes only PCMU to the offerer. Because PCMU was in O1 and is now the only codec in A1, it is used, and no transcoding is used between the endpoints.


O2: G726-16 iLBC PCMA PCMU
G726-16 and iLBC mode=20 are added and ordered to put G726-16 at the top. Both added codecs are allocated dynamic payload types in the order they appear in the add line. The Net-Net SBC re-uses payload type 96 because telephone-event with payload type 96 was stripped out. PCMA is removed because its not allowed. PCMU is allowed and put on the end of the list because of order rules.
O1: PCMU PCMA
Only PCMU and PCMA remain because the Net-Net SBC matches a forced codec. All non-forced codecs are removed, i.e., G729 and tel-event are removed
Ingress Realm
Egress Realm
Offerer
Offer O0
m=audio 3000 RTP/AVP 18 8 0 96
a=rtpmap:96 telephone-event/8000
Ingress Policy
O1
Egress Policy
Offer O2
m=audio 4000 RTP/AVP 97 96 0
a=rtpmap:97 G726-16/8000
a=rtpmap:96 iLBC/8000
Answerer
Net-Net SBC
Result
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
A1
Egress Policy
Answer A0
m=audio 5000 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
Result: PCMU
Top A1 codec is in O1, thus no transcoding. Result is intersection of A1 and O1
A1: PCMU
PCMA is stripped out because its not allowed.

4. In the following diagram, G726-16 and telephone-event are offered to the Oracle Enterprise Session Border Controller. Ingress policy allows both codecs. The egress policy adds iLBC, and then orders the codecs according to the order-codecs parameter.

   The SDP answer agreed to use all codecs, but reorders them with G726-16 in the top position. Because G726-16 is the top codec in A1, and it is also present in O1, it is used for this call without any transcoding.


O2: G726-16 tel-event iLBC
G726-16 is not added because it was already offered. iLBC mode=20 is added and ordered to put G726-16 at the top. iLBC is allocated first unused dynamic payload type (97)
O1: G726-16 tel-event
Both codecs are allowed
Ingress Realm
Egress Realm
Offerer
Offer O0
m=audio 3000 RTP/AVP 100 96
a=rtpmap:100 G726-16/8000
a=rtpmap:96 telephone-event/8000
Ingress Policy
O1
Egress Policy
Offer O2
m=audio 4000 RTP/AVP 100 96 97
a=rtpmap:100 G726-16/8000
a=rtpmap:96 telephone-event/8000
a=rtpmap:97 iLBC/8000
Answerer
Net-Net SBC
Result
m=audio 6000 RTP/AVP 100 96
a=rtpmap:100 G726-16/8000
a=rtpmap:96 telephone-event/8000
A1
Egress Policy
Answer A0
m=audio 5000 RTP/AVP 100 97 96
a=rtpmap:100 G726-16/8000
a=rtpmap:97 iLBC/8000
a=rtpmap:96 telephone-event/8000
Result: 100 96
Top A1 codec is in O1, thus no transcoding. Result is intersection of A1 and O1
A1: G726-16 iLBC tel-event
All codecs are allowed

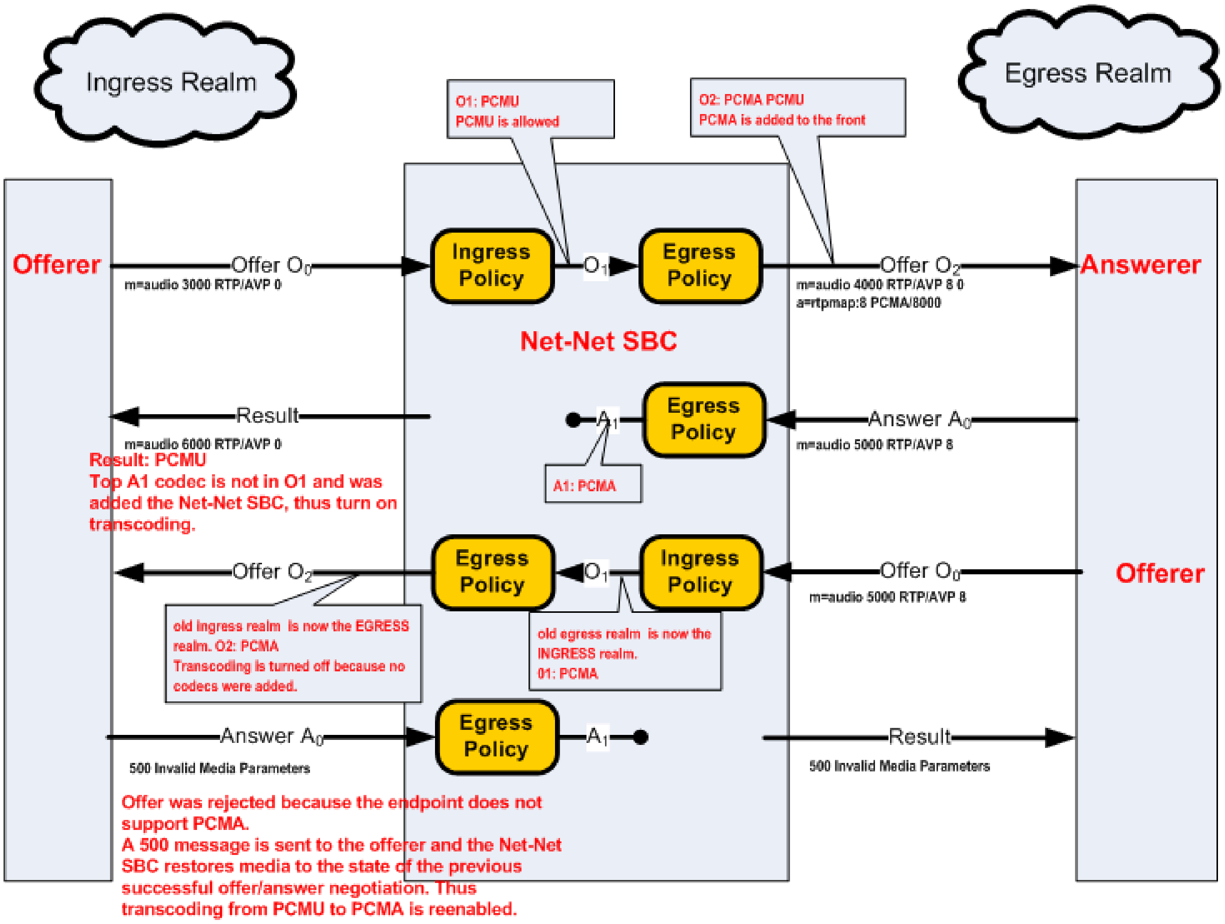
Voice Scenario 3

Voice scenario 3 involves reINVITEs. The following ingress and egress policies are used for scenario 3.

Ingress Policy		Egress Policy	
allow-codecs	PCMU G729	allow-codecs	*
add-codecs-on-egress		add-codecs-on-egress	PCMA

Ingress Policy		Egress Policy	
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

In the following diagram, the answerer sends a reINVITE after a previous transcoding session was established. The original offerer and answerer swap roles. The new offerer rejects the SDP offer and the call reverts to the state negotiated in the original SDP negotiation.


Ingress Realm
Egress Realm
O1: PCMU
PCMU is allowed
O2: PCMA PCMU
PCMA is added to the front
Offerer
Offer O_0
m=audio 3000 RTP/AVP 0
Ingress Policy
O_1
Egress Policy
Offer O_2
m=audio 4000 RTP/AVP 8 0
a=rtpmap:8 PCMA/8000
Answerer
Net-Net SBC
Result
m=audio 6000 RTP/AVP 0
Result: PCMU
Top A1 codec is not in O1 and was added the Net-Net SBC, thus turn on transcoding.
A_1
Egress Policy
A1: PCMA
Answer A_0
m=audio 5000 RTP/AVP 8
Offer O_2
Egress Policy
O_1
Ingress Policy
Offer O_0
m=audio 5000 RTP/AVP 8
Offerer
old ingress realm is now the EGRESS realm. O2: PCMA
Transcoding is turned off because no codecs were added.
old egress realm is now the INGRESS realm.
01: PCMA
Answer A_0
500 Invalid Media Parameters
Egress Policy
A_1
Result
500 Invalid Media Parameters
Offer was rejected because the endpoint does not support PCMA.
A 500 message is sent to the offerer and the Net-Net SBC restores media to the state of the previous successful offer/answer negotiation. Thus transcoding from PCMU to PCMA is reenabled.

RFC 2833 Transcoding

RFC 2833 defines an RTP payload that functions interchangeably with DTMF Digits, Telephony Tones and Telephony Signals. The Oracle Enterprise Session Border Controller can monitor audio stream for in-band DTMF tones and then can convert them to data-based telephone-events, as sent in RFC2833 packets. This section explains how the Oracle Enterprise Session Border Controller transcodes between these RTP-based telephone events and in-band DTMF tones carried by G711. DTMF tones can only be transported in non compressed codecs. The Oracle Enterprise Session Border Controller supports two DTMFable non-compressed codecs: PCMU (G711µ) and PCMA (G711A).

Note: The following line is added to SDP whenever telephone-event is added on egress: a=fmtp:101 0-15

The following two scenarios describe when telephone-event to DTMF transcoding takes place:

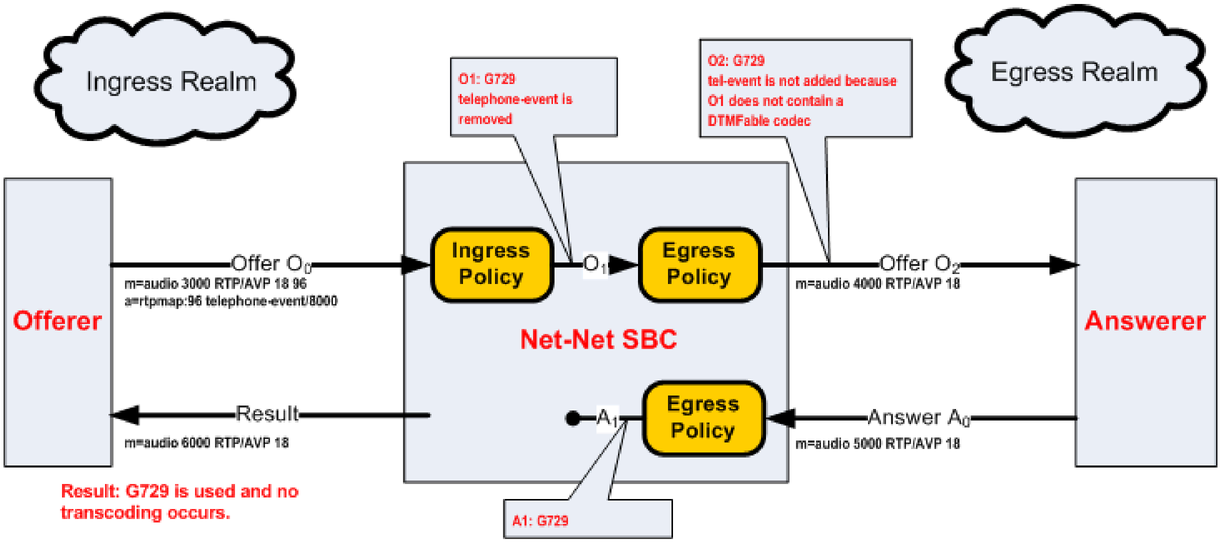
RFC 2833 Scenario 1

The following ingress and egress policies are used for scenario 1.

Ingress Policy		Egress Policy	
allow-codecs	* telephone-event:no	allow-codecs	* PCMA:no
add-codecs-on-egress		add-codecs-on-egress	telephone-event
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	
dtmf-in-audio	preferred	dtmf-in-audio	preferred

1. In the following diagram, telephone event was offered by the offerer but was stripped by ingress policy. telephone-event was not added by the egress policy because the remaining audio codec in O1 was not DTMFable. G729 was the only codec forwarded on to the answerer.

   The SDP answer agreed to use the remaining audio codec, G729. A0 is unaltered by egress policy, and forwarded as the Result to the offerer. Therefore, G729 is used in both the ingress and egress realms, the call does not support RFC 2833, and the call is not transcoded.



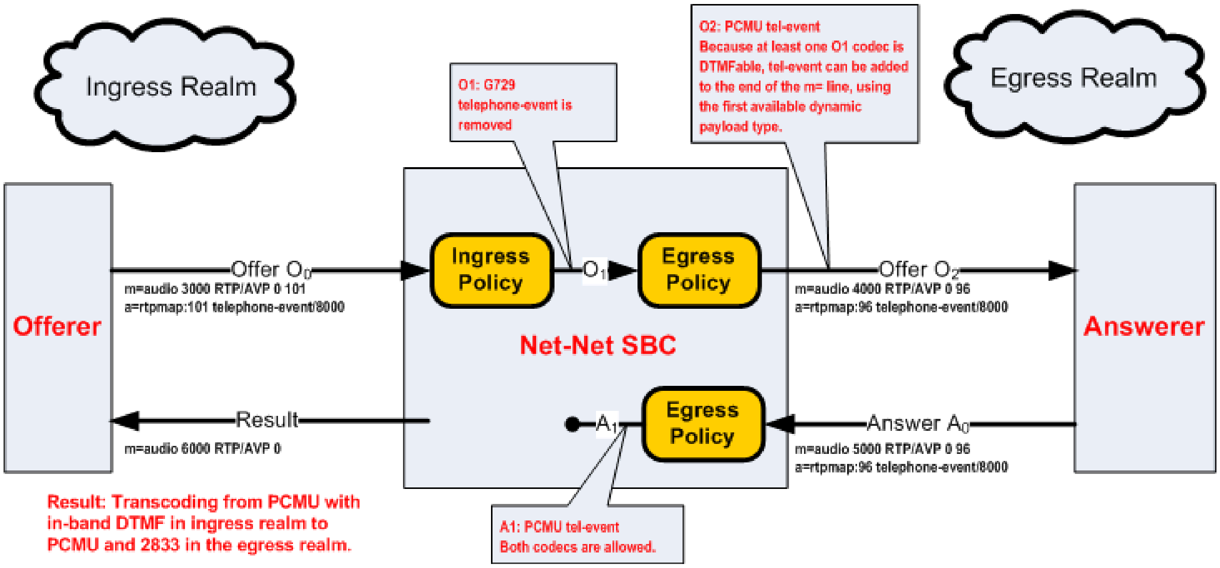
O1: G729
telephone-event is removed

O2: G729
tel-event is not added because O1 does not contain a DTMFable codec

A1: G729

Result: G729 is used and no transcoding occurs.

2. In the following diagram, telephone event was offered by the offerer but was stripped by ingress policy. telephone-event was added by the egress policy because the remaining audio codec in O1 was DTMFable. PCMU and telephone-event are then forwarded on to the answerer. Note that the telephone-event payload type is added with the lowest available dynamic type number.


Ingress Realm
Egress Realm
Offerer
Answerer
Net-Net SBC
Ingress Policy
Egress Policy
Egress Policy
Offer O_0
m=audio 3000 RTP/AVP 0 101
a=rtpmap:101 telephone-event/8000
O_1
Offer O_2
m=audio 4000 RTP/AVP 0 96
a=rtpmap:96 telephone-event/8000
Result
m=audio 6000 RTP/AVP 0
A_1
Answer A_0
m=audio 5000 RTP/AVP 0 96
a=rtpmap:96 telephone-event/8000

O1: G729
telephone-event is removed

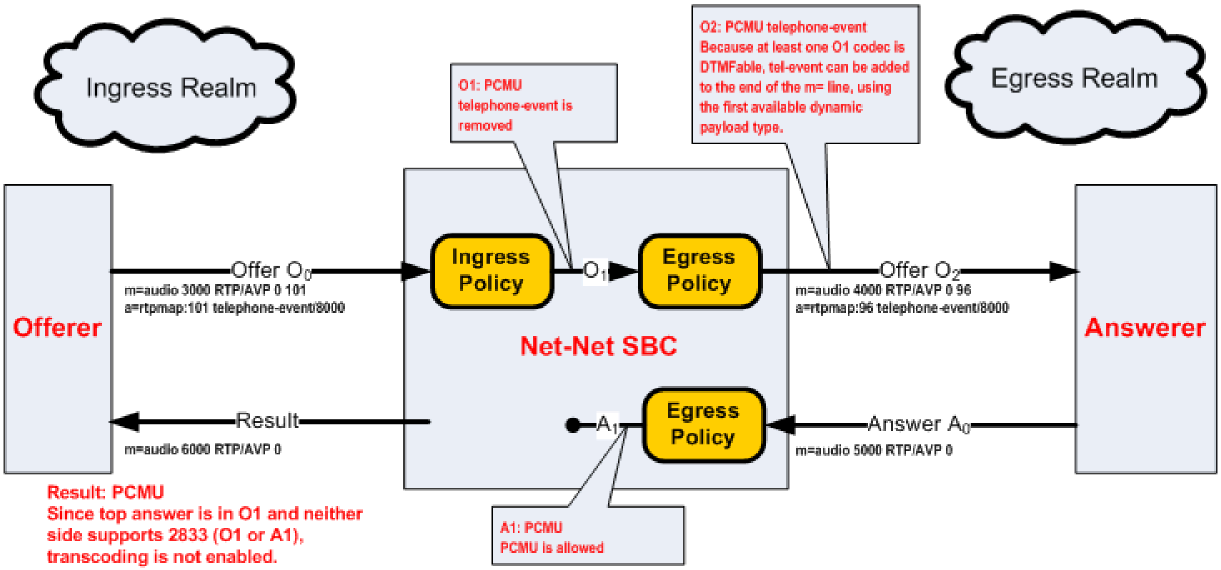
O2: PCMU tel-event
Because at least one O1 codec is DTMFable, tel-event can be added to the end of the m= line, using the first available dynamic payload type.

A1: PCMU tel-event
Both codecs are allowed.

Result: Transcoding from PCMU with in-band DTMF in ingress realm to PCMU and 2833 in the egress realm.

This case illustrates when the answerer supports audio and RFC 2833, but the offerer supports audio with inband DTMF. The Oracle Enterprise Session Border Controller transcodes between RFC2833 in the egress realm to in-band DTMF on the ingress realm.

3. In the following diagram, telephone event was offered by the offerer but was stripped by ingress policy. telephone-event is added by the egress policy because the remaining audio codec in O1 was DTMFable. PCMU and telephone-event are then forwarded on to the answerer. Note that the telephone-event payload type is added with the lowest available dynamic type number.



O1: PCMU
telephone-event is removed

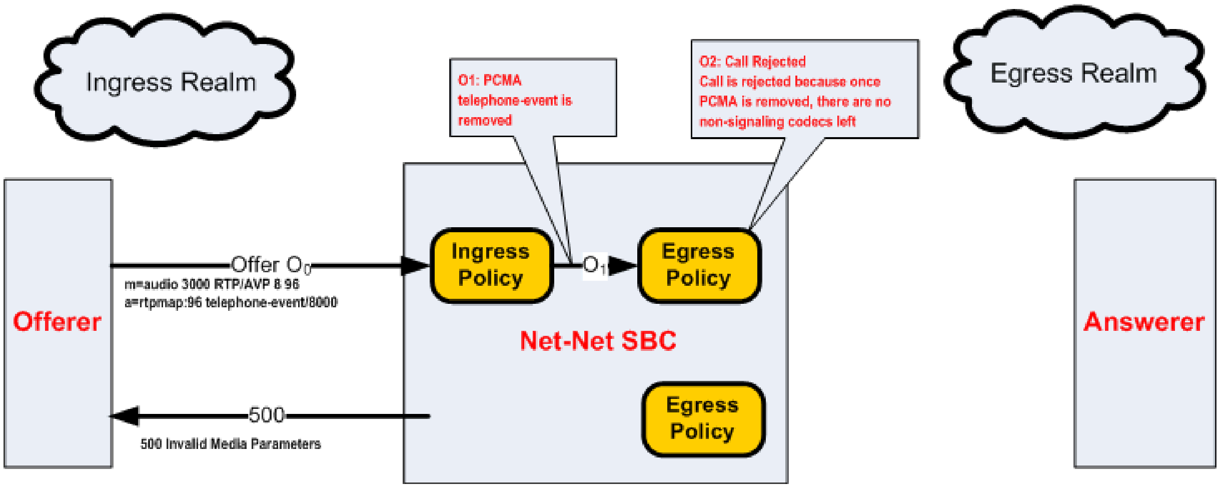
O2: PCMU telephone-event
Because at least one O1 codec is DTMFable, tel-event can be added to the end of the m= line, using the first available dynamic payload type.

A1: PCMU
PCMU is allowed

Result: PCMU
Since top answer is in O1 and neither side supports 2833 (O1 or A1), transcoding is not enabled.

The SDP answer only agreed to use PCMU. When A0 reaches the egress policy, it is passed along through the Oracle Enterprise Session Border Controller to the offerer. Because telephone-event was not answered by the answerer and not supported in O1, it can't be used. Transcoding is therefore not used for this call.

4. In the following diagram, telephone event was offered by the offerer and was stripped by ingress policy. Since PCMA was also stripped by the egress policy, leaving no non-signaling codecs, the call is rejected. A 500 message is sent back to the offerer.



```
m=audio 3000 RTP/AVP 8 96
a=rtpmap:96 telephone-event/8000
```

```
500 Invalid Media Parameters
```

O1: PCMA
telephone-event is
removed

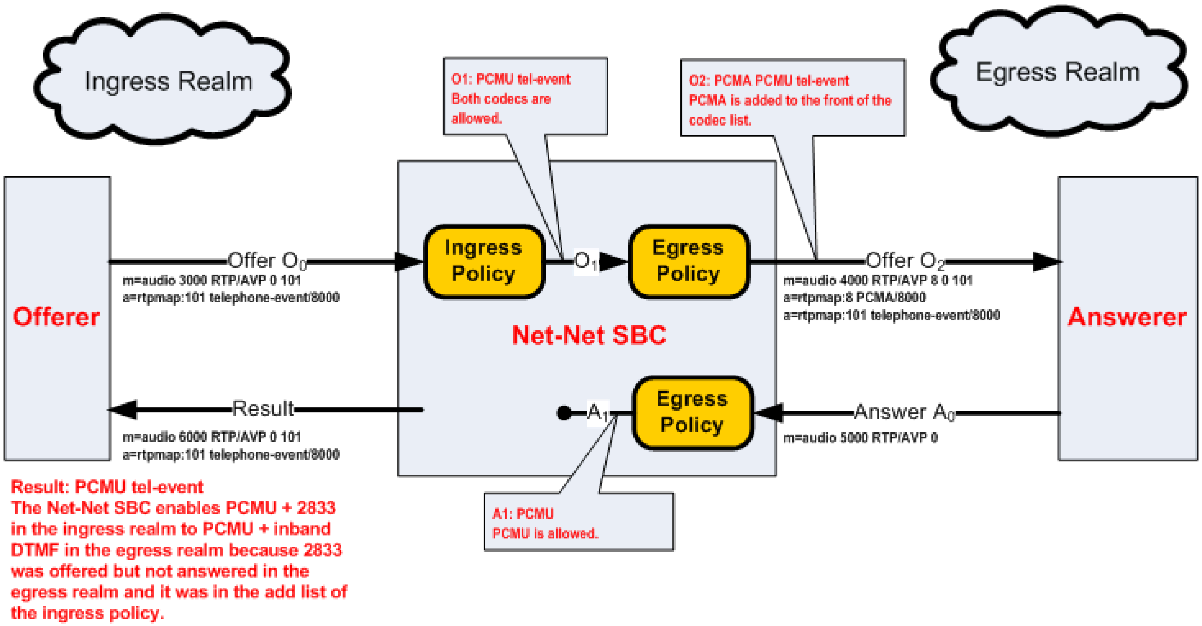
O2: Call Rejected
Call is rejected because once
PCMA is removed, there are no
non-signaling codecs left

RFC 2833 Scenario 2

The following ingress and egress policies are used for RFC2833 scenario 2.

Ingress Policy		Egress Policy	
allow-codecs	*	allow-codecs	*
add-codecs-on-egress	telephone-event	add-codecs-on-egress	PCMU
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	
dtmf-in-audio	preferred	dtmf-in-audio	preferred

1. In the following diagram, telephone event and PCMU are offered by the offerer. They are both passed to O1, and PCMA is added as it is sent to the answerer. The SDP answer, A0 disables all codecs but PCMU.



```
m=audio 3000 RTP/AVP 0 101
a=rtpmap:101 telephone-event/8000
```

```
m=audio 4000 RTP/AVP 8 0 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
```

```
m=audio 6000 RTP/AVP 0 101
a=rtpmap:101 telephone-event/8000
```

```
m=audio 5000 RTP/AVP 0
```

O1: PCMU tel-event
Both codecs are
allowed.

O2: PCMA PCMU tel-event
PCMA is added to the front of the
codec list.

A1: PCMU
PCMU is allowed.

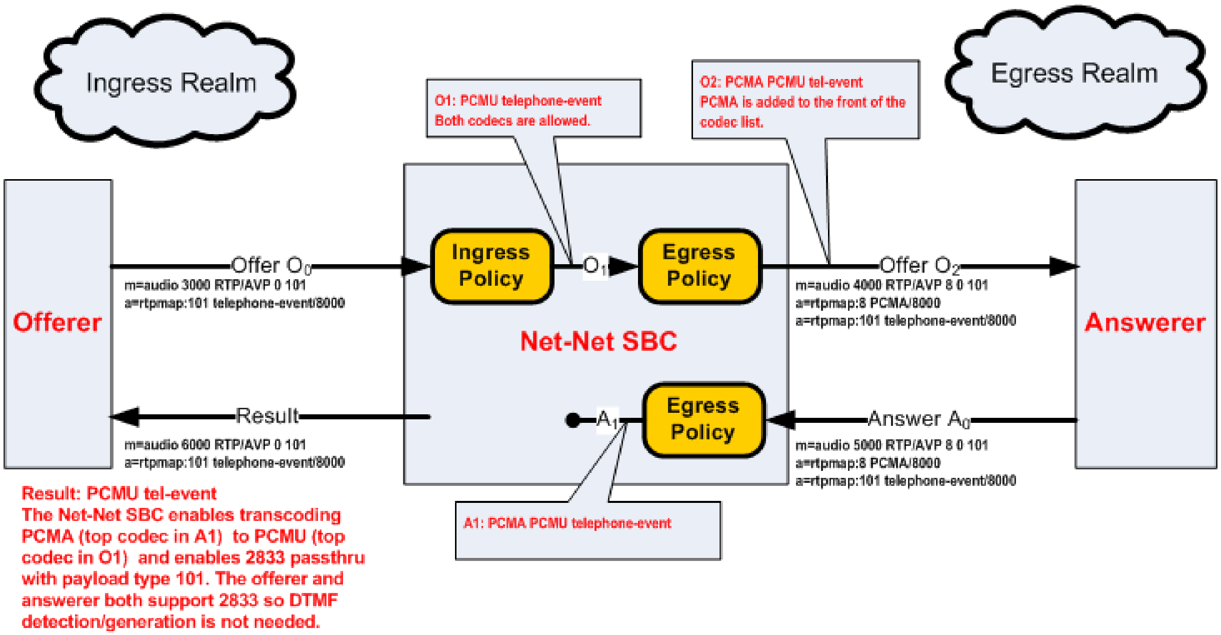
Result: PCMU tel-event
The Net-Net SBC enables PCMU + 2833
in the ingress realm to PCMU + inband
DTMF in the egress realm because 2833
was offered but not answered in the
egress realm and it was in the add list of
the ingress policy.

The Oracle Enterprise Session Border Controller adds telephone-event to the result because it is listed on the ingress policy’s add-codecs-on-egress parameter and present in the offerer’s SDP.

Note: This is the only time the add list of an ingress policy is utilized as a check.

The result SDP includes PCMU and telephone-event in the ingress realm, which is transcoded to PCMU with in-band DTMF in the egress realm.

2. In the following diagram, telephone event and PCMU are offered by the offerer. They are both passed to O1, and PCMA is added as it is sent to the answerer. The SDP answer supports all three codecs offered, with PCMA added on top.


Ingress Realm
Egress Realm
O1: PCMU telephone-event
Both codecs are allowed.
O2: PCMA PCMU tel-event
PCMA is added to the front of the codec list.
Offerer
Offer O_0
m=audio 3000 RTP/AVP 0 101
a=rtpmap:101 telephone-event/8000
Ingress Policy
O_1
Egress Policy
Offer O_2
m=audio 4000 RTP/AVP 8 0 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
Answerer
Net-Net SBC
Result
m=audio 6000 RTP/AVP 0 101
a=rtpmap:101 telephone-event/8000
A_1
Egress Policy
Answer A_0
m=audio 5000 RTP/AVP 8 0 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
Result: PCMU tel-event
The Net-Net SBC enables transcoding PCMA (top codec in A1) to PCMU (top codec in O1) and enables 2833 passthru with payload type 101. The offerer and answerer both support 2833 so DTMF detection/generation is not needed.
A1: PCMA PCMU telephone-event

The answerer responds with PCMA as the preferred codec in A0. The Oracle Enterprise Session Border Controller compares A1 to O1 to make the transcoding decision. PCMA is the top codec in A1 and is transcoded to PCMU, the top codec in O1. Also, because telephone-event is supported by both sides of the call, it is passed through without any transcoding necessary.

This case illustrates when both endpoints are capable of sending and receiving telephone-event. Regardless of whether the audio portion of the call is transcoded, the telephone-event messages are passed through the system untouched, thus not requiring transcoding resources. This is known as telephone-event pass-through.

FAX Transcoding

FAXes are transmitted in a call as either T.30 and T.38 media. T.30 FAX is binary in-band media carried over G.711. The Oracle Enterprise Session Border Controller can transcode between T.38 and a faxable codec. The supported faxable codecs are PCMU and PCMA.

T.30 can only be transported in non-compressed codecs. The two non-compressed codecs supported by the Oracle Enterprise Session Border Controller are PCMU (G711µ) and PCMA (G711A). If a transcoding realm does not support an uncompressed codec, T.30 can not be supported in that realm. Alternatively, G711FB may be allowed specifically for FAX only.

The Oracle Enterprise Session Border Controller uses an internal codec called G711FB (G711 - Fall Back) that is an umbrella codec of all FAXable codecs. G711FB will default to PCMU for the purpose of offering a faxable codec. You can remap G711FB to PCMA by configuring the media-profile for it appropriately. G711FB’s only use is for FAX transcoding.

FAX transcoding is triggered when you configure the add on egress parameter with either T.38 or G711FB. In a FAX scenario, when the codec policy adds either T.38 or G711FB, a new m= line is added to the SDP. When adding T.38, the new m= line specifies the T.38 codec. When adding G711FB, the new m= line specifies PCMU (or alternatively PCMA).

Once added, m= lines can not be deleted in the context of a call. The Oracle Enterprise Session Border Controller maintain all m= lines between itself and an endpoint throughout the course of call. All m= lines not in use can be disabled by setting their receive port to 0, but they can not be removed from the SDP.

Defining G711FB

G711 Fall Back (G711FB) is an internal codec that encompasses PCMU and PCMA for carrying fax information FAXable codecs. The G711FB codec must be configured either way for when the Oracle Enterprise Session Border Controller inserts a FAXable codec in SDP. G711FB is only used for FAX transcoding scenarios.

To define G711 FB, create a media profile configuration element named g711fb and set the payload-type to 0 or 8.

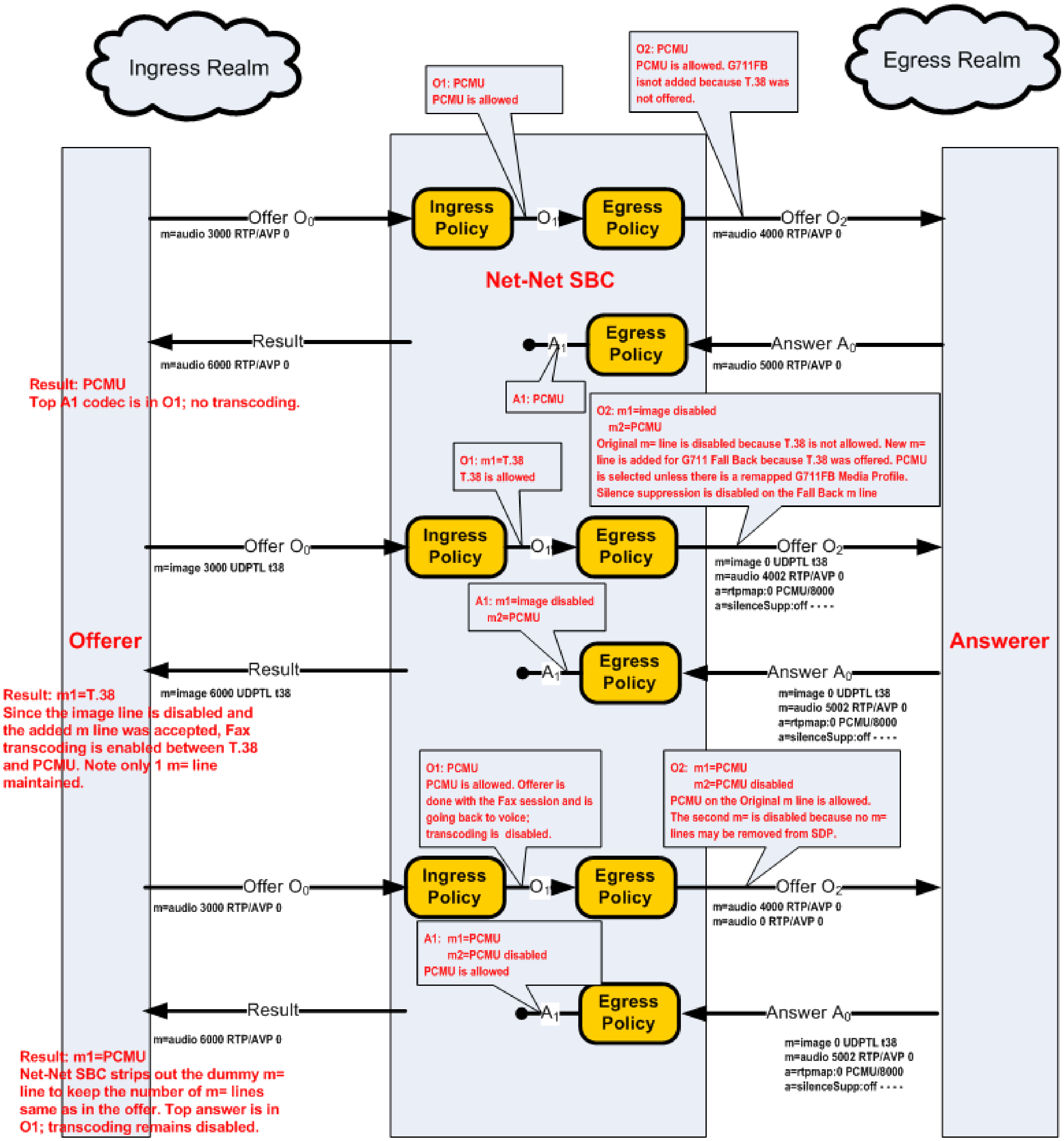
Codec (supported bit rates)	RTP Payload Type	Default Ptime (ms)	Supported Ptime (ms)
T.38	N/A	30	10, 20, 30
G711FB (64 kbps)	0, 8	30	10, 20, 30

FAX Scenario 1

The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		Egress Policy	
allow-codecs	*	allow-codecs	T.38:no
add-codecs-on-egress		add-codecs-on-egress	G711FB
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled
packetization-time		packetization-time	

In the following diagram, there are three offer-answer exchanges. Initially a PCMU-to-PCMU session is negotiated. Next, a T.38 to PCMU session is negotiated. Finally, the session reverts to non-transcoded PCMU to PCMU state.



O1: PCMU
PCMU is allowed

O2: PCMU
PCMU is allowed. G711FB isnot added because T.38 was not offered.

m=audio 3000 RTP/AVP 0

m=audio 4000 RTP/AVP 0

m=audio 6000 RTP/AVP 0

m=audio 5000 RTP/AVP 0

A1: PCMU

Result: PCMU
Top A1 codec is in O1; no transcoding.

O2: m1=image disabled
m2=PCMU
Original m= line is disabled because T.38 is not allowed. New m= line is added for G711 Fall Back because T.38 was offered. PCMU is selected unless there is a remapped G711FB Media Profile. Silence suppression is disabled on the Fall Back m line

O1: m1=T.38
T.38 is allowed

m=image 3000 UDPTL t38

```
m=image 0 UDPTL t38
m=audio 4002 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=silenceSupp:off - - - -
```

A1: m1=image disabled
m2=PCMU

m=image 6000 UDPTL t38

```
m=image 0 UDPTL t38
m=audio 5002 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=silenceSupp:off - - - -
```

Result: m1=T.38
Since the image line is disabled and the added m line was accepted, Fax transcoding is enabled between T.38 and PCMU. Note only 1 m= line maintained.

O1: PCMU
PCMU is allowed. Offerer is done with the Fax session and is going back to voice; transcoding is disabled.

O2: m1=PCMU
m2=PCMU disabled
PCMU on the Original m line is allowed. The second m= is disabled because no m= lines may be removed from SDP.

m=audio 3000 RTP/AVP 0

```
m=audio 4000 RTP/AVP 0
m=audio 0 RTP/AVP 0
```

A1: m1=PCMU
m2=PCMU disabled
PCMU is allowed

m=audio 6000 RTP/AVP 0

```
m=image 0 UDPTL t38
m=audio 5002 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=silenceSupp:off - - - -
```

Result: m1=PCMU
Net-Net SBC strips out the dummy m= line to keep the number of m= lines same as in the offer. Top answer is in O1; transcoding remains disabled.

FAX Scenario 2

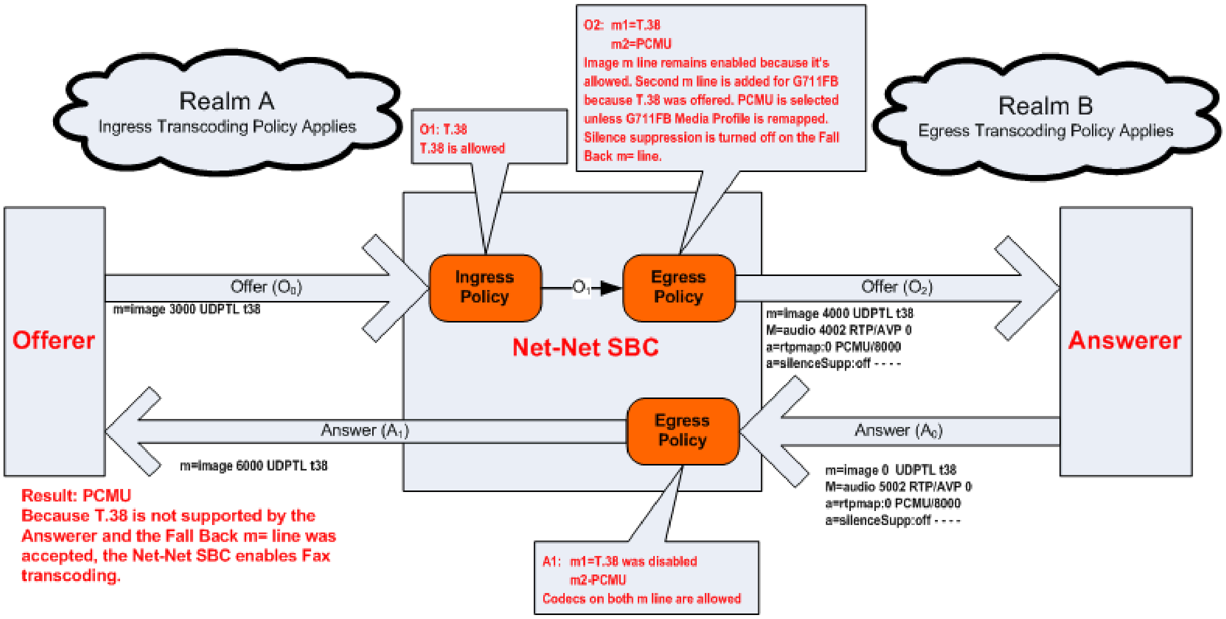
The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		Egress Policy	
allow-codecs	*	allow-codecs	*
add-codecs-on-egress		add-codecs-on-egress	G711FB
order-codecs		order-codecs	
force-ptime	disabled	force-ptime	disabled

Ingress Policy		Egress Policy	
packetization-time		packetization-time	

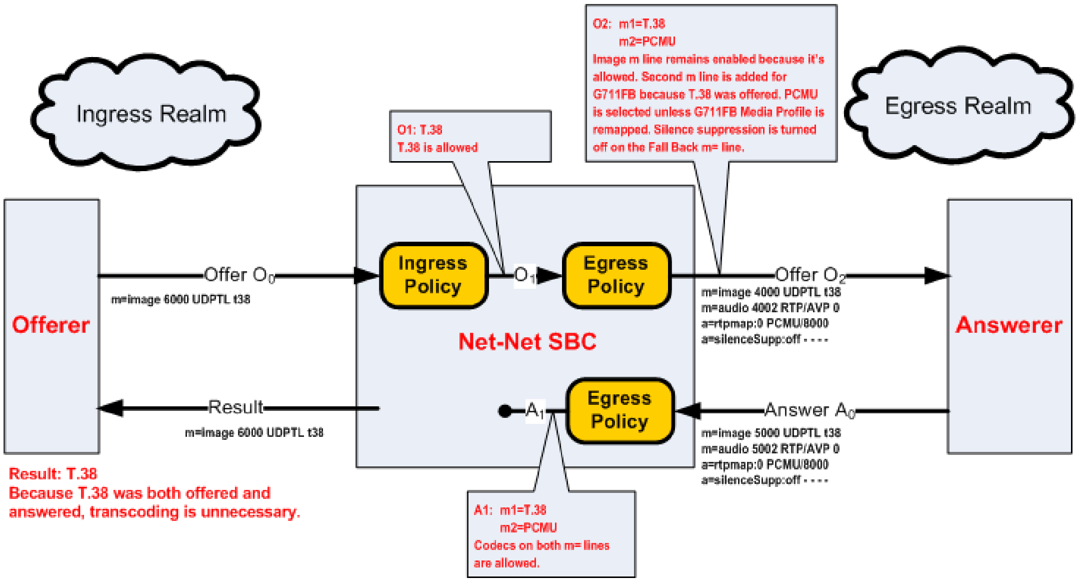
1. In the following diagram, T.38 is offered to the Oracle Enterprise Session Border Controller . A second m= line was added to O1 that included a G711FB codec (PCMU).

   The SDP answer agreed to PCMU, but disabled T.38. When the Oracle Enterprise Session Border Controller forwarded the SDP in A1 to the answerer, it stripped the second m= line. Because A1 rejects T.38 m= line, but accepts the PCMU m= line, FAX transcoding is enabled.


Realm A
Ingress Transcoding Policy Applies
O1: T.38
T.38 is allowed
O2: m1=T.38
m2=PCMU
Image m line remains enabled because it's allowed. Second m line is added for G711FB because T.38 was offered. PCMU is selected unless G711FB Media Profile is remapped. Silence suppression is turned off on the Fall Back m= line.
Realm B
Egress Transcoding Policy Applies
Offerer
Offer (O_0)
m=image 3000 UDPTL t38
Ingress Policy
O_1
Egress Policy
Net-Net SBC
Offer (O_2)
m=image 4000 UDPTL t38
M=audio 4002 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=silenceSupp:off - - - -
Answerer
Answer (A_1)
m=image 6000 UDPTL t38
Egress Policy
Answer (A_0)
m=image 0 UDPTL t38
M=audio 5002 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=silenceSupp:off - - - -
Result: PCMU
Because T.38 is not supported by the Answerer and the Fall Back m= line was accepted, the Net-Net SBC enables Fax transcoding.
A1: m1=T.38 was disabled
m2-PCMU
Codecs on both m line are allowed

2. In the following diagram, T.38 is offered to the Oracle Enterprise Session Border Controller . A second m= line was added to O1 that included a G711FB codec (PCMU).

   The SDP answer agreed to PCMU and T.38. Because both O1 and A1 support T.38, the call proceeds without transcoding.


Ingress Realm
O1: T.38
T.38 is allowed
O2: m1=T.38
m2=PCMU
Image m line remains enabled because it's allowed. Second m line is added for G711FB because T.38 was offered. PCMU is selected unless G711FB Media Profile is remapped. Silence suppression is turned off on the Fall Back m= line.
Egress Realm
Offerer
Offer O_0
m=image 6000 UDPTL t38
Ingress Policy
O_1
Egress Policy
Net-Net SBC
Offer O_2
m=image 4000 UDPTL t38
m=audio 4002 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=silenceSupp:off - - - -
Answerer
Result
m=image 6000 UDPTL t38
A_1
Egress Policy
Answer A_0
m=image 5000 UDPTL t38
m=audio 5002 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=silenceSupp:off - - - -
Result: T.38
Because T.38 was both offered and answered, transcoding is unnecessary.
A1: m1=T.38
m2=PCMU
Codecs on both m= lines are allowed.

Transrating

The Oracle Enterprise Session Border Controller can transrate media as it exits the Oracle Enterprise Session Border Controller into the network. Transrating is also known as forced packetization time (ptime), and is used to enforce a configured ptime within a realm. Transrating is often desirable when devices in a realm can only accept media with a specific ptime, or to optimize bandwidth.

If this feature is configured, the media portion of a call is transrated regardless of which codecs are ultimately chosen for each realm as long as they are transcodable. This allows realms that have devices that can only use a single packetization interval to interwork with devices that may or may not have the same packetization capabilities.

You must enable force-ptime in the egress codec policy and then specify the packetization time to force. When force ptime is enabled, it implicitly masks all codecs not of the specified packetization time that are listed in that codec policy's allow codecs and add codecs on egress parameters. For example, if force ptime is enabled with a packetization time of 20 ms, then no G723 codecs (which are only available at 30 and 60 ms) may be active via codec policy in that realm.

Transrating occurs when forced-ptime is enabled and the offered and answered ptimes do not match and the top non-Signaling codec of A1 and top non Signaling codec of O1 are Transcodable.

Note: Answered ptime A1 does not have to be equal to the ptime inserted into the outgoing offer O2; it just has to be different than the offer the Oracle Enterprise Session Border Controller received (O1).

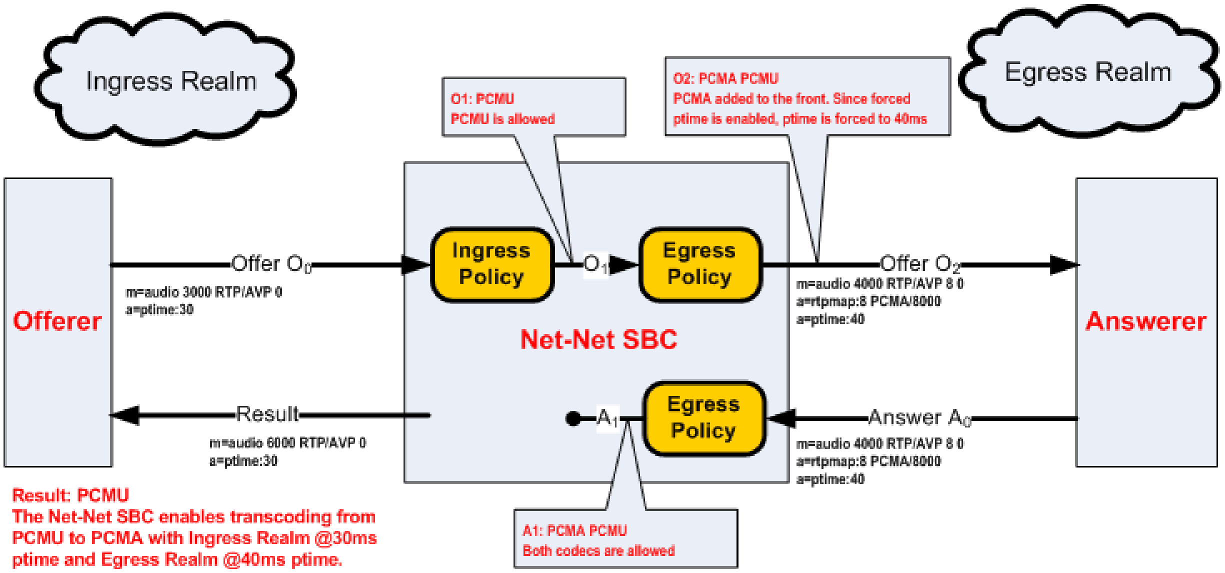
Transrating Scenario 1

The following ingress and egress policies are used for this FAX scenario.

Ingress Policy		**Egress Policy**	
allow-codecs	*	allow-codecs	*
add-codecs-on-egress		add-codecs-on-egress	PCMA
order-codecs	G723 *	order-codecs	
force-ptime	disabled	force-ptime	enabled
packetization-time		packetization-time	40

1. In the following diagram, PCMU is offered in the ingress realm with 30ms ptime, and the egress realm is forced to use 40ms ptime. PCMA is added as the top codec for the egress realm.

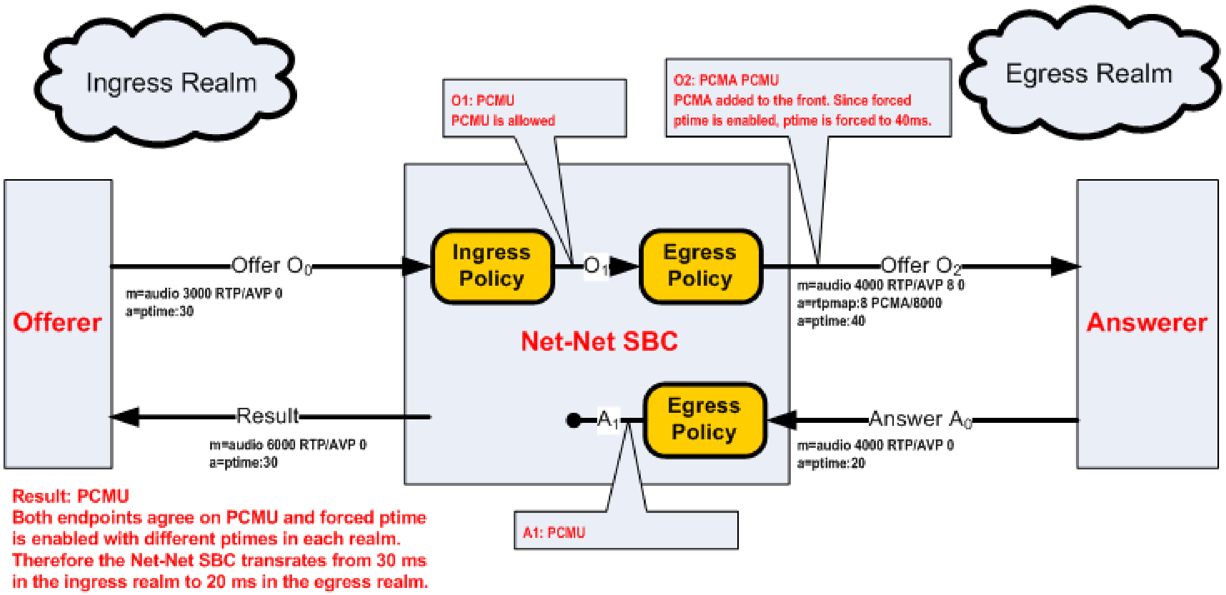
   The Oracle Enterprise Session Border Controller enables transcoding between the ingress realm (PCMU) and the egress realm (PCMA) and the ptimes as negotiated are also maintained.



Result: PCMU
The Net-Net SBC enables transcoding from PCMU to PCMA with Ingress Realm @30ms ptime and Egress Realm @40ms ptime.

2. In the following diagram, PCMU is offered in the ingress realm with a ptime of 30ms, and forced to 40 ms in the egress realm by policy.

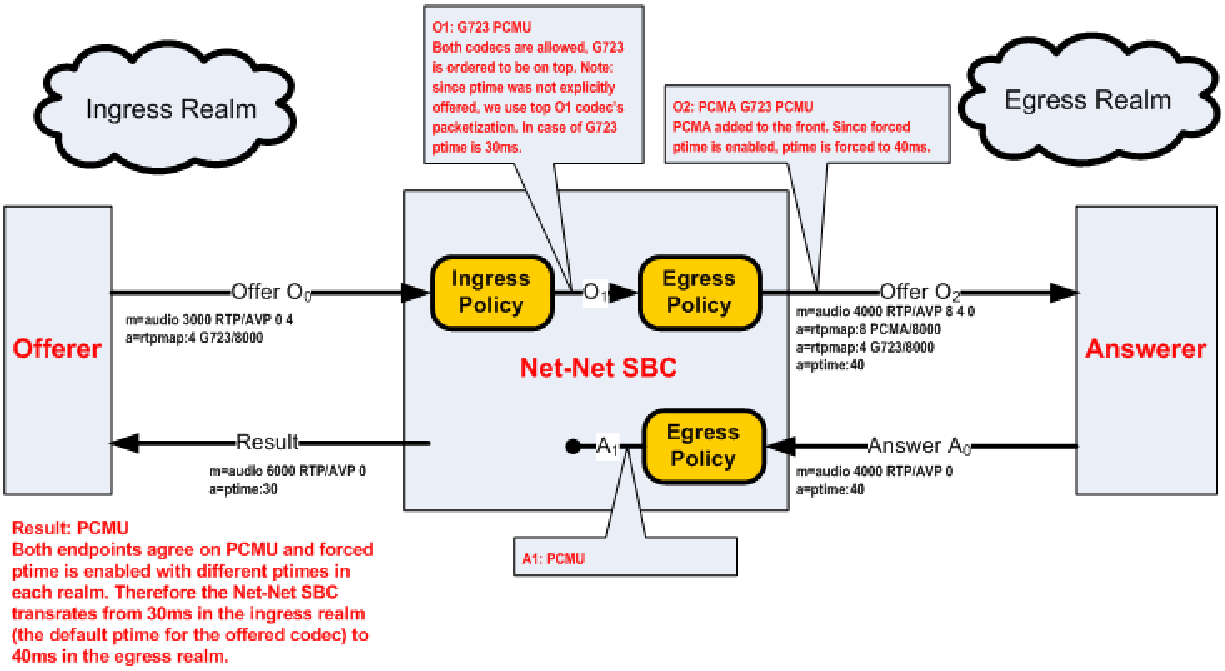
   The answerer chooses to use PCMU with a 20 ms ptime. Thus the call is not transcoded, but it is transrated from 30ms in the ingress realm to 20ms in the egress realm.



Result: PCMU
Both endpoints agree on PCMU and forced ptime is enabled with different ptimes in each realm. Therefore the Net-Net SBC transrates from 30 ms in the ingress realm to 20 ms in the egress realm.

3. In the following diagram, PCMU and G723 are offered in Realm A. The top codec's ptime (30ms) is implied as the one for the ingress realm. The Oracle Enterprise Session Border Controller adds PCMA to the SDP offer with a 40ms ptime.

   The answerer chooses to use PCMU with a 40 ms ptime. Thus the call is transrated from 30ms in the ingress realm to 40ms in the egress realm.



```
m=audio 3000 RTP/AVP 0 4
a=rtpmap:4 G723/8000
```

```
m=audio 4000 RTP/AVP 8 4 0
a=rtpmap:8 PCMA/8000
a=rtpmap:4 G723/8000
a=ptime:40
```

```
m=audio 6000 RTP/AVP 0
a=ptime:30
```

```
m=audio 4000 RTP/AVP 0
a=ptime:40
```

O1: G723 PCMU
Both codecs are allowed, G723 is ordered to be on top. Note: since ptime was not explicitly offered, we use top O1 codec's packetization. In case of G723 ptime is 30ms.

O2: PCMA G723 PCMU
PCMA added to the front. Since forced ptime is enabled, ptime is forced to 40ms.

A1: PCMU

Result: PCMU
Both endpoints agree on PCMU and forced ptime is enabled with different ptimes in each realm. Therefore the Net-Net SBC transrates from 30ms in the ingress realm (the default ptime for the offered codec) to 40ms in the egress realm.

Default Media Profiles

The Oracle Enterprise Session Border Controller contains a set of default media profiles that define characteristics of well-known IANA codecs. You can not view the default media profiles' configurations, but you can override them by configuring identically-named media profile configuration elements.

Transcodable codecs are a subset of the default media profiles which the Oracle Enterprise Session Border Controller can transcode between.

Transcodable Codecs

The following list shows the transcodable codecs which the Oracle Enterprise Session Border Controller can add to SDP. These codecs all reflect default media profiles for their given names.

- PCMU
- PCMA
- G729
- G729A
- iLBC
- telephone-event
- T.38
- G726
- G726-16
- G726-24
- G726-32
- G726-40
- G722
- G723
- GSM
- AMR
- AMR-WB
- G711-FB

When creating an override media profile from the previously listed codec, case is ignored. Also, GSM is GSM-FR.

Transcodable Codec Details

The following table lists the supported codecs, RTP payload number, default ptime, and supported ptimes.

Codec (supported bit rates)	RTP Payload Type	Default Ptime (ms)	Supported Ptime (ms)
G711 PCMU (64 kbps)	0	20	10, 20, 30, 40, 50, 60
G711 PCMA (64 kbps)	8	20	10, 20, 30, 40, 50, 60
G722	9	20	20
G723.1 (5.3/6.4 kbps)	4	30	30, 60
G729/A/B (8 kbps)	18	20	10, 20, 30, 40, 50, 60, 70, 80, 90
G.726-32 (32 kbps)	2, 96-127	20	10,20,30,40,50,60
GSM FR (13 kbps)	3	20	20
G726-16 (16 kbps)	96-127	20	10,20,30,40,50,60
G726-24 (24 kbps)	96-127	20	10,20,30,40,50,60
G726-40 (40 kbps)	96-127	20	10,20,30,40,50,60
iLBC-15.20K (15.2 kbps)	96-127	20	20
iLBC-13.33K (13.33 kbps)	96-127	30	30
AMR (4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.2, 12.2 kbps)	96-127	20	20, 40
AMR-WB (G.722.2) (6.6, 8.85, 12.65, 14.25, 15.85, 18.25, 19.85, 23.05, 23.85 kbps)	96-127	20	20, 40

Preferred Default Payload Type

When the Oracle Enterprise Session Border Controller adds a codec with a dynamic payload type to SDP, it uses the lowest unused payload number. You can configure a preferred payload type for a dynamic codec by creating an override media profile. This makes the Oracle Enterprise Session Border Controller use your preferred payload type for insertion into SDP. If you configure a dynamic codec to use a preferred payload type, and that payload type is already in use, the codec will still be inserted into SDP, but with the first available dynamic payload type.

For example, you create a media profile for telephone-event with a payload type of 101. If telephone-event is added to SDP, and payload type 101 is already in use in the SDP, the Oracle Enterprise Session Border Controller will use the first available payload type in the 96-127 range when adding telephone-event.

Redefining Codec Packetization Time

You can configure a media profile with a packetization time (ptime) that overrides the codec's default ptime. Transcoding functions look up and use default ptimes when not specified in offered or answered SDP. Default ptime for most audio codecs is 20ms; some however are 30ms.

To change the default ptime for a codec, you must create a media profile that overwrites the default ptime parameter with your new packetization time. When SDP is received with no 'a= ptime' attribute or when adding the codec to egress SDP, the newly configured ptime is used.

New default ptime for a media profile is entered by typing "ptime=<x>" in the parameters parameter, where <x> is the new default packetization time.

mptime Support for Packet Cable

The SDP specification lacks the ability to specify unique packetization times per codec when more than one codec is listed in an m= line. The ptime attribute is not related to a specific codec but to the entire m= line. When multiple codecs appear on a single m= line, the PacketCable mptime attribute can specify different packetization times for each codec.

The Oracle Enterprise Session Border Controller adheres to PKT-SP-NCS1.5-I01-050128 and PKT-SP-EC-MGCP-I06-021127 for processing and generating mptime. The mptime line uses an integer to indicate the packetization time for each corresponding codec in the m= line. The dash character, "-", on an mptime line is used for non-packetized codecs, such as CN or telephone-event.

If the Oracle Enterprise Session Border Controller receives an invalid mptime, it is ignored and removed. If a valid mptime is received in the incoming SDP, its values will be used for packetization times of each corresponding codec and a valid mptime line will be sent in the outgoing SDP.

Valid:

```
m=audio 10000 RTP/AVP 0 96 8
a=mptime:20 - 30
a=rtpmap:96 telephone-event/8000
```

Valid: 'ptime' attribute is ignored

```
m=audio 10000 RTP/AVP 0 8
a=mptime:20 30
a=ptime:30
```

Invalid: dash cannot be first mptime value

```
m=audio 10000 RTP/AVP 96 0
a=mptime: - 20
```

When Oracle Enterprise Session Border Controller includes an mptime in an outgoing SDP, it will also always add a ptime attribute with the value of the most preferred codec. This is done to increase the interoperability with devices that do not support mptime.

AMR-NB and AMR-WB Specifications

The Oracle Enterprise Session Border Controller supports Adaptive Multi-Rate Narrow Band & Wide Band codecs. All configurations of this codec, as indicated by SDP, are transcodable except when the following SDP parameters are enabled:

- robust-sorting
- interleaving

When AMR is configured in a codec policy’s add-codecs-on-egress parameter, it is forwarded from the Oracle Enterprise Session Border Controller with the following default settings:

- 12.2 kbps (AMR-NB)
- 23.85 kbps (AMR-WB)
- RTP/IF1 format
- No redundant packets
- bandwidth efficient default payload
- No CRC frame
- 20ms default ptime

Note: AMR and AMR-WB each require a separate license.

Configuring Transcoding

Codec Policy Configuration

Transcoding is configured by creating codec policies and referencing them from a realm configuration.

The parameters that you can configure are name, allow-codecs, add-codecs-on-egress, order-codecs, and ptime. The following section provides brief explanations of how these parameters work, and how you configure each of them.

Note: A single codec policy can be reused for any number of realms.

To access the configuration parameters for codec policies:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter.

   ```
   configure# media-manager
   ```

3. Type **codec-policy** and press Enter.

   ```
   media-manager# codec-policy
   ```

   From this point, you can start configuring your codec policy.

Naming Codec Policies

The codec policy’s name is important not only because it uniquely identifies the policy, but because it is the name you will enter into your realm configuration’s **codec-policy** parameter. It is important to apply the correct policy to the appropriate realm.

To set the codec policy’s name:

name—Set the name for this codec policy, and note it for future reference when you apply codec policies to realms. This parameter is required, and has no default.

```
codec-policy# name private
```

Removing Allowing and Adding Codecs

The Oracle Enterprise Session Border Controller removes and allows codecs using the **allow-codecs** parameter.

- allow-codecs—The **allow-codecs** parameter takes a list of codecs that you want to pass through the Oracle Enterprise Session Border Controller and can explicitly allow them to remain in the SDP for the next step; codecs not matching the items on this list are removed. This parameter is required.
- add-codecs-on-egress—The **add-codecs-on-egress** parameter sets the codecs that the Oracle Enterprise Session Border Controller adds to an offer if that codec is not already there. This parameter applies only to the egress policy.

  For allow-codecs, order-codecs, and add codecs to codec policies:

You can configure and edit these two transcoding parameters as ACLI lists, meaning that there are **add** and **delete** commands associated with each. You type the name of the parameter, choose the operation you want to perform on the list (adding or deleting), and then specify the data that you want to add or remove.

The examples in the procedure that follows show you how to add to the lists you are configuring. To remove items from the **allow-codecs** list, simply replace the **add** command you see in these example with **delete** and the items you want to remove.

If you want to overwrite previous values, you can enter the command, a Space and the items in the list enclosed in quotes (“).

1. **allow-codecs**—Enter a list of codecs that are allowed to pass through the Oracle Enterprise Session Border Controller . To allow all codecs, enter an asterisk (*).

   ```
   codec-policy# allow-codecs *
   ```

   When multiple items are added, enclose them in quotes. For example:

   ```
   codec-policy# allow-codecs G729 G711 AMR
   ```

2. **add-codecs-on-egress**—Enter the codecs that you want added to the SDP offer for the egress codec policy. If you leave this parameter blank, then the Oracle Enterprise Session Border Controller will not add codecs to the SDP answer. This parameter cannot be wildcarded.

   ```
   codec-policy# add-codecs-on-egress G729
   ```

   If you need to modify the list of configured codecs, you must enter the complete list at once.

Ordering Codecs

Codec policy can specify the order that codecs appear in the SDP offer or answer.

To configure an order which codecs appear in the offer:

order-codecs—Enter the order in which you want codecs to appear in the SDP offer or answer. You can enter them in any of the ways described in the preceding explanation.

```
codec-policy# order-codecs G711 * G729
```

Transrating Configuration

The following procedure explains how to configure transrating for a codec policy. This codec policy must be applied as an egress codec policy.

To configure forced ptime for a codec policy:

1. In Superuser mode, type **configure terminal** and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **media-manager** and press Enter.

   ```
   ACMEPACKET(configure)# media-manager
   ACMEPACKET(media-manager)#
   ```

3. Type **network-parameters** and press Enter.

   ```
   ACMEPACKET(media-manager)# codec-policy
   ACMEPACKET(codec-policy)#
   ```

4. If you are adding support for this feature to a pre-existing configuration, then you must select the specific configuration instance, using the ACLI **select** command.

   ```
   ACMEPACKET(codec-policy)# select 1
   ```

   You can now configure forced ptime.

5. **force-ptime**—Set this parameter to **enabled** to enable forced ptime for this codec policy.
6. **packetization-time**—Enter the ptime in ms to use in the realm where this codec policy is active.
7. Save your work using the ACLI **done** command.

Applying a Codec Policy to a Realm

Once you have configured a codec policy, you apply it to a realm by configuration name.

To apply a codec policy to a realm:

1. In Superuser mode, type configure terminal and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type media-manager and press Enter.

```
configure# media-manager
```

3. Type **realm-config** and press Enter.

```
media-manager# realm-config
```

4. **codec-policy**—Enter the name of the codec policy that you want to apply to this realm. This value is the same as the one you entered in the name parameter for the codec policy you want to use for this realm. There is no default for this parameter.

```
realm-config# codec-policy private
```

Media Profile Configuration

Media profiles must be created and then defined when you want to override the Oracle Enterprise Session Border Controller ’s default media profiles.

ACLI Configuration Instructions and Examples

The parameters that you can configure are name, allow-codecs, add-codecs-on-egress, order-codecs, and ptime. The following section provides brief explanations of how these parameters work, and how you configure each of them.

Creating User-Defined Ptime per Codec

To change the Oracle Enterprise Session Border Controller ’s default ptime for a specific codec, you must create a media profile configuration element. In the **parameter** parameter, you set the ptime to the value of your choosing.

Note: The frames-per-packet parameter in the media profile configuration element is NOT used for setting a user defined ptime for that codec.

To configure a new ptime value for a codec:

1. In Superuser mode, type **configure terminal** and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **media-profile** and press Enter.

```
ACMEPACKET(session-router)# media-profile
ACMEPACKET(media-profile)#
```

   If you are adding ptime to a pre-existing media profile, then you must select (using the ACLI select command) the configuration that you want to edit. If you are adding ptime to an undefined media profile, you must create it first.

4. **name**—Type the name of the codec for which you are creating a new default ptime.

```
ACMEPACKET(media-profile)# name pcmu
```

5. **payload-type**—Enter the well-known payload type for this codec.

```
ACMEPACKET(media-profile)# payload-type 0
```

6. **parameter**—Set the ptime by typing **parameter**, a Space, **ptime=**, the new ptime value. Then press Enter. For example:

```
ACMEPACKET(media-profile)# parameter ptime=40
```

7. Save your work using the ACLI **done** command.

Media Type Subnames

You can define multiple versions of a media profile for a single codec by using the subnames feature. You can then reference the new media profile by a combination of the media profile name and media profile subname.

Some media types are not unique per just their value in an SDP m= line, they must be uniquely identified by looking at additional SDP parameters. For example, you can define a media profile for G729, when only the parameter and value **`annexb=yes`** is present in the SDP. By creating a media profile + subname that defines both a media type and parameter, you can perform various operations on G729 only when **`annexb=yes`** is encountered.

Some applications of media type subnames are:

- maintaining different versions of the same codec with different bandwidth ceilings
- maintaining different versions of the same codec with different ptimes
- grouping codecs by using customer as a subname
- grouping codecs by using realm as a subname

SDP Parameter Matching

This feature matches parameters in the **`a=fmtp`**, codec-specific SDP **`a=`** line. It does not try to match a global **`m=`** line attribute like **`a=ptime`**.

Using Subnames with Codec Policies

Media profiles are defined and referenced in the ACLI by a name and subname in the following format

```
<name>::<subname>
```

If no subname has been created for a media profile, you may continue using the media profile name without any subname specifier.

For example, to remove a media profile and subname configured as PCMU::customer1 from all SDP entering the egress realm, you would configure the codec policy **`allow-codecs`** parameter as follows:

```
allow-codecs PCMU::customer1:no
```

Subname Syntax and Wildcarding

You can wildcard one or both portions (name and subname) of a media type and subname pair:

- When you wildcard the **`name`** portion of the value, you can provide a specific subname that the Oracle Enterprise Session Border Controller uses to find matching media profiles.
- When you wildcard the subname portion of the value, you can provide a specific **`name`** that the Oracle Enterprise Session Border Controller uses to find matching media profiles.

The following table defines and explains subname wildcarding and syntax:

Syntax	Example Value	Description
<name>	PCMU	Matches any and all media profiles with the name value configured as PCMU. This entry has the same meaning as a value with this syntax: <name>::*.
<name>::	PCMU::	Matches a media profile with the name with the name value configured as PCMU with an empty subname parameter.
<name>::*	PCMU::*	Matches any and all media profiles with the name value configured as PCMU with any and all subname configured.
<name>::<subname>	PCMU::64k	Matches a media profiles with the name with the name value configured as PCMU with the subname parameter set to 64k.
*	*	Matches anything, but does not have to be a defined media profile.
::	*::*	Matches any and all media profiles, but requires the presence of media profile configurations.

Syntax	Example Value	Description
*::<subname>	*::64k	Matches all media profiles with this subname. You might have a group of media profiles with different names, but the same subname value.
*::	*::	Matches any media profiles with an empty subname parameter.
::	::	Invalid
::*	::*	Invalid

Wildcarding add-codecs-on-egress

It is important to note that you may not configure **add-codecs-on-egress** with a wildcarded subname in a codec policy. You may only add a specific instance of a media type.

Valid:

```
add-codecs-on-egress PCMU
add-codecs-on-egress PCMU::customer1
```

Invalid:

```
add-codecs-on-egress PCMU::*
```

Media Type Subname Configuration

To use media type subnames with a codec policy, you must first configure a media profile and subname. Then you can configure a codec policy with a media type and subname pair for your application

To configure a media type and subname:

1. In Superuser mode, type **configure terminal** and press Enter.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type **session-router** and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```
3. Type **media-profile** and press Enter.

   ```
   ACMEPACKET(session-router)# media-profile
   ACMEPACKET(media-profile)#
   ```
4. **name**—Type the name of the codec for which you are creating a new default ptime.

   ```
   ACMEPACKET(media-profile)# name g729
   ```
5. **subname**—Enter a description for the use of this subname

   ```
   ACMEPACKET(media-profile)# subname annexb-yes
   ```

   You may now configure this subname's unique attributes. PCMU is created with ptime of 30 in this example.
6. **parameter**—Set the ptime by typing **parameter**, a Space, **ptime=**, the new ptime value. Then press Enter. For example:

   ```
   ACMEPACKET(media-profile)# parameter annexb=yes
   ```

   Note: Remember to configure all additional, required media profile parameters, or they will inherit default values.
7. Save your work using the ACLI **done** command.

Codec Policy with Media Type Subname Configuration

To configure a codec policy with a media type with subname:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
ACMEPACKET(configure)# media-manager
```

3. Type **codec-policy** and press Enter.

```
ACMEPACKET(media-manager)# codec-policy
```

4. Use the ACLI **select** command to select a codec policy.

```
ACMEPACKET(codec-policy)# select 1
```

   You may now enter a media profile with subname to any parameter in the codec policy that accepts a media profile.

5. **allow-codecs**—Enter a list of codecs that this codec policy allows or denies from passing through the Oracle Enterprise Session Border Controller . To allow all codecs, enter an asterisk (*).

```
ACMEPACKET(codec-policy)# allow-codecs g729::annexb-yes:no
```

6. Save and activate your configuration.

Maintenance and Troubleshooting

show mbcd errors

The **show mbcd errors** command displays statistics related to MBCD task errors. The following fields are explained:

- XCode Internal Errors—Number of uncategorized errors due to Transcoding session error.
- XCode Alloc Errors—Number of times that buffer allocation failed for transcoding tasks.
- XCode Update Errors—Number of errors encountered when attempting to update an entry in the Transcoding table upon receipt of the first packet for a media flow.
- XCode Delete Errors—Number of errors encountered when attempting to delete an entry in the Transcoding table.
- XCode Over Cap Errors—Number of Transcoding sessions denied once session capacity is reached.
- XCode Over License Cap—Number of Transcoding sessions denied once license capacity is reached.

```
ACMESYSTEM # show mbcd errors
13:22:50-126
MBC Errors/Events              ---- Lifetime ----
                        Recent      Total  PerMax
Client Errors                0          0       0
Client IPC Errors            0          0       0
Open Streams Failed          0          0       0
Drop Streams Failed          0          0       0
Exp Flow Events              0          0       0
Exp Flow Not Found           0          0       0
Transaction Timeouts         0          0       0
Server Errors                0          0       0
Server IPC Errors            0          0       0
Flow Add Failed            180        180     180
Flow Delete Failed           0          0       0
Flow Update Failed           0          0       0
Flow Latch Failed            0          0       0
Pending Flow Expired         0          0       0
ARP Wait Errors              0          0       0
Exp CAM Not Found            0          0       0
Drop Unknown Exp Flow        0          0       0
Drop/Exp Flow Missing        0          0       0
Exp Notify Failed            0          0       0
Unacknowledged Notify        0          0       0
```

```
  Invalid Realm                0          0       0
  No Ports Available           0          0       0
  Insufficient Bandwidth       0          0       0
  Stale Ports Reclaimed        0          0       0
  Stale Flows Replaced         0          0       0
  Telephone Events Gen         0          0       0
  Pipe Alloc Errors            0          0       0
  Pipe Write Errors            0          0       0
  Not Found In Flows           0          0       0
  XCode Internal Errors        0          0       0
  XCode Alloc Errors           0          0       0
  XCode Update Errors          0          0       0
  XCode Delete Errors          0          0       0
  XCode Over Cap Errors      180        180     180
  XCode Over License Cap       0          0       0
  SRTP Capacity Exceeded       0          0       0
```

show xcode api-stats

The **`show xcode api-stats`** command shows the client and server side message counts for the XClient and XServer software components. The main messages are allocate, update, and free of the transcoding resource. The command uses a 100 second window to show recent counts within the sliding window as well as the total and per max (maximum in a sliding window interval). This command is useful for comparing the client and server side counts and seeing where errors may have occurred with the transcoding resources.

```
ACMEPACKET#show xcode api-stats
                      --------- Client --------   --------- Server --------
Message/Event         Recent      Total  PerMax   Recent      Total  PerMax
                      ------  ---------  ------   ------  ---------  ------
Allocs                     0       5197    4897        0       6355    6055
Updates                    0       1776    1676        0        888     788
Frees                      0       6355    6015        0       6355    6015
Error-Allocs               0          0       0        0         45      45
Error-Updates              0          0       0        0        888     888
Error-Frees                0          0       0        0          0       0
Total                      0      13328   12588        0      14531   13791
```

show xcode dbginfo

The debug information command shows the packet API statistics for the host to DSP path. There is one session/ connection opened with each DSP. The command displays the total packet counts as well as the round trip time statistics for the packets. The recent field shows the count since the last time the command was executed

```
ACMEPACKET#show xcode dbginfo
Startup Time    : 2006-09-08 01:11:50.522
Last Clear Time : 2006-09-08 01:11:50.522
Last Read Time  : 2006-09-08 17:14:52.351
Current Time    : 2006-09-08 17:14:52.351
Up Time         : 0 Days, 16 Hours 3 Minutes 2 Seconds
                             -- Life Time --    -- Recent --
PktApiStats:
    OpenConnectionCnt    =       2
    OpenSessionCnt       =       2
    TotalPktSentCnt      =   21051               21051
    TotalPktRecvCnt      =   21041               21041
    TotalPktRecvEventCnt =       0                   0
    TotalPktRecvDataCnt  =       0                   0
    TotalPktRejectCnt    =       5                   5
    TotalPktTimeoutCnt   =       0                   0
    TotalPktInvalidCnt   =       0                   0
    TotalPktDropCnt      =       0                   0
    TotalPktDropEventCnt =       0                   0
    TotalPktDropDataCnt  =       0                   0
    TotalPktLateRspCnt   =       0                   0
    LowestRoundTripMs               =       1
    HighestRoundTripMs              =    2010
    LowestExtractTimeMs             =       1
    HighestExtractTimeMs            =   13320
    HighestTransportRxTimeMs        =       0
    ulHighestTransportNoRxTimeMs    =       0
```

show sipd codecs

The **show sipd codecs <realm ID>** command displays media-processing statistics per SIP traffic. This command displays statistics per realm and requires a realm argument.

Session Based Statistics

Three statistics are session based. They are the transcoded, transrated, and transparent counts.

- transcoded—counts of sessions that use the Transcoding NIU's TCUs to transcode between two or more codes.
- transrated—counts of sessions that use the Transcoding NIU's TCUs to change the packetization interval among dialogs in the session.
- transparent—counts of sessions that require no TCU hardware intervention (all end-to-end media uses the same codec)

A value of "none" which is not counted in the statistics is set when there is no media at all or media is not yet negotiated. Sessions within the same realm are counted only once.

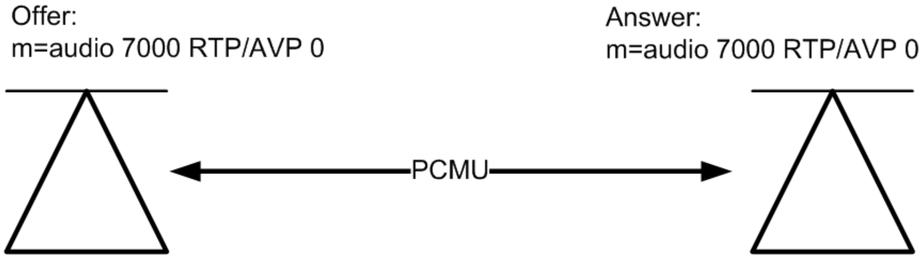
These are meter type counters, and thus have an "active" count as well as total lifetime values. The media-processing state of the session only can increase in precedence (highest=transcoded, transrated, transparent, lowest=none). Thus, if a session begins as transcoded, and then is re-negotiated to transparent later by a re-INVITE, it is still considered transcoded. However, if a session begins as transparent, it can go to transcoded by a re-INVITE. In such a case, the total counts for both transparent and transcoded would be incremented. If there are several media lines, the highest precedence is used for the session.

Flow Based Statistics

The remaining 16 lines of the **show sip codecs** command track the number of codecs in established sessions. The 'Other' type refers to unknown codecs. Only the Recent-Total and other lifetime columns are populated; the Active and Recent High are not applicable. Theses counts represent each SDP m= line emanating in the queried realm. Refer to the following examples:

Example 1

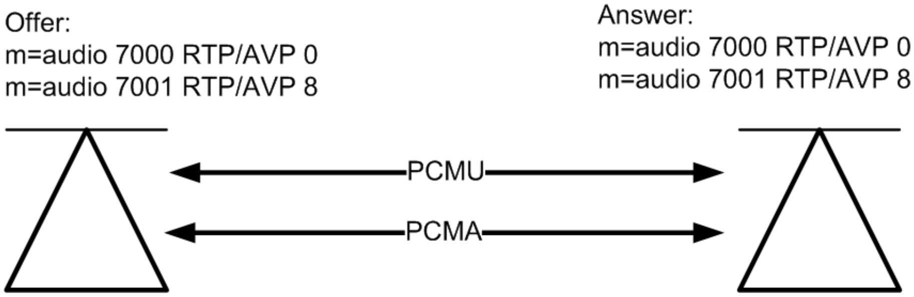
The following diagram shows an intra-realm session with one audio stream using the PCMU codec. Once the session is established, the PCMU count in the **show sip codecs** output is 2.



If the session originator and terminator in the previous diagram exist in two different realms, you must execute the **show sip codecs** command twice, once for each realm. A single PCMU count will be reflected in each respective query because only one m= line emanates from each realm.

Example 2

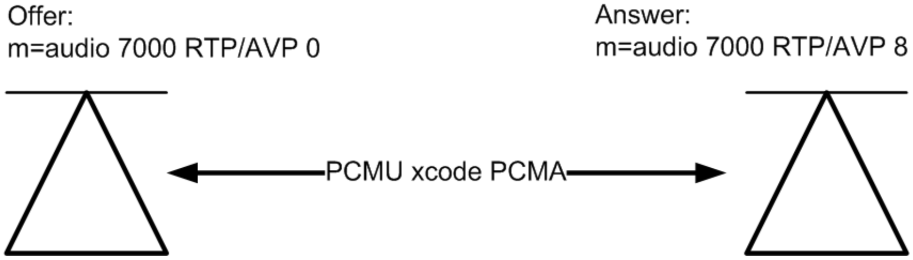
The following diagram shows an intra-realm session with two audio streams. Each stream uses a different codec. Once the session is established, the PCMU count in the **show sip codecs** output is 2, and the PCMA count is 2.



If the session originator and terminator in the previous diagram exist in two different realms, you must execute the **`show sip codecs`** command twice, once for each realm. A single PCMU count and a single PCMA count will be reflected in each respective query because two m= lines emanate from each realm.

Example 3

The following diagram shows an intra-realm transcoding scenario where the originator and terminator are using different audio codecs. The Oracle Enterprise Session Border Controller is performing transcoding functions, which are invisible to the endpoints. Once the session is established, the PCMU count in the **`show sip codecs`** output is 1, and the PCMA count is 1.



If the session originator and terminator in the previous diagram exist in two different realms, you must execute the **`show sip codecs`** command twice, once for each realm. A single PCMU count appears in one query and a single PCMA count appears in the other query because only one m= line emanate from each realm.

An example **`show sip codecs <realm ID>`** command follows:

```
ACMEPACKET# show sipd codecs realm01
18:30:41-595 Realm realm01
Codec Statistics
                          ---- Recent ---- -------- Lifetime --------
                    Active    High   Total      Total  PerMax    High
Transcoded               0       0       0          0       0       0
Transrated               0       0       0          0       0       0
Transparent              0       0       0          0       0       0
PCMU Count               -       -       0          0       0
PCMA Count               -       -       0          0       0
G722 Count               -       -       0          0       0
G723 Count               -       -       0          0       0
G726-16 Count            -       -       0          0       0
G726-24 Count            -       -       0          0       0
G726-32 Count            -       -       0          0       0
G726-40 Count            -       -       0          0       0
G728 Count               -       -       0          0       0
G729 Count               -       -       0          0       0
GSM Count                -       -       0          0       0
iLBC Count               -       -       0          0       0
H261 Count               -       -       0          0       0
H263 Count               -       -       0          0       0
T38 Count                -       -       0          0       0
AMR Count                -       -       0          0       0
AMR-WB Count             -       -       0          0       0
```

```
EVRC Count               -       -       0          0       0
Other Count              -       -       0          0       0
```

show xcode dsp-events

The DSP events command shows all the asynchronous events received from the DSP. These could be anything from CPU monitoring events to digit detection events. Asynchronous error alarms are called alerts and are counted and displayed separately. This command shows the total as well as per DSP event counts and a 10 event trace of the most recent events from the device. Only counters for installed devices are displayed.

```
ACMEPACKET#show xcode dsp-events
displayDspEvents
RX Event Log:
Total Events: 0
Total Undefined Events: 0
Total Alerts: 0
Device: 14
Alert Count: 0
Event Count: 0
Undefined Event Count: 0
Event Trace:
Device: 18
Alert Count: 0
Event Count: 0
Undefined Event Count: 0
```

show xcode load

The **`show xcode load`** command shows the current transcoding module (TCM) load both in number of sessions and percent loading. The load percentage depends on the precise mix of codecs, ptimes, and features enabled on the active sessions. The maximum lifetime load is also displayed. Uninstalled TCMs are marked with dashes.

```
ACMEPACKET#show xcode load
Total Sessions:       0
                    ----- Load -----
         ID  #Sess  Current  Maximum
         ==  =====  =======  =======
TCM    :  0      -       -         -
TCM    :  1      -       -         -
TCM    :  2      -       -         -
TCM    :  3      -       -         -
TCM    :  4      -       -         -
TCM    :  5      -       -         -
TCM    :  6      -       -         -
TCM    :  7      0    0.00%   99.81%
TCM    :  8      -       -         -
TCM    :  9      0    0.00%   99.81%
TCM    : 10      -       -         -
TCM    : 11      -       -         -
```

show xcode session-all

The show xcode session-all command displays all of the currently active sessions by their unique session id.

```
ACMEPACKET#show xcode session-all
15:22:51
Requesting xclient sessions table
        Total Active Sessions: 200
        Displaying sessions 1 to 100:
        Session Id: 0x10007
        Session Id: 0x10008
        Session Id: 0x10009
        Session Id: 0x1000a
        Session Id: 0x1000b
```

Note: When there are more than 100 active sessions , the command now displays only active sessions 1 to 100 as opposed to all the active session:

show xcode session-byid

The session-byid command gives more detailed information about the session. The session-byid command displays the configuration of each channel as well as a number of packet statistics for each channel. This same information can be looked up by IP address and port by using the session-byip command. If only the configuration portion is required, use the session-config command with the session id as the argument. This command is entered as:

```
show xcode     session-byid <session_id>
```

For example:

```
ACMEPACKET#show xcode session-byid 0xf006e
################# SESSION  0xf006e ##################
Channel 0:
  DSP device          = 14
  Source MAC          = 00:08:25:a0:9a:f3
  Destination MAC     = 00:0e:0c:b7:32:e2
  VLAN ID             = 0
  Egress Interface    = 0
  Src IP:Port         = 172.16.0.235:24448
  Dst IP:Port         = 172.16.0.87:16000
  Src RTCP IP:Port    = 172.16.0.235:24449
  Dst RTCP IP:Port    = 172.16.0.87:16001
  Codec               = G711_ULAW_PCM
  Payload Type        = 0
  Pkt Interval        = 20 msec
  2833 Payload Type   = DISABLED
  Xtone Mode          = XTONE_XTHRU
  Status              = DISABLED
DSP Counters:
  RxInPktCnt                  474
  RxInByteCnt                 75840
  RxOutPktCnt                 749
  RxInSidPktCnt               0
  RxNoPktCnt                  275
  RxBadPktTypeCnt             0
  RxBadRtpPayloadTypeCnt      0
  RxBadPktHdrFormatCnt        0
  RxBadPktLengthCnt           0
  RxMisorderedPktCnt          0
  RxBadPktChecksumCnt         0
  RxUnderrunSlipCnt           0
  RxOverrunSlipCnt            0
  RxLastVocoderType           0
  RxVocoderChangeCnt          0
  RxMaxDetectedPdv            168
  RxDecdrRate                 15
  RxJitter:
    CurrentDelay              160
    EstimatedDelay            0
    ClkDriftingDelta          0
    ClkDriftingCorrectionCnt  0
    InitializationCnt         1
  RxCircularBufferWriteErrCnt 0
  RxApiEventCnt               0
  TxCurrentVocoderType        0
  TxInPktCnt                  749
  TxOutPktCnt                 750
  TxOutByteCnt                120000
  TxInBadPktPayloadCnt        0
  TxTimestampGapCnt           0
  TxTdmWriteErrCnt            0
  RxToneDetectedCnt           0
  RxToneRelayEventPktCnt      0
  RxToneRelayUnsupportedCnt   0
  TxToneRelayEventPktCnt      0
  TxApiEventCnt               0
  TxNoRtpEntryPktDropCnt      0
  ConnectionWaitAckFlag       1
  RxMipsProtectionDropCnt     0
  TxMipsProtectionDropCnt     0
Channel 1:
  DSP device          = 14
  Source MAC          = 00:08:25:a0:9a:f4
  Destination MAC     = 00:1b:21:7a:29:b1
  VLAN ID             = 0
  Egress Interface    = 2
  Src IP:Port         = 192.168.0.235:24448
  Dst IP:Port         = 192.168.0.87:32000
  Src RTCP IP:Port    = 192.168.0.235:24449
  Dst RTCP IP:Port    = 192.168.0.87:32001
  Codec               = G729_A
  Payload Type        = 18
  Pkt Interval        = 20 msec
  2833 Payload Type   = DISABLED
  Xtone Mode          = XTONE_XTHRU
  Status              = DISABLED
DSP Counters:
  RxInPktCnt                  748
  RxInByteCnt                 14960
  RxOutPktCnt                 751
  RxInSidPktCnt               0
  RxNoPktCnt                  3
  RxBadPktTypeCnt             0
  RxBadRtpPayloadTypeCnt      0
  RxBadPktHdrFormatCnt        0
  RxBadPktLengthCnt           0
  RxMisorderedPktCnt          0
  RxBadPktChecksumCnt         0
  RxUnderrunSlipCnt           0
  RxOverrunSlipCnt            0
  RxLastVocoderType           6
  RxVocoderChangeCnt          0
  RxMaxDetectedPdv            171
  RxDecdrRate                 15
  RxJitter:
    CurrentDelay              160
    EstimatedDelay            0
    ClkDriftingDelta          0
    ClkDriftingCorrectionCnt  0
    InitializationCnt         1
  RxCircularBufferWriteErrCnt 0
  RxApiEventCnt               0
  TxCurrentVocoderType        6
  TxInPktCnt                  748
  TxOutPktCnt                 748
  TxOutByteCnt                14960
  TxInBadPktPayloadCnt        0
  TxTimestampGapCnt           0
  TxTdmWriteErrCnt            0
  RxToneDetectedCnt           0
  RxToneRelayEventPktCnt      0
  RxToneRelayUnsupportedCnt   0
  TxToneRelayEventPktCnt      0
  TxApiEventCnt               0
  TxNoRtpEntryPktDropCnt       0
  ConnectionWaitAckFlag        0
  RxMipsProtectionDropCnt      0
  TxMipsProtectionDropCnt      0
```

show xcode xlist

The show xcode xlist command displays the TCMs with the number of DSPs on each module, the number of active sessions, and the load percentage. It also displays the state such as Active or Boot Failure. Uninstalled TCMs are indicated by a dash.

```
ACMEPACKET#show xcode xlist
     ID  DSPs  #Sess  Load  State
     ==  ====  =====  ====  ========
TCM:  0     -      -    -
TCM:  1     -      -    -
TCM:  2     -      -    -
TCM:  3     -      -    -
TCM:  4     -      -    -
TCM:  5     -      -    -
TCM:  6     -      -    -
TCM:  7     2      0    0%  Active
TCM:  8     -      -    -
TCM:  9     2      0    0%  Active
TCM: 10     -      -    -
TCM: 11     -      -    -
```

Logs

A new log file named `log.xserv` is output for debugging the transcoding feature on the Oracle Enterprise Session Border Controller. This log records the API between the host software and the DSPs, and any errors that are encountered.

Alarms

The transcoding feature employs several hardware and software alarms to alert the user when the system is not functioning properly or overload conditions are reached.

Name/ID	Severity/ Health Degredation	Cause(s)	Log Message	Traps Generated
No DSPs Installed/65587	Minor/0	A minor alarm is raised when a transcoding capable NIU is installed but no transcoding modules (TCMs) are installed.	No DSP’s present on TNIU	apSysMgmtHardwareErrorTrap
DSP Slot Misconfiguration/ 65588	Warning/0	A warning alarm is triggered if a T-NIU PHY type is detected and the transcoding modules are not installed in consecutive slots, starting in slot 0.	DSP’s need to be installed in consecutive slots lower to highest	apSysMgmtHardwareErrorTrap

Name/ID	Severity/ Health Degredation	Cause(s)	Log Message	Traps Generated
DSP Boot Failure Alarm/65584	Critical/100	A critical alarm is raised when a DSP device (XX) fails to boot properly at system initialization. The DSPs boot from flash so boot failure will be detected only when the DSPs fail to respond to initial Ethernet control messages	DSP#XX Boot Failure!	apSysMgmtHardwareErrorTrap
DSP Comms Timeout Alarm/ 65586	Critical/100	A DSP communications timeout alarm will be raised when a DSP (XX) fails to respond after 2 seconds with 3 retry messages. This condition indicates either the DSP has become unresponsive or the control path to the DSP is not functioning.	DSP Timeout on Device #XX	apSysMgmtHardwareErrorTrap
DSP Alert Alarm/ 65585	Critical/100	A DSP Alert alarm is an asynchronous event indicating a problem with the health of the DSP such as a halted DSP core. This will trigger a critical alarm. The software will attempt to reset the DSP and gather diagnostic information about the crash. This information will be saved in the /logs directory.	DSP Core Halt on Device #XX!	apSysMgmtHardwareErrorTrap
DSP Temperature/ 65583	Clear 85°C Warning 86°C / 5 Minor 90°C / 25	A DSP over-temperature alarm will be raised when a DSP device exceeds the temperature threshold. If the temperature exceeds		apSysMgmtHardwareErrorTrap

Name/ID	Severity/ Health Degredati on	Cause(s)	Log Message	Traps Generated
	Major 95°C/ 50 Critical 100°C/ 100	90°C, a minor alarm will be raised. If it exceeds 95°C, a major alarm will be raised. f the temperature exceeds 100°C, a critical alarm will be raised. The alarm is cleared if the temperature falls below 85°C.		
Transcoding Capacity Threshold Alarm/ 131158	Clear 80% Warning 95%	A warning alarm will be raised when the transcoding capacity exceeds a high threshold of 95%. The alarm will be cleared after the capacity falls below a low threshold of 80%. This alarm warns the user that transcoding resources are nearly depleted. This alarm is not health affecting.	Transcoding capacity at YY (over threhold of 95)	apSysMgmtHardwareErrorTrap
Licensed AMR Transcoding Capacity Threshold Alarm/ 131159	Clear 80% Warning 95%	A warning alarm is triggered if the AMR transcoding capacity exceeds a high threshold of 95%. The alarm clears after the capacity falls below a low threshold of 80%. This alarm is not health affecting.	AMR Transcoding capacity at YY (over threhold of 95)	apSysMgmtHardwareErrorTrap
Licensed AMR-WB Transcoding Capacity Threshold Alarm/ 131160	Clear 80% Warning 95%	A warning alarm is triggered if the AMR-WB transcoding capacity exceeds a high threshold of 95%. The alarm clears after the capacity falls below a low threshold of 80%. This alarm is not health affecting.	AMR-WB Transcoding capacity at YY (over threhold of 95)	apSysMgmtHardwareErrorTrap

Transcoding Capacity Traps

The Oracle Enterprise Session Border Controller sends the apSysMgmtGroupTrap as transcoding capacity nears its limit. This trap is sent and cleared for three conditions:

- Total DSP usage exceeds 95%
- Total AMR sessions exceed 95% of licensed capacity
- Total AMR-WB sessions exceed 95% of licensed capacity

The apSysMgmtGroupTrap contains the condition observed (apSysMgmtTrapType) and the corresponding value reached (apSysMgmtTrapValue).

```
apSysMgmtGroupTrap         NOTIFICATION-TYPE
    OBJECTS        { apSysMgmtTrapType, apSysMgmtTrapValue }
    STATUS           current
    DESCRIPTION
          " The trap will generated if value of the monitoring object
          exceeds a certain threshold. "
    ::= { apSystemManagementNotifications 1 }
```

When the resource usage retreats below a defined threshold, the Oracle Enterprise Session Border Controller sends an apSysMgmtGroupClearTrap.

```
apSysMgmtGroupClearTrap         NOTIFICATION-TYPE
    OBJECTS        { apSysMgmtTrapType }
    STATUS           current
    DESCRIPTION
          " The trap will generated if value of the monitoring object
          returns to within a certain threshold.  This signifies that
          an alarm caused by that monitoring object has been cleared. "
    ::= { apSystemManagementNotifications 2 }
```

The following table summarizes trigger and clear conditions for transcoding capacity alerts as sent in the the apSysMgmtGroupTrap:

Monitored Transcoding Resource	SNMP Object & OID in apSysMgmtTrapType	Trap Sent	Clear Trap Sent
Total DSP Usage	apSysXCodeCapacity 1.3.6.1.4.1.9148.3.2.1.1.34	95%	80%
AMR License Capacity Usage	apSysXCodeAMRCapacity 1.3.6.1.4.1.9148.3.2.1.1.35	95%	80%
AMR-WB License Capacity Usage	apSysXCodeAMRWBCapacity 1.3.6.1.4.1.9148.3.2.1.1.36	95%	80%

The following SNMP Objects are inserted into the apSysMgmtTrapType when sending and clearing a transcoding capacity trap. They are not used anywhere else in the system and return no data when the Oracle Enterprise Session Border Controller is issued an SNMP GET.

Note: The Oracle Enterprise Session Border Controller will return no data if you perform an SNMP GET on apSysXCodeCapacity (1.3.6.1.4.1.9148.3.2.1.1.34), apSysXCodeAMRCapacity (1.3.6.1.4.1.9148.3.2.1.1.35), or apSysXCodeAMRWBCapacity (1.3.6.1.4.1.9148.3.2.1.1.36). These objects are only used when populating transcoding-related SNMP traps.

SNMP

Acme Packet Codec and Transcoding MIB (ap-codec.mib)

The following table describes the SNMP GET query names for the Oracle Codec and Transcoding MIB (ap-codec.mib).

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apCodecMIBObjects (1.3.6.1.4.1.9148.3.7.1)		
Object Identifier Name: apCodecRealmStatsTable (1.3.6.1.4.1.9148.3.7.1.1)		
Object Identifier Name: apCodecRealmStatsEntry (1.3.6.1.4.1.9148.3.7.1.1.1)		
apCodecRealmCountOther	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.1	Count of the SDP media streams received in the realm which negotiated to a codec not defined in this table.
apCodecRealmCountPCMU	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.2	Count of SDP media streams received in the realm which negotiated to the PCMU codec.
apCodecRealmCountPCMA	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.3	Count of SDP media streams reveived in the realm which negotiated to the PCMA codec.
apCodecRealmCountG722	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.4	Count of SDP media streams reveived in the realm which negotiated to the G722 codec.
apCodecRealmCountG723	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.5	Count of SDP media streams reveived in the realm which negotiated to the G723 codec.
apCodecRealmCountG726-16	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.6	Count of SDP media streams reveived in the realm which negotiated to the G726-16 codec.
apCodecRealmCountG726-24	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.7	Count of SDP media streams reveived in the realm which negotiated to the G726-24 codec.
apCodecRealmCountG726-32	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.8	Count of SDP media streams reveived in the realm which negotiated to the G726-32 codec.
apCodecRealmCountG726-40	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.9	Count of SDP media streams reveived in the realm which negotiated to the G726-40 codec.
apCodecRealmCountG728	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.10	Count of SDP media streams reveived in the realm which negotiated to the G728 codec.
apCodecRealmCountG729	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.11	Count of SDP media streams reveived in the realm which negotiated to the G729 codec.
apCodecRealmCountGSM	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.12	Count of SDP media streams reveived in the realm which negotiated to the GSM codec.

SNMP GET Query Name	Object Identifier Name: Number	Description
apCodecRealmCountILBC	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.13	Count of SDP media streams reveived in the realm which negotiated to the iLBC codec.
apCodecRealmCountAMR	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.14	Count of SDP media streams reveived in the realm which negotiated to the AMR codec.
apCodecRealmCountEVRC	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.15	Count of SDP media streams reveived in the realm which negotiated to the EVRC codec.
apCodecRealmCountH261	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.16	Count of SDP media streams reveived in the realm which negotiated to the H261 codec.
apCodecRealmCountH263	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.17	Count of SDP media streams reveived in the realm which negotiated to the H.263 codec.
apCodecRealmCountT38	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.18	Count of SDP media streams reveived in the realm which negotiated to the T.38 codec.
apCodecRealmCountAMRWB	apCopdecRealmStatsEntry 1.3.6.1.4.1.9148.3.7.1.1.1.19	Count of SDP media streams reveived in the realm which negotiated to the AMR-WB codec.
Object Identifier Name: apTranscodingMIBObjects (1.3.6.1.4.1.9148.3.7.2)		
Object Identifier Name: apCodecTranscodingRealmStatsTable (1.3.6.1.4.1.9148.3.7.2.1)		
Object Identifier Name: apTranscodingRealmStatsEntry (1.3.6.1.4.1.9148.3.7.2.1.1)		
apCodecRealmSessionsTransparent	apCodecTranscodingRealmStatsEntry: 1.3.6.1.4.1.9148.3.7.2.1.1.1	Number of sessions in the realm that did not use any DSP resources for transcoding or transrating.
apCodecRealmSessionsTransrated	apCodecTranscodingRealmStatsEntry: 1.3.6.1.4.1.9148.3.7.2.1.1.2	Number of sessions in the realm that had a common codec but used DSP resources to modify packetization rate.
apCodecRealmSessionsTranscoded	apCodecTranscodingRealmStatsEntry: 1.3.6.1.4.1.9148.3.7.2.1.1.3	Number of sessions in the realm that had used DSP resources to transcode between codecs.
Object Identifier Name: apSysMgmtMIBSessionObjects (1.3.6.1.4.1.9148.3.2.1.2)		
Object Identifier Name: apSigRealmStatsTable (1.3.6.1.4.1.9148.3.2.1.2.4)		
Object Identifier Name: apSigRealmStatsEntry (1.3.6.1.4.1.9148.3.2.1.2.4.1)		
apSigRealmStatsRealmName	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.2	Nmae of the realm the following for which the following statistics are being calculated.
apSigRealmStatsCurrentActiveSessionsInbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.3	Number of current active inbouind sessions.

SNMP GET Query Name	Object Identifier Name: Number	Description
apSigRealmStatsCurrentSessionRateInbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.4	Current inbound session rate in CPS.
apSigRealmStatsCurrentActiveSessionsOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.5	Number of current active outbound sessions.
apSigRealmStatsCurrentSessionRateOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.6	Current outbound session rate in CPS.
apSigRealmStatsTotalSessionsInbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.7	Total number of inbound sessions.
apSigRealmStatsTotalSessionsNotAdmittedInbouind	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.8	Total number of inbound sessions rejected due to insufficient bandwidth.
apSigRealmStatsPeriodHighInbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.9	Highest number of concurrent inbound sessions during the period.
apSigRealmStatsAverageRateInbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.10	Average rate of inbound sessions during the period in CPS.
apSigRealmStatsTotalSessionsOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.11	Total number of outbound sessions.
apSigRealmStatsTotalSessionsNotAdmittedOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.12	Total number of outbound sessions rejected due to insufficient bandwidth.
apSigRealmStatsPeriodHighOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.13	Highest number of concurrent outbound sessions during the period.
apSigRealmStatsAverageRateOutbound	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.14	Average rate of outbound sessions during the period in CPS.
apSigRealmStatsMaxBurstRate	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.15	Maximum burst rate of traffic measured during the period (combined inbound and outbound).
apSigRealmStatsPeriodSeizures	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.16	Total number of seizures during the period.
apSigRealmStatsPeriodAnswers	apSigRealmStatsEntry:	Total number of answered sessions during the period.

SNMP GET Query Name	Object Identifier Name: Number	Description
	1.3.6.1.4.1.9148.3.2.1.2.4.1.17	
apSigRealmStatsPeriodASR	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.18	Answer-to-seizure ratio, expressed as a percentage. For example, a value of 90 represents 90% or .90.
apSigRealmStatsAverageLatency	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.19	Average observed one-way signaling latency during the period in milliseconds.
apSigRealmStatsMaxLatency	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.20	Maximum observed one-way signaling latency during the period in milliseconds.
apSigRealmStatsMinutesLeft	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.21	Number of montly-minutes left in the pool per calendar month for a given realm.
apSigRealmStatsMinutesReject	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.22	Peg counts of number of rejected calls due to monthly-minutes constraints exceeded.
apSigRealmStatsShortSessions	apSigRealmStatsEntry: 1.3.6.1.4.1.9148.3.2.1.2.4.1.23	Lifetime number of sessions whose duration was less than the configured short session duration.

Acme Packet System Management MIB (ap-smgmt.mib)

The following VARBINDs are used in Transcoding related traps. They may not be polled and retrieved using an SNMP GET.

SNMP Object Name	Object Identifier Name: Number	Description
Object Identifier Name: apSysMgmtMIBObjects (1.3.6.1.4.1.9148.3.2.1)		
Object Identifier Name: apSysMgmtGeneralObjects (1.3.6.1.4.1.9148.3.2.1.1)		
apSysXCodeCapacity	apSysMgmtGeneralObjects 1.3.6.1.4.1.9148.3.2.1.1.34	Percentage of Digital Signal Processor (DSP) transcoding utilization (trap generated above 95% and cleared below 80%)
apSysXCodeAMRCapacity	apSysMgmtGeneralObjects 1.3.6.1.4.1.9148.3.2.1.1.35	Percentage of licensed Adaptive Multi-Rate (AMR) transcoding utilization (trap generated above 95% and cleared below 80%)
apSysXCodeAMRWBCapacity	apSysMgmtGeneralObjects	Percentage of licensed Adaptive Multi-Rate

SNMP Object Name	Object Identifier Name: Number	Description
	1.3.6.1.4.1.9148.3.2.1.1.36	Wideband (AMR-WB) transcoding utilization (trap generated above 95% and cleared below 80%)
apSysXCodeEVRCCapacity	apSysMgmtGeneralObjects 1.3.6.1.4.1.9148.3.2.1.1.39	Percentage of Enhanced Variable Rate Codec (EVRC) transcoding utilization (trap generated above 95% and cleared below 80%)
apSysXCodeEVRCBCapacity	apSysMgmtGeneralObjects 1.3.6.1.4.1.9148.3.2.1.1.40	Percentage of licensed Enhanced Variable Rate Code B (EVRC-B) transcoding utilization (trap generated above 95% and cleared below 80%)

2

DTMF Transfer and Support

DTMF Interworking

Multimedia devices and applications can exchange user-input DTMF information end-to-end over IP networks. The Oracle Enterprise Session Border Controller provides the capabilities required to interconnect networks and devices that use different DTMF indication signaling protocols.

DTMF Indication

There are three ways to convey DTMF information for packet-based communications:

- DTMF audio tones: DTMF digit waveforms are encoded inline with voice packets. This method only works with uncompressed audio codecs like G.711. Compressed audio codecs like G.729 and G.723 are incompatible with DTMF audio. DTMF audio is also referred to as in-band tones.
- Out-of-band signaling events:

  SIP INFO messages with Content-Type: application/dtmf-relay define out-of-band signaling events for transmitting DTMF information. SIP INFO messages separate DTMF digits from the voice stream and send them in their own signaling message.
- RTP named telephony events (NTE): RFC 2833 telephone-events are a standard that describes how to transport DTMF tones in RTP packets according to section 3 of RFC 2833. Of the three RTP payload formats available, the Oracle Enterprise Session Border Controller supports RTP NTE.

RFC 2833 telephone-event

RFC 2833 specifies a way of encoding DTMF-indications in RTP media streams. It does not encode the audio of the tone itself, instead data represents the sent tone. RFC 2833 can be used with SIP.

RFC 2833 defines the format of NTE RTP packets used to transport DTMF digits and other telephony events between two peer endpoints. With the NTE method, the endpoints perform per-call negotiation of the DTMF transfer method. They also negotiate to determine the payload type value for the NTE RTP packets.

The NTE payload takes the place of codec data in a standard RTP packet. The payload type number field of the RTP packet header identifies the contents as 2833 NTE. The payload type number is negotiated per call. The local device sends the payload type number to use for RFC 2833 packets using SDP, which tells the other side what payload type number to use when sending the named event packets to the local device. Most devices use payload type number 101 for RFC 2833 packets, although no default is specified in the standard.

The RFC 2833 packet’s RTP header also makes use of the timestamp field. Because events often last longer than the RFC 2833 packets sending interval, the timestamp of the first 2833 packet for an event represents the beginning reference time for subsequent RFC 2833 packets for that same event. For events that span multiple RTP packets, the RTP timestamp identifies the beginning of the event. As a result, several RTP packets might carry the same timestamp.

SIP INFO Messages

SIP INFO messages can send indications of DTMF audio tones between peers as part of the signaling path of the call. Upon receipt of a SIP INFO message with DTMF content, the gateway generates the specified DTMF tone on the receiving end of the call.

DTMF Transfer Processing Overview

Enabling 2 UAs to communicate different DTMF indications with each other is facilitated in two steps. First, the Oracle Enterprise Session Border Controller takes an active role in the SDP negotiation between two UAs, this is the capability negotiation step. You can configure your system to suggest the DTMF indication methods that each endpoint can and can not use.

The second step is translation evaluation. After the SDP negotiation is complete, based on system configuration and the media support on each call leg, the Oracle Enterprise Session Border Controller performs DTMF indication conversion or passive forwarding of DTMF indication messages between UAs.

Capability Negotiation

SDP capability negotiation is the first phase of enabling DTMF transfer. The completion of the SDP offer/answer exchange yields a set of supported codecs between each UA and the Oracle Enterprise Session Border Controller .

For DTMF transfer consideration, SDP manipulation is directed by parameters set in one of three configuration elements:

- codec policy
- signaling interface’s RFC 2833 mode
- session agent’s RFC 2833 mode

The Oracle Enterprise Session Border Controller performs SDP manipulation (addition, removal, or modification of supported codecs) toward the called party first by any applicable codec policies. If one or more of the actions which define telephone-event Modification by Codec Policy occurs, then SDP manipulation is only performed by the codec policy configuration, not by RFC 2833 mode parameters in the signaling interface or session agent.

If a codec policy attached to either the ingress or egress realm triggers SDP manipulation, then the other realm uses codec policy for any telephone-event SDP manipulation; none of the RFC 2833 Mode configurations are used for SDP manipulation.

Note: If the call does not trigger the evaluation of any codec policies, all DTMF transfer processing is only subject to RFC 2833 Mode rules.

If none of the telephone-event Modification by Codec occur, then the Oracle Enterprise Session Border Controller performs SDP manipulation according to the RFC 2833 Mode parameters in the signaling interface or session agent.

SDP Manipulated by Codec Policy

When a call is received, any applicable codec policies are applied and evaluated. If telephone-event SDP is modified by codec policies, then SDP manipulation by RFC 2833 Mode is not performed for either side of the call.

telephone-event Modification by Codec Policy

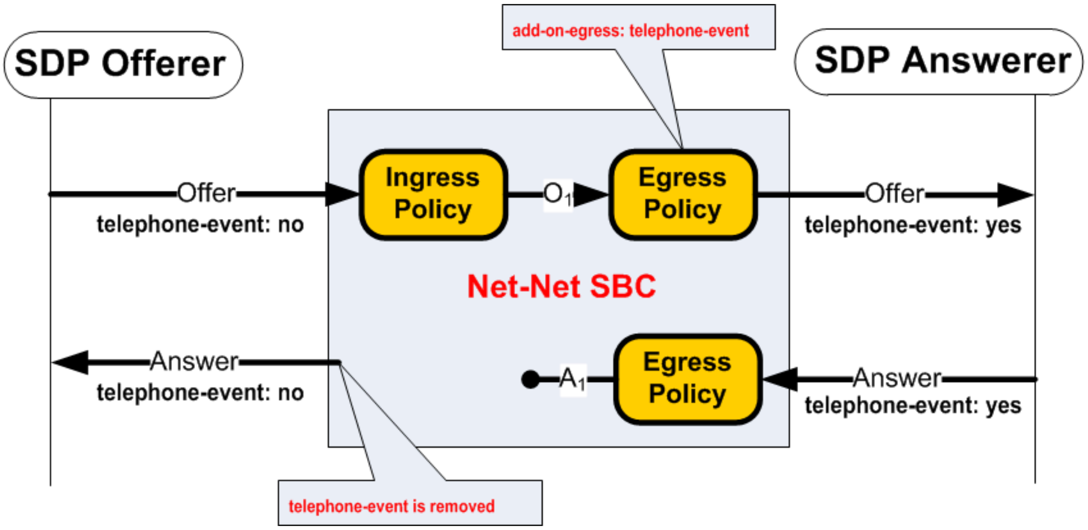
For qualifying if telephone-event modification in SDP was performed by codec policy, one of the following events had to have happened:

- Codec policy explicitly deleted telephone-event by configuring **`allow telephone-event:no`**

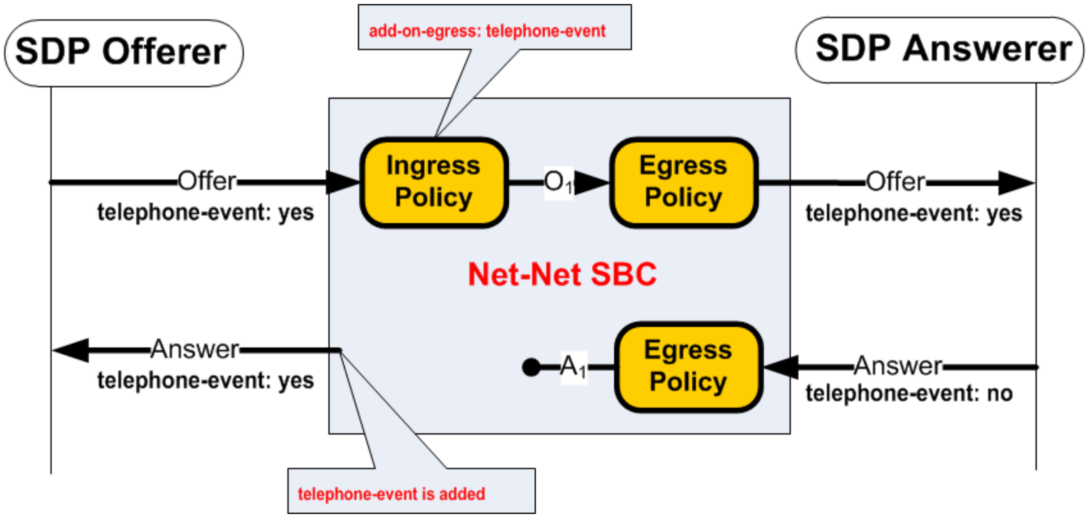
- Codec policy explicitly added telephone-event by configuring `add-on-egress telephone-event`
- Codec policy implicitly denied telephone-event by allowing one or more codecs but not adding telephone-event to the allow list
- Codec policy has `audio:no` configured in the allow list

The following three cases highlight how codec policies can manipulate telephone-event SDP. Once any of these cases occurs, SDP manipulation by RFC 2833 Mode parameter will not be performed.

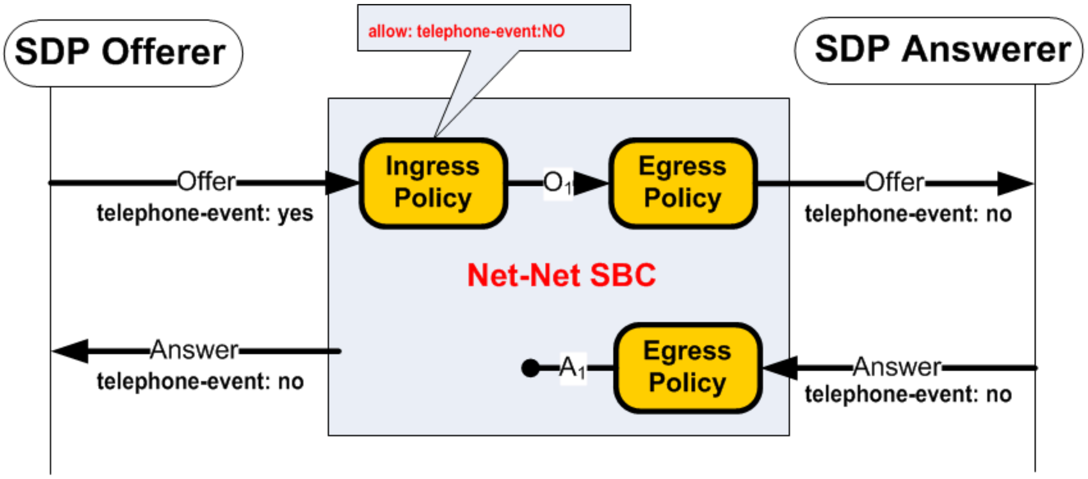
1. telephone-event added to SDP: The codec policy adds telephone-event to the SDP sent to the egress realm. The UA supports telephone-event.



2. telephone-event maintained in SDP: The codec policy maintains the offered telephone-event in the SDP sent into the egress realm. Although telephone-event was not answered in the egress realm, telephone-event is added back to the offerer-side SDP because of the add-on-egress setting in the ingress realm.



3. telephone-event removed from SDP: codec policy removes telephone-event from the initial SDP offer. telephone-event is not forwarded to the Answerer, and subsequently not returned from the answerer, or forwarded again to the offerer.



SDP Manipulated by RFC 2833 Mode

If none of the telephone-event Modification by Codec Policy events occur, as previously explained, SDP may still be modified by RFC 2833 Mode parameter if preferred or dual mode is configured.

The RFC 2833 mode parameter functions similarly to the add-on-egress parameter; it suggests telephone-event support, by adding it in SDP if not already there. This parameter must be set to preferred or dual to add telephone-event to SDP.

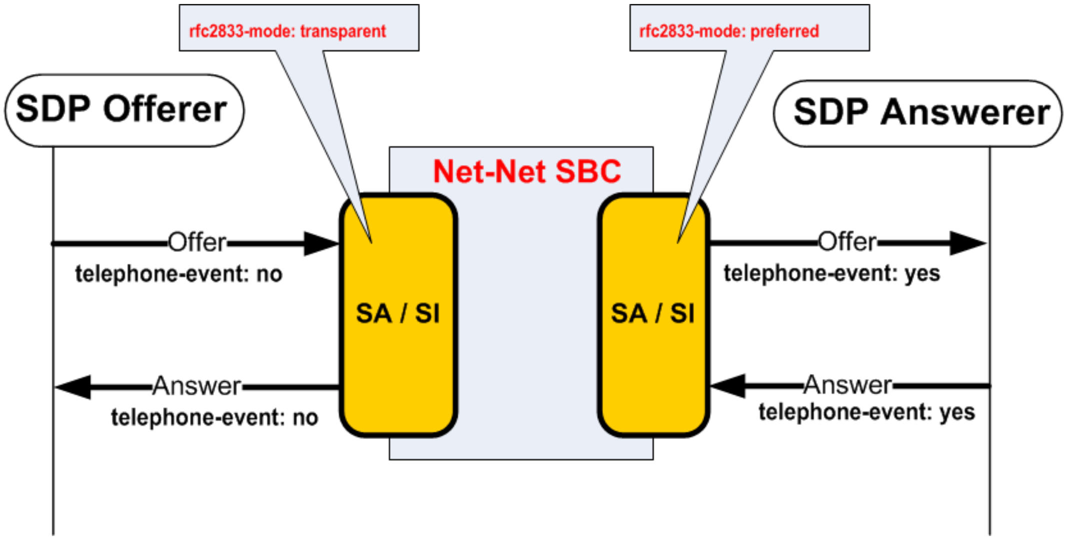
Transparent RFC 2833 Support

Setting a signaling interface or session agent's RFC 2833 mode to transparent disables the addition of RFC 2833 telephone-event to SDP upon egress. The Oracle Enterprise Session Border Controller passes the offered SDP capabilities to the next-hop signaling element.

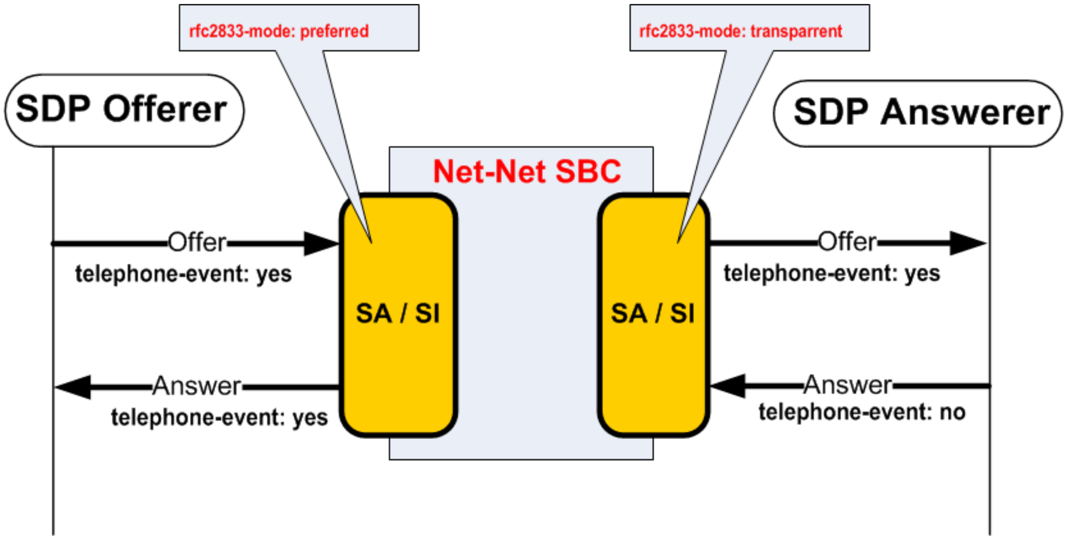
Preferred RFC 2883 Support

Setting a signaling interface or session agent's RFC 2833 mode to preferred indicates that the RFC 2833 telephone-event DTMF transfer method is the preferred method for sending a DTMF indication. In the capability negotiation phase a telephone-event media type will be inserted in the outgoing SDP offer, if it was not present in the original offer.

1. In the following example RFC 2833 mode is set to preferred on the egress side of the call. Because there is no telephone-event in the SDP, and RFC 2833 mode is set to preferred, the Oracle Enterprise Session Border Controller adds telephone-event to the SDP offer.



2. In the following example, RFC 2833 is set to preferred mode on the SDP offer side of the call. The Oracle Enterprise Session Border Controller maintains the telephone-event support even though telephone-event is not supported on the SDP answerer's side of the call.



RFC 2833 Payload Type Mapping

The Oracle Enterprise Session Border Controller does not require that call legs use the same media type for telephone-event. If each call leg uses a different media type value, the Oracle Enterprise Session Border Controller facilitates payload type mapping to ensure the telephone-event media stream be reliably transported across the call.

- On the SDP offer side, when the Oracle Enterprise Session Border Controller returns its SDP answer, it uses the same media type that the SDP offerer offered.
- The Oracle Enterprise Session Border Controller forwards the originally offered telephone-event media type to the SDP answerer. If telephone-event was added by RFC 2833 mode, the Oracle Enterprise Session Border Controller adds telephone-event with the media type value configured in the RFC 2833 payload parameter. If telephone-event was added by a codec policy, the Oracle Enterprise Session Border Controller adds telephone-event with the media type value configured in the media profile.
- If the SDP answerer returns a new value for telephone-event, the Oracle Enterprise Session Border Controller still supports RFC 2833 on that side of the call and uses the media type that the answerer sent.

Translation Evaluation

After SDP has been negotiated, the Oracle Enterprise Session Border Controller determines what types of DTMF translation takes place for the call. The Oracle Enterprise Session Border Controller sequentially evaluates the following rules for each call leg to determine what DTMF indication type it will forward to an endpoint.

1. RFC 2833—When the SDP offer/answer exchange resolves to both the Oracle Enterprise Session Border Controller and the endpoint supporting RFC 2833 on one side of the call, the Oracle Enterprise Session Border Controller will send DTMF indications in RFC 2833 format.
2. DTMF audio tones—Three conditions must be met for the Oracle Enterprise Session Border Controller to support DTMF audio tones, as transcoded from another DTMF indication form:
   - The applicable codec policy's dtmf in audio parameter is set to preferred
   - The endpoint and Oracle Enterprise Session Border Controller have negotiated to a DTMFable codec (G711)
   - Transcoding resources are available

   Note: Because of rule number one, rule number two can not happen if RFC 2833 is supported in SDP —Only one media-based DTMF transfer method, RFC 2833 or DTMF audio tones may be used on a call leg.

3. If neither RFC 2833 nor DTMF Audio tones are supported on a call leg, as a result of SDP negotiation, then the Oracle Enterprise Session Border Controller forwards DTMF indication messages to that side in signaling message format (SIP INFO).

   In the following images that illustrate DTMF transfer scenarios, a gears icon appears when relevant. This icon indicates that the Oracle Enterprise Session Border Controller performs DTMF indication processing, creating DTMF audio tones or RFC 2833 telephone-event messages from another form of DTMF indication.

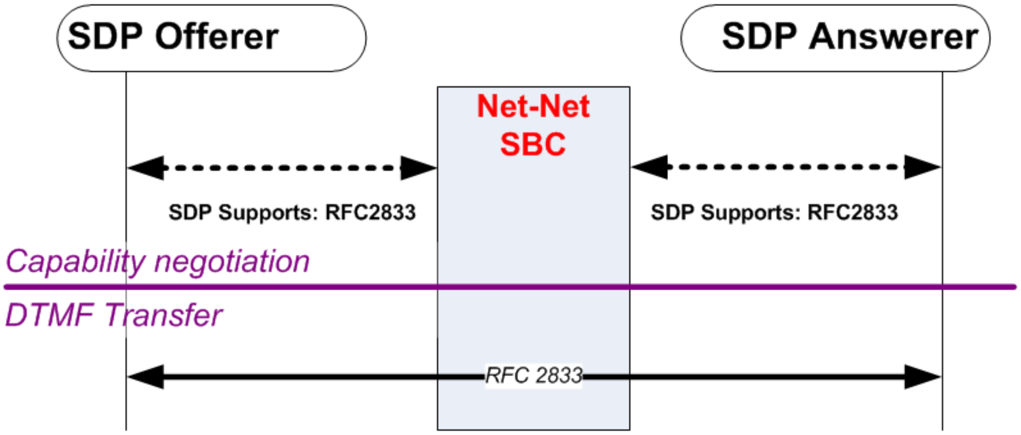
RFC 2833 Sent by Offerer

In the following three examples, the SDP offerer sends DTMF indication messages in RFC 2833 format. The SDP answerer can receive DTMF indications in the format identified in each example.

RFC 2833 to RFC 2833

When the SDP offer and answer sides of a call both support RFC 2833, the Oracle Enterprise Session Border Controller forwards RFC 2833 messages between both sides of the call. No processing is used to transform these DTMF-indication massages to another format.

Note: When the audio stream is transcoded, DTMF audio is completely removed from the audio stream.



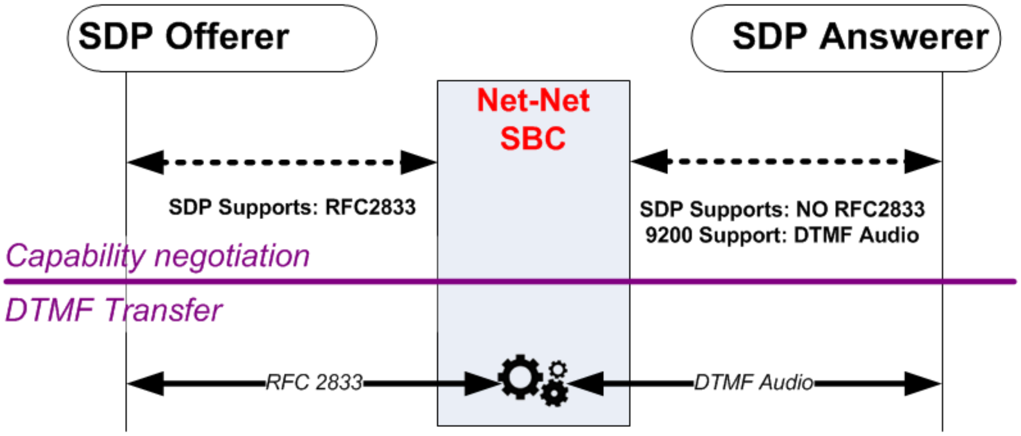
A SIP INFO message received from either the offerer or answerer is forwarded unconverted to the other side of the call.

If there is no audio transcoding enabled for this call, and the egress side is set to dual, a received SIP INFO message will not be converted to both RFC 2833 and SIP INFO messaged for sending to the other side of the call.

If DTMF audio tones are received from either the offerer or answer, they are forwarded unconverted to the other side (when the audio portion of the call is not transcoded).

RFC 2833 to DTMF Audio Tones

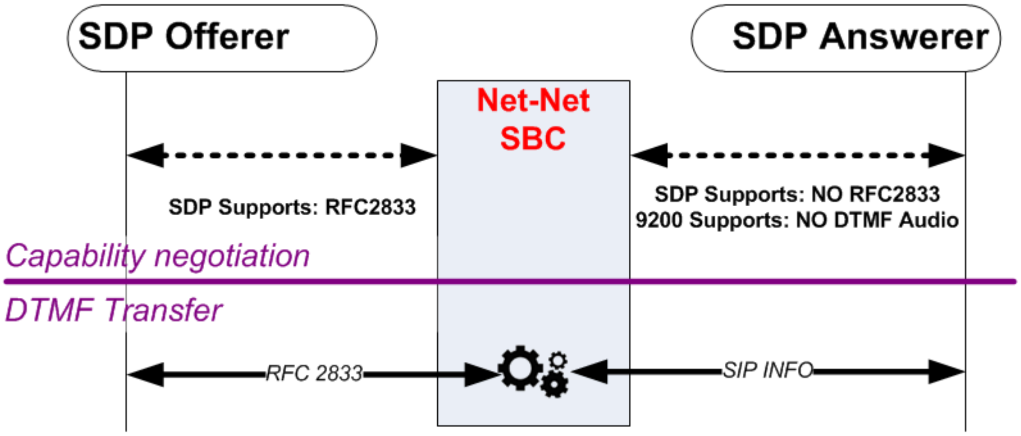
When the SDP offer side supports RFC 2833, and the SDP answer side supports the three DTMF Audio Tone conditions and does not support RFC 2833, the Oracle Enterprise Session Border Controller converts from RFC 2833 to DTMF audio tones for the call.



A SIP INFO message received by the Oracle Enterprise Session Border Controller from either the offerer or answerer is converted into the DTMF transfer method that the previous diagram shows for the egress side of the message. In this case, transcoding resources are used.

RFC 2833 to SIP INFO

When the SDP offer side supports RFC 2833 and the SDP answer side does not support the DTMF conditions nor RFC 2833, the Oracle Enterprise Session Border Controller converts from RFC 2833 to SIP INFO.



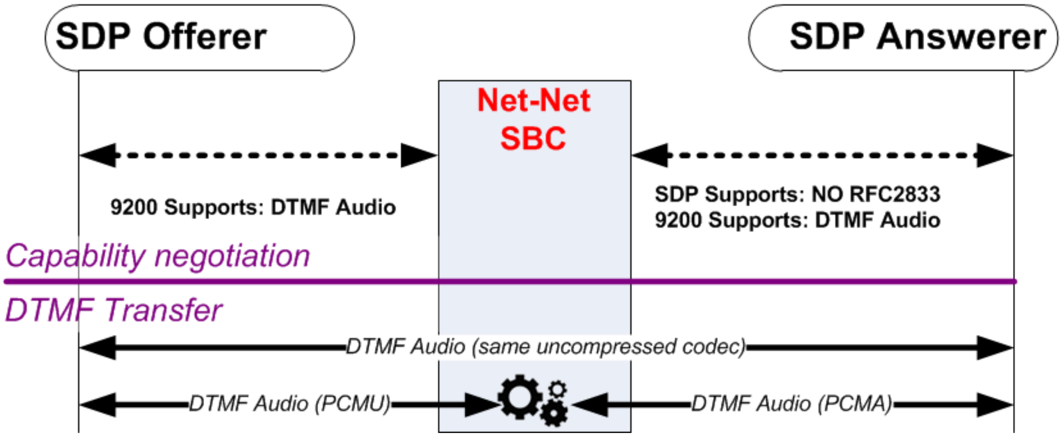
If a SIP message is received from the offerer, it is forwarded unconverted to the answerer.

DTMF Audio Tones Sent by Offerer

In the following three examples, the SDP offerer sends DTMF indication messages in DTMF audio tones format. The SDP answerer can receive DTMF indications in the format identified in each example.

DTMF Audio to DTMF Audio

If the SDP offer and answer sides both support the same type of G711 codec, the audio stream is forwarded between the two sides without processing.

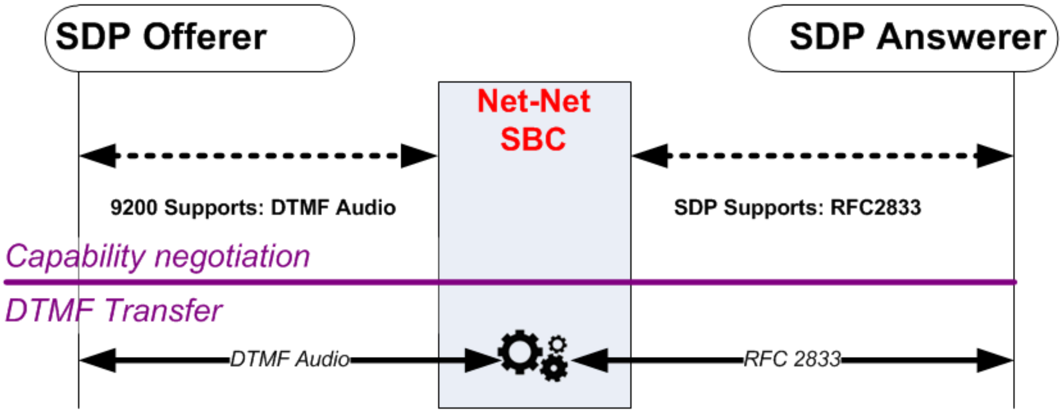


If the two sides of the call support DTMF audio tones, but use different audio codecs, and the SDP answer side supports the three DTMF Audio Tone conditions and does not support RFC 2833 then the Oracle Enterprise Session Border Controller will preserve DTMF audio tone indication across the call.

Transcoding resources are used only if different audio codecs are used or the Override Preferred DTMF Audio feature is enabled.

DTMF Audio to RFC 2833

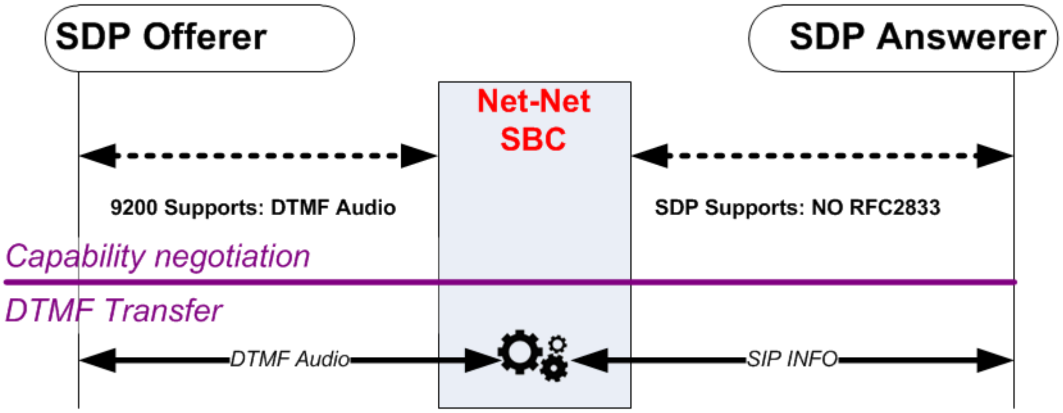
When the SDP offer side supports DTMF audio tones, and the SDP answer side supports RFC 2833, and transcoding resources are available, and does NOT support either or both of the first two DTMF Audio tone conditions, then the Oracle Enterprise Session Border Controller will convert incoming DTMF audio tones to outgoing RFC 2833 packets.



Transcoding resources are always required in this scenario.

DTMF Audio to SIP

When the SDP offer side supports DTMF audio tones, and the SDP answer side does not support RFC 2833, and does not support the three DTMF Audio Tone conditions, then the Oracle Enterprise Session Border Controller converts incoming DTMF audio tones to SIP INFOmessages.



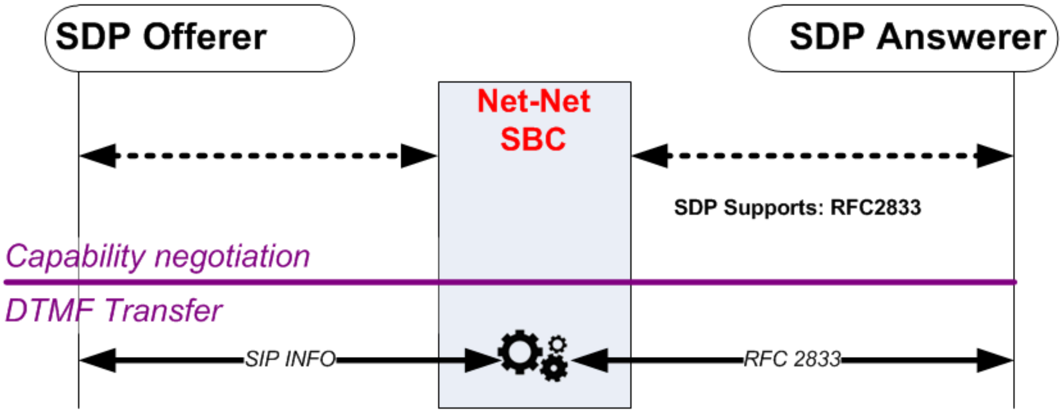
Transcoding resources are always required in this scenario.

SIP INFO Sent By Offerer

In the following three examples, the SDP offerer sends DTMF indication messages in SIP INFO message format. The SDP answerer can receive DTMF indications in the format identified in each example.

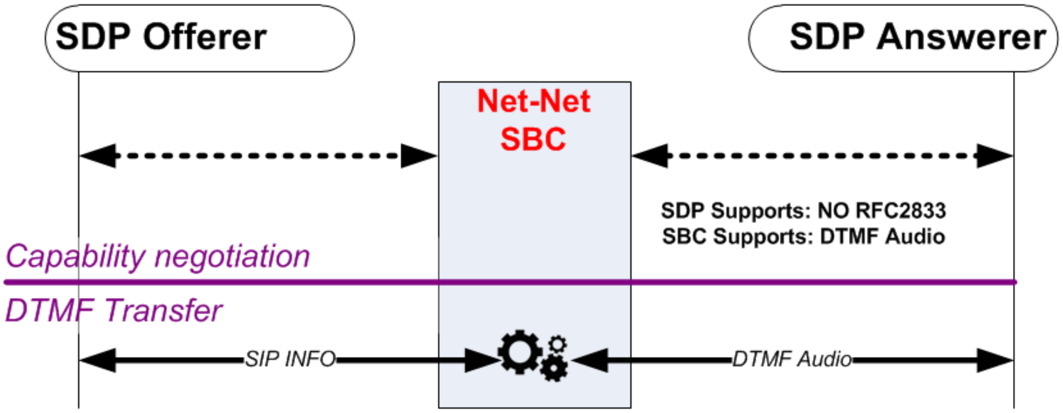
SIP INFO to RFC 2833

When the SDP offer side sends a SIP INFO message, and the SDP answer side supports RFC 2833, then the Oracle Enterprise Session Border Controller will convert incoming SIP INFO messages to outgoing RFC 2833 packets.



SIP INFO to DTMF Audio

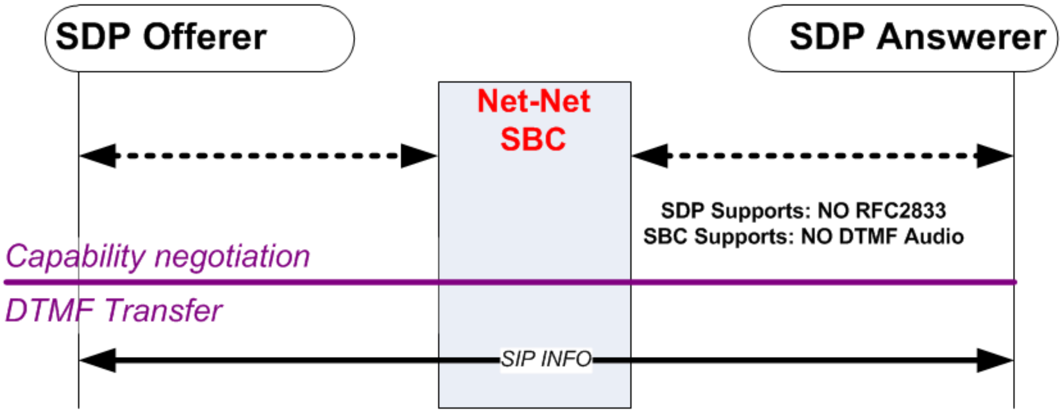
SIP INFO will only be converted to DTMF audio tones only if RFC 2833 is not supported, dtmf-in-audio is enabled, the answer side supports a G711 codec, and transcoding resources are available.



Transcoding resources are always required in this scenario.

SIP INFO to SIP INFO

When the SDP offer side sends a SIP INFO message and the SDP answer side does not support RFC 2833 and does not support the three DTMF audio tone conditions, the SIP INFO message will always be forwarded as the like SIP INFO message.



Dual Mode

Dual mode is used to send both RFC 2833 and the protocol-specific DTMF indication to a UA when possible: SIP INFO from a SIP interface.

To consider dual mode scenarios, the Oracle Enterprise Session Border Controller sets up the call SDP. At the conclusion of the capability negotiation, the Oracle Enterprise Session Border Controller is configured to support DTMF Audio tones or RFC 2833 independently for each side of the call.

When the call leg supports RFC 2833 as the means of DTMF transfer, the Oracle Enterprise Session Border Controller checks if the SIP interface's (or session agent's) RFC 2833-mode parameter is configured to dual. If it is, the Oracle Enterprise Session Border Controller sends both RFC 2833 and SIP INFO messages to the UA on this side of the call.

Note: Whether RFC 2833 support was initiated between the Oracle Enterprise Session Border Controller and the UA by a codec policy or by the rfc2833-mode parameter, the Oracle Enterprise Session Border Controller looks to the rfc-2833 parameter to consider if dual mode is supported.

When the call leg supports DTMF audio tones as the means of DTMF transfer, the Oracle Enterprise Session Border Controller checks if the codec policy's dtmf-in-audio parameter is configured to dual. If it is, the Oracle Enterprise Session Border Controller sends both DTMF audio tones and SIP INFOmessages to the UA on this side of the call.

P-Dual-Info Header

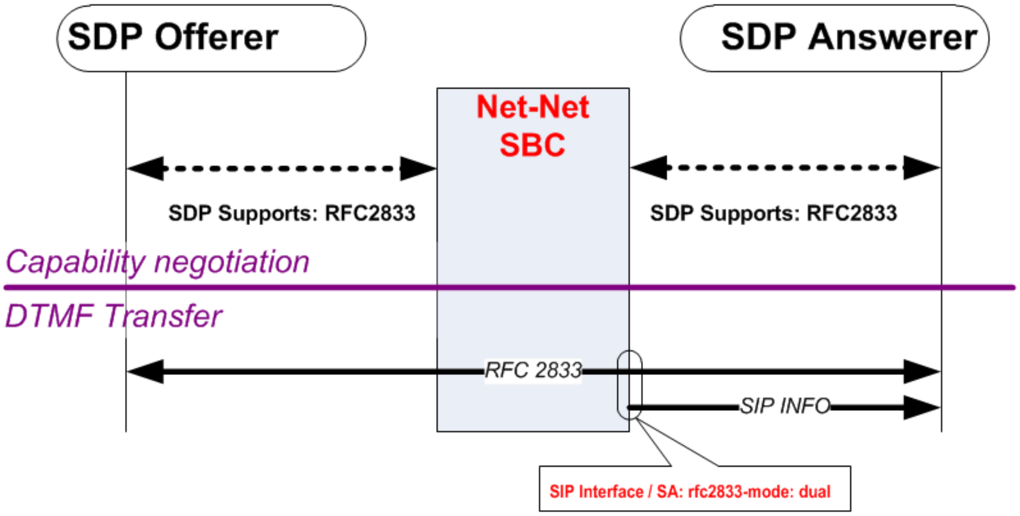
When the Oracle Enterprise Session Border Controller forward both media DTMF indication and signaling based DTMF indication for the same received DTMF indication, a P-Dual-Info header is added to the forwarded signaling message. You can configure the appearance of the header with the **`dual-info`** option. The default header appearance is:

```
P-dual-info: true
```

P-Dual-Info headers are only inserted into SIP INFO messages.

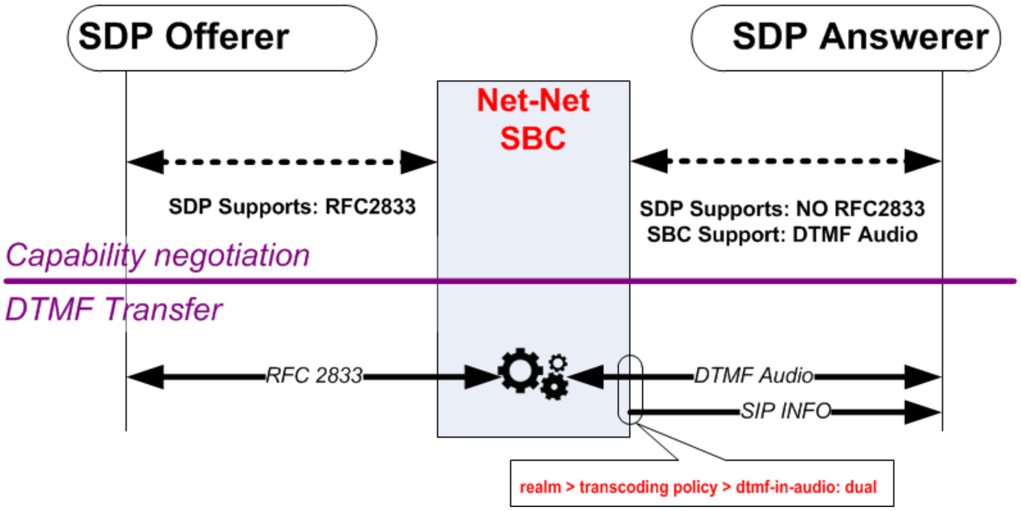
Example 1

In this example, RFC 2833 is supported on the egress side of the call. The egress SIP interface or session agent's rfc2833-mode is set to dual mode. When the Oracle Enterprise Session Border Controller forwards RFC 2833 to the SDP answerer, it also creates and forwards a corresponding SIP INFO message toward the target.



Example 2

In this example RFC 2833 telephone-event is not supported on the egress side of the call, but DTMF audio tones are. If the Oracle Enterprise Session Border Controller receives an RFC 2833 message, it is converted to DTMF audio tones. When the Oracle Enterprise Session Border Controller forwards DTMF audio tones to the SDP answerer, it also creates and forwards a corresponding SIP INFO message toward the target.



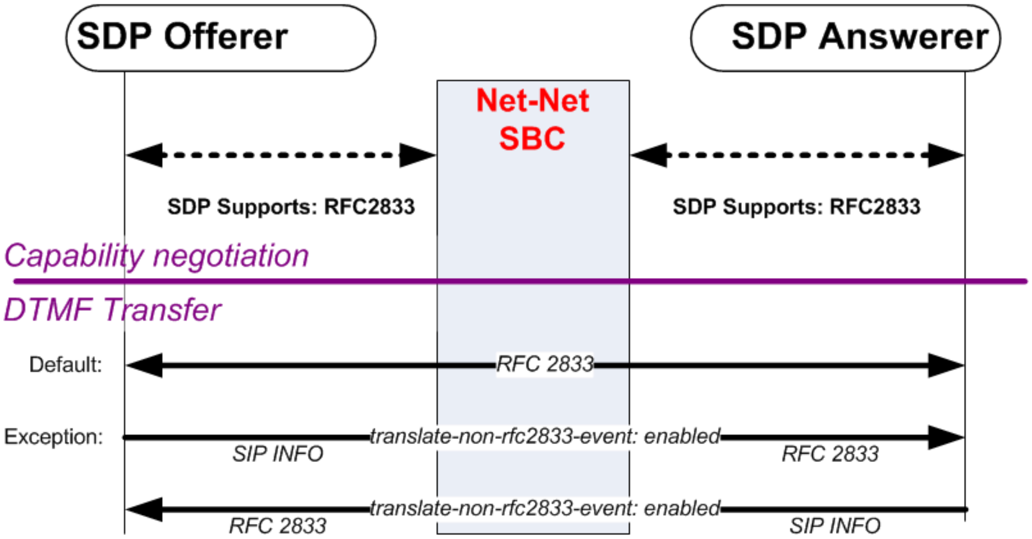
Identical Inband with Signaling DTMF Transfer Exception

Endpoints may send signaling-based DTMF indication (SIP INFO) to the Oracle Enterprise Session Border Controller at any time. Most of the time they are passed through the Oracle Enterprise Session Border Controller unchanged. You can enable an exception to this behavior forcing signaling messages to either RFC 2833 or DTMF audio tones depending on the call’s DTMF transfer mode. The two parameters to enable their respective exceptions are located in the media manager configuration.

Note: These behavior exceptions are only applicable when both sides of the call support and prefer the same DTMF transfer mode.

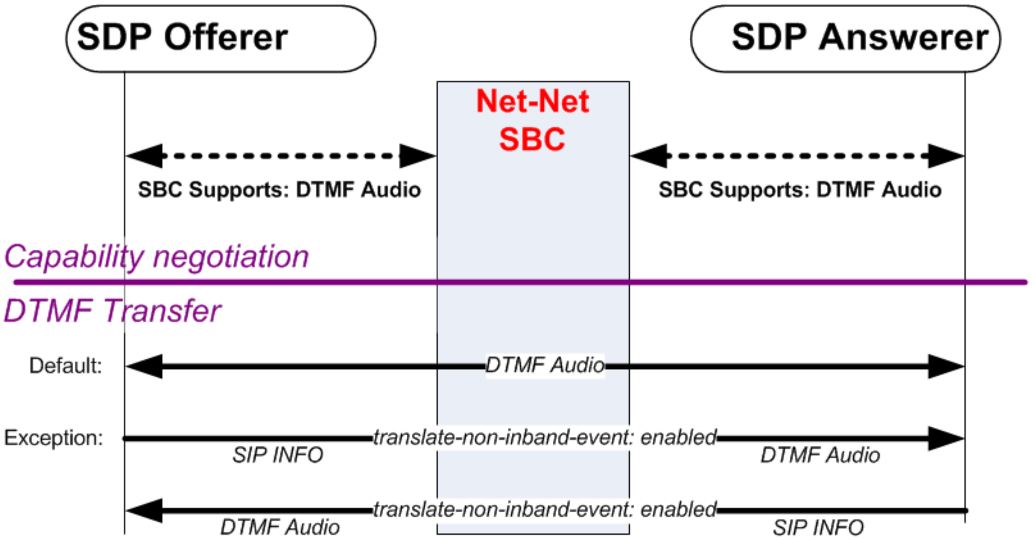
Override Preferred RFC 2833

When RFC 2833 is supported on both sides of the call, it is the preferred method of DTMF indication transport. To override this behavior, enable the translate non rfc 2833 parameter.



Override Preferred DTMF Audio

When DTMF audio tones are supported on both sides of the call, it is the preferred method of DTMF indication transport. To override this behavior, enable the translate non inband event parameter.



Note: Enabling one or both of these exceptions can cost DTMF translation resources whether they occur in the network processors or on transcoding modules. The translate-non-inband-event exception is especially costing because it reserves transcoding for all calls that resolve to support for DTMFable to DTMFable codecs.

DTMF Transfer for Spiral Calls

A spiral call occurs when a call’s signaling messages loop back through the Oracle Enterprise Session Border Controller. Most commonly the signaling path is from one UA, through the Oracle Enterprise Session Border Controller, to a call server, back through the Oracle Enterprise Session Border Controller, to another UA. The media path is from one UA, through the Oracle Enterprise Session Border Controller, to the other UA. For DTMF indication processing, only the call legs between the endpoints and Oracle Enterprise Session Border Controller are considered.

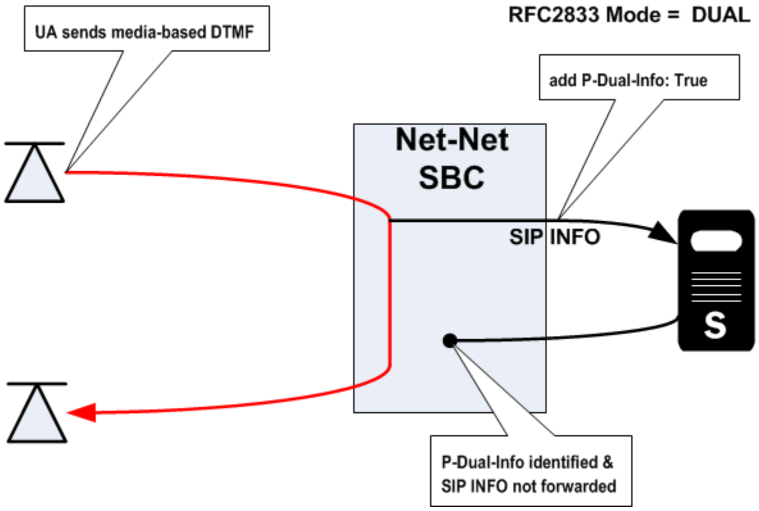
The Oracle Enterprise Session Border Controller evaluates that the signaling path to and from the call server terminates on the same IP address and port on the Oracle Enterprise Session Border Controller. This means that it’s a spiral call. In addition, the media IP addresses and ports in the SDP indicate that the Oracle Enterprise Session Border Controller does not need to send the media into the call server's realm.

An issue occurs when DTMF indication is relayed either by RFC 2833 or DTMF audio tones for a spiral call. Since the DTMF indication is in the media path, the call server remains unaware of the signaling; no media-based DTMF indication digits will ever reach the call server. In order to include the call server in the DTMF-indication signaling, you should set the dtmf-in-audio and/or rfc-2833 mode used for the realm or signaling interface where the call server is to dual.

P-Dual-Info Header

The P-Dual-Info header is used in a spiral call scenario, when the call leg to (and from) the call server is set to dual. While the media portion of a spiral call goes from endpoint to endpoint through the Oracle Enterprise Session Border Controller, the signaling portion of the call loops through a call server.

The Oracle Enterprise Session Border Controller inserts a P-Dual-Info header into a SIP info message sent to the call server, which in turn forwards the SIP INFO message back to the Oracle Enterprise Session Border Controller. Seeing the P-Dual-Info header, the Oracle Enterprise Session Border Controller knows not to forward this SIP INFO message to the target endpoint because it would be a duplication of DTMF indication already sent to the endpoint in media format.


UA sends media-based DTMF
RFC2833 Mode = DUAL
add P-Dual-Info: True
Net-Net SBC
SIP INFO
S
P-Dual-Info identified & SIP INFO not forwarded

DTMF Transfer Hardware Processing

DTMF transfer processing, the conversion between two DTMF transfer types, occurs in either the transcoding NIU's Digital Signal Processors (DSPs) or the Network Processors (NPs). Understanding where the processing takes place is important to determine which subsystem uses extra processing load per conversion.

There are a few rules you can use to determine which subsystem performs the DTMF transfer processing:

- If audio transcoding is enabled for the call, DTMF transfer processing occurs in the transcoding modules.
- If DTMF audio tones are generated from RFC 2833 or from signaling messages (SIP INFO), DTMF transfer processing occurs in the transcoding modules.
- If the global translate non inband event parameter is enabled, DTMF transfer processing occurs in the transcoding modules.
- If signaling to RFC 2833 processing occurs in either direction of the call, and the previous 3 conditions are not valid, DTMF transfer processing occurs in the NPs.

DTMF Transfer Configuration

RFC 2833 Session Agent Configuration

Session agents, used as a way to classify and act on a subset of a signaling interface's traffic, also have **rfc2833-mode** and **rfc2833-payload** parameters. The configurations of these parameters overrides the configuration of the same-named parameters on the signaling interface where the session agent resides. You can set the **rfc2833-mode** parameter to **none** for a session agent to revert to the parent signaling interface's two RFC 2833 settings.

ACLI Configuration and Instructions

This section explains how to configure the RFC 2833 mode on a signaling interface and on a session agent configured for that signaling interface. The session agent's configuration takes precedence over the signaling interface, unless the session agent's rfc2833-mode is set to none. In that case, the signaling interface's configuration is used for applicable traffic.

SIP Interface

To configure the RFC 2833 mode on a SIP interface:

1. In Superuser mode, type **configure terminal** and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press Enter.

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-interface** and press Enter.

```
ACMEPACKET(session-router)# sip-interface
```

4. If you are adding support for this feature to a pre-existing SIP interface, then you must select the specific configuration instance, using the ACLI **select** command.
5. **rfc2833-mode**—Set this parameter to either **transparent**, **preferred**, or **dual** based upon the behavior your want for this SIP interface.
   - transparent—does not add RFC 2833 telephone-event into SDP if not present, and does not prefer.
   - preferred—adds RFC 2833 telephone-event media type into SDP and prefers to use this method for DTMF indication.
   - dual—adds RFC 2833 telephone-event media type into SDP and sends both SDP and signaling-based DTMF indications if possible.
6. **rfc2833-payload**—Set this parameter to the media-type value you wish to use when inserting RFC 2833 telephone-events into an SDP offer. 101 is the generally accepted media type for RFC 2833 telephone-events.
7. Save and activate your configuration.

Session Agent

Session agent RFC 2833 mode configurations override those on the signaling interface where they exit. The **none** parameter is used to defer to the signaling interface.

To configure the RFC 2833 mode on a session agent:

1. In Superuser mode, type **configure terminal** and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type **session-agent** and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. Select the session agent where you want this feature.

```
ACMEPACKET(session-agent)# select 1
```

5. **`rfc2833-mode`**—Set this parameter to either **`none, transparent`**, **`preferred`**, or **`dual`** based upon the behavior your want for this SIP interface.

   - none—defaults to the behavior of the SIP interface for traffic that matches this session agent.
   - transparent—does not add RFC 2833 telephone-event into SDP if not present, and does not prefer.
   - preferred—adds RFC 2833 telephone-event media type into SDP and prefers to use this method for DTMF indication.
   - dual—adds RFC 2833 telephone-event media type into SDP and sends both SDP and signaling-based DTMF indications if possible.

6. **`rfc2833-payload`**—Set this parameter to the media-type value you wish to use when inserting RFC 2833 telephone-events into an SDP offer. 101 is the generally accepted media type for RFC 2833 telephone-events.
7. Save and activate your configuration.

Codec Policy

To configure a codec policy to support DTMF audio tones, as transcoded:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type media-manager and press Enter.

```
configure# media-manager
```

3. Type **`codec-policy`** and press Enter.

```
ACMEPACKET(media-manager)# codec-policy
ACMEPACKET(codec-policy)#
```

4. If you are adding support for this feature to a pre-existing configuration, then you must select the specific configuration instance, using the ACLI **`select`** command.

```
ACMEPACKET(codec-policy)# select
<name>:
1: private
2: public
selection:1
ACMEPACKET(codec-policy)#
```

5. **`dtmf-in-audio`**—Set this parameter to **`disabled`**, **`preferred`**, or **`dual`** based upon how the Oracle Enterprise Session Border Controller should support the conversion of signaling messages or RFC 2833 to DTMF Audio tones in the realm where this codec policy is active.

   - disabled—does not support DTMF audio tones as transcoded in this realm.
   - preferred—supports DTMF audio tones as transcoded in this realm.
   - dual—supports both transcoded DTMF audio tones and signaling-based DTMF indications if possible.

6. Save and activate your configuration.

Translate Non2833 Event Behavior

To configure the exceptional behavior:

1. In Superuser mode, type **`configure terminal`** and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type **`media-manager`** and press Enter to access the media-level configuration elements.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type **media-manager** and press Enter to begin configuring this feature.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. **translate-non-rfc2833-event**—Set this parameter to enabled to use non-default behavior described in Override Preferred RFC 2833.
5. **translate-non-inband-event**—Set this parameter to enabled to use non-default behavior described in Override Preferred DTMF Audio.
6. Save and activate your configuration.

P-dual-info Header Appearance

Customizing the P-Dual-Info header is performed globally from the sip config.

To configure how the P-dual-info header appears:

1. In Superuser mode, type **configure terminal** and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press Enter to access the system-level configuration elements.

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-config** and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-config
ACMEPACKET(sip-config)#
```

4. Type options followed by a Space.
5. After the Space, type the P-Dual-Info header information in the following format:

```
+dual-info=”<header-name>”
```

For example:

```
ACMEPACKET(sip-config)# options dual-info=P-Dual-Info
```

6. Save your work using the ACLI **done** command.

RFC 2833 Customization

RTP Timestamp

As a media flow with injected RFC 2833 telephone-event packets exits the Oracle Enterprise Session Border Controller, the newly generated RTP packets are timestamped in one of two ways. If RFC 2833 generation is preformed in the NPs, the default method of creating the timestamp for a generated RFC 2833 packet is to use the previous RTP packet’s timestamp and add 1.

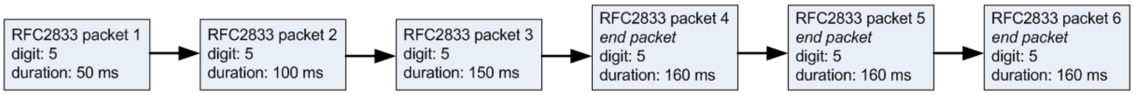
Alternatively the Oracle Enterprise Session Border Controller, can estimate the actual time that the injected RFC 2833 telephone-event packet will leave the system and use that. This method tags the injected DTMF indication packet more accurately than the than previous packet + 1 method. As an additional bonus, the packet’s checksum is regenerated. This alternate timestamp creation behavior is configured by setting the **rfc2833-timestamp** parameter to enabled.

When RFC 2833 telephone-event generation is preformed by the transcoding modules, the packets’ timestamps are the set to the time that the packets leave the Oracle Enterprise Session Border Controller,.

RFC 2833 telephone-event duration intervals

If an incoming SIP INFO message’s DTMF indication duration is unspecified, the Oracle Enterprise Session Border Controller, uses a default 250 ms duration for the generated RFC 2833 telephone-event. Otherwise, the SIP INFO’s specified event duration is used. RFC 2833 telephone-event packets are still generated at 50 ms intervals upon egress.

At the conclusion of the DTMF indication, the three end-event packets are sent. The packet arrangement when the user presses the digit 5 for 160ms, with the default 50ms interval follows:



When either no DTMF event duration is specified, or the event duration is less than the 50ms default minimum, you can set the default RFC 2833 telephone-event duration using **`default-2833-duration`** parameter in the media manager configuration. This is the value that the Oracle Enterprise Session Border Controller, uses for the duration of a telephone event when none is specified in the incoming message. The **`default-2833-duration`**'s valid range is 50-5000ms. The Oracle Enterprise Session Border Controller, also uses this configured value when it receives a SIP INFO message with a duration less than the minimum signal duration.

You can configure the minimum duration at which RFC 2833 telephone-events are generated by the Oracle Enterprise Session Border Controller, using the **`min-signal-duration`** option in the media manager configuration, thus changing the lower threshold of the **`default-2833-duration`** parameter from 50 ms to your own value. If the duration the Oracle Enterprise Session Border Controller, receives is less than the threshold, it uses the value configured in the **`default-2833-duration`** parameter.

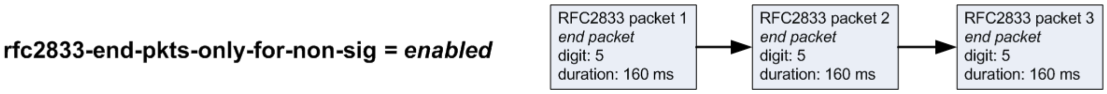
Note: Timestamp changes and duration changes only take effect when the 2833 timestamp (rfc-2833-timestamp) is enabled in the media manager configuration. If you enable the rfc-2833-timestamp parameter, but do not configure the default-2833-duration parameter, the default-2833-duration parameter defaults to 100 ms.

RFC 2833 End Packets

When the Oracle Enterprise Session Border Controller, generates RFC 2833 telephone-event packets, they are forwarded from the egress interface every 50 ms by default. Each packet includes the digit and the running total of time the digit is held. Thus DTMF digits and events are sent incrementally to avoid having the receiver wait for the completion of the event.

At the conclusion of the signaled event, three end packets stating the total event time are sent. This redundancy compensates for RTP being an unreliable transport protocol.

You can configure your Oracle Enterprise Session Border Controller, to generate either the entire start-interim-end RFC 2833 packet sequence or only the last three end 2833 packets for non-signaled digit events using the **`rfc2833-end-pkts-only-for-non-sig`** parameter. If the parameter were enabled, the RFC 2833 telephone-event packets for the same event would appear are represented by the following graphic.



ACLI Instructions and Examples

To configure RFC 2833 customization:

1. In Superuser mode, type **`configure terminal`** and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type **`media-manager`** and press Enter to access the media-level configuration elements.

```
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)#
```

3. Type **`media-manager`** and press Enter to begin configuring this feature.

```
ACMEPACKET(media-manager)# media-manager
ACMEPACKET(media-manager-config)#
```

4. **`rfc2833-timestamp`**—Set this parameter to enabled to use the estimated real departure timestamp on an injected RFC 2833 telephone-event packet.

5. **`default-2833-duration`**—Set this parameter to the default value you wish to use when the Oracle Enterprise Session Border Controller, creates or generates DTMF-indication messages.
6. **`rfc2833-end-pkts-only-for-non-sig`**—Set this parameter to enabled for the Oracle Enterprise Session Border Controller, to only send the three RFC 2833 telephone-event end-packets to indicate a total event duration, rather than the running total from time=0.
7. **`options`**—Set the options parameter by typing **`options`**, a Space, the option name **`min-signal-duration=x`** (where **`x`** is the value in milliseconds you want to use for the threshold) with a plus sign in front of it. Then press Enter.
8. Save and activate your configuration.

3

Other Release Features

RFC 4028 Session Timers

The Oracle Enterprise Session Border Controller, supports RFC 4028 Session Timers. In this role, it acts as a B2BUA between two endpoints and then enforces the timer values on each call leg independently. The RFC 4028 abstract states:

This document defines an extension to the Session Initiation Protocol (SIP). This extension allows for a periodic refresh of SIP sessions through a re-INVITE or UPDATE request. The refresh allows both user agents and proxies to determine whether the SIP session is still active. The extension defines two new header fields:

- Session-Expires—which conveys the lifetime of the session
- Min-SE—which conveys the minimum allowed value for the session timer

The following parameters in the **session-timer-profile** configuration element are used for this feature:

- **session-expires**—The value of the session expires header in seconds
- **min-se**—The value of the Min-SE header in seconds (this is a minimum session expires value)
- **force-reinvite**—Sets if the Oracle Enterprise Session Border Controller will send a reINVITE to refresh the session timer when applicable.
- **request-refresher**—Set on the outbound side of a call what the Oracle Enterprise Session Border Controller sets the refresher parameter to. Valid values are **uac**, **uas**, or **none**.
- **response-refresher**—Set on the inbound side the value of the refresher parameter in the 200OK message. Valid values are **uac** or **uas**.

In this section, the notion that a UAC or UAS supports Session Timers is indicated by the presence of the Supported: timer header and option tag.

Ingress Call Leg

Setting 200 OK’s Session-Expire value

The session timer is based on the negotiation between each side’s session expires value. The final value on the ingress leg is returned by the Oracle Enterprise Session Border Controller, to the UAC, unless there is an error.

The Oracle Enterprise Session Border Controller, can always reduce the session expires value it returns to the UAC. It checks that the Session-Expires: header is larger than the SIP Interface’s min-se value. The Oracle Enterprise Session Border Controller then compares the received Session-Expires: header to the configured session-expires configuration element and uses the lower value for the 200 OK's Session-Expires: header. If this outbound Session-Expires: value is lower than the received Min-SE: header, it will be bumped up to the Min-SE: header's value.

If the Oracle Enterprise Session Border Controller,'s Min-SE value is larger than the Session-Expires: header, a 422 (Session Interval Too Small) message is returned to the UAC containing the Oracle Enterprise Session Border Controller,'s configured Min-SE value.

When the UAC supports (but does not require) Session Timers and the Oracle Enterprise Session Border Controller, does not support session timers, a 200 OK is returned to the UAC with no indication of session timer support.

Refresher

The initial UAC, the side that sends the INVITE, can set itself to be the refresher (uac) or the Oracle Enterprise Session Border Controller, as the refresher (uas). Whoever is the refresher is indicated in the 200 OK. If the UAC does not specify any refresher, the Oracle Enterprise Session Border Controller, uses it's response-refresher value in the 200 OK. If that value is set to uas, the Oracle Enterprise Session Border Controller, creates and sends a re-INVITE toward the UAC with previously negotiated session expiration values.

Once the Oracle Enterprise Session Border Controller, becomes the refresher, it does not relinquish that role. Then, when the Oracle Enterprise Session Border Controller, sends refresh requests, it does not change any parameters (refresher role & timers) from the initial request negotiation.

UAC does not Support Session Timers

If the UAC's initial request does not include a Session-Expires: header, then the 200 OK will include the **`session-timer-profile > session-expires`** value on the ingress leg in the Session-Expires: header.

The Oracle Enterprise Session Border Controller, also inserts the refresher parameter as configured. The orientation of UAC/UAS on the Oracle Enterprise Session Border Controller,'s view of a call leg can change if later in the call flow the endpoint designates the Oracle Enterprise Session Border Controller as the refresher.

Note: When the request doesn't support Session Timers, the Oracle Enterprise Session Border Controller,'s reply adds session timer support according to configuration.

If the Oracle Enterprise Session Border Controller, receives a message with a Require: timer header, and the inbound SIP interface or the final UAS do not support session timers, a 420 (Bed Extension) is returned to the UAC.

Egress Call Leg

Outbound INVITE Message

When the Oracle Enterprise Session Border Controller,'s outbound interface is configured with session timers, it forwards an INVITE to the UAS with the following headers:

Session-Expires—Oracle Enterprise Session Border Controller, inserts the outbound SIP interface's session-timer parameter

Session-Expires refresher parameter—Oracle Enterprise Session Border Controller, inserts the request-refresh parameter

Min-SE—Oracle Enterprise Session Border Controller, inserts the outbound SIP interface's session-timer parameter

Supported—Supported header has the timer option tag

Note: Require/Proxy-Require—If the timer parameter is present in the Require or Proxy-Require: header field in the request received from the UAC, it will be removed.

No Session Timer Configuration

If the ingress SIP interface supports session timers, and the original INVITE from the UAC included session timer support, the INVITE request sent to the UAS will have no session timer support. However, the Supported: timer header will be created. This ensures that the Oracle Enterprise Session Border Controller, does not get a 421 response for 'timer' from the UAS.

If the ingress SIP interface supports session timers, and the UAC’s initial INVITE did not include session timer support, then the INVITE sent to the UAS will have no session timer support (headers) as well.

If the ingress SIP interface does not support session timers, the INVITE is forwarded with no Session Timer alteration.

UAS Initial Response

Upon receiving a 200 OK from the UAS, if the response specifies uac as the refresher, the 200 OK includes a Session-Expires header and specifies uac as the refresher, the Oracle Enterprise Session Border Controller, will assume the refresher role. If the 300 OK does not include a Session-Expires header, and the egress interface supports session timers, then the Oracle Enterprise Session Border Controller, assumes the refresher role.

UAS Returns Errors

422 Session Interval Too Small—The Oracle Enterprise Session Border Controller, in response sends the request again with new values in the 'Session-Expires' header field based on the 'Min-SE' value present in the 422 response.

421 Extension Required for 'timer'—This response can only happen if none of the other three entities (UAC, ingress SIP interface and egress SIP interface) support session timers. The 421 is forwarded through the system to the original UAC.

420 Bad Extension for 'timer'—This response should never happen because the Oracle Enterprise Session Border Controller, will never send Require: timer header. But the event this error is received, it will be forwarded to the original UAC.

Session Refreshes

On either side of the call, the Oracle Enterprise Session Border Controller, can be responsible for initiating the session refreshes or responding to the session refreshes.

Oracle Enterprise Session Border Controller, as Refresher

The Oracle Enterprise Session Border Controller, sends the refresh request when half the session expiration has elapsed. The Oracle Enterprise Session Border Controller, always wants to remain the refresher and maintain the initially agreed upon session expiration timers.

Creating the Refresh Message

The refresh message takes the form of a re-INVITE when the force-reinvite parameter in the session timer profile is enabled. If this parameter is disabled and the remote end supports UPDATE requests, an UPDATE message will be sent.

UPDATE messages contain no SDP information.

Re-INVITE messages contain the SDP that is the same as what was sent before.

The refresh request’s Session-Expires: header value is set to the existing value for the session. The refresher parameter is set to uac since the Oracle Enterprise Session Border Controller, acts like a UAC for this refresh transaction. The Min-SE header is also included.

Processing the Refresh Response

The session expires value in the 2xx response is accepted and the timer restarts.

If the remote end does not include any session expiration parameters, the Oracle Enterprise Session Border Controller, continues to support session timers, and assumes that the refresh interval is the same as before.

Any response that is 422 Session Interval Too Small is handled as expected. The Oracle Enterprise Session Border Controller, resends the refresh request again with new values based on the 422 response. Any other response to the refresh request that is not a dialog/usage destroying response is treated like a 200 OK response.

Subsequent refresh requests are created and sent after half the previous refresh interval. If non-2xx, dialog / usage destroying responses are received, the Oracle Enterprise Session Border Controller, reduces the following refresh intervals by half, as long as the final interval is not less than 32 seconds. The Oracle Enterprise Session Border Controller, then uses this period for sending refresh requests until it successfully receives a 2xx response.

Oracle Enterprise Session Border Controller, as Refresh Responder

Processing the Refresh

The refresh request is processed similarly to the initial request regarding the session timer parameters. The session timer for this call leg is restarted when the Oracle Enterprise Session Border Controller, when it sends the 200 OK response for the refresh request.

Forwarding the Refresh

When the Oracle Enterprise Session Border Controller, receives an UPDATE request, it is forwarded to the other end since the Oracle Enterprise Session Border Controller, cannot determine whether this request is only for session refreshing, or for other purposes as well.

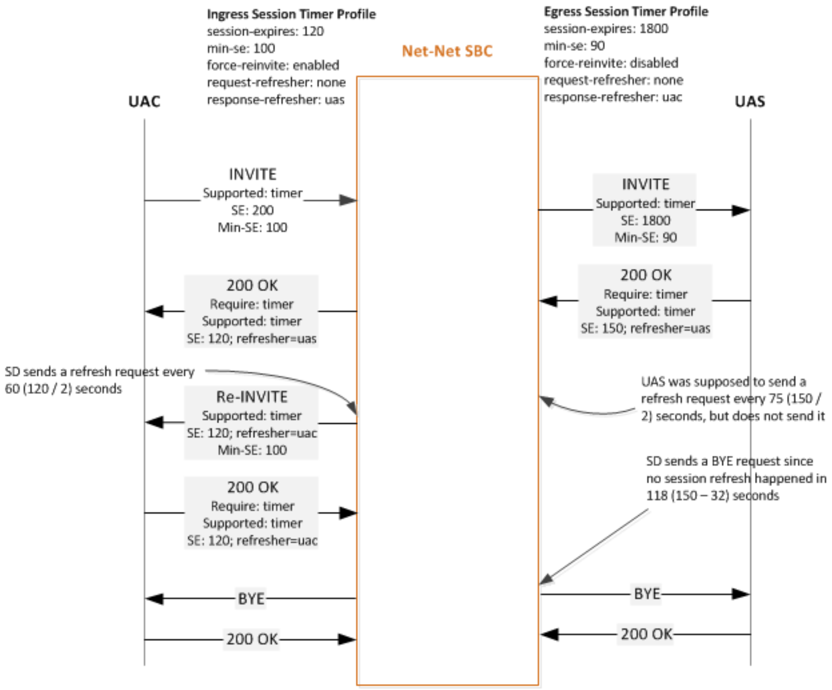
When the Oracle Enterprise Session Border Controller, receives a re-INVITE request, it will determine whether this request needs to be suppressed, or should it be forwarded to the other end.

Timer Expiration

If the Oracle Enterprise Session Border Controller, fails to receive a session refresh request before the session expiration, the session will be terminated before the full session time. This is computed according to:

$$\text{real expiration time} = \text{session expiration period} - \min\left(\frac{\text{session expiration period}}{3}, 32\right)$$

After the real expiration time elapses, the Oracle Enterprise Session Border Controller, sends a BYE request in both directions to terminate the session.



Interaction with SIP Features

Consider the following sections that have interactions with RFC 4028 Support.

sip-config option session-timer-support

A configured session-timers-profile on a SIP interface overrides the session-timer-support option in the SIP config. The Oracle Enterprise Session Border Controller, can still act in proxy mode for some calls and B2BUA for other calls considering which SIP interfaces session timer profiles are not configured for.

sip-feature Support

When the UAC sends a Require: timer header is in the initial request and the Oracle Enterprise Session Border Controller, does not support session timers, and no sip-feature configuration element is configured for 'timer' for that realm, the Oracle Enterprise Session Border Controller, replies with a 420 (Bad Extension) response for 'timer'.

When the UAC sends a Require: timer header is in the initial request and the Oracle Enterprise Session Border Controller, does support session timers, the timer tag is removed from the Require: header even if a sip-feature configuration element is configured for 'timer'. This also applies for the Proxy-Require: header.

Acme Packet recommends you do not configure a sip feature configuration element while using the Session Timers feature.

sip-interface option suppress-reinvite

SIP re-INVITE suppression is automatically enabled for a SIP interface when session timers are enabled. This behavior prevents re-INVITEs whose purpose is only for session refreshes from being forwarded to the other call leg. The first re-INVITE received from a UAS on the terminating call leg will be passed to the UAC on the originating call leg since the Oracle Enterprise Session Border Controller, has no prior INVITE request coming from the UAS to match against.

Re-INVITEs are suppressed only when the Oracle Enterprise Session Border Controller, receives the same INVITE request back-to-back without any intervening re-INVITE request in the opposite direction, or an UPDATE or PRACK request in either direction.

Note: The SIP reINVITE Supression parameters are not replicated on the standby system in an HA environment. The first re-INVITE after a switchover will be forwarded to the far end.

Examples

Ex	Messages on Originating Call Leg	Ingress SIP Interface Config	Ingress SIP Interface Config	Messages on Terminating Call Leg
1	INVITE → Supported: timer SE: 200 ———————— ← 200 OK Require: timer SE: 200; refresher=uas	session-expires: 500 min-se: 200 request-refresher: none response-refresher: uas this element becomes refresher	session-expires: 500 min-se: 400 request-refresher: none response-refresher: uas	INVITE→ Supported: timer SE: 500 Min-se: 400 ———————— ← 200 OK Require: timer SE: 400; refresher=uas
2	INVITE → Supported: timer SE: 1200; refresher=uas	session-expires: 500 min-se: 200 request-refresher: none	session-expires: 500 min-se: 400 request-refresher: uas	INVITE → Supported: timer SE: 500; refresher=uas

Ex	Messages on Originating Call Leg	Ingress SIP Interface Config	Ingress SIP Interface Config	Messages on Terminating Call Leg
	———————————— ← 200 OK Require: timer SE: 500; refresher=uas	response-refresher: uac this element becomes refresher	response-refresher: uas this element becomes refresher	Min-se: 400 ———————————— ← 200 OK
3	INVITE → Supported: timer SE: 1200; refresher=uac Min-se: 800 ———————————— ← 200 OK Require: timer SE: 800; refresher=uac	session-expires: 500 min-se: 200 request-refresher: none response-refresher: uas	No session timer configuration this element becomes refresher	INVITE → Supported: timer ———————————— ← 200 OK Require: timer SE: 400; refresher=uac
4	INVITE → Supported: timer SE: 200; refresher=uac ———————————— ← 200 OK Require: timer SE: 200; refresher=uac	session-expires: 500 min-se: 200 request-refresher: none response-refresher: uas	No session timer configuration	INVITE → Supported: timer ———————————— ← 200 OK
5	INVITE → Supported: timer SE: 200 ———————————— ← 200 OK	No session timer configuration	session-expires: 500 min-se: 400 request-refresher: uas response-refresher: uas this element becomes refresher	INVITE → Supported: timer SE: 500; refresher=uas Min-se: 400 ———————————— ← 200 OK
6	INVITE → Supported: timer SE: 200 ———————————— ← 200 OK Require: timer	No session timer configuration ESD behavior stays same as current behavior	No session timer configuration	INVITE → Supported: timer SE: 200 ———————————— ← 200 OK Require: timer

Ex	Messages on Originating Call Leg	Ingress SIP Interface Config	Ingress SIP Interface Config	Messages on Terminating Call Leg
	SE: 400; refresher=uas			SE: 400; refresher=uas
7	INVITE → Require: timer SE: 200 ———————————————— ← 420 Unsupported: timer	No SIP feature for timer No session timer configuration SD behavior stays same as current behavior	No session timer configuration	
8	INVITE → Require: timer SE: 200 ———————————————— ← 420 Unsupported: timer	SIP feature configured for timer No session timer configuration SD behavior stays same as current behavior	No session timer configuration	INVITE → Required: timer SE: 200 ———————————————— ← 420 Unsupported: timer
9	INVITE → Require: timer SE: 200 ———————————————— ← 200 OK	SIP feature configured for timer No session timer configuration	session-expires: 500 min-se: 400 request-refresher: none response-refresher: uas this element becomes refresher	INVITE → Supported: timer SE: 500 Min-se: 400 ———————————————— ← 200 OK Require: timer SE: 500; refresher=uac
10	INVITE → Require: timer SE: 200 ———————————————— ← 200 OK Require: timer SE: 200; refresher=uac	No SIP feature for timer session-expires: 500 min-se: 200 request-refresher: none response-refresher: uac	session-expires: 500 min-se: 400 request-refresher: uas response-refresher: uas	INVITE → Supported: timer SE: 500; refresher=uas Min-se: 400 ———————————————— ← 200 OK Require: timer SE: 500; refresher=uas
11	INVITE → Require: timer	No SIP feature for timer No session timer configuration	session-expires: 500 min-se: 400	

Ex	Messages on Originating Call Leg	Ingress SIP Interface Config	Ingress SIP Interface Config	Messages on Terminating Call Leg
	SE: 200 ———————————— —— ← 420 Unsupported: timer		request-refresher: none response-refresher: uas	
12	INVITE → SE: 200 ———————————— —— ← 200 OK SE: 500; refresher=uas	session-expires: 500 min-se: 500 request-refresher: none response-refresher: uas this element becomes refresher	No session timer configuration	INVITE → ———————————— —— ← 200 OK
13	INVITE → ———————————— —— ← 200 OK	No session timer configuration	session-expires: 500 min-se: 400 request-refresher: none response-refresher: uas	INVITE → Supported: timer SE: 500 Min-se: 400 ———————————— —— ← 200 OK Require: timer SE: 400; refresher=uas
14	INVITE → ———————————— —— ← 421 Require: timer	No session timer configuration SD behavior stays same as current behavior	No session timer configuration	
15	INVITE → Supported: timer SE: 200 ———————————— —— ← 422 Min-se: 400	session-expires: 500 min-se: 400		
16	INVITE → Supported: timer SE: 200	session-expires: 800 min-se: 90 request-refresher: none response-refresher: uac	session-expires: 800 min-se: 90 request-refresher: none response-refresher: uac	INVITE → Supported: timer SE: 800 Min-se: 90

Ex	Messages on Originating Call Leg	Ingress SIP Interface Config	Ingress SIP Interface Config	Messages on Terminating Call Leg
	————————— —— ← 200 OK Require: timer SE: 200; refresher=uac			————————— —— ← 422 Min-se: 900 ————————— —— INVITE → Supported: timer SE: 900 Min-se: 900 ————————— —— ← 200 OK Require: timer SE: 900; refresher=uas

RADIUS Interim record Generation

When refresh requests (UPDATE or Re-INVITE) are sent by the Oracle Enterprise Session Border Controller, no RADIUS Interim records are generated because session parameters do not change when these requests are sent.

When UPDATE requests are received by the Oracle Enterprise Session Border Controller, no RADIUS Interim records are generated.

When Re-INVITE requests are received by the Oracle Enterprise Session Border Controller, RADIUS Interim records are generated if the generate-interim parameter is enabled.

Session Timer Profile Configuration

To configure a session timer profile object:

1. In Superuser mode, type **configure terminal** and press Enter.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)#
```

2. Type **session-router** and press Enter.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type **session-timer-profile** and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-timer-profile
ACMEPACKET(session-timer-profile)#
```

4. **name**—Enter a name for this session timer profile.
5. **session-expires**—Enter the session timer value in seconds you wish this object to use natively.
6. **min-se**—Enter the minimum session timer value in seconds for this object.
7. **force-reinvite**—Leave the default of enabled for the Oracle Enterprise Session Border Controller, to always use reINVITEs for session refreshes. Set this parameter to disabled for the Oracle Enterprise Session Border Controller, to try using UPDATEs for session refreshes.

8. **`request-refresher`**—Set this to the value to insert in the refresher parameter in the Session-Expires: header on the originating call leg that the Oracle Enterprise Session Border Controller, includes in the 200 OK response message. Valid values are **`uac`** and **`uas`**.
9. **`response-refresher`**—Set this to the value to insert in the refresher parameter in the Session-Expires: header on the terminating call leg that the Oracle Enterprise Session Border Controller, includes in the INVITE message. Valid values are **`uac`**, **`uas`**, **`none`**.
10. Type **`done`** to save your work and continue.

Session Timer Profile to a SIP Interface Configuration

To apply a session timer profile to a SIP interface:

1. In Superuser mode, type **`configure terminal`** and press Enter.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)#
   ```
2. Type **`session-router`** and press Enter.

   ```
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)#
   ```
3. Type **`sip-interface`** and press Enter.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

   If you are adding this feature to an existing configuration, then you will need to select the configuration you want to edit.
4. **`session-timer-profile`**—Enter the name of a session timer profile object you have configured and want applied to this SIP interface.
5. Type **`done`** to save your work and continue.

Verify Config Validation

The Oracle Enterprise Session Border Controller,'s verify-config function checks that the value configured in all sip-interfaces' session-timer-profiles correspond to a configured session-timer-profile name. The following is be generated when this check fails:

```
ERROR: sip-interface [id] has reference to session-timer-profile [xyz] which
does not exist
```

show sipd status

The show sipd status command now contains new statistic, called Refreshes Sent which reflects the number of refresh requests that the Oracle Enterprise Session Border Controller, has sent. For example:

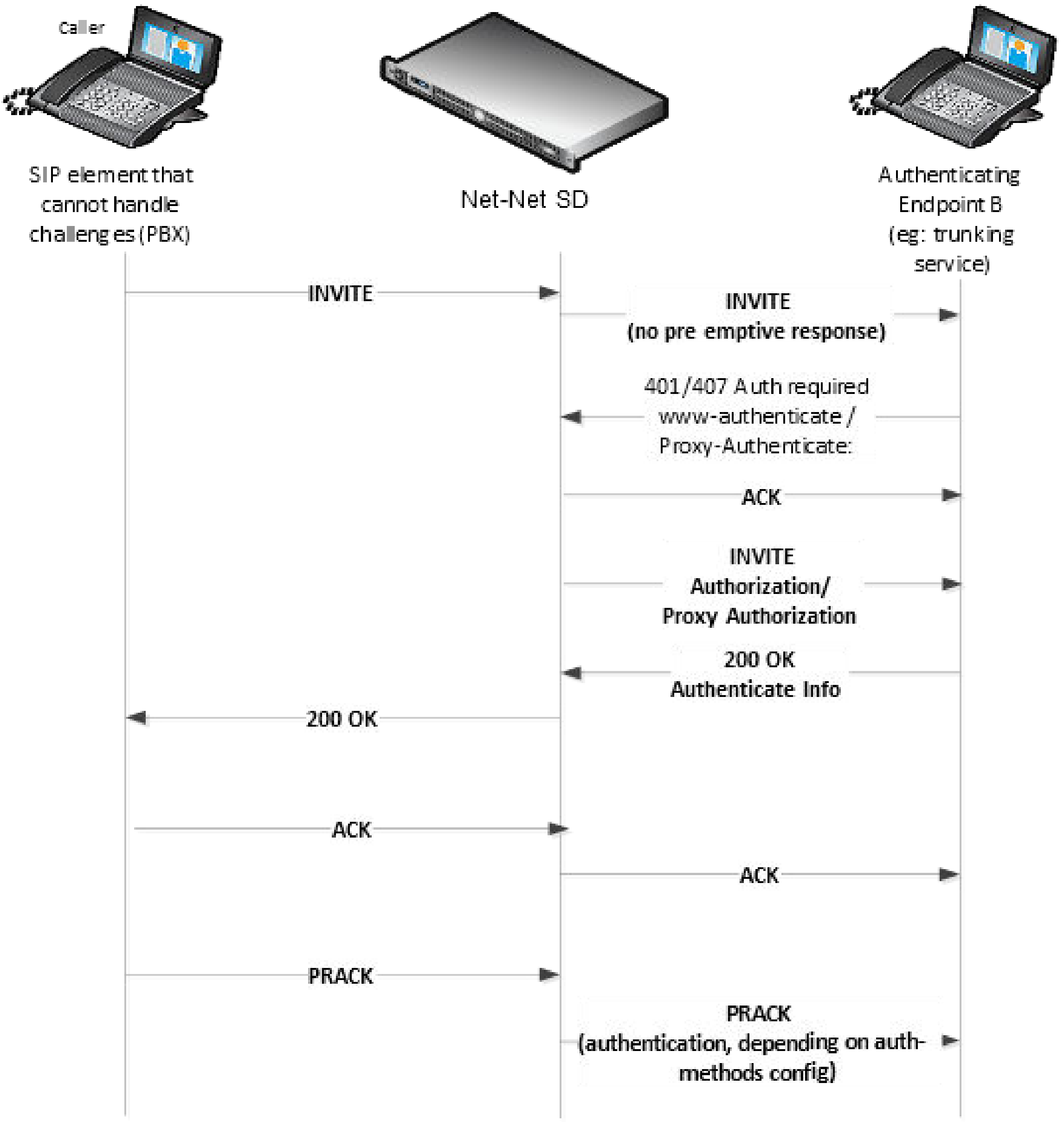
```
ACMEPACKET#show sipd status
SIP Status                -- Period -- -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Sessions             0       1       0          2       1       1
Subscriptions        0       0       0          0       0       0
Dialogs              0       2       0          4       2       2
CallID Map           0       2       0          4       2       2
Rejections           -       -       0          0       0
ReINVITEs            -       -       0          2       1
ReINV Suppress       -       -       0          1       1
Media Sessions       0       1       0          2       1       1
Media Pending        0       0       0          0       0       0
Client Trans         0       3       2         10       3       3
Server Trans         0       0       0          6       3       3
Resp Contexts        0       0       0          6       3       3
Saved Contexts       0       0       0          0       0       0
```

```
Sockets              2       2       0          2       2       2
Req Dropped          -       -       0          0       0
Refreshes Sent       0       0       0          0       0       0
DNS Trans            0       0       0          0       0       0
DNS Sockets          0       0       0          0       0       0
DNS Results          0       0       0          0       0       0
Rejected Msgs        0       0       0          0       0       0
```

Digest Authentication with SIP

Digest authentication for Session Initiation Protocol (SIP) is a type of security feature on the Oracle Enterprise Session Border Controller that provides a minimum level of security for basic Transport Control Protocol (TCP) and User Datagram Protocol (UDP) connections. Digest authentication verifies that both parties on a connection (host and endpoint client) know a shared secret (a password). This verification can be done without sending the password in the clear.

Digest authentication is disabled by default on the Oracle Enterprise Session Border Controller. When digest authentication is enabled, the Oracle Enterprise Session Border Controller (host) responds to authentication challenges from SIP trunking Service Providers (endpoint client). The Oracle Enterprise Session Border Controller performs authentication for each IP-PBX initiating the call. However, the authentication challenge process takes place between the host and the client only since the IP-PBX cannot handle authentication challenges. The following illustration shows the digest authentication process.


Caller
SIP element that cannot handle challeng es (PBX)
Net-Net SD
Authenticating Endpoint B (eg: trunking service)
INVITE
INVITE (no pre emptive response)
401/407 Auth required www-authenticate / Proxy-Authenticate:
ACK
INVITE Authorization/ Proxy Authorization
200 OK Authenticate Info
200 OK
ACK
ACK
PRACK
PRACK (authentication, depending on auth-methods config)

The digest authentication scheme is based on a simple challenge-response paradigm. A valid response contains a checksum (by default, the MD5 checksum) of the “username” and password. In this way, the password is never sent in the clear.

By default, the Oracle Enterprise Session Border Controller uses cached credentials for all requests within the same dialog, once the authentication session is established with a 200OK from the authenticating SIP element. If the in-dialog-methods attribute contains a value, it specifies the requests that have challenge-responses inserted within a dialog.

In digest authentication with SIP, the following can happen:

- More than one authenticating SIP element (IP-PBX) may be the destination of requests.
- More than one authentication challenge can occur in a SIP message. This can occur when there are additional authenticating SIP elements behind the first authenticating SIP element.

- The Oracle Enterprise Session Border Controller distinguishes whether the IP-PBX is capable of handling the challenge. If Digest Authentication is disabled (no auth-attributes configured) on the Session Agent, the challenge is passed back to the IP-PBX.

  Note: If there are multiple challenges in the request, and if the Oracle Enterprise Session Border Controller has only some of the cached credentials configured, the Oracle Enterprise Session Border Controller adds challenge-responses for the requests it can handle, and does not pass the challenge back to the IP-PBX.

Challenge-Responses in Requests not in the Dialog

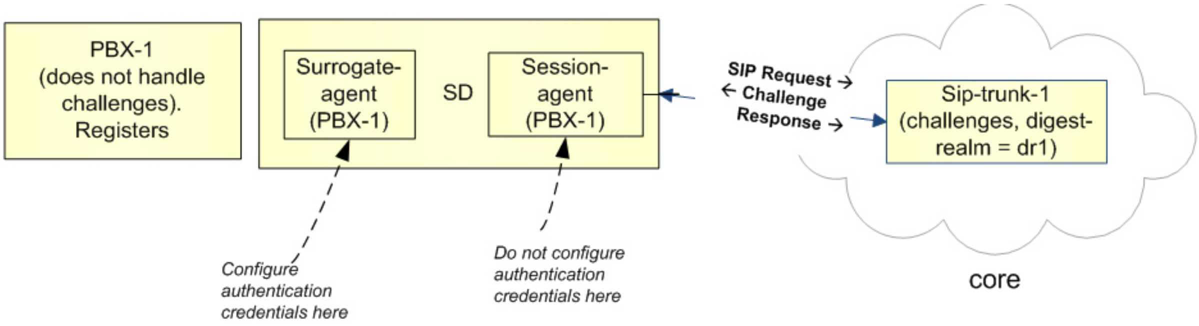
A digest authentication session starts from the client response to a www-authenticate/proxy-authenticate challenge and lasts until the client receives another challenge in the protection space defined by the auth-realm. Credentials are not cached across dialogs; however, if a User Agent (UA) is configured with the auth-realm of its outbound proxy, when one exists, the UA may cache credentials for that auth-realm across dialogs.

Note: Existing Oracle Enterprise Session Border Controller behavior with surrogate-agents is that they cache credentials from REGISTER for INVITE sessions only if the Oracle Enterprise Session Border Controller is considered a UA sending to its outbound proxy.

Surrogate Agents and the Oracle Enterprise Session Border Controller

In the case where a surrogate-agent is configured for the IP-PBX, you do not have to configure digest authentication attributes in the session-agent object for the same IP-PBX. The surrogate-agent authentication configuration takes precedence over the session-agent authentication configuration and so it is ignored.

The following illustration shows an example of a surrogate-agent with a session-agent in the network.



Configuring Digest Authentication

In the Oracle Enterprise Session Border Controller ACLI, you can access the Digest Authentication object at the path session-router->session-agent->**`auth-attribute`**. If enabled, the Digest Authentication process uses the attributes and values listed in this table.

Note: If enabling Digest Authentication, all attributes listed below are required except for the in-dialog-methods attribute which is optional.

The following table lists the digest authentication object

```
ACMEPACKET(auth-attribute)# show
        auth-attribute
                auth-realm                     realm01
                username                       user
                password                       ********
                in-dialog-methods              ACK INVITE SUBSCRIBE
```

To configure digest authentication on the Oracle Enterprise Session Border Controller:

1. In Superuser mode, type configure terminal and press Enter.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press Enter to access the session router-related objects.

```
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)#
```

3. Type session-agent and press Enter to access the session agent-related attributes.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. Type auth-attribute and press Enter to access the digest authentication-related attributes.

```
ACMEPACKET(session-agent)# auth-attribute
ACMEPACKET(auth-attribute)#
```

5. auth-realm — Enter the name (realm ID) of the host realm initiating the authentication challenge. This value defines the protected space in which the digest authentication is performed. Valid value is an alpha-numeric character string. Default is blank.

```
ACMEPACKET(auth-attribute)# auth-realm realm01
```

6. username — Enter the username of the client. Valid value is an alpha-numeric character string. Default is blank.

```
ACMEPACKET(auth-attribute)# username user
```

7. password — Enter the password associated with the username of the client. This is required for all LOGIN attempts. Password displays while typing but is saved in clear-text (i.e., *****). Valid value is an alpha-numeric character string. Default is blank.

```
ACMEPACKET(auth-attribute)# password *******
```

8. in-dialog-methods — Enter the in-dialog request method(s) that digest authentication uses from the cached credentials. Specify request methods in a list form separated by a space enclosed in parentheses. Valid values are:

- INVITE | BYE | ACK | CANCEL | OPTIONS | SUBSCRIBE | PRACK | NOTIFY | UPDATE | REFER

```
ACMEPACKET(auth-attribute)# in-dialog-methods (ack invite subscribe)
```

Note: The methods not in this list are still resubmitted if a 401/407 response is received by the Oracle Enterprise Session Border Controller.

If you do not specify any in-dialog-method value(s), digest authentication does not add challenge-responses to in-dialog requests within a dialog.

This attribute setting applies to in-dialog requests only.

Additional Notes

The following are additional notes that describe the digest authentication process:

- The Oracle Enterprise Session Border Controller always challenges the first LOGIN request, and initial authentication begins with that request. The recalculated authorization key — the credentials — are then included in every subsequent request.
- If the Oracle Enterprise Session Border Controller does not receive any communication from the client within the expiration period, the Oracle Enterprise Session Border Controller logs the client out and tears down the transport connection. Faced with interface loss, the Oracle Enterprise Session Border Controller default behavior is to flush all warrant information from the target database. This response necessitates that the client first login/re-register with the Oracle Enterprise Session Border Controller, and then repopulate the empty database using a series of ADD requests. This behavior ensures that client and Oracle Enterprise Session Border Controller target databases are synchronized.

  Alternatively, when faced with interface loss, the Oracle Enterprise Session Border Controller can retain all warrant information within the target database. This response necessitates only that the client first login/re-register with the Oracle Enterprise Session Border Controller. After successful registration the client should, but is not required to, use a series of GET, ADD, and DELETE requests to ensure that the Oracle Enterprise Session Border Controller and client target databases are synchronized.

- The Oracle Enterprise Session Border Controller ignores the Authentication-Info header that comes in the 200OK response after digest authentication is complete. The Oracle Enterprise Session Border Controller receives a 401/407 response from the client. However, some surrogate-agents may process the Authentication-Info header in a single challenge.

Digest Authentication and High Availability

The Oracle Enterprise Session Border Controller supports digest authentication in high availability (HA) environments. The session-agent configuration, which includes the digest authentication parameters on the primary Oracle Enterprise Session Border Controller, are replicated on the HA Oracle Enterprise Session Border Controller. However, cached credentials on the primary device are not replicated on the HA device.

SDP Insertion for (Re)INVITEs

If your network contains some SIP endpoints that do not send SDP in ReINVITEs but also contains others that refuse INVITEs without SDP, this feature can facilitate communication between the two types. The Oracle Enterprise Session Border Controller can insert SDP into outgoing INVITE messages when the corresponding, incoming INVITE does not contain SDP.

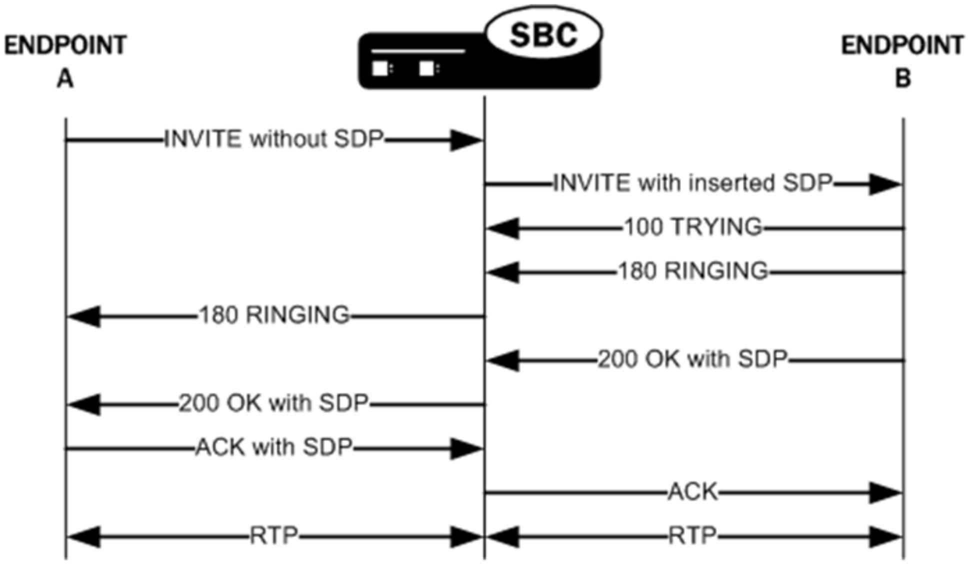
You can also use this feature when the network devices used in H.323-SIP interworking do not include SDP in the INVITEs sent to SIP endpoints. In this case, the Oracle Enterprise Session Border Controller can insert SDP in the outgoing INVITE messages it forwards to the next hop.

This feature works for either INVITEs, ReINVITEs, or both.

This section explains how the SDP insertion feature works for INVITEs and ReINVITEs. The examples used this section are both pure SIP calls. Even when you want to use this feature for IWF calls, though, you configure it for the SIP side.

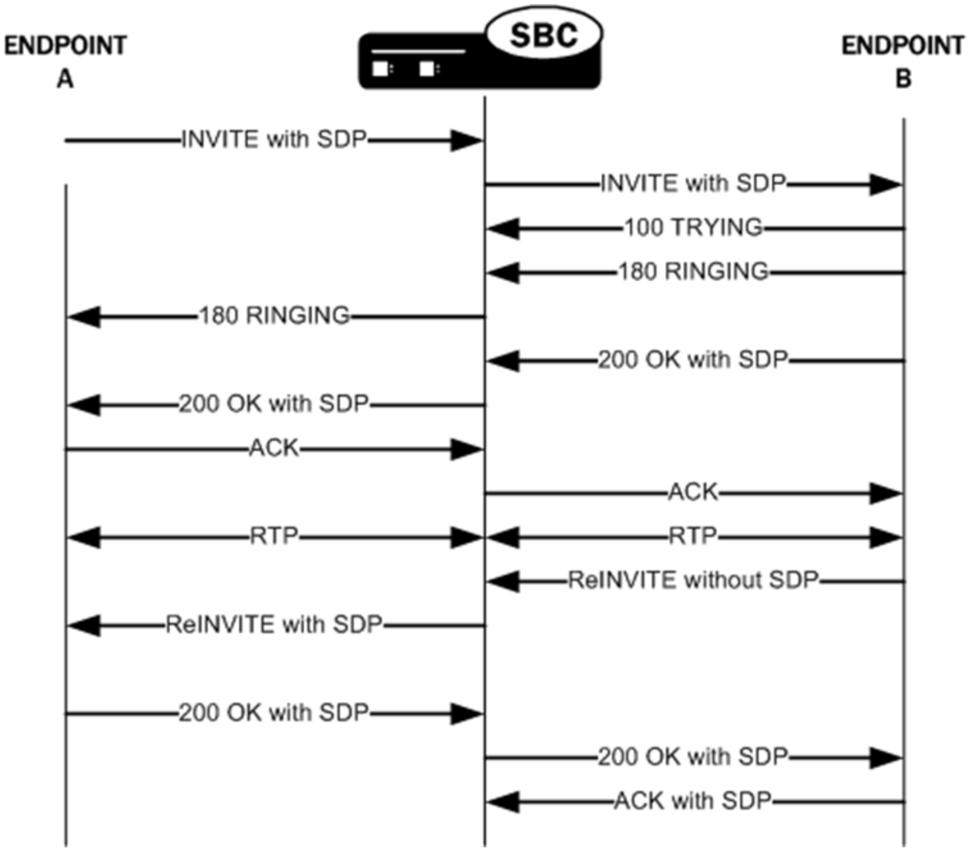
SDP Insertion for SIP INVITES

Appropriately configured, the Oracle Enterprise Session Border Controller inserts SDP into an outgoing INVITE when the corresponding incoming INVITE has none. Because no SDP information is available for the session, the Oracle Enterprise Session Border Controller uses a media profile from a list of them you configure and then apply for SDP insertion.



SDP Insertion for SIP ReINVITEs

The section explains SDP insertion for ReINVITEs, using a case where SIP session has been established with an initial INVITE containing SDP. In the diagram below, you can see the initial INVITE results in a negotiated media stream. But after the media stream is established, Endpoint B sends a ReINVITE without SDP to the Oracle Enterprise Session Border Controller. In this case, the Oracle Enterprise Session Border Controller uses the negotiated media information from the initial INVITE to insert when the ReINVITE has no SDP. It then sends this ReINVITE with inserted SDP to the next hop signaling entity.



SDP Insertion for SIP INVITE Configuration

To work properly, SDP insertion for SIP invites requires you to set a valid media profile configuration.

To enable SDP insertion for INVITEs:

1. Access the **sip-interface** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

2. Select the **sip-interface** object to edit.

```
ACMEPACKET(sip-interface)# select
<RealmID>:
1: realm01 172.172.30.31:5060

selection: 1
ACMEPACKET(sip-interface)#
```

3. **add-sdp-invite**—Change this parameter from disabled (default), and set it to **invite**.
4. **add-sdp-profile**—Enter a list of one or more media profile configurations you want to use when the Oracle Enterprise Session Border Controller inserts SDP into incoming INVITEs that have no SDP. The media profile contains media information the Oracle Enterprise Session Border Controller inserts in outgoing INVITE.

   This parameter is empty by default.
5. Type **done** to save your configuration.

SDP Insertion for SIP ReINVITE Configuration

In this scenario, the Oracle Enterprise Session Border Controller uses the media information negotiated early in the session to insert after it receives an incoming ReINVITE without SDP. The Oracle Enterprise Session Border Controller then sends the ReINVITE with inserted SDP to the next hop signaling entity. You do not need the media profiles setting for ReINVITEs.

To enable SDP insertion for ReINVITEs:

1. Access the **sip-interface** configuration element.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

2. Select the **sip-interface** object to edit.

```
ACMEPACKET(sip-interface)# select
<RealmID>:
1: realm01 172.172.30.31:5060

selection: 1
ACMEPACKET(sip-interface)#
```

3. **add-sdp-invite**—Change this parameter from disabled (default), and set it to **reinvite**.
4. Type **done** to save your configuration.

A

Local CSV Orientation

Appendix A contains three tables that show where, in locally-generated CSV files, specific VSAs appear. There is one table for each of the following type of record: Start, Stop, Interim.

Start Record CSV Placement

CSV Placement	Attribute Name	ACME VSA ID
1	Acct-Status-Type	
2	NAS-IP-Address	
3	NAS-Port	
4	Acct-Session-Id	
5	Acme-Session-Ingress-CallId	3
6	Acme-Session--Egress-CallId	4
7	Acme-Session-Protocol-Type	43
8	Acme-Session-Forked-Call-Id	171
9	Acme-Session--Generic-Id	40
10	Calling-Station-Id	
11	Called-Station-Id	
12	h323-setup-time	
13	h323-connect-time	
14	Acme-Egress-Network-Interface-Id	139
15	Acme-Egress-Vlan-Tag-Value	140
16	Acme-Ingress-Network-Interface-Id	137
17	Acme-Ingress-Vlan-Tag-Value	138
18	Acme-Session-Egress-Realm	42

CSV Placement	Attribute Name	ACME VSA ID
19	Acme-Session-Ingress-Realm	41
20	Acme-FlowId_FS1_F	1
21	Acme-FlowType_FS1_F	2
22	Acme-Flow-In-Realm_FS1_F	10
23	Acme-Flow-In-Src-Addr_FS1_F	11
24	Acme-Flow-In-Src-Port_FS1_F	12
25	Acme-Flow-In-Dst-Addr_FS1_F	13
26	Acme-Flow-In-Dst-Port_FS1_F	14
27	Acme-Flow-Out-Realm_FS1_F	20
28	Acme-Flow-Out-Src-Addr_FS1_F	21
29	Acme-Flow-Out-Src-Port_FS1_F	22
30	Acme-Flow-Out-Dst-Addr_FS1_F	23
31	Acme-Flow-Out-Dst-Port_FS1_F	24
32	Acme-FlowID_FS1_R	78
33	Acme-FlowType_FS1_R	79
34	Acme-Flow-In-Realm_FS1_R	80
35	Acme-Flow-In-Src-Addr_FS1_R	81
36	Acme-Flow-In-Src-Port_FS1_R	82
37	Acme-Flow-In-Dst-Addr_FS1_R	83
38	Acme-Flow-In-Dst-Port_FS1_R	84
39	Acme-Flow-Out-Realm_FS1_R	85
40	Acme-Flow-Out-Src-Addr_FS1_R	86
41	Acme-Flow-Out-Src-Port_FS1_R	87
42	Acme-Flow-Out-Dst-Addr_FS1_R	88
43	Acme-Flow-Out-Dst-Port_FS1_R	89
44	Acme-FlowID_FS2_F	90
45	Acme-FlowType_FS2_F	91
46	Acme-Flow-In-Realm_FS2_F	92
47	Acme-Flow-In-Src-Addr_FS2_F	93
48	Acme-Flow-In-Src-Port_FS2_F	94
49	Acme-Flow-In-Dst-Addr_FS2_F	95
50	Acme-Flow-In-Dst-Port_FS2_F	96
51	Acme-Flow-Out-Realm_FS2_F	97
52	Acme-Flow-Out-Src-Addr_FS2_F	98

CSV Placement	Attribute Name	ACME VSA ID
53	Acme-Flow-Out-Src-Port_FS2_F	99
54	Acme-Flow-Out-Dst-Addr_FS2_F	100
55	Acme-Flow-Out-Dst-Port_FS2_F	101
56	Acme-FlowID_FS2_R	112
57	Acme-FlowType_FS2_R	113
58	Acme-Flow-In-Realm_FS2_R	114
59	Acme-Flow-In-Src-Addr_FS2_R	115
60	Acme-Flow-In-Src-Port_FS2_R	116
61	Acme-Flow-In-Dst-Addr_FS2_R	117
62	Acme-Flow-In-Dst-Port_FS2_R	118
63	Acme-Flow-Out-Realm_FS2_R	119
64	Acme-Flow-Out-Src-Addr_FS2_R	120
65	Acme-Flow-Out-Src-Port_FS2_R	121
66	Acme-Flow-Out-Dst-Addr_FS2_R	122
67	Acme-Flow-Out-Dst-Port_FS2_R	123
68	Acme-Session-Charging-Vector	54
69	Acme-Session-Charging-Function_Address	55
70	Acme-Firmware-Version	56
71	Acme-Local-Time-Zone	57
72	Acme-Post-Dial-Delay	58
73	Acme-Primary-Routing-Number	64
74	Acme-Originating-Trunk-Group	65
75	Acme-Terminating-Trunk-Group	66
76	Acme-Originating-Trunk-Context	67
77	Acme-Terminating-Trunk-Context	68
78	Acme-P-Asserted-ID	69
79	Acme-Ingress-Local-Addr	74
80	Acme-Ingress-Remote-Addr	75
81	Acme-Egress-Local-Addr	76
82	Acme-Egress-Remote-Addr	77
83	Acme-SIP-Diversion	70
84	Acme-Egress-Final-Routing-Number	134
85	Acme-Session-Ingress-RPH	135
86	Acme-Session-Egress-RPH	136

CSV Placement	Attribute Name	ACME VSA ID
87	Acme-Custom-VSA-200	200
88	Acme-Custom-VSA-201	201
89	Acme-Custom-VSA-202	202
90	Acme-Custom-VSA-203	203
91	Acme-Custom-VSA-204	204
92	Acme-Custom-VSA-205	205
93	Acme-Custom-VSA-206	206
94	Acme-Custom-VSA-207	207
95	Acme-Custom-VSA-208	208
96	Acme-Custom-VSA-209	209
97	Acme-Custom-VSA-210	210
98	Acme-Custom-VSA-211	211
99	Acme-Custom-VSA-212	212
100	Acme-Custom-VSA-213	213
101	Acme-Custom-VSA-214	214
102	Acme-Custom-VSA-215	215
103	Acme-Custom-VSA-216	216
104	Acme-Custom-VSA-217	217
105	Acme-Custom-VSA-218	218
106	Acme-Custom-VSA-219	219
107	Acme-Custom-VSA-220	220
108	Acme-Custom-VSA-221	221
109	Acme-Custom-VSA-222	222
110	Acme-Custom-VSA-223	223
111	Acme-Custom-VSA-224	224
112	Acme-Custom-VSA-225	225
113	Acme-Custom-VSA-226	226
114	Acme-Custom-VSA-227	227
115	Acme-Custom-VSA-228	228
116	Acme-Custom-VSA-229	229
117	Acme-Custom-VSA-230	230
118	Acme-Flow-Calling-Media-Stop-Time_FS1	231
119	Acme-Flow-Called-Media-Stop-Time_FS1	232
120	Acme-Flow-Calling-Media-Stop-Time_FS2	233

CSV Placement	Attribute Name	ACME VSA ID
121	Acme-Flow-Called-Media-Stop-Time_FS2	234
122	Acme-FlowMediaType_FS1_F	142
123	Acme-FlowMediaType_FS1_R	143
124	Acme-FlowMediaType_FS2_F	144
125	Acme-FlowMediaType_FS2_R	145
126	Acme-CDR-Sequence-Number	59

Interim Record CSV Placement

CSV Placement	Attribute Name	ACME VSA ID
1	Acct-Status-Type	
2	NAS-IP-Address	
3	NAS-Port	
4	Acct-Session-Id	
5	Acme-Session-Ingress-CallId	3
6	Acme-Session--Egress-CallId	4
7	Acme-Session-Protocol-Type	43
9	Acme-Session-Forked-Call-Id	171
8	Acme-Session--Generic-Id	40
10	Calling-Station-Id	
11	Called-Station-Id	
12	h323-setup-time	
13	h323-connect-time	
14	Acme-Egress-Network-Interface-Id	139
15	Acme-Egress-Vlan-Tag-Value	140
16	Acme-Ingress-Network-Interface-Id	137
17	Acme-Ingress-Vlan-Tag-Value	138
18	Acme-Session-Egress-Realm	42
19	Acme-Session-Ingress-Realm	41
20	Acme-FlowId_FS1_F	1
21	Acme-FlowType_FS1_F	2
22	Acme-Flow-In-Realm_FS1_F	10
23	Acme-Flow-In-Src-Addr_FS1_F	11
24	Acme-Flow-In-Src-Port_FS1_F	12
25	Acme-Flow-In-Dst-Addr_FS1_F	13

CSV Placement	Attribute Name	ACME VSA ID
26	Acme-Flow-In-Dst-Port_FS1_F	14
27	Acme-Flow-Out-Realm_FS1_F	20
28	Acme-Flow-Out-Src-Addr_FS1_F	21
29	Acme-Flow-Out-Src-Port_FS1_F	22
30	Acme-Flow-Out-Dst-Addr_FS1_F	23
31	Acme-Flow-Out-Dst-Port_FS1_F	24
32	Acme-Calling-RTCP-Packets-Lost_FS1	32
33	Acme-Calling-RTCP-Avg-Jitter_FS1	33
34	Acme-Calling-RTCP-Avg-Latency_FS1	34
35	Acme-Calling-RTCP-MaxJitter_FS1	35
36	Acme-Calling-RTCP-MaxLatency_FS1	36
37	Acme-Calling-RTP-Packets-Lost_FS1	37
38	Acme-Calling-RTP-Avg-Jitter_FS1	38
39	Acme-Calling-RTP-MaxJitter_FS1	39
40	Acme-Calling-Octets_FS1	28
41	Acme-Calling-Packets_FS1	29
42	Acme-Calling-R-Factor	151
43	Acme-Calling-MOS	152
44	Acme-FlowID_FS1_R	78
45	Acme-FlowType_FS1_R	79
46	Acme-Flow-In-Realm_FS1_R	80
47	Acme-Flow-In-Src-Addr_FS1_R	81
48	Acme-Flow-In-Src-Port_FS1_R	82
49	Acme-Flow-In-Dst-Addr_FS1_R	83
50	Acme-Flow-In-Dst-Port_FS1_R	84
51	Acme-Flow-Out-Realm_FS1_R	85
52	Acme-Flow-Out-Src-Addr_FS1_R	86
53	Acme-Flow-Out-Src-Port_FS1_R	87
54	Acme-Flow-Out-Dst-Addr_FS1_R	88
55	Acme-Flow-Out-Dst-Port_FS1_R	89
56	Acme-Called-RTCP-Packets-Lost_FS1	46
57	Acme-Called-RTCP-Avg-Jitter_FS1	47
58	Acme-Called-RTCP-Avg-Latency_FS1	48
59	Acme-Called-RTCP-MaxJitter_FS1	49

CSV Placement	Attribute Name	ACME VSA ID
60	Acme-Called-RTCP-MaxLatency_FS1	50
61	Acme-Called-RTP-Packets-Lost_FS1	51
62	Acme-Called-RTP-Avg-Jitter_FS1	52
63	Acme-Called-RTP-MaxJitter_FS1	53
64	Acme-Called-Octets_FS1	44
65	Acme-Called-Packets_FS1	45
66	Acme-Called-R-Factor	153
67	Acme-Called-MOS	154
68	Acme-FlowID_FS2_F	90
69	Acme-FlowType_FS2_F	91
70	Acme-Flow-In-Realm_FS2_F	92
71	Acme-Flow-In-Src-Addr_FS2_F	93
72	Acme-Flow-In-Src-Port_FS2_F	94
73	Acme-Flow-In-Dst-Addr_FS2_F	95
74	Acme-Flow-In-Dst-Port_FS2_F	96
75	Acme-Flow-Out-Realm_FS2_F	97
76	Acme-Flow-Out-Src-Addr_FS2_F	98
77	Acme-Flow-Out-Src-Port_FS2_F	99
78	Acme-Flow-Out-Dst-Addr_FS2_F	100
79	Acme-Flow-Out-Dst-Port_FS2_F	101
80	Acme-Calling-RTCP-Packets-Lost_FS2	104
81	Acme-Calling-RTCP-Avg-Jitter_FS2	105
82	Acme-Calling-RTCP-Avg-Latency_FS2	106
83	Acme-Calling-RTCP-MaxJitter_FS2	107
84	Acme-Calling-RTCP-MaxLatency_FS2	108
85	Acme-Calling-RTP-Packets-Lost_FS2	109
86	Acme-Calling-RTP-Avg-Jitter_FS2	110
87	Acme-Calling-RTP-MaxJitter_FS2	111
88	Acme-Calling-Octets_FS2	102
89	Acme-Calling-Packets_FS2	103
90	Acme-FlowID_FS2_R	112
91	Acme-FlowType_FS2_R	113
92	Acme-Flow-In-Realm_FS2_R	114
93	Acme-Flow-In-Src-Addr_FS2_R	115

CSV Placement	Attribute Name	ACME VSA ID
94	Acme-Flow-In-Src-Port_FS2_R	116
95	Acme-Flow-In-Dst-Addr_FS2_R	117
96	Acme-Flow-In-Dst-Port_FS2_R	118
97	Acme-Flow-Out-Realm_FS2_R	119
98	Acme-Flow-Out-Src-Addr_FS2_R	120
99	Acme-Flow-Out-Src-Port_FS2_R	121
100	Acme-Flow-Out-Dst-Addr_FS2_R	122
101	Acme-Flow-Out-Dst-Port_FS2_R	123
102	Acme-Called-RTCP-Packets-Lost_FS2	126
103	Acme-Called--RTCP-Avg-Jitter_FS2	127
104	Acme-Called--RTCP-Avg-Latency_FS2	128
105	Acme-Called--RTCP-MaxJitter_FS2	129
106	Acme-Called-RTCP-MaxLatency_FS2	130
107	Acme-Called-RTP-Packets-Lost_FS2	131
108	Acme-Called-RTP-Avg-Jitter_FS2	132
109	Acme-Called-RTP-MaxJitter_FS2	133
110	Acme-Called-Octets_FS2	124
111	Acme-Called-Packets_FS2	125
112	Acme-Session-Charging-Vector	54
113	Acme-Session-Charging-Function_Address	55
114	Acme-Firmware-Version	56
115	Acme-Local-Time-Zone	57
116	Acme-Post-Dial-Delay	58
117	Acme-Primary-Routing-Number	64
118	Acme-Originating-Trunk-Group	65
119	Acme-Terminating-Trunk-Group	66
120	Acme-Originating-Trunk-Context	67
121	Acme-Terminating-Trunk-Context	68
122	Acme-P-Asserted-ID	69
123	Acme-Ingress-Local-Addr	74
124	Acme-Ingress-Remote-Addr	75
125	Acme-Egress-Local-Addr	76
126	Acme-Egress-Remote-Addr	77
127	Acme-SIP-Diversion	70

CSV Placement	Attribute Name	ACME VSA ID
128	Acme-Intermediate_Time	63
129	Acct-Session-Time	
130	Acme-Egress-Final-Routing-Number	134
131	Acme-Session-Ingress-RPH	135
132	Acme-Session-Egress-RPH	136
133	Acme-Custom-VSA-200	200
134	Acme-Custom-VSA-201	201
135	Acme-Custom-VSA-202	202
136	Acme-Custom-VSA-203	203
137	Acme-Custom-VSA-204	204
138	Acme-Custom-VSA-205	205
139	Acme-Custom-VSA-206	206
140	Acme-Custom-VSA-207	207
141	Acme-Custom-VSA-208	208
142	Acme-Custom-VSA-209	209
143	Acme-Custom-VSA-210	210
144	Acme-Custom-VSA-211	211
145	Acme-Custom-VSA-212	212
146	Acme-Custom-VSA-213	213
147	Acme-Custom-VSA-214	214
148	Acme-Custom-VSA-215	215
149	Acme-Custom-VSA-216	216
150	Acme-Custom-VSA-217	217
151	Acme-Custom-VSA-218	218
152	Acme-Custom-VSA-219	219
153	Acme-Custom-VSA-220	220
154	Acme-Custom-VSA-221	221
155	Acme-Custom-VSA-222	222
156	Acme-Custom-VSA-223	223
157	Acme-Custom-VSA-224	224
158	Acme-Custom-VSA-225	225
159	Acme-Custom-VSA-226	226
160	Acme-Custom-VSA-227	227
161	Acme-Custom-VSA-228	228

CSV Placement	Attribute Name	ACME VSA ID
162	Acme-Custom-VSA-229	229
163	Acme-Custom-VSA-230	230
164	Acme-Flow-Calling-Media-Stop-Time_FS1	231
165	Acme-Flow-Called-Media-Stop-Time_FS1	232
166	Acme-Flow-Calling-Media-Stop-Time_FS2	233
167	Acme-Flow-Called-Media-Stop-Time_FS2	234
168	Acme-FlowMediaType_FS1_F	142
169	Acme-FlowMediaType_FS1_R	143
170	Acme-FlowMediaType_FS2_F	144
171	Acme-FlowMediaType_FS2_R	145
172	Acme-CDR-Sequence-Number	59

Interim Unsuccessful Record CSV Placement

CSV Placement	AttributeName	ACME VSA ID
1	Acct-Status-Type	
2	NAS-IP-Address	
3	NAS-Port	
4	Acct-Session-Id	
5	Acme-Session-Ingress-CallId	3
6	Acme-Session--Egress-CallId	4
7	Acme-Session-Protocol-Type	43
9	Acme-Session-Forked-Call-Id	171
8	Acme-Session--Generic-Id	40
10	Calling-Station-Id	
11	Called-Station-Id	
12	h323-setup-time	
13	h323-connect-time	
14	Acme-Egress-Network-Interface-Id	139
15	Acme-Egress-Vlan-Tag-Value	140
16	Acme-Ingress-Network-Interface-Id	137
17	Acme-Ingress-Vlan-Tag-Value	138
18	Acme-Session-Egress-Realm	42
19	Acme-Session-Ingress-Realm	41
20	Acme-FlowID_FS1_F	1

CSV Placement	AttributeName	ACME VSA ID
21	Acme-FlowType_FS1_F	2
22	Acme-Flow-In-Realm_FS1_F	10
23	Acme-Flow-In-Src-Addr_FS1_F	11
24	Acme-Flow-In-Src-Port_FS1_F	12
25	Acme-Flow-In-Dst-Addr_FS1_F	13
26	Acme-Flow-In-Dst-Port_FS1_F	14
27	Acme-Flow-Out-Realm_FS1_F	20
28	Acme-Flow-Out-Src-Addr_FS1_F	21
29	Acme-Flow-Out-Src-Port_FS1_F	22
30	Acme-Flow-Out-Dst-Addr_FS1_F	23
31	Acme-Flow-Out-Dst-Port_FS1_F	24
32	Acme-Calling-RTCP-Packets-Lost_FS1	32
33	Acme-Calling-RTCP-Avg-Jitter_FS1	33
34	Acme-Calling-RTCP-Avg-Latency_FS1	34
35	Acme-Calling-RTCP-MaxJitter_FS1	35
36	Acme-Calling-RTCP-MaxLatency_FS1	36
37	Acme-Calling-RTP-Packets-Lost_FS1	37
38	Acme-Calling-RTP-Avg-Jitter_FS1	38
39	Acme-Calling-RTP-MaxJitter_FS1	39
40	Acme-Calling-Octets_FS1	28
41	Acme-Calling-Packets_FS1	29
42	Acme-Calling-R-Factor	151
43	Acme-Calling-MOS	152
44	Acme-FlowID_FS1_R	78
45	Acme-FlowType_FS1_R	79
46	Acme-Flow-In-Realm_FS1_R	80
47	Acme-Flow-In-Src-Addr_FS1_R	81
48	Acme-Flow-In-Src-Port_FS1_R	82
49	Acme-Flow-In-Dst-Addr_FS1_R	83
50	Acme-Flow-In-Dst-Port_FS1_R	84
51	Acme-Flow-Out-Realm_FS1_R	85
52	Acme-Flow-Out-Src-Addr_FS1_R	86
53	Acme-Flow-Out-Src-Port_FS1_R	87
54	Acme-Flow-Out-Dst-Addr_FS1_R	88

CSV Placement	AttributeName	ACME VSA ID
55	Acme-Flow-Out-Dst-Port_FS1_R	89
56	Acme-Called-RTCP-Packets-Lost_FS1	46
57	Acme-Called-RTCP-Avg-Jitter_FS1	47
58	Acme-Called-RTCP-Avg-Latency_FS1	48
59	Acme-Called-RTCP-MaxJitter_FS1	49
60	Acme-Called-RTCP-MaxLatency_FS1	50
61	Acme-Called-RTP-Packets-Lost_FS1	51
62	Acme-Called-RTP-Avg-Jitter_FS1	52
63	Acme-Called-RTP-MaxJitter_FS1	53
64	Acme-Called-Octets_FS1	44
65	Acme-Called-Packets_FS1	45
66	Acme-Called-R-Factor	153
67	Acme-Called-MOS	154
68	Acme-FlowID_FS2_F	90
69	Acme-FlowType_FS2_F	91
70	Acme-Flow-In-Realm_FS2_F	92
71	Acme-Flow-In-Src-Addr_FS2_F	93
72	Acme-Flow-In-Src-Port_FS2_F	94
73	Acme-Flow-In-Dst-Addr_FS2_F	95
74	Acme-Flow-In-Dst-Port_FS2_F	96
75	Acme-Flow-Out-Realm_FS2_F	97
76	Acme-Flow-Out-Src-Addr_FS2_F	98
77	Acme-Flow-Out-Src-Port_FS2_F	99
78	Acme-Flow-Out-Dst-Addr_FS2_F	100
79	Acme-Flow-Out-Dst-Port_FS2_F	101
80	Acme-Calling-RTCP-Packets-Lost_FS2	104
81	Acme-Calling-RTCP-Avg-Jitter_FS2	105
82	Acme-Calling-RTCP-Avg-Latency_FS2	106
83	Acme-Calling-RTCP-MaxJitter_FS2	107
84	Acme-Calling-RTCP-MaxLatency_FS2	108
85	Acme-Calling-RTP-Packets-Lost_FS2	109
86	Acme-Calling-RTP-Avg-Jitter_FS2	110
87	Acme-Calling-RTP-MaxJitter_FS2	111
88	Acme-Calling-Octets_FS2	102

CSV Placement	AttributeName	ACME VSA ID
89	Acme-Calling-Packets_FS2	103
90	Acme-FlowID_FS2_R	112
91	Acme-FlowType_FS2_R	113
92	Acme-Flow-In-Realm_FS2_R	114
93	Acme-Flow-In-Src-Addr_FS2_R	115
94	Acme-Flow-In-Src-Port_FS2_R	116
95	Acme-Flow-In-Dst-Addr_FS2_R	117
96	Acme-Flow-In-Dst-Port_FS2_R	118
97	Acme-Flow-Out-Realm_FS2_R	119
98	Acme-Flow-Out-Src-Addr_FS2_R	120
99	Acme-Flow-Out-Src-Port_FS2_R	121
100	Acme-Flow-Out-Dst-Addr_FS2_R	122
101	Acme-Flow-Out-Dst-Port_FS2_R	123
102	Acme-Called-RTCP-Packets-Lost_FS2	126
103	Acme-Called--RTCP-Avg-Jitter_FS2	127
104	Acme-Called--RTCP-Avg-Latency_FS2	128
105	Acme-Called--RTCP-MaxJitter_FS2	129
106	Acme-Called-RTCP-MaxLatency_FS2	130
107	Acme-Called-RTP-Packets-Lost_FS2	131
108	Acme-Called-RTP-Avg-Jitter_FS2	132
109	Acme-Called-RTP-MaxJitter_FS2	133
110	Acme-Called-Octets_FS2	124
111	Acme-Called-Packets_FS2	125
112	Acme-Firmware-Version	56
113	Acme-Local-Time-Zone	57
114	Acme-Post-Dial-Delay	58
115	Acme-Primary-Routing-Number	64
116	Acme-Originating-Trunk-Group	65
117	Acme-Terminating-Trunk-Group	66
118	Acme-Originating-Trunk-Context	67
119	Acme-Terminating-Trunk-Context	68
120	Acme-P-Asserted-ID	69
121	Acme-Ingress-Local-Addr	74
122	Acme-Ingress-Remote-Addr	75

CSV Placement	AttributeName	ACME VSA ID
123	Acme-Egress-Local-Addr	76
124	Acme-Egress-Remote-Addr	77
125	Acme-SIP-Diversion	70
126	Acme-Intermediate_Time	63
127	Acct-Session-Time	46
128	Acme-Egress-Final-Routing-Number	134
129	Acme-Session-Disposition	60
130	Acme-Disconnect-Initiator	61
131	Acme-Disconnect-Cause	62
132	Acme-SIP-Status	71
133	Acme-Custom-VSA-200	200
134	Acme-Custom-VSA-201	201
135	Acme-Custom-VSA-202	202
136	Acme-Custom-VSA-203	203
137	Acme-Custom-VSA-204	204
138	Acme-Custom-VSA-205	205
139	Acme-Custom-VSA-206	206
140	Acme-Custom-VSA-207	207
141	Acme-Custom-VSA-208	208
142	Acme-Custom-VSA-209	209
143	Acme-Custom-VSA-210	210
144	Acme-Custom-VSA-211	211
145	Acme-Custom-VSA-212	212
146	Acme-Custom-VSA-213	213
147	Acme-Custom-VSA-214	214
148	Acme-Custom-VSA-215	215
149	Acme-Custom-VSA-216	216
150	Acme-Custom-VSA-217	217
151	Acme-Custom-VSA-218	218
152	Acme-Custom-VSA-219	219
153	Acme-Custom-VSA-220	220
154	Acme-Custom-VSA-221	221
155	Acme-Custom-VSA-222	222
156	Acme-Custom-VSA-223	223

CSV Placement	AttributeName	ACME VSA ID
157	Acme-Custom-VSA-224	224
158	Acme-Custom-VSA-225	225
159	Acme-Custom-VSA-226	226
160	Acme-Custom-VSA-227	227
161	Acme-Custom-VSA-228	228
162	Acme-Custom-VSA-229	229
163	Acme-Custom-VSA-230	230
164	Acme-Flow-Calling-Media-Stop-Time_FS1	231
165	Acme-Flow-Called-Media-Stop-Time_FS1	232
166	Acme-Flow-Calling-Media-Stop-Time_FS2	233
167	Acme-Flow-Called-Media-Stop-Time_FS2	234
168	Acme-FlowMediaType_FS1_F	142
169	Acme-FlowMediaType_FS1_R	143
170	Acme-FlowMediaType_FS2_F	144
171	Acme-FlowMediaType_FS2_R	145
172	Acme-CDR-Sequence-Number	59

Stop Record CSV Placement

CSV Placement	Attribute Name	ACME VSA ID
1	Acct-Status-Type	
2	NAS-IP-Address	
3	NAS-Port	
4	Acct-Session-Id	
5	Acme-Session-Ingress-CallId	3
6	Acme-Session--Egress-CallId	4
7	Acme-Session-Protocol-Type	43
8	Acme-Session-Forked-Call-Id	171
9	Acme-Session--Generic-Id	40
10	Calling-Station-Id	
11	Called-Station-Id	
12	Acct-Terminate-Cause	
13	Acct-Session-Time	
14	h323-setup-time	
15	h323-connect-time	

CSV Placement	Attribute Name	ACME VSA ID
16	h323-disconnect-time	
17	h323-disconnect-cause	
18	Acme-Egress-Network-Interface-Id	139
19	Acme-Egress-Vlan-Tag-Value	140
20	Acme-Ingress-Network-Interface-Id	137
21	Acme-Ingress-Vlan-Tag-Value	138
22	Acme-Session-Egress-Realm	42
23	Acme-Session-Ingress-Realm	41
24	Acme-FlowId_FS1_F	1
25	Acme-FlowType_FS1_F	2
26	Acme-Flow-In-Realm_FS1_F	10
27	Acme-Flow-In-Src-Addr_FS1_F	11
28	Acme-Flow-In-Src-Port_FS1_F	12
29	Acme-Flow-In-Dst-Addr_FS1_F	13
30	Acme-Flow-In-Dst-Port_FS1_F	14
31	Acme-Flow-Out-Realm_FS1_F	20
32	Acme-Flow-Out-Src-Addr_FS1_F	21
33	Acme-Flow-Out-Src-Port_FS1_F	22
34	Acme-Flow-Out-Dst-Addr_FS1_F	23
35	Acme-Flow-Out-Dst-Port_FS1_F	24
36	Acme-Calling-RTCP-Packets-Lost_FS1	32
37	Acme-Calling-RTCP-Avg-Jitter_FS1	33
38	Acme-Calling-RTCP-Avg-Latency_FS1	34
39	Acme-Calling-RTCP-MaxJitter_FS1	35
40	Acme-Calling-RTCP-MaxLatency_FS1	36
41	Acme-Calling-RTP-Packets-Lost_FS1	37
42	Acme-Calling-RTP-Avg-Jitter_FS1	38
43	Acme-Calling-RTP-MaxJitter_FS1	39
44	Acme-Calling-Octets_FS1	28
45	Acme-Calling-Packets_FS1	29
46	Acme-Calling-R-Factor	151
47	Acme-Calling-MOS	152
48	Acme-FlowID_FS1_R	78
49	Acme-FlowType_FS1_R	79

CSV Placement	Attribute Name	ACME VSA ID
50	Acme-Flow-In-Realm_FS1_R	80
51	Acme-Flow-In-Src-Addr_FS1_R	81
52	Acme-Flow-In-Src-Port_FS1_R	82
53	Acme-Flow-In-Dst-Addr_FS1_R	83
54	Acme-Flow-In-Dst-Port_FS1_R	84
55	Acme-Flow-Out-Realm_FS1_R	85
56	Acme-Flow-Out-Src-Addr_FS1_R	86
57	Acme-Flow-Out-Src-Port_FS1_R	87
58	Acme-Flow-Out-Dst-Addr_FS1_R	88
59	Acme-Flow-Out-Dst-Port_FS1_R	89
60	Acme-Called-RTCP-Packets-Lost_FS1	46
61	Acme-Called-RTCP-Avg-Jitter_FS1	47
62	Acme-Called-RTCP-Avg-Latency_FS1	48
63	Acme-Called-RTCP-MaxJitter_FS1	49
64	Acme-Called-RTCP-MaxLatency_FS1	50
65	Acme-Called-RTP-Packets-Lost_FS1	51
66	Acme-Called-RTP-Avg-Jitter_FS1	52
67	Acme-Called-RTP-MaxJitter_FS1	53
68	Acme-Called-Octets_FS1	44
69	Acme-Called-Packets_FS1	45
70	Acme-Called-R-Factor	153
71	Acme-Called-MOS	154
72	Acme-FlowID_FS2_F	90
73	Acme-FlowType_FS2_F	91
74	Acme-Flow-In-Realm_FS2_F	92
75	Acme-Flow-In-Src-Addr_FS2_F	93
76	Acme-Flow-In-Src-Port_FS2_F	94
77	Acme-Flow-In-Dst-Addr_FS2_F	95
78	Acme-Flow-In-Dst-Port_FS2_F	96
79	Acme-Flow-Out-Realm_FS2_F	97
80	Acme-Flow-Out-Src-Addr_FS2_F	98
81	Acme-Flow-Out-Src-Port_FS2_F	99
82	Acme-Flow-Out-Dst-Addr_FS2_F	100
83	Acme-Flow-Out-Dst-Port_FS2_F	101

CSV Placement	Attribute Name	ACME VSA ID
84	Acme-Calling-RTCP-Packets-Lost_FS2	104
85	Acme-Calling-RTCP-Avg-Jitter_FS2	105
86	Acme-Calling-RTCP-Avg-Latency_FS2	106
87	Acme-Calling-RTCP-MaxJitter_FS2	107
88	Acme-Calling-RTCP-MaxLatency_FS2	108
89	Acme-Calling-RTP-Packets-Lost_FS2	109
90	Acme-Calling-RTP-Avg-Jitter_FS2	110
91	Acme-Calling-RTP-MaxJitter_FS2	111
92	Acme-Calling-Octets_FS2	102
93	Acme-Calling-Packets_FS2	103
94	Acme-FlowID_FS2_R	112
95	Acme-FlowType_FS2_R	113
96	Acme-Flow-In-Realm_FS2_R	114
97	Acme-Flow-In-Src-Addr_FS2_R	115
98	Acme-Flow-In-Src-Port_FS2_R	116
99	Acme-Flow-In-Dst-Addr_FS2_R	117
100	Acme-Flow-In-Dst-Port_FS2_R	118
101	Acme-Flow-Out-Realm_FS2_R	119
102	Acme-Flow-Out-Src-Addr_FS2_R	120
103	Acme-Flow-Out-Src-Port_FS2_R	121
104	Acme-Flow-Out-Dst-Addr_FS2_R	122
105	Acme-Flow-Out-Dst-Port_FS2_R	123
106	Acme-Called-RTCP-Packets-Lost_FS2	126
107	Acme-Called--RTCP-Avg-Jitter_FS2	127
108	Acme-Called--RTCP-Avg-Latency_FS2	128
109	Acme-Called--RTCP-MaxJitter_FS2	129
110	Acme-Called-RTCP-MaxLatency_FS2	130
111	Acme-Called-RTP-Packets-Lost_FS2	131
112	Acme-Called-RTP-Avg-Jitter_FS2	132
113	Acme-Called-RTP-MaxJitter_FS2	133
114	Acme-Called-Octets_FS2	124
115	Acme-Called-Packets_FS2	125
116	Acme-Session-Charging-Vector	54
117	Acme-Session-Charging-Function-Address	55

CSV Placement	Attribute Name	ACME VSA ID
118	Acme-Firmware-Version	56
119	Acme-Local-Time-Zone	57
120	Acme-Post-Dial-Delay	58
121	Acme-Primary-Routing-Number	64
122	Acme-Originating-Trunk-Group	65
123	Acme-Terminating-Trunk-Group	66
124	Acme-Originating-Trunk-Context	67
125	Acme-Terminating-Trunk-Context	68
126	Acme-P-Asserted-ID	69
127	Acme-Ingress-Local-Addr	74
128	Acme-Ingress-Remote-Addr	75
129	Acme-Egress-Local-Addr	76
130	Acme-Egress-Remote-Addr	77
131	Acme-SIP-Diversion	70
132	Acme-Session-Disposition	60
133	Acme-Disconnect-Initiator	61
134	Acme-Disconnect-Cause	62
135	Acme-SIP-Status	71
136	Acme-Egress-Final-Routing-Number	134
137	Acme-Session-Ingress-RPH	135
138	Acme-Session-Egress-RPH	136
139	Acme-Refer-Call-Transfer-Id	141
140	Acme-Custom-VSA-200	200
141	Acme-Custom-VSA-201	201
142	Acme-Custom-VSA-202	202
143	Acme-Custom-VSA-203	203
144	Acme-Custom-VSA-204	204
145	Acme-Custom-VSA-205	205
146	Acme-Custom-VSA-206	206
147	Acme-Custom-VSA-207	207
148	Acme-Custom-VSA-208	208
149	Acme-Custom-VSA-209	209
150	Acme-Custom-VSA-210	210
151	Acme-Custom-VSA-211	211

CSV Placement	Attribute Name	ACME VSA ID
152	Acme-Custom-VSA-212	212
153	Acme-Custom-VSA-213	213
154	Acme-Custom-VSA-214	214
155	Acme-Custom-VSA-215	215
156	Acme-Custom-VSA-216	216
157	Acme-Custom-VSA-217	217
158	Acme-Custom-VSA-218	218
159	Acme-Custom-VSA-219	219
160	Acme-Custom-VSA-220	220
161	Acme-Custom-VSA-221	221
162	Acme-Custom-VSA-222	222
163	Acme-Custom-VSA-223	223
164	Acme-Custom-VSA-224	224
165	Acme-Custom-VSA-225	225
166	Acme-Custom-VSA-226	226
167	Acme-Custom-VSA-227	227
168	Acme-Custom-VSA-228	228
169	Acme-Custom-VSA-229	229
170	Acme-Custom-VSA-230	230
171	Acme-Flow-Calling-Media-Stop-Time_FS1	231
172	Acme-Flow-Called-Media-Stop-Time_FS1	232
173	Acme-Flow-Calling-Media-Stop-Time_FS2	233
174	Acme-Flow-Called-Media-Stop-Time_FS2	234
175	Acme-FlowMediaType_FS1_F	142
176	Acme-FlowMediaType_FS1_R	143
177	Acme-FlowMediaType_FS2_F	144
178	Acme-FlowMediaType_FS2_R	145
179	Acme-CDR-Sequence-Number	59