Oracle® Enterprise Session Border Controller

Call Traffic Monitoring Guide
Release E-CZ7.5.0

August 2017

Notices

Contents

Preface: About this Guide

The Oracle Communications Enterprise Session Border Controller Call Traffic Monitoring Guide provides information about monitoring the call traffic on your system.

Documentation Set

The E-CZ7.5.0 documentation set differs from previous releases with the addition of separate guides for installation, call traffic monitoring, and header manipulation rules. The content for the new guides was previously located in the *ACLI Configuration Guide*, which no longer contains such information. The documentation set also includes new guides for Federal Information Processing Standard (FIPS) compliance and the Admin Security licenses.

The following table describes the documents included in the ESBC E-CZ7.5.0 documentation set.

Document Name	Document Description
ACLI Configuration Guide	Contains conceptual and procedural information for configuring, administering, and troubleshooting the ESBC.
Administrative Security Guide	Contains conceptual and procedural information for supporting the Admin Security license, the Admin Security ACP license, and JITC on the ESBC.
Call Traffic Monitoring Guide	Contains conceptual and procedural information for configuration using the tools and protocols required to manage call traffic on the ESBC.
FIPS Compliance Guide	Contains conceptual and procedural information about FIPS compliance on the ESBC.
HMR Guide	Contains conceptual and procedural information for header manipulation. Includes rules, use cases, configuration, import, export, and examples.
Installation and Platform Preparation Guide	Contains conceptual and procedural information for system provisioning, software installations, and upgrades.
Release Notes	Contains information about the E-Cz7.5.0 release, including platform support, new features, caveats, known issues, and limitations.
Web GUI User Guide	Contains conceptual and procedural information for using the tools and features of the ESBC Web GUI.

Related Documentation

The following table lists other documentation related to using the ESBC. You can find the listed documents on http://docs.oracle.com/en/industries/communications/ in the "Session Border Controller Documentation" and "Acme Packet" libraries.

Document Name	Document Description
Accounting Guide	Contains information about the ESBC accounting support, including details about RADIUS accounting.
ACLI Reference Guide	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Acme Packet 1100 Hardware Installation Guide	Contains information about the hardware components and features of the Acme Packet 1100, as well as conceptual and procedural information for installation, start-up, operation, and maintenance.
Acme Packet 3900 Hardware Installation Guide	Contains information about the hardware components and features of the Acme Packet 3900, as well as conceptual and procedural information for installation, start-up, operation, and maintenance.
Acme Packet 4500 Hardware Installation Guide	Contains information about the hardware components and features of the Acme Packet 4500, as well as conceptual and procedural information for installation, start-up, operation, and maintenance.
Acme Packet 4600 Hardware Installation Guide	Contains information about the hardware components and features of the Acme Packet 4600, as well as conceptual and procedural information for installation, start-up, operation, and maintenance.
Acme Packet 6300 Hardware Installation Guide	Contains information about the hardware components and features of the Acme Packet 6300, as well as conceptual and procedural information for installation, start-up, operation, and maintenance.
HDR Resource Guide	Contains information about the ESBC Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Maintenance and Troubleshooting Guide	Contains information about ESBC 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 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.

Document Name	Document Description
Security Guide	Contains information about security considerations and best practices from a network and application security perspective for the ESBC family of products.

Revision History

Date	Description
August 2017	• Initial Release

1

Session Recording

SelectiveCall Recording SIPREC

The SIPREC protocol is the protocol used to interact between a Session Recording Client (SRC) (the role performed by Oracle Communications Enterprise Session Border Controller) and a Session Recording Server (SRS) (a 3rd party call recorder or Oracle Communications Interactive Session Recorder's Record and Store Server (RSS)). It controls the recording of media transmitted in the context of a communications session (CS) between multiple user agents.

SIPREC provides a selective-based call recording solution that increases media and signaling performance on a recording server, more robust failovers, and the ability to selectively record. SIPREC also isolates the RSS from the communication session.

The SRC starts a recording session for every call within a configured realm. All call filtering, if desired, must be accomplished by the recording server. The recording server performs the filtering and selection of which sessions it should record.

SIPREC for Active Recording

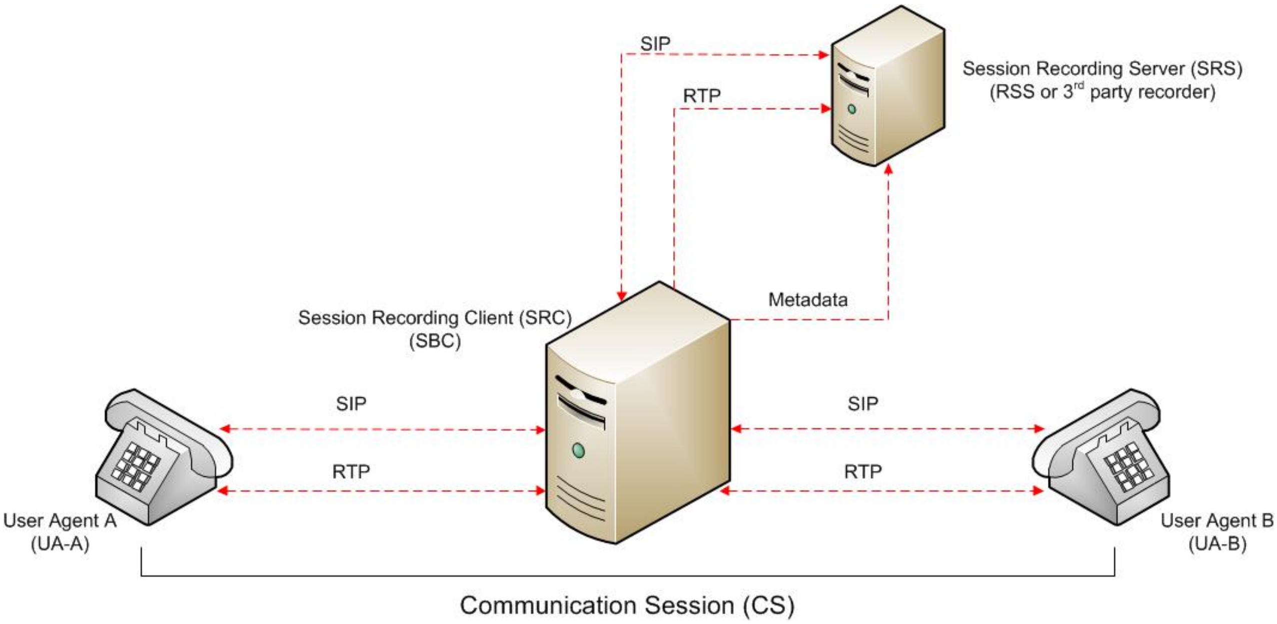
SIPREC supports active recording, where the Oracle Communications Enterprise Session Border Controller acting as the SRC, purposefully streams media to the Oracle Communications Interactive Session Recorder's RSS (or 3rd party call recorder) acting as the SRS. The SRC and SRS act as SIP User Agents (UAs). The SRC provides additional information to the SRS to describe the communication sessions, participants and media streams for the recording session to facilitate archival and retrieval of the recorded information.

The Oracle Communications Enterprise Session Border Controller acting as the SRC, is the source for the recorded media. The Oracle Communications Enterprise Session Border Controller consumes configuration information describing the ecosystem within which it operates. The interface, realm and session agent configuration objects specify the SIPREC configuration. A SIP UA can elect to allow or disallow any network element from recording its media.

During the establishment of a SIP Session, the Oracle Communications Enterprise Session Border Controller determines if SIPREC is configured for recording the call. If so, it then duplicates the media prior to initiating the session with the SRS. (Media replication is set up prior to the recording session). The SRS may choose to record, not record, or cancel the recording session, and then communicates via SIP signaling to the Oracle Communications Enterprise Session Border Controller. If the call is not to be recorded, the SRS signals termination of the recording session.

The Oracle Communications Enterprise Session Border Controller maintains SIPREC metadata information associated with recording sessions. The recording session metadata describes the current state of the recording session and its communication session(s). It is updated when a change of state in the communication session(s) is observed by the Oracle Communications Enterprise Session Border Controller. The SRS is responsible for maintaining call history, etc. The Oracle Communications Enterprise Session Border Controller creates and logs call detail records (CDRs) in the current manner, the 3rd party SRS vendor may collate this information if desired. (For more information about the contents of metadata, see *Metadata Contents*).

The following illustration shows two endpoints, User Agent A (UA-A) and User Agent B (UA-B). Their session is being recorded by an SRC (the Oracle Communications Enterprise Session Border Controller) and an SRS.


SIP
RTP
Session Recording Server (SRS)
(RSS or 3rd party recorder)
Metadata
Session Recording Client (SRC)
(SBC)
SIP
RTP
SIP
RTP
User Agent A
(UA-A)
User Agent B
(UA-B)
Communication Session (CS)

Preserve SIPREC with SIP REFER Header

When the Oracle Communications Enterprise Session Border Controller (ESBC) generates a new INVITE as part of terminating a SIP REFER, the ESBC evaluates the SIPREC configuration of the realms and session agents involved in the new call leg and responds accordingly. The REFER and Transfer mechanism automatically preserves the UCID, XUCID, GUID, GUCID, and UUI in the metadata, and can forward this information to the Session Recording Server. The ESBC can Start, Stop, Pause, and Resume SIPREC sessions in response to any re-INVITE, UPDATE, new INVITE, REFER, or specified SIP Response Message.

The ESBC can establish a new session or update the existing session with the SIPREC server in the following ways.

- When the A-B call leg SA-realm-sipinterface is configured for SIPREC, and the B-C call leg SA-realm-sipinterface is not configured for SIPREC, the ESBC sends metadata to the Session Recording Server to stop the recording on the sessionID associated with the original call.
- When both the A-B call leg and the B-C call leg have the same SIPREC configuration on their SA-realm-sipinterface, the ESBC sends metadata to the Session Recording Server to stop Party A participation and start Party C participation within the same sessionID.
- When the A-B and B-C call legs have a different SIPREC configurations on their SA-realm-sipinterface, the ESBC sends metadata to the A-B call leg Session Recording Server to stop the current recording session and sends metadata to the B-C call leg Session Recording Server to start a new recording session with a new sessionID.

Configuring SIPREC

This section defines the information required to configure SIPREC on the Oracle Communications Enterprise Session Border Controller. It also provides a sample procedure for configuring SIPREC using the Acme Packet Command Line Interface (ACLI).

Session Recording Server (SRS)

The Oracle Communications Interactive Session Recorder's RSS acts as the SRS in the network. A **session-recording-server** attribute under the **session-router** object in the Oracle Communications Enterprise Session Border Controller ACLI allows you to enable/disable the SRS. This object is the session recording server that receives replicated media and records signaling. Additional parameters for SRS are configured under the **session-agent**, **realm-config**, and **sip-interface** objects. The rules of precedence for which the Oracle Communications Enterprise Session Border Controller uses these parameters are:
session-agent takes precedence over the **realm-config**, and **realm-config** takes precedence over **sip-interface**.

Each SRS is associated with a **realm-config**. The realm specifies the source interface from which replicated traffic originates. The destination is an IP Port parameter (IP address or hostname with an optional port) that defines the SIP address (request URI) of the actual SRS.

For an additional level of security, Oracle recommends the SRS be configured in its own realm so as to apply a set of access control lists (ACLs) and security for the replicated communication.

Although the Oracle Communications Enterprise Session Border Controller supports large UDP packets, Oracle recommends the **sip-interface** associated with the SRS realm, be provisioned with a TCP port.

Session Recording Group

The Oracle Communications Enterprise Session Border Controller uses the **session-recording-group** attribute under the **session-router** object in the ACLI to set high availability (HA) for 3rd party call recorders. Using this object, you can define a collection of one or more SRSs. The Oracle Communications Enterprise Session Border Controller utilizes SIP's transport mechanism and keeps track of statistics on each SRS to manage the distribution of traffic and load balancing. (For more information on Oracle Communications Enterprise Session Border Controller load balancing in session recording groups, see *Load Balancing*). When multiple SRSs are in a session recording group, the Oracle Communications Enterprise Session Border Controller uses heuristics to intelligently route the recording dialog to one or more SRSs utilizing the selection strategy.

The **simultaneous-recording-servers** configuration attribute controls the number of simultaneous SIP dialogs that the Oracle Communications Enterprise Session Border Controller establishes to the SRSs in the session recording group per communication session. For instance, if a session recording group contains 3 SRSs, and **simultaneous-recording-servers** is set to **2**, the recording agent initiates a SIP INVITE to the next two SRSs based on the session recording group strategy. In this way, duplicative recording sessions are instantiated, allowing for recording redundancy in multiple SRSs or within a session recording group.

Note: The Oracle Communications Enterprise Session Border Controller streams media to all SRSs. Each SRS chooses whether or not to ignore the media by returning a recvonly(receive only) media line. This permits an SRS to select specific media to record in the recording session, as well as determine whether or not to record the media.

The number of simultaneous recording servers does not dictate the number of recording devices required to be active for a communication session. If two SRSs exist in a session recording group and **simultaneous-recording-servers** is set to **2**, if at least one recording device to any of the servers completes, the recording server is treated as being established.

Load Balancing

The Oracle Communications Enterprise Session Border Controller supports recording server load balancing across members of a session recording group using the following strategies:

Note: SRS groups support “round-robin” and “hunt” strategies only.

[**`Round-robin`**]: The Oracle Communications Enterprise Session Border Controller remembers the last SRS that was used. Each new recording session selects the next SRS in the session recording group. When simultaneous-recording-servers is greater than 1, the next n recording servers are selected from the session recording group.

[**`hunt`**]: The Oracle Communications Enterprise Session Border Controller successively attempts to contact SRSs in the session recording group until a successful recording dialog is established with the SRS, starting from the first SRS in the session recording group. The Oracle Communications Enterprise Session Border Controller attempts to contact each SRS in the session reporting group once. When contact is exhausted, the recording device is considered failed. A SIP failure (response greater than 399, timeout or TCP setup failure) causes the Oracle Communications Enterprise Session Border Controller to attempt the next possible SRS. When simultaneous-recording-servers is greater than 1, the Oracle Communications Enterprise Session Border Controller attempts to establish n recording devices in a hunting fashion.

Session Recording Group within Logical Remote Entities

Each logical remote entity (session-agent, realm-config and sip-interface) has a **`session-recording-server attribute.`**This attribute is a reference to a specific SRS configuration and can be used to specify a session recording group instead. If a session recording group is specified instead of an SRS, the session recording group name must be prefixed with "**`SRG:`**" followed by the session recording group name. This distinguishes between an SRS being referenced and a session recording group being referenced.

With SIPREC, if an SRS or session recording group is configured on both the ingress and egress logical remote entities, both the ingress and egress SRS/session recording groups are used. This means that the Oracle Communications Enterprise Session Border Controller records the media between participants twice (or more) - once for the ingress recorders and once for the egress recorders.

If both the ingress and egress SRS/session recording group are the same, the Oracle Communications Enterprise Session Border Controller makes an optimization and only records the media once. Even if the ingress session recording group is the same exact set of SRSs as the egress session recording group (but with a different name), the Oracle Communications Enterprise Session Border Controller replicates media to both destinations. However, if the same set of SRSs has the exact same identifier, the Oracle Communications Enterprise Session Border Controller sends media to one and not both SRSs.

Selective Recording

SIPREC defines a number of use cases for which the Oracle Communications Enterprise Session Border Controller can record communication sessions. These use cases include the use of selective based recording. A **`selective recording`** is one in which a unique recording server is created per communication session.

Note: The Oracle Communications Enterprise Session Border Controller does not support persistent recording.

For SRSs using selective recording, recording servers are unique per session recording group. For each selective SRS in a session recording group, during the setup of a new communication session, the recording metadata is the same for each recording device. The SRC initiates a new SIP INVITE to the SRS carrying the metadata for that new recording server. The recording agent terminates the SIP dialog at the time that the recording session ends.

The lifetime of a recording session extends beyond the lifetime of the recorded communication. The SRC (Oracle Communications Enterprise Session Border Controller) re-uses the recording session ID in the metadata instead of creating a new ID for each recording.

High Availability (HA) Support

An Oracle Communications Enterprise Session Border Controller using SIPREC supports HA in the network. The Oracle Communications Enterprise Session Border Controller replicates all metadata states between the active and standby Oracle Communications Enterprise Session Border Controllers. Any recording dialogs in progress do not survive the failover, but all calls in progress are preserved. Additionally, the recording dialogs are replicated as well to the failed over Oracle Communications Enterprise Session Border Controller so that in-dialog SIP requests continue to function.

Each recorded communication session replicated to a single SRS counts as two calls instead of one. The Oracle Communications Enterprise Session Border Controller creates two flows between the two participants and two additional flows to the SRS for each of the parent flows.

SIPREC Configuration Procedure

The following configuration example assumes the Oracle Communications Enterprise Session Border Controller has the session recording license enabled on the Oracle Communications Enterprise Session Border Controller. Changes to the call session recording configuration for SIPREC are dynamic. Active calls in progress remain unaffected by the configuration changes. New calls, however, utilize the changes after a **Save** and **Activate** of the configuration.

The following attributes must be configured:

- session-recording-server
- **session-recording-group** (for RSS or 3rd party SRS high availability (HA) only)

and at least one of the following attributes:

- realm-config
- session-agent
- sip-interface

Session-recording-server Attribute

To configure the session-recording-server attribute:

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-recording-server** and press Enter to access the session recording server-related attributes.

```
ACMEPACKET(session-router)# session-recording-server
ACMEPACKET(session-recording-server)#
```

4. **name** — Enter a unique name for the session recording server. This name can be referenced when configuring realm-config, session-agent, and sip-interface. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-server)# name SRS1
```

5. **(optional) description** — Enter a description for the session recording server. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-server)# description <recording server name>
```

6. **realm** — Enter the realm for which the session recording server belongs. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-server)# realm <realm name>
```

Note: Oracle recommends that the session recording server be configured in its own realm.

7. **mode** — Enter the recording mode for the session recording server. Valid values are:
   - **selective** (default) - Unique recording server created per communication session
   - **persistent** - Not supported.

   ```
   ACMEPACKET(session-recording-server)# recording-mode selective
   ```
8. **destination** — Enter the destination IP address with IP port (port specification is optional) that defines the SIP address (request URI) of the session recording server. Enter values in the format 0.0.0.0:<port number>. Default is no value specified.

   ```
   ACMEPACKET(session-recording-server)# destination 172.34.2.3:5060
   ```
9. **port** — Enter the port number to contact the session recording server. Valid values are 1024 to 65535. Default is 5060.
10. **transport-method** — Enter the protocol that the session recording server uses to accept incoming packets from the session reporting client on the network. Default is DynamicTCP. Valid values are:
    - “” - No transport method used. Same as leaving this parameter value blank.
    - UDP - User Datagram Protocol (UDP) is used for transport method.
    - UDP+TCP - UDP and Transmission Control Protocol (TCP) are used for transport method.
    - DynamicTCP - One TCP connection for EACH session is used for the transport method.
    - StaticTCP - Only one TCP connection for ALL sessions is used for the transport method. This option saves resource allocation (such as ports) during session initiation.
    - DynamicTLS - One Transport Layer Security (TLS) connection for EACH session is used for the transport method.
    - StaticTLS - Only one TLS connection for ALL sessions is used for the transport method. This option saves resource allocation (such as ports) during session initiation.
    - DTLS - Datagram TLS is used for the transport method.
    - TLS+DTLS - TLS and DTLS are used for the transport method.
    - StaticSCTP - Only one Stream Control Transmission Protocol (SCTP) connection for ALL sessions is used for the transport method. This option saves resource allocation (such as ports) during session initiation.

      ```
      ACMEPACKET(session-recording-server)# protocol UDP
      ```
11. Enter **done** to save the session recording configuration.

    ```
    ACMEPACKET(session-recording-server)# done
    ```

Configure Session-Recording-Group

The Oracle Communications Enterprise Session Border Controller (ESBC) uses the session-recording-group attribute under session-router to define a collection of session recording servers.

- Enable the SIP Session Recording licence. See "Getting Started."
- Configure multiple session recording servers. See "Session-recording-server Attribute."
- Determine the load balancing strategy that you want the ESBC to use. See "Load Balancing."

In the configuration, you list the session recording servers that you want in the group, select a load balancing strategy, and set the number of simultaneous SIP dialogs.

1. Access the **session-recording-group** configuration element.

   ```
   ORACLEORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# session-recording-group
   ORACLE(session-recording-group)#
   ```
2. Do the following:

Attribute	Instructions
Name	Enter a unique name for the session recording group. You may need this name when configuring

Attribute	Instructions
	realm-config, session-agent, and sip-interface. Valid values: Alpha-numeric characters.
Description (Optional)	Enter a description for the session recording group. Valid values: Alpha-numeric characters.
Session recording servers	Enter the names of the session recording servers that belong to this session recording group. You must enter multiple server names. Valid values: Alpha-numeric characters.
Strategy	Enter the load balancing strategy that you want the ESBC to use when sending recordings to the session reporting server. • Round robin—Go to the next session recording server on the list, since the last session. • Hunt—Look for a session recording server, starting with the first one on the list.
Simultaneous recording servers	Enter the number of simultaneous SIP dialogs that the ESBC establishes to the session recording servers in the session recordng group per communication session. Valid values: 1 - 10. Default: 0.

3. Type **done** to save the configuration.

Realm-config Attribute

Use the following procedure to configure the realm-config attribute and enable session recording:

1. Access the **realm-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# media-manager
ORACLE(media-manager)# realm-config
ORACLE(realm-config)#
```

2. **session-recording-server** — Enter the name of the session-recording server or the session-recording-group in the realm associated with the session reporting client (Oracle Communications Enterprise Session Border Controller). Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(realm-config)# session-recording-server <srs-name>

or

ACMEPACKET(realm-config)# session-recording-server SRG:<group-name>
```

Note: The value for this attribute is the name you specified in Step 4 of the *Session-recording-server Attribute* or Step 4 of the *Session-recording-group Attribute (for HA only)*. If specifying a session-recording-group, you must precede the group name with "SRG:".

3. **session-recording-required** — Enter whether you want a call to be accepted by the Oracle Communications Enterprise Session Border Controller when recording is not available. The default value is **disabled**.

- **Enabled** — Restricts call sessions from being initiated when a recording server is not available.
- **Disabled** (default) — Allows call sessions to initiate even when the recording server is not available.

Note: Oracle recommends that the **session-recording-required** parameter remain disabled.

4. **session-max-life-limit** — Enter the maximum interval in seconds before the SBC must terminate long duration calls. The value supercedes the value of **session-max-life-limit** in the **sip-interface** and *sip-config* configuration elements and is itself superceded by the value of **session-max-life-limit** in the *session-agent* configuration element. The default value is 0 (off/ignored).

   test

5. Type **done** to save your configuration.

Session-agent Attribute

To configure the session-agent attribute and enable session recording:

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

   ```
   ORACLE# configure terminal
   ```

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

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

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

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

4. **`session-recording-server`** — Enter the name of the session-recording server or the session-recording-group to apply to the session recording client (Oracle Communications Enterprise Session Border Controller). Valid values are alpha-numeric characters. Default is no value specified.

   ```
   ORACLE(session-agent)# session-recording-server <srs-name>
   ```

   or

   ```
   ORACLE(session-agent)# session-recording-server SRG:<group-name>
   ```

   Note: The value for this attribute is the name you specified in Step 4 of the *Session-recording-server Attribute* or Step 4 of the *Session-recording-group Attribute (for HA only)*. If specifying a session-recording-group, you must precede the group name with **`SRG:`**.

5. **`session-recording-required`** — Enter whether or not you want a call to be accepted by the Oracle Communications Enterprise Session Border Controller if recording is not available. Valid values are:

   - **`Enabled`** - Restricts call sessions from being initiated when a recording server is not available.
   - **`Disabled`** (default)- Allows call sessions to initiate even if the recording server is not available.

   ```
   ORACLE(session-agent)# session-recording-required disabled
   ```

   Note: Oracle recommends that the session-recording-required parameter remain disabled.

6. Enter **`exit`** to exit the session agent configuration.

   ```
   ORACLE(session-agent)# exit
   ```

7. Enter **`exit`** to exit the session router configuration.

   ```
   ORACLE(session-router)# exit
   ```

8. Enter **`exit`** to exit the configure mode.

   ```
   ORACLE(configure)# exit
   ```

9. Enter **`save-config`** to save the session agent configuration.

   ```
   ORACLE# save-config
   ```

10. Enter **`activate-config`** to activate the session agent configuration.

    ```
    ORACLE# activate-config
    ```

Sip-interface Attribute

To configure the sip-interface attribute and enable session recording:

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

   ```
   ORACLE# configure terminal
   ```

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

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

3. Type **`sip-interface`** and press Enter to access the SIP interface-related attributes.

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

4. **`session-recording-server`** — Enter the name of the session-recording server or the session-recording-group to apply to the SIP interface on the session recording client (Oracle Communications Enterprise Session Border Controller). Valid values are alpha-numeric characters. Default is no value specified.

   ```
   ORACLE(sip-interface)# se ss on-recording-server SRG:<session recording
   server name or session-recording group name>
   ```

   Note: The value for this attribute is the name you specified in Step 4 of the *Session-recording-server Attribute* or Step 4 of the *Session-recording-group Attribute (for HA only)*.

5. **`session-recording-required`** — Enter whether or not you want a call to be accepted by the Oracle Communications Enterprise Session Border Controller if recording is not available. Valid values are:

   - **`Enabled`** - Restricts call sessions from being initiated when a recording server is not available.
   - **`Disabled`** (default)- Allows call sessions to initiate even if the recording server is not available.

   ```
   ORACLE(sip-interface)# session-recording-required disabled
   ```

   Note: Oracle recommends that the session-recording-required parameter remain disabled.

6. Enter **`exit`** to exit the SIP interface configuration.

   ```
   ORACLE(sip-interface)# exit
   ```

7. Enter **`exit`** to exit the session router configuration.

   ```
   ORACLE(session-router)# exit
   ```

8. Enter **`exit`** to exit the configure mode.

   ```
   ORACLE(configure)# exit
   ```

9. Enter **`save-config`** to save the SIP interface configuration.

   ```
   ORACLE# save-config
   ```

10. Enter **`activate-config`** to activate the SIP interface configuration.

    ```
    ORACLE# activate-config
    ```

SIPREC Ping

This SIPREC ping is a signal that the Oracle Communications Enterprise Session Border Controller transmits to the connected SRS requesting a response pertaining to the message type that you specify for the ping-method. It uses the ping-interval to determine how long it should wait before sending another ping to the SRS.

You can check the connectivity by configuring the following parameters:

- **`Ping method`**- SIP message or method for which to ping the SRS.
- **`Ping interval`**- Amount of time, in seconds, that the Oracle Communications Enterprise Session Border Controller waits before it pings the SRS in subsequent intervals. For example, if this parameter is

set for 60 seconds, the Oracle Communications Enterprise Session Border Controller pings the SRS every 60 seconds.

Once configured the Oracle Communications Enterprise Session Border Controller uses this feature to perform SIP-based pinging to determine if the SRS is reachable or not.

Configuring SIPREC Ping on theOracle Communications Enterprise Session Border Controller.

To configure SIPREC ping on the Oracle Communications Enterprise Session Border Controller, you use the ping-method and the ping-interval objects under call-recording-server. Use the following procedure to configure SIPREC ping on the Oracle Communications Enterprise Session Border Controller.

To configure SIPREC ping:

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 call-recording-server and press Enter.

```
ACMEPACKET(session-router)# call-recording-server
ACMEPACKET(call-recording-server)#
```

4. ping-method—Enter the message or method type for which the Net-Net ESD uses in a ping request to the SRS to determine if it is reachable or not. Default is blank. Valid values are:

BYE	OPTIONS
UPDATE	SUBSCRIBE
CANCEL	NOTIFY

5. ping-interval—Enter the amount of time, in seconds, that the Net-Net ESD waits before it pings the SRS in subsequent intervals. Valid values are 0 to 99999. Default is 0 (zero). The setting of zero disables the ping interval.
6. Type done and press Enter.

```
ACMEPACKET(call-recording-server)# done
ACMEPACKET(call-recording-server)#
```

7. Type exit and press Enter.

```
ACMEPACKET(call-recording-server)# exit
ACMEPACKET(session-router)#
```

8. Type exit and press Enter.

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

9. Save the configuration.

Example SIPREC Ping Configuration

The following is an example of a SIPREC ping configuration.

```
call-recording-server# show
     name                 SRS1
     description          session recording server
     realm                realmA
     mode                 selective
     destination          132.43.5.6
     port                 5060
```

```
    transport-method       DynamicTCP
    ping-method            OPTIONS
    ping-interval          60
```

In the above example, the Net-Net ESD sends a ping request to the SRS using the OPTIONS value every 60 seconds to determine if the SRS is reachable or not.

Metadata Contents

The recording metadata contains a set of related elements which define the recording session. A recording session may contain zero or more communication sessions and/or communication session groups. A communication session represents a call instance; a communication session group represents a related group of communication sessions. A recording session is composed of a sequence of complex element types. Not all element types are required to describe a recording session initiated from the Oracle Communications Enterprise Session Border Controller. The recording session XML schema defines the following element types:

- **`dataMode`** - partial or complete metadata description (required)
- **`group`** - a collection of related communication sessions
- **`session`** - a single communication session of two or more participants (required)
- **`participant`** - a SIP endpoint representation (required)
- **`stream`** - a media stream
- **`extensiondata`** - application specific data outside of the SIPREC scope.

The recording agent generates dataMode, session, participant, and stream elements. Extension data is attached to other elements within the metadata through the use of the parent attribute. The recording metadata is defined as a sequence of element types; therefore all associations between elements are represented as references to element identifiers.

The state of the metadata within a recording session reflects the state of the communication session(s) which is being recorded. SIPREC implements stop-times and reason codes when communication sessions end within a recording session. Once a communication session, participant, or media stream has been marked as 'stopped' and accepted by the SRS, the metadata item is removed from the current metadata state. In addition, media lines within the SDP or the recording session may be re-used/re-labeled for reuse if new communication sessions and media streams are created within the recording session.

The XML schema for the recording metadata is defined in the IETF draft RFC *draft-ram-siprec-metadata-format-02* [7].

The ACLI command to show recorded metadata is **`show rec`**. For more information on this command see the section, *Show rec*.

Show Commands for Recording Sessions

The Oracle Communications Enterprise Session Border Controller allows you to utilize the following **`show`** commands via the ACLI to display statistical information about recording sessions:

- show rec
- show rec redundancy

Show rec

The **`show rec`** command displays the count of all metadata objects in sessions managed by the recording agent. These statistics include metadata monitored over an active **`period`** of time and over a lifetime period (where lifetime totals reflect from the last reboot of the Oracle Communications Enterprise Session Border Controller to the present time). The following example shows the use of this command.

1. Log into the Oracle Communications Enterprise Session Border Controller as a User or Superuser.

```
ACMEPACKET> enable
ACMEPACKET(enable)#
```

2. Type **show rec** and press Enter to display the recording metadata statistics. The following output is an example of the show rec command.

```
ACMEPACKET(enable)# show rec
```

Show rec output

```
13:49:44-81645
Recording Agent Status        -- Period -- -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Rec Sessions         0       1       1          1       1       1
Comm Groups          0       0       0          0       0       0
Comm Sessions        0       1       1          1       1       1
Media Streams        0       2       2          2       2       2
Participants         0       2       2          2       2       2
```

The following table describes the metadata objects in the show rec command output.

Object	Description
Rec Sessions	Number of recording sessions during an active period of time and over a lifetime period.
Comm Groups	Number of active communication session recording groups during an active period of time and over a lifetime period.
Comm Sessions	Number of active communication sessions during an active period of time and over a lifetime period.
Media Streams	Number of active media streams during an active period of time and over a lifetime period.
Participants	Total number of participants in session recordings during an active period of time and over a lifetime period.

Show rec redundancy

The **show rec redundancy** command displays information for session recording server statistics when the Oracle Communications Enterprise Session Border Controller is configured for HA. These statistics include metadata monitored over an active period of time and over a lifetime period (where lifetime totals reflect from the last reboot of the Oracle Communications Enterprise Session Border Controller to the present time) on both the primary and redundant Oracle Communications Enterprise Session Border Controller. The following example shows the use of this command.

1. Log into the Oracle Communications Enterprise Session Border Controller as a User or Superuser.

```
ACMEPACKET> enable
ACMEPACKET(enable)#
```

2. Type **show rec redundancy** and press Enter to display the session recording server statistics for Oracle Communications Enterprise Session Border Controllers in HA mode. The following output is an example of the show rec redundancy command.

```
ACMEPACKET(enable)# show rec redundancy
```

Show rec redundancy output

Primary System

```
13:49:44-81645
Recording Agent Status        -- Period -- -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Rec Sessions         0       1       1          1       1       1
Comm Groups          0       0       0          0       0       0
Comm Sessions        0       1       1          1       1       1
Media Streams        0       2       2          2       2       2
Participants         0       2       2          2       2       2
```

```
Redundant System
13:49:44-81646
Recording Agent Status        -- Period -- -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Rec Sessions         0       1       1          1       1       1
Comm Groups          0       0       0          0       0       0
Comm Sessions        0       1       1          1       1       1
Media Streams        0       2       2          2       2       2
Participants         0       2       2          2       2       2
```

The following table describes the session recording server statistics in the **`show rec redundancy`** command output.

Object	Description
Rec Sessions	Number of recording sessions during an active period of time and over a lifetime period.
Comm Groups	Number of active communication session recording groups during an active period of time and over a lifetime period.
Comm Sessions	Number of active communication sessions during an active period of time and over a lifetime period.
Media Streams	Number of active media streams during an active period of time and over a lifetime period.
Participants	Total number of participants in session recordings during an active period of time and over a lifetime period.

Codec Negotiation

In a SIPREC environment, it is assumed that the recording ecosystem provides transcoding media servers for which media calls can be redirected to, relieving the issue of codec matching from the recording servers. However, if transcoding media servers are not provided, the responsibility for transcoding falls on the recording server or the recording client in a SIPREC environment. The Oracle Communications Enterprise Session Border Controller/SRC is required to impose some policy decisions on the codec negotiation between the three, or more, end-points. Specifically, the codec negotiation between the two participants and the recording server is subject to additional policy actions.

The SDP answer from the SRS may not agree with the media flows established in the communication session between UA-A and UA-B. If UA-A and UA-B agree to use G729, yet the SRS's answer indicates no support for G729, the SRS is then unable to interpret the media streams. The SDP offer forwarded to the called party (in this case UA-B) limits the codec choices to those supported by the SRS.

Note: The recording agent forwards the original codec offer to the SRS prior to sending the invite to the UA-B. The SRS responds with the SDP answer, indicating the codec list most desirable to the SRS. The codec list in the answer is then forwarded to UA-B. This allows three parties in a conference call to participate in the negotiation of the codecs among the supported formats only.

SIPREC Call Flows

This section provides examples of call flow scenarios that can occur in a SIPREC environment. SIP recording call flow examples include:

For Selective Recording:

- *Normal Call (recording required)*
- *Normal Call (recording not required)*
- *Early Media Call (recording not required)*
- *REFER Pass-Through Call (REFER handled by User Agent)*

- *REFER Call (REFER handled by the Oracle Communications Enterprise Session Border Controller)*
- *SRS Indicates Busy in Call (recording not required)*
- *Call Transfer scenario*

Note: REFER is a SIP method indicating that the recipient (identified by the Request-URI) should contact a third party using the contact information provided in the request.

SIPREC Re-INVITE Collision and Back-off Support

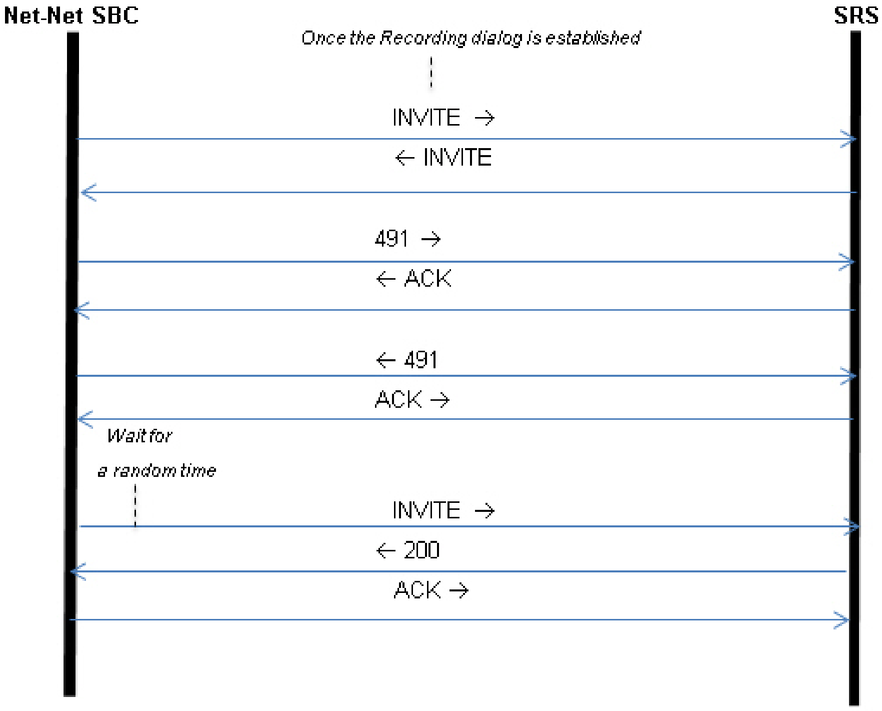
The Oracle SBC acts a back-to-back User Agent (B2BUA) in all call scenarios. However with SIPREC, the Oracle SBC acts as a User Agent Client (UAC) when connected with a session recording server (SRS). Therefore, SIP requests can originate from the Oracle SBC.

During a recording session, when the SRS establishes a recording dialog, the Oracle SBC and the SRS may send Re-INVITES to each other with updated information. When the Oracle SBC receives an INVITE while it is still waiting for the response to a previous INVITE it sent out, this produces an INVITE collision.

To avoid an INVITE collision, the Oracle SBC now sends a 491 Request Pending response back to the SRS and then waits for a random amount of time before re-trying the INVITE. It also acknowledges (ACK) any 491 response received from the other side. RFC 3261 and RFC 6141 describes the way the User Agent (UA) resolves the INVITE collision. The random wait time is chosen based on the following guidelines from RFC 3261:

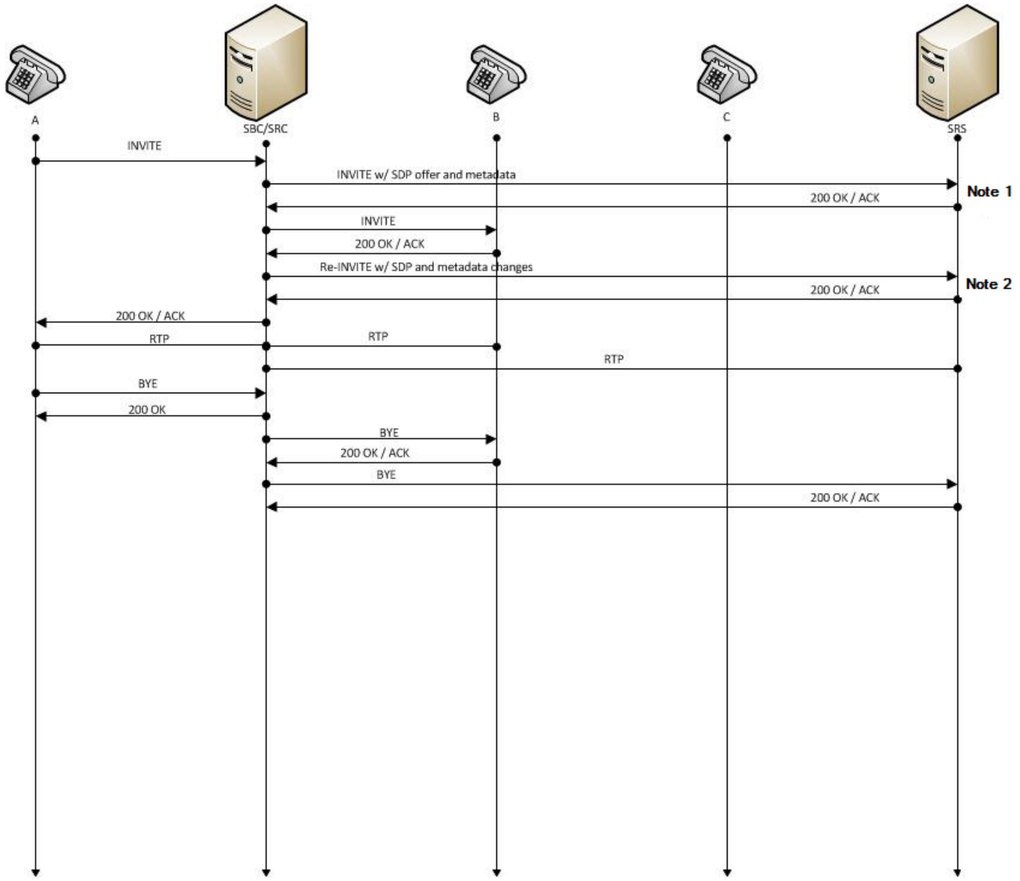
- If the UAC is the owner of the Call-ID of the dialog ID (i.e., it generated the value), T (the wait time) is a randomly chosen value between 2.1 and 4 seconds in units of 10 milliseconds.
- If the UAC is not the owner of the Call-ID of the dialog ID (i.e., it did not generate the value), T (the wait time) is a randomly chosen value between 0 and 2 seconds in units of 10 milliseconds.

The following call flow diagram shows the Oracle SBC’s feature to avoid INVITE collision.



Selective Recording
Normal Call (recording required)

The following illustration shows a normal call using selective recording with recording required. For SDP and Metadata information in Notes 1 and 2 , see *Sample SDP and Metadata*.



SIP INVITE, “recording required”, selective recording

I

Call Flow Description	
① UA-A sends INVITE to Oracle Communications Enterprise Session Border Controller.	⑩ RTP stream initiated between Oracle Communications Enterprise Session Border Controller and UA-B.
② Oracle Communications Enterprise Session Border Controller forwards INVITE with SDP and metadata to SRS.	⑪ RTP stream initiated between Oracle Communications Enterprise Session Border Controller and SRS.
③ SRS responds with OK to Oracle Communications Enterprise Session Border Controller.	⑫ UA-A sends BYE to Oracle Communications Enterprise Session Border Controller.
④ Oracle Communications Enterprise Session Border Controller sends INVITE to UA-B.	⑬ Oracle Communications Enterprise Session Border Controller responds with OK to UA-A.
⑤ UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.	⑭ Oracle Communications Enterprise Session Border Controller sends BYE to Oracle Communications Enterprise Session Border Controller.
⑥ Oracle Communications Enterprise Session Border Controller sends re-INVITE with SDP and metadata changes to SRS.	⑮ Oracle Communications Enterprise Session Border Controller responds with OK to UA-A.

Call Flow Description	
⑦ SRS responds with OK to Oracle Communications Enterprise Session Border Controller.	⑯ Oracle Communications Enterprise Session Border Controller sends BYE to UA-B.
⑧ Oracle Communications Enterprise Session Border Controller forwards OK response to UA-A.	⑰ UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.
⑨ RTP stream initiated between UA-A and Oracle Communications Enterprise Session Border Controller.	⑱ Oracle Communications Enterprise Session Border Controller sends BYE to SRS.
	⑲ SRS responds with OK to Oracle Communications Enterprise Session Border Controller.

Sample SDP and Metadata

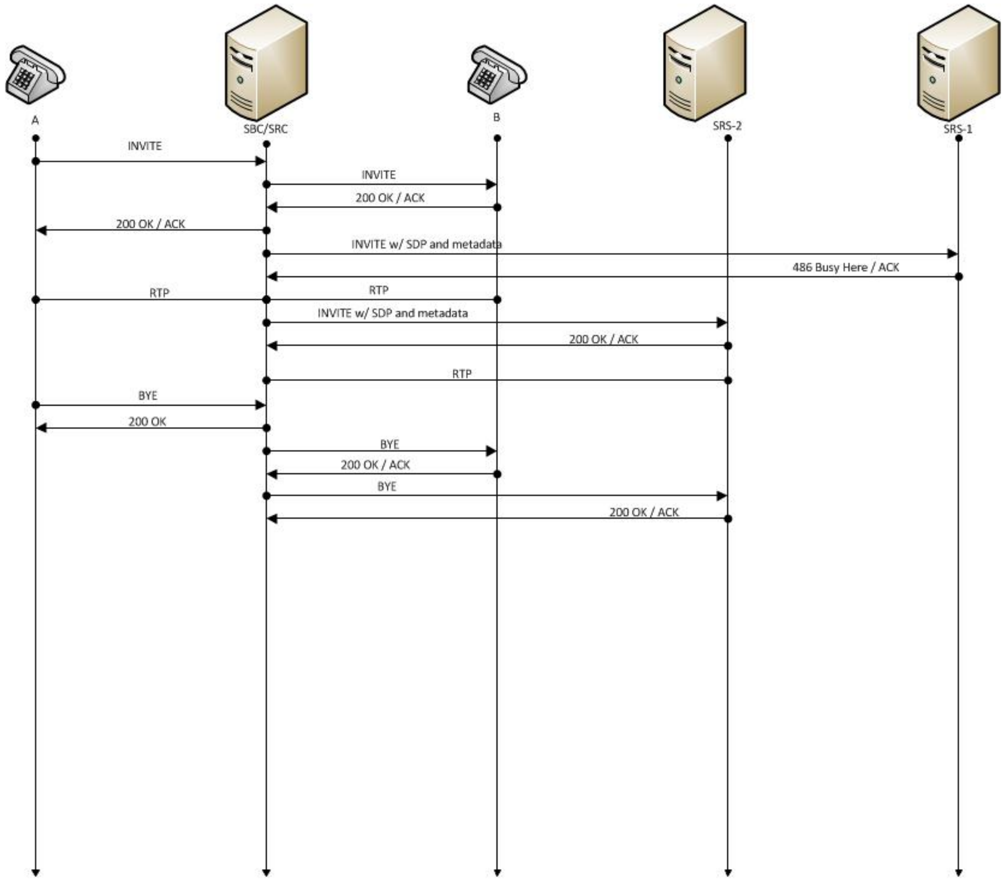
The following sample SDP and Metadata pertain to Notes 1 and 2 in the previous Call Flow diagram.

```
--[Note 1]------------------------------
Content-Type: application/sdp
v=0
o=- 171 213 IN IP4 10.0.0.2
s=-
c=IN IP4 10.0.0.1
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:1

Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session
<?xml version='1.0' encoding='UTF-8'?>
<recording xmlns='urn:ietf:params:xml:ns:recording'>
        <dataMode>complete</dataMode>
        <session id="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <start-time>2011-06-27T17:03:57</start-time>
        </session>
        <participant id="urn:uuid:10ac9063-76b7-40bb-4587-08ba290d7327"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <aor>sip:sipp@168.192.24.40</aor>
                <name>sipp </name>
                <send>urn:uuid:07868c77-ef8e-4d6f-6dd5-a02ff53a1329</send>
                <start-time>2011-06-27T17:03:57</start-time>
        </participant>
        <participant id="urn:uuid:797c45f5-e765-4b12-52b0-d9be31138529"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <aor>sip:service@168.192.24.60</aor>
                <name>sut </name>
        </participant>
        <stream id="urn:uuid:4a72a1ed-abb2-4d7c-5f4d-6d4c36e2d4ec"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <mode>separate</mode>
                <start-time>2011-06-27T17:03:57</start-time>
                <label>1</label>
        </stream>
</recording>


--[Note 2]------------------------------
```

```
Content-Type: application/sdp
v=0
o=- 171 213 IN IP4 10.0.0.2
s=-
c=IN IP4 10.0.0.1
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:1
m=audio 6002 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:2

Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session
<?xml version='1.0' encoding='UTF-8'?>
<recording xmlns='urn:ietf:params:xml:ns:recording'>
        <dataMode>partial</dataMode>
        <session id="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <start-time>2011-06-27T17:03:57</start-time>
        </session>
        <participant id="urn:uuid:797c45f5-e765-4b12-52b0-d9be31138529"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <aor>sip:service@168.192.24.60</aor>
                <name>sut </name>
                <send>urn:uuid:4a72a1ed-abb2-4d7c-5f4d-6d4c36e2d4ec</send>
                <start-time>2011-06-27T17:03:58</start-time>
        </participant>
        <stream id="urn:uuid:07868c77-ef8e-4d6f-6dd5-a02ff53a1329"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <mode>separate</mode>
                <start-time>2011-06-27T17:03:58</start-time>
                <label>2</label>
        </stream>
</recording>
```

Normal Call (recording not required)

The following illustration shows a normal call using selective recording with recording optional.

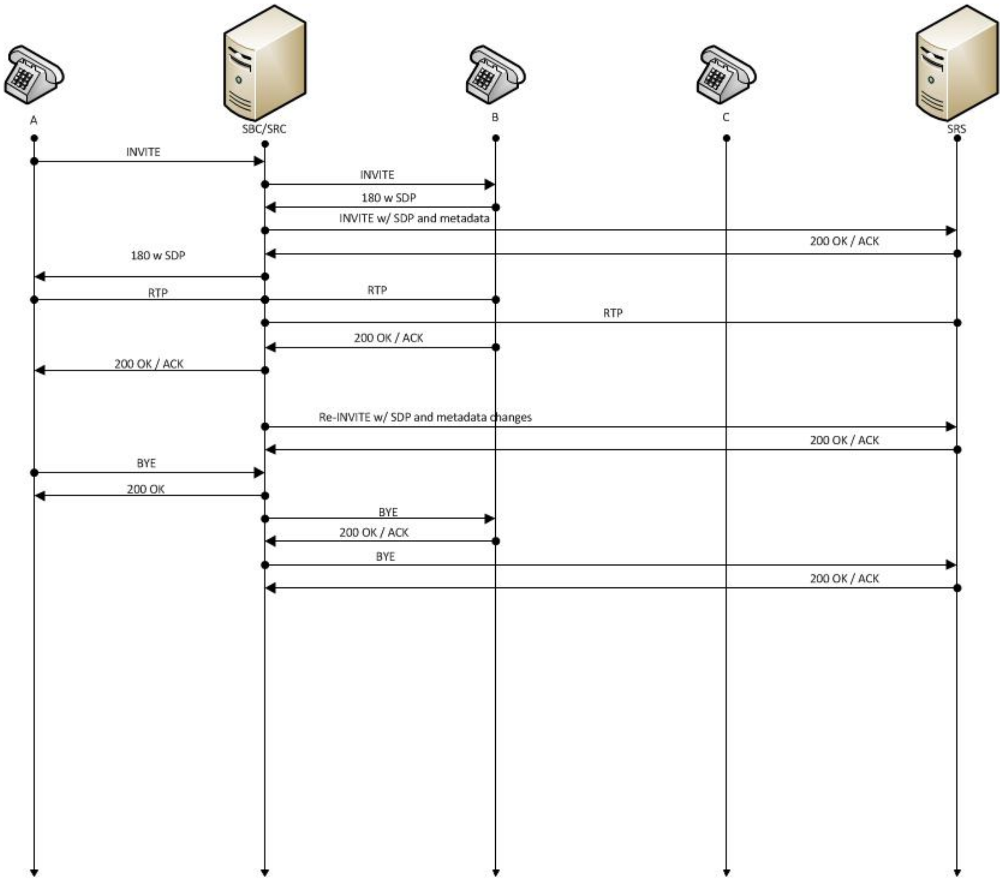


SIP INVITE, “recording not required”

Call Flow Description	
① UA-A sends INVITE to Oracle Communications Enterprise Session Border Controller.	⑧ RTP stream initiated between Oracle Communications Enterprise Session Border Controller and SRS.
② Oracle Communications Enterprise Session Border Controller forwards INVITE to UA-B.	⑨ UA-A sends BYE to Oracle Communications Enterprise Session Border Controller.
③ UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.	⑩ Oracle Communications Enterprise Session Border Controller responds with OK to UA-A.
④ Oracle Communications Enterprise Session Border Controller forwards OK response to UA-A.	⑪ Oracle Communications Enterprise Session Border Controller sends BYE to UA-B.
⑤ Oracle Communications Enterprise Session Border Controller sends INVITE with SDP and metadata to SRS.	⑫ UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.
⑥ SRS responds with OK to Oracle Communications Enterprise Session Border Controller.	⑬ Oracle Communications Enterprise Session Border Controller sends BYE to SRS.
⑦ RTP stream initiated between UA-A, Oracle Communications Enterprise Session Border Controller, and UA-B.	⑭ SRS responds with OK to Oracle Communications Enterprise Session Border Controller.

Early Media Call (recording not required)

The following illustration shows an early media call using selective recording with recording optional.



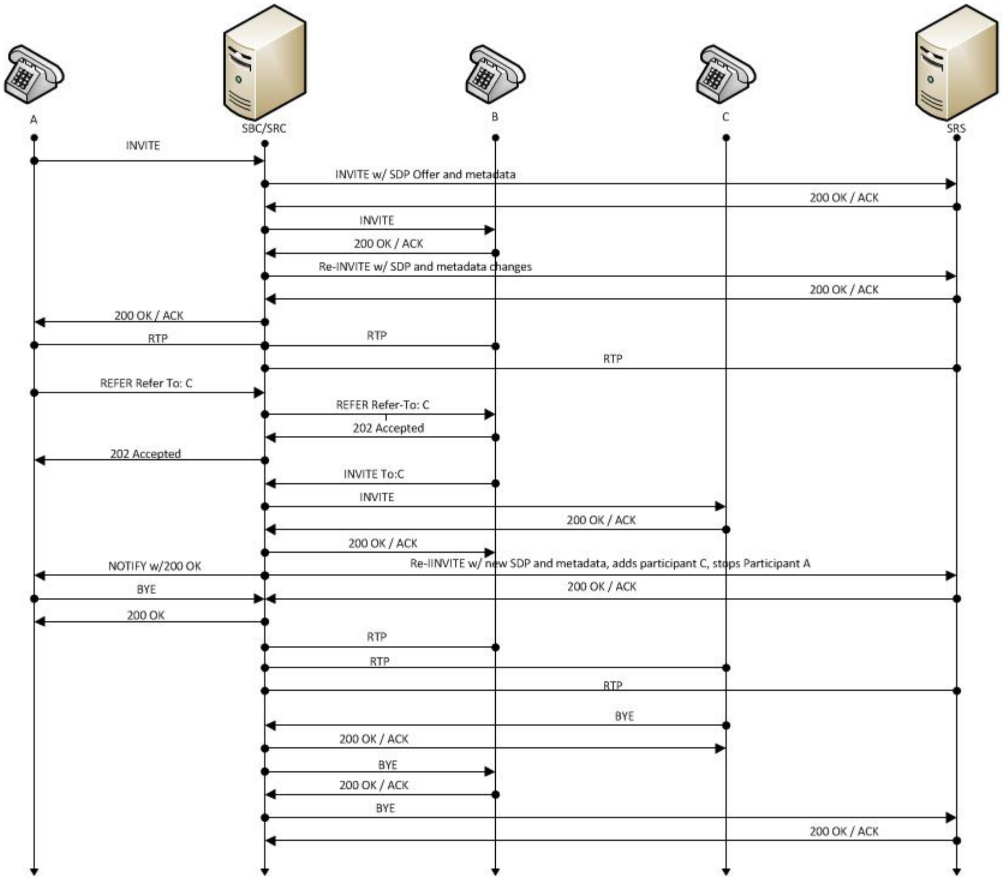
SIP INVITE, early media , “recording not required”, selective recording

Call Flow Description	
① UA-A sends INVITE to Oracle Communications Enterprise Session Border Controller.	⑩ UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.
② Oracle Communications Enterprise Session Border Controller forwards INVITE to UA-B.	⑪ Oracle Communications Enterprise Session Border Controller forwards OK to UA-A.
③ UA-B sends 180 and SDP to Oracle Communications Enterprise Session Border Controller.	⑫ Oracle Communications Enterprise Session Border Controller sends re-INVITE with SDP and metadata changes to SRS.
④ Oracle Communications Enterprise Session Border Controller sends INVITE with SDP and metadata to SRS.	⑬ SRS responds with OK to Oracle Communications Enterprise Session Border Controller.
⑤ SRS responds with OK to Oracle Communications Enterprise Session Border Controller.	⑭ UA-A sends BYE to Oracle Communications Enterprise Session Border Controller.
⑥ Oracle Communications Enterprise Session Border Controller sends 180 with SDP to UA-A.	⑮ Oracle Communications Enterprise Session Border Controller responds with OK to UA-A.

Call Flow Description	
⑦ RTP stream initiated between Oracle Communications Enterprise Session Border Controller and UA-A.	⑯ Oracle Communications Enterprise Session Border Controller sends BYE to UA-B.
⑧ RTP stream initiated between Oracle Communications Enterprise Session Border Controller and UA-B.	⑰ UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.
⑨ RTP stream initiated between Oracle Communications Enterprise Session Border Controller and SRS.	⑱ Oracle Communications Enterprise Session Border Controller sends BYE to SRS.
	⑲ SRS responds with OK to Oracle Communications Enterprise Session Border Controller.

REFER Pass-Through Call (REFER handled by User Agent)

The following illustration shows a REFER pass-through call using selective recording and the User Agent (UA) handling the REFER on the call. Recording is required in this call flow.



SIP REFER, UA handles REFER, “recording required”, selective recording

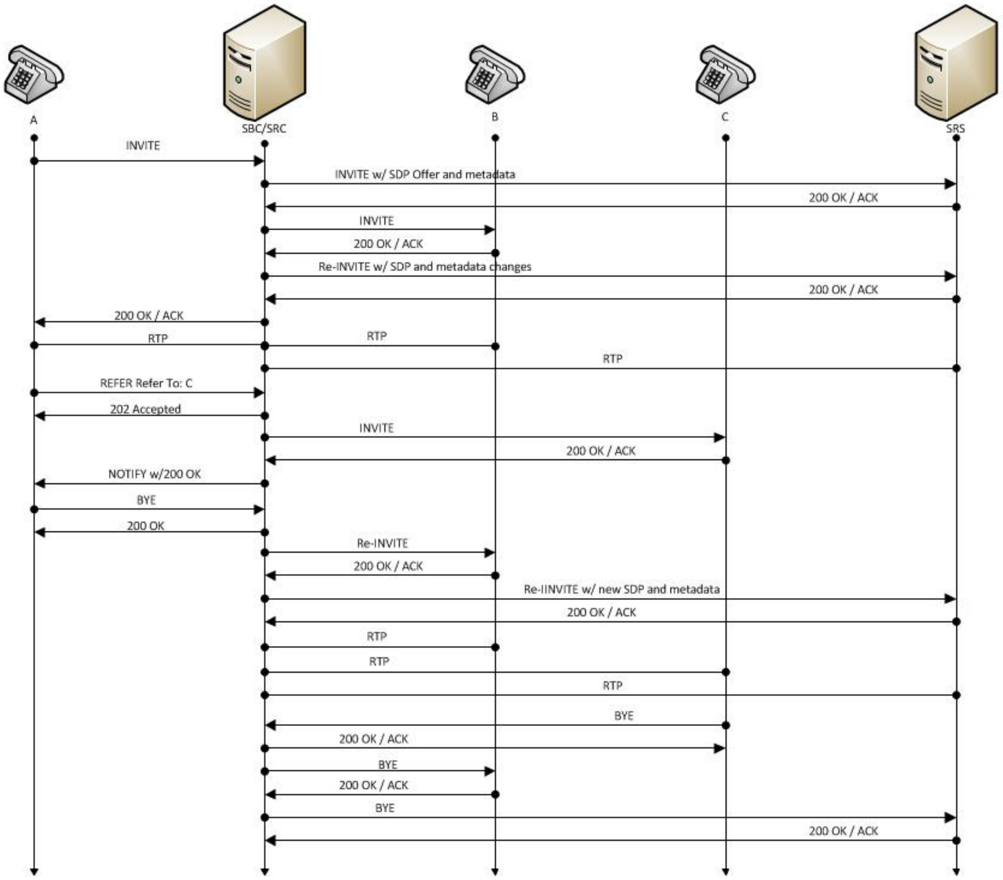
Call Flow Description	
1 - UA-A sends INVITE to Oracle Communications Enterprise Session Border Controller.	18 - UA-C responds with OK to Oracle Communications Enterprise Session Border Controller.

Call Flow Description	
2 - Oracle Communications Enterprise Session Border Controller forwards INVITE with SDP Offer and metadata to SRS.	19 - Oracle Communications Enterprise Session Border Controller forwards OK response to UA-B.
3 - SRS responds with OK to Oracle Communications Enterprise Session Border Controller.	20 - Oracle Communications Enterprise Session Border Controller sends NOTIFY with OK reponse to UA-A.
4 - Oracle Communications Enterprise Session Border Controller sends INVITE to UA-B.	21 - Oracle Communications Enterprise Session Border Controller sends re-INVITE to SRS with new SDP and metadata, adds participant C, stops participant A .
5 - UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.	22 - SRS responds with OK to Oracle Communications Enterprise Session Border Controller.
6 - Oracle Communications Enterprise Session Border Controller sends re-INVITE with SDP and metadata changes to SRS.	23 - UA-A sends BYE to Oracle Communications Enterprise Session Border Controller.
7 - SRS responds with OK to Oracle Communications Enterprise Session Border Controller.	24 - Oracle Communications Enterprise Session Border Controller responds with OK to UA-A.
8 - Oracle Communications Enterprise Session Border Controller forwards OK response to UA-A.	25 - Oracle Communications Enterprise Session Border Controller responds with OK to UA-A.
9 - RTP stream initiated between UA-A and Oracle Communications Enterprise Session Border Controller.	26 - RTP stream initiated between Oracle Communications Enterprise Session Border Controller and UA-B.
10 - RTP stream initiated between Oracle Communications Enterprise Session Border Controller and UA-B.	27 - RTP stream initiated between Oracle Communications Enterprise Session Border Controller and UA-C.
11 - RTP stream initiated between Oracle Communications Enterprise Session Border Controller and SRS.	28 - RTP stream initiated between Oracle Communications Enterprise Session Border Controller and SRS.
12 - UA-A sends REFER-TO: C to Oracle Communications Enterprise Session Border Controller.	29 - UA-C sends BYE to Oracle Communications Enterprise Session Border Controller.
13 - Oracle Communications Enterprise Session Border Controller forwards REFER-TO: C to UA-B.	30 - Oracle Communications Enterprise Session Border Controller responds with OK to UA-C.
14 - UA-B responds with 202 ACCEPTED to Oracle Communications Enterprise Session Border Controller.	31 - Oracle Communications Enterprise Session Border Controller sends BYE to UA-B.
15 - Oracle Communications Enterprise Session Border Controller forwards 202 ACCEPTED to UA-A.	32 - UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.
16 - UA-B sends INVITE TO: C to Oracle Communications Enterprise Session Border Controller.	33 - Oracle Communications Enterprise Session Border Controller sends BYE to SRS

Call Flow Description	
17 - Oracle Communications Enterprise Session Border Controller sends INVITE to UA-C.	34 - SRS responds with OK to Oracle Communications Enterprise Session Border Controller.

REFER Call (REFER handled by Oracle Communications Enterprise Session Border Controller)

The following illustration shows a call using selective recording and the Session Border Controller (Oracle Communications Enterprise Session Border Controller) handling the REFER on the call. Recording is required in this call flow.



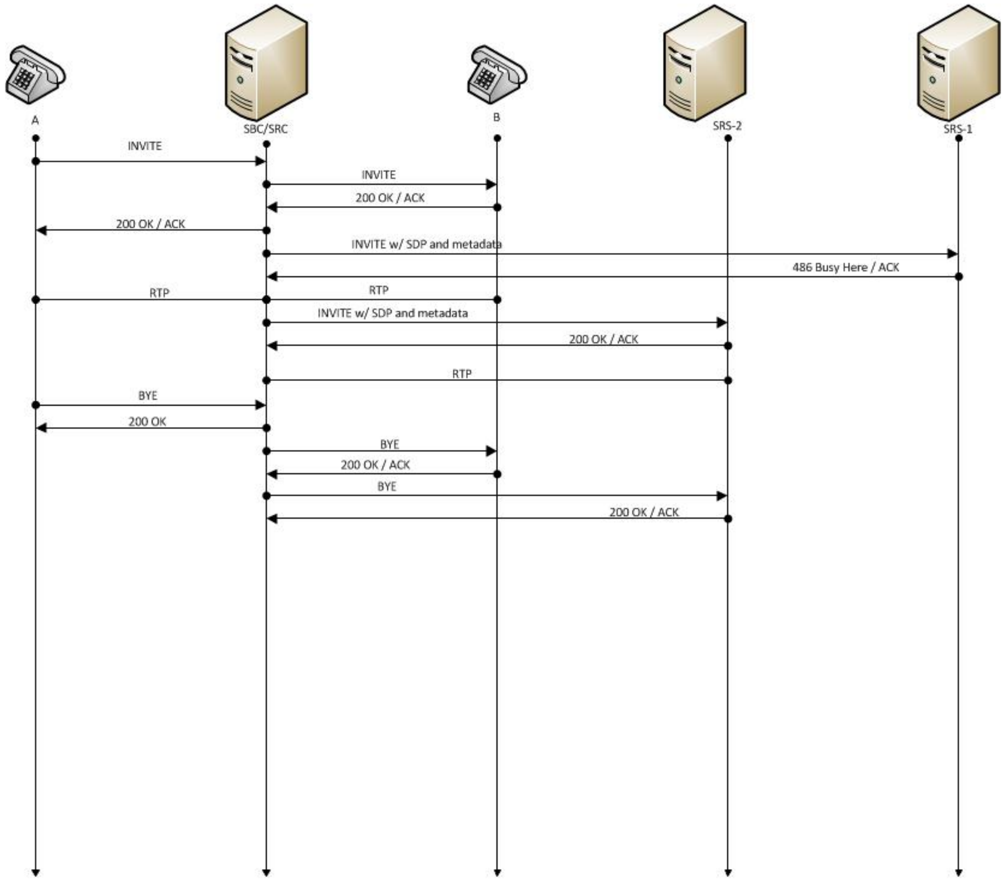
SIP REFER, SBC absorbs REFER, “recording required”, selective recording

Call Flow Description	
1 - UA-A sends INVITE to Oracle Communications Enterprise Session Border Controller.	16 - Oracle Communications Enterprise Session Border Controller sends NOTIFY with OK response to UA-A.
2 - Oracle Communications Enterprise Session Border Controller forwards INVITE with SDP Offer and metadata to SRS.	17 - UA-A sends BYE to Oracle Communications Enterprise Session Border Controller.
3 - SRS responds with OK to Oracle Communications Enterprise Session Border Controller.	18 - Oracle Communications Enterprise Session Border Controller responds with OK to UA-A.
4 - Oracle Communications Enterprise Session Border Controller sends INVITE to UA-B.	19 - Oracle Communications Enterprise Session Border Controller sends re-INVITE to UA-B.

Call Flow Description	
5 - UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.	20 - UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.
6 - Oracle Communications Enterprise Session Border Controller sends re-INVITE with SDP and metadata changes to SRS.	21 - Oracle Communications Enterprise Session Border Controller sends re-INVITE to SRS with new SDP and metadata.
7 - SRS responds with OK to Oracle Communications Enterprise Session Border Controller.	22 - SRS responds with OK to Oracle Communications Enterprise Session Border Controller.
8 - Oracle Communications Enterprise Session Border Controller forwards OK response to UA-A.	23 - RTP stream initiated between Oracle Communications Enterprise Session Border Controller and UA-B.
9 - RTP stream initiated between UA-A and Oracle Communications Enterprise Session Border Controller.	24 - RTP stream initiated between Oracle Communications Enterprise Session Border Controller and UA-C.
10 - RTP stream initiated between Oracle Communications Enterprise Session Border Controller and UA-B.	25 - RTP stream initiated between Oracle Communications Enterprise Session Border Controller and SRS.
11 - RTP stream initiated between Oracle Communications Enterprise Session Border Controller and SRS.	26 - UA-C sends BYE to Oracle Communications Enterprise Session Border Controller.
12 - UA-A sends REFER-TO: C to Oracle Communications Enterprise Session Border Controller.	27 - Oracle Communications Enterprise Session Border Controller responds with OK to UA-C.
13 - Oracle Communications Enterprise Session Border ControllerOracle Communications Enterprise Session Border Controller responds with 202 ACCEPTED to UA-A.	28 - Oracle Communications Enterprise Session Border Controller sends BYE to UA-B.
14 - Oracle Communications Enterprise Session Border Controller sends INVITE to UA-C.	29 - UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.
15 - UA-C responds with OK to Oracle Communications Enterprise Session Border Controller.	30 - Oracle Communications Enterprise Session Border Controller sends BYE to SRS.
	31 - SRS responds with OK to Oracle Communications Enterprise Session Border Controller.

SRS Indicates Busy in Call (recording not required)

The following illustration shows the Session Recording Server (SRS) is BUSY for a call session. Recording is not required in this call flow.



SIP INVITE, "recording not required"

Call Flow Description	
① UA-A sends INVITE to Oracle Communications Enterprise Session Border Controller.	⑨ Oracle Communications Enterprise Session Border Controller sends INVITE to SRS2 with SDP and metadata.
② Oracle Communications Enterprise Session Border Controller forwards INVITE to UA-B.	⑩ SRS2 responds with OK to Oracle Communications Enterprise Session Border Controller.
③ UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.	⑪ RTP stream initiated between Oracle Communications Enterprise Session Border Controller and SRS2.
④ Oracle Communications Enterprise Session Border Controller forwards OK response to UA-A.	⑫ UA-A sends BYE to Oracle Communications Enterprise Session Border Controller.
⑤ Oracle Communications Enterprise Session Border Controller sends INVITE to SRS1 with SDP and metadata.	⑬ Oracle Communications Enterprise Session Border Controller responds with OK to UA-A.
⑥ SRS1 responds to Oracle Communications Enterprise Session Border Controller with 436 BUSY HERE.	⑭ Oracle Communications Enterprise Session Border Controller sends BYE to UA-B.
⑦ RTP stream initiated between UA-A andOracle Communications Enterprise Session Border Controller.	⑮ UA-B responds with OK to Oracle Communications Enterprise Session Border Controller.

Call Flow Description	
⑧ RTP stream initiated between Oracle Communications Enterprise Session Border Controller and UA-B.	⑯ Oracle Communications Enterprise Session Border Controller sends BYE to SRS2.
	⑰ SRS2 responds with OK to Oracle Communications Enterprise Session Border Controller.

Local Media Playback

Commonly, ringback is the media playback of a certain tone informing callers their calls are in progress. In typical deployments, remote endpoints or media servers handle ringback generation, leaving the Oracle Communications Enterprise Session Border Controller to proxy RTP. When endpoints or media servers do not support ringback generation, the Oracle Communications Enterprise Session Border Controller becomes responsible for producing it.

Note: The Oracle Communications Enterprise Session Border Controller supports a maximum of 100 simultaneous playbacks.

You can configure the Oracle Communications Enterprise Session Border Controller to generate ringback locally, meaning it can produce RTP media on a media flow. The most common use for enabling the system to produce RTP on a media flow is to support locally generated ringback. Since you can also use this capability for music-on-hold, announcements, and interrupting media for notifications, this Oracle Communications Enterprise Session Border Controller capability is referred to as local playback.

Local playback is controlled through the ACLI using the Local Media Playback SPL configuration. For more information about SPLs, and configuring the Local Media Playback SPL Plug-in, see Chapter 23, Session Plug-in Language (SPL).

Supported Capabilities and Caveats

The Oracle Communications Enterprise Session Border Controller supports the following playback scenarios:

1. Playback on 183 Session Progress
2. Playback on REFER
3. Playback on header, where the header is P-Acme-Playback

Local media playback is not supported for these Oracle Communications Enterprise Session Border Controller capabilities:

- SRTP
- Call recording
- SIPREC

Local playback does not work in call flows for which media is released. Concurrent playbacks are limited to 100.

Media Setup & Playback

For each session requiring media playback, the Oracle Communications Enterprise Session Border Controller sets up media that supports standard RTP parameters. From the original media (if present), the Oracle Communications Enterprise Session Border Controller preserves the synchronization source (SSRC), timestamp, and sequence number.

For all playback options besides playback-on-header, the playback duration is continuous, meaning that the media file loops if it fails to cover the entire playback period.

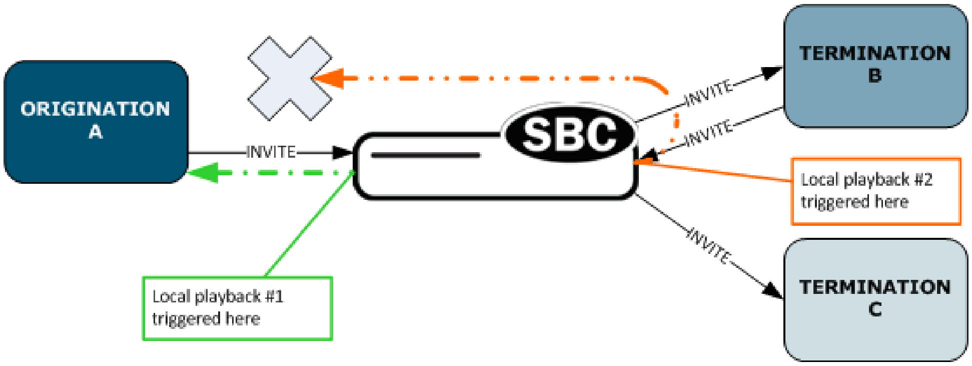
The playback duration for the playback on header scenario changes according to the settings in the P-Acme-Playback header. This header contains a duration=<ms-value | once | continuous> value, so you can customize the playback duration.

- Continuous—Media playback continues until the point at which the playback scenario defines its stop; media file loops if it fails to cover the entire playback duration.
- Once—The file plays until either the playback scenario defines its stop or until the media file runs out.
- Ms-value—Playback continues for a specific duration (in milliseconds) and will loop if necessary.

Once playback is in progress, the Oracle Communications Enterprise Session Border Controller mutes the session in the playback direction so that only the playback media can be heard.

Media Spirals

Certain call flows cause media to traverse the Oracle Communications Enterprise Session Border Controller multiple times, resulting in media spirals. For local playback, this means that multiple playback files can be triggered to play. In situations like this, the Oracle Communications Enterprise Session Border Controller uses the playback closest to the endpoint receiving the media playback. Origination A in the diagram below is played Local playback #1, even though the scenario also triggers Local playback #2.



Supported Playback Scenarios

This section discusses playback scenarios the Oracle Communications Enterprise Session Border Controller supports and identifies triggers for playback. When more than one trigger appears, the Oracle Communications Enterprise Session Border Controller acts on the one closest to the playback endpoint.

These scenarios are defined by the SPL options parameters you configure for realms, session agents, and SIP interfaces. For more information about configuring these options, see Chapter 23, Session Plug-in Language (SPL).

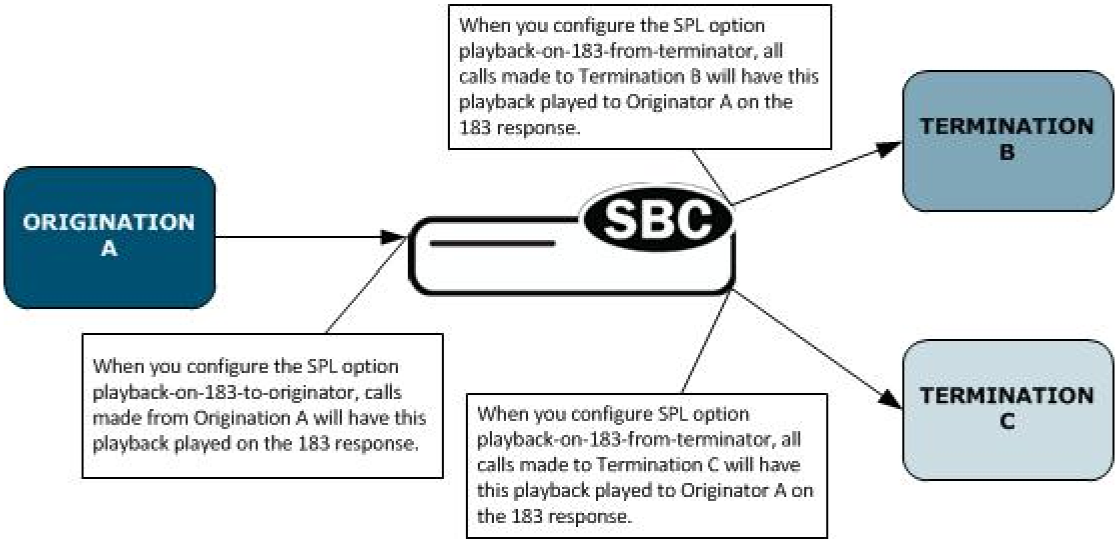
- playback-on-183-to-originator—Playback enabled upon the receipt of a 183 Session Progress destined for the originator and stops when a either a (200-299 or 400-699) final response is sent.
- playback-on-183-from-terminator—Playback enabled upon the receipt of a 183 Session Progress response is received from the terminator and stops when a (200-299 or 400-699) final response is received.
- playback-on-refer—Playback enabled for the caller being transferred when the Oracle Communications Enterprise Session Border Controller receives a REFER message that is locally terminated (i.e., processed on the Oracle Communications Enterprise Session Border Controller on REFER completion).
- playback-on-header—Starts or stops playback based on the presence of the P-Acme-Playback header and its definitions.

Note: The Oracle Communications Enterprise Session Border Controller supports a maximum of 100 simultaneous playbacks.

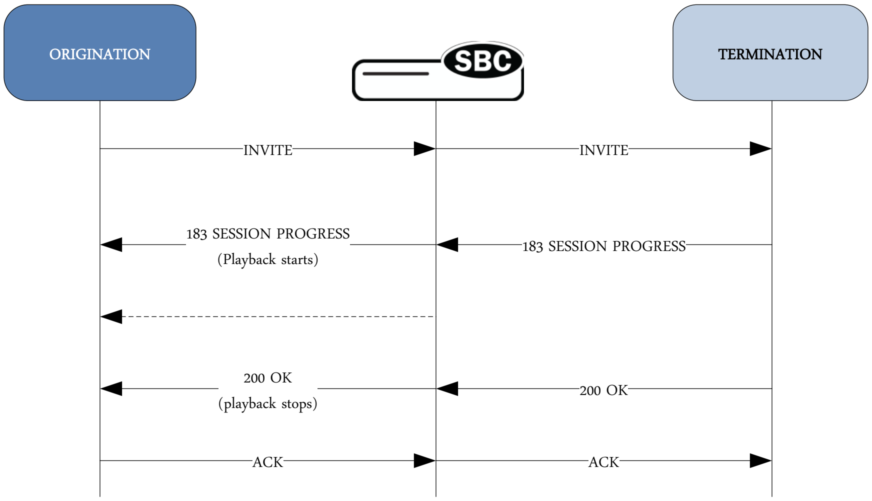
Playback on 183 Session Progress

This scenario is triggered by setting the SPL options parameter to either playback-on-183-to-originator or playback-on-183-from-terminator in realms, session agents, or SIP interfaces. When both options trigger, playback-on-183-to-originator takes precedence. This scenario triggers only for 183 Session Progress responses to initial INVITEs, not for re-INVITEs.

- playback-on-183-to-originator—Starts playback upon the receipt of a 183 Session Progress destined for the originator and stops when a either a (200-299 or 400-699) final response is sent. When you configure this option, every call sent from the originator triggers this playback.
- playback-on-183-from-terminator—Starts playback upon the receipt of a 183 Session Progress response is received from the terminator and stops when a (200-299 or 400-699) final response is received. When you configure this option, every call sent to the terminator triggers this playback.


When you configure the SPL option playback-on-183-from-terminator, all calls made to Termination B will have this playback played to Originator A on the 183 response.
TERMINATION B
ORIGINATION A
SBC
When you configure the SPL option playback-on-183-to-originator, calls made from Origination A will have this playback played on the 183 response.
When you configure SPL option playback-on-183-from-terminator, all calls made to Termination C will have this playback played to Originator A on the 183 response.
TERMINATION C

A call flow for the playback-on-183-from-terminator scenario looks like this:

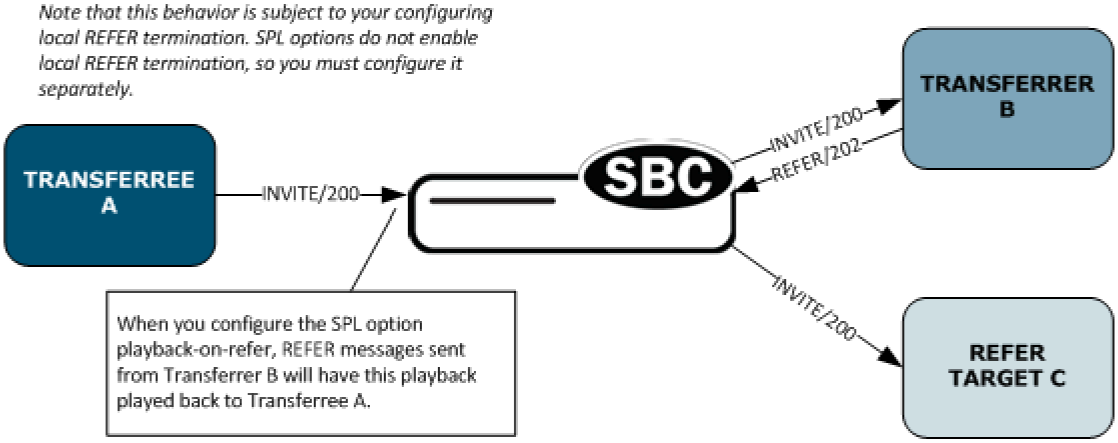


REFER

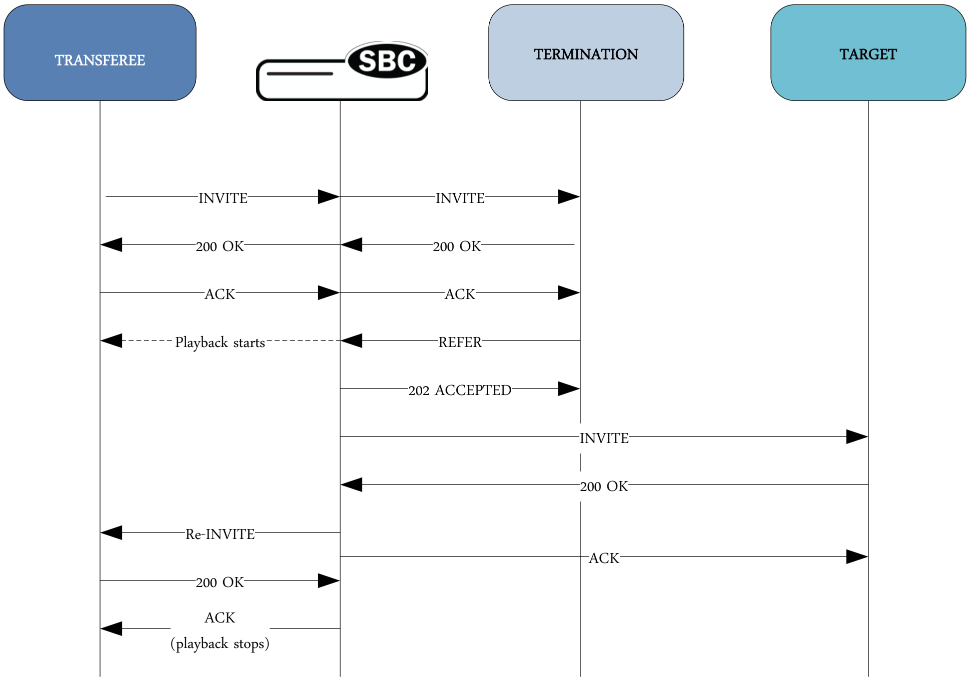
Setting the SPL options parameter to playback-on-refer enables a REFER message to trigger playback. You configure this option for the realm, session agent, or SIP interface for the transferrer, not for the transferee or the REFER target.

The REFER scenario requires that the Oracle Communications Enterprise Session Border Controller performs local REFER termination, i.e., that it terminates the REFER locally. The SPL options you configure do not implement this behavior: You must configure local REFER termination separately. Proxying a REFER message is not a trigger.

Playback begins when the Oracle Communications Enterprise Session Border Controller receives the REFER message, and stops when the REFER operation is deemed complete with a final response (200-299 or 400-699).


Note that this behavior is subject to your configuring local REFER termination. SPL options do not enable local REFER termination, so you must configure it separately.
TRANSFERREE A
INVITE/200
SBC
INVITE/200
REFER/202
TRANSFERRER B
INVITE/200
REFER TARGET C
When you configure the SPL option playback-on-refer, REFER messages sent from Transferrer B will have this playback played back to Transferee A.

A call flow for the playback-on-refer scenario looks like this:

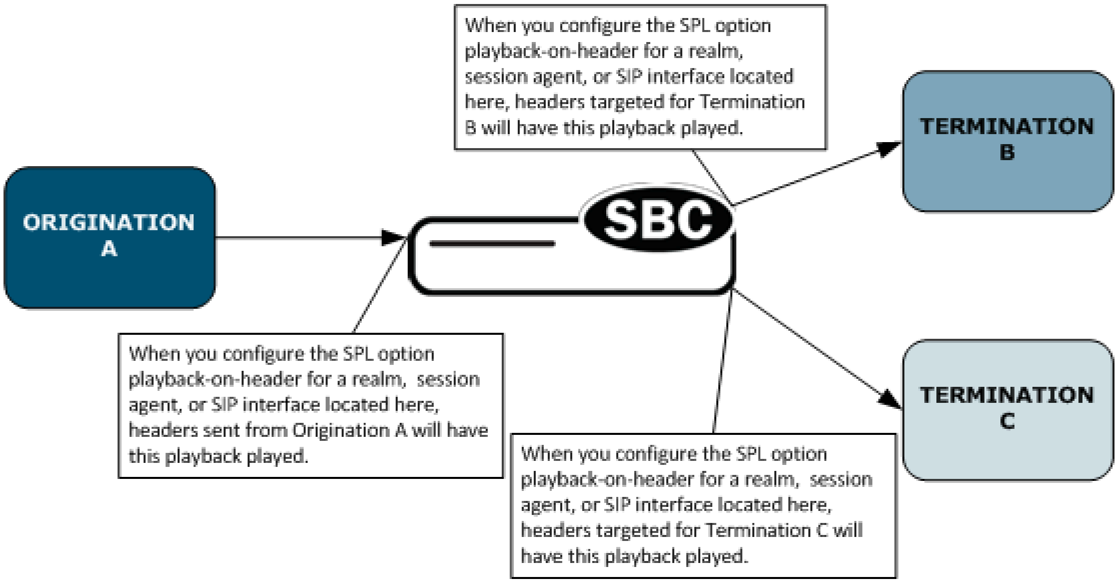


Playback Header

Setting the SPL options parameter to playback-on-header triggers in the presence of the P-Acme-Playback header. You can configure the option on either the side receiving the header message or the side from which the message will be sent. If both trigger, then the configuration closest to the playback direction takes precedence.

This header can be part of any request or response, but playback can only start once media has been established. Playback stops automatically with a final response (200-299 or 400-699), unless explicitly turned off or another playback header requesting it to stop is received.

The Oracle Communications Enterprise Session Border Controller deletes the P-Acme-Playback after processing if the SPL option is configured for the call (either incoming or outgoing).


When you configure the SPL option playback-on-header for a realm, session agent, or SIP interface located here, headers targeted for Termination B will have this playback played.
TERMINATION B
ORIGINATION A
SBC
When you configure the SPL option playback-on-header for a realm, session agent, or SIP interface located here, headers sent from Origination A will have this playback played.
TERMINATION C
When you configure the SPL option playback-on-header for a realm, session agent, or SIP interface located here, headers targeted for Termination C will have this playback played.

The header looks like this:

```
P-Acme-Playback: start
    ;media=media1
    ;duration=continuous
    ;direction=both
    ;stop-on-final-resp=true
```

Header Element	Description
<start|stop>	Required Defines starting and stopping playback. • start: starts playback • stop: stops playback
[;media=<media-name>]	Optional Defines the name of the playback configuration to play. If unspecified, the playback configuration that was triggered by the header will play.
[;duration=<ms-value|once|continuous>]	Optional Defines the duration of playback. If unspecified, the value will be taken from the playback-config that was triggered. • ms-value: time value in milliseconds • once: plays playback media one time • continuous: loops the playback media
[;direction=<originator|terminator|both>]	Optional Defines the direction from which to play media. If unspecified, playback will begin in the realm, session agent, or SIP interface from which the header was received. • originator: plays in the west flow (original caller) • terminator: plays in the east flow (original callee) • both: plays in both directions

Header Element	Description
[;stop-on-final-resp=<true | false>]	Optional Defines whether or not to stop playing media upon the final response. If unspecified, this parameter is true. • true: stops playback automatically on a final response • false: stop only after a stop header is received or media terminated

ACLI Configuration and Examples

For configuring the Local Media Playback SPL options on the Oracle Communications Enterprise Session Border Controller, see Chapter 23, Session Plug-in Language (SPL).

Considerations for HA Nodes

On switchover, media set-up for playback is preserved, which requires negotiated codec and ptime for playback be transferred to the stand-by system in an HA node. However, any playback in progress will not be continued on switchover.

While standard configuration replication handles transferring configuration information between the active and standby systems, media playback files (in /code/media) must be loaded onto the standby.

Alarms

These are the alarms for local playback. They are MAJOR in severity, and do not impact the system health score.

Alarm	Description
Playback media file not found or couldn’t be loaded	Raised when a configuration is activated if the system cannot find a media file referenced configuration or if the system is unable to load the media file. This alarm clears automatically when a file is correctly referenced or when it is loaded properly. You might encounter this alarm if you have established playback configuration, but have not loaded the appropriate playback files to /code/media.
Playback could not be started due to capacity limit	Raised at call time when system has reached its maximum number of playbacks (100). This alarm must be cleared manually.
Playback could not be started due to unsupported codec	Raised at call time when there is a mismatch of codecs between those in available files and one that must be played. This alarm must be cleared manually.

Monitoring

You can use the **show mbcd statistics** command to displays the number of media playbacks that are currently alive:

```
ORACLE# show mbcd statistics
MBCD Status                      -- Period -- -------- Lifetime --------
```

```
                      Active    High   Total      Total  PerMax    High
Media Playback             0       5       5          6       0       6
```

You can use the **show mbcd errors** command to track the number of playback failures:

```
ORACLE# show mbcd errors
MBC Errors/Events              ---- Lifetime ----
                        Recent      Total  PerMax
Media Playback Fails         0          0       0
Playback Exh Resources       0          0       0
Playback Flow Inactive       0          0       0
Playback Mismatch            0          0       0
```

2

Oracle Communications Operations Monitor Deployments

The Oracle® Communications Operations Monitor (OCOM) Mediation Engine is a platform that collects SIP, DIAMETER, DNS and ENUM protocol message traffic received from OCOM Probes. You can configure the Oracle Communications Enterprise Session Border Controller to run an onboard Probe. Probes can also run on COTS hardware collecting packets, for example, from span/monitor ports on Ethernet switches. A Probe takes the protocol packets, prepends a receive timestamp and other information, encapsulates the packets, and passes them to the OCOM mediation engine via a secure connection. After receiving protocol traffic from a Probe, mediation engine stores the traffic in an internal database, and analyzes aggregated data to provide comprehensive multi-level monitoring, troubleshooting, and interoperability information.

In contrast to the Packet-Trace feature, message logging is performed by software, which sends a copy of sent/received messages over UDP, or by saving such messages in a local file. The copy includes a timestamp, port/vlan information, and IP:port information, but all in ASCII format. Message Logging is performed after all decryption, meaning that SIP/TLS traffic cam be monitored. Because remote message logging sends the protocol messages over UDP, there is no guarantee or confirmation of delivery.

The Oracle Communications Enterprise Session Border Controller provides support for a user-configurable capability that enables the system to function as an OCOM Probe. Acting as a Probe, or as an exporter, the Oracle Communications Enterprise Session Border Controller can:

1. Establish an authenticated, persistent, reliable TCP connection between itself and one or more OCOM Mediation Engines.
2. Optionally ensure message privacy by encrypting the TCP connection using TLS.
3. Use the TCP connection to send a UTC-timestamped, unencrypted copy of a protocol message to the OCOM Mediation Engine(s).
4. Accompany the copied message with related data to include: the port/vlan on which the message was sent/received, local and remote IP:port information, and the transport layer protocol.

IPFIX

The Oracle Communications Enterprise Session Border Controller uses the IPFIX suite of standards to export protocol message traffic and related data to the Oracle Communications Operations Monitor (OCOM) Mediation Engine.

- RFC 5101, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information*
- RFC 5102, *Information Model for IP Flow Information Export*
- RFC 5470, *Architecture for IP Flow Information Export*
- RFC 5655, *Specification of the IP Flow Information Export (IPFIX) File Format*
- RFC 5815, *Definitions of Managed Objects for IP Flow Information Export*

The IPFIX standards describe the use of templates to format the export of specific types of protocol traffic. The Oracle Communications Enterprise Session Border Controller and the OCOM Mediation Engine share ten (10) pre-defined templates that facilitate protocol message exchange, and subsequent processing and analysis by the OCOM Engine.

The pre-defined templates are:

- incoming SIP/DNS over UDP
- incoming SIP over TCP
- incoming SIP over SCTP
- incoming DNS over UDP (entire IP and UDP header not included)
- outgoing SIP/DNS over UDP
- outgoing SIP over TCP
- outgoing SIP over SCTP
- outgoing DNS over UDP (entire IP and UDP header not included)
- media qos and flow record
- IPFIX handshake (used for connection establishment)

Incremental QoS Updates

The Interim Quality of Service (QoS) Update setting supported on the Acme Packet 4600 and the Acme Packet 6300 provides a more granular view of voice quality for troubleshooting by providing updates in 10 second increments. Without the Interim QoS Update setting selected, the Oracle Communications Enterprise Session Border Controller (ESBC) probe provides an average Mean Opinion Score (MOS) only at the end of the call. A troubleshooter cannot see what occurred in other parts of the call. For example, suppose your employee or agent complains of poor voice quality that occurred in the middle of the call, but the average MOS score at the end of the call is 4.40. The troubleshooter might determine that the quality is acceptable, without knowing that the score in the middle of the call is 2.50. The Interim QoS Update setting provides MOS scores every 10 seconds, and with more granular data to help troubleshooting efforts.

Standalone Oracle Communications Operations Monitor (OCOM) probes, such as those that run OCOM software on Linux COTS servers, provide MOS scores in 10 second time chunks. With the Interim QoS Update setting selected, the data presented in OCOM looks similar whether coming from an ESBC probe, OCOM probe, or both. To configure the devices to sample voice quality information in 10 second increments, select **Interim QoS update** in system-config.

The ESBC provides the following data, per ten second interval.

- start + end time of the stream
- IP 5-tuple information to correlate to SIP sessions
- correlation information if available
- SSRC of the RTP stream (to be checked)
- Codec type
- Codec change information (if codecs changed)

The ESBC provides the following data, per ten second chunk.

- jitter
- min/avg/max

- histogram (optional), e.g. # of packets with jitter <5ms, <10ms, <20ms, ... >100ms.
- packet loss
- # of packets received
- # of packets lost
- discarded packets (optional, received 50+ms too late)
- R-factor (optional)
- MOS value (optional)

The ESBC delivers voice quality details, as follows:

- Per RTP stream.
- In 10 second increments, where the increment starts on a full minute based on the NTP clock (not the start time of the stream).
- Intervals not covering the full 10 seconds do not return a MOS value.

Licensing

Native EOM probes include the Interim QoS Update function. Using the feature requires a Media Quality Extension (MQE) license. The base Enterprise license includes the function. Service Provider customers must purchase the MQE license in addition to the base Service Provider OCOM license.

Oracle Communications Operations Monitor (OCOM) Configuration

Oracle Communications Enterprise Session Border Controller configuration on the consists of the following steps.

1. Configuration of one or more Oracle Communications Enterprise Session Border Controller/OCOM exporter/collector pairs.
2. Optional assignment of a TLS profile to an exporter/collector pair.

Oracle Communications Operations Monitor (OCOM)

Use the following procedure to configure OCOM:

1. From superuser mode, use the following ACLI sequence to access comm-monitor configuration mode. From comm-monitor mode, you establish a connection between the Oracle Communications Enterprise Session Border Controller, acting as a exporter of protocol message traffic and related data, and an OCOM Mediation Engine, acting as an information collector.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# system
   ACMEPACKET(system)# system-config
   ACMEPACKET(system-config)# comm-monitor
   ACMEPACKET(comm-monitor)#
   ```

2. Use the **state** parameter to enable or disable communication monitoring.

   Communication monitoring is disabled by default.

   ```
   ACMEPACKET(comm-monitor)# state enabled
   ACMEPACKET(comm-monitor)#
   ```

3. Use the **sbc-group-id** parameter to assign an integer value to the Oracle Communications Enterprise Session Border Controller, in its role as an information exporter.

   Retain the default value (0) or assign another integer value.

   ```
   ACMEPACKET(comm-monitor)# sbc-group-id 5
   ACMEPACKET(comm-monitor)#
   ```

4. If the network interface specified in Step 8 is a media interface, you can optionally use TLS to encrypt the exporter/collector connection.

   To enable TLS encryption, use the **tls-profile** parameter to identify a TLS profile to be assigned to the network interface. The absence of an assigned TLS profile (the default state) results in unencrypted transmission.

   Refer to *TLS Profile Configuration* for configuration details.

   ```
   ACMEPACKET(comm-monitor)# tls-profile commMonitor
   ACMEPACKET(comm-monitor)#
   ```

5. Use the **qos-enable** parameter to enable or disable to export of RTP, SRTP, and QOS data flow information.

   ```
   ACMEPACKET(comm-monitor)# qos-enable enabled
   ACMEPACKET(comm-monitor)#
   ```

6. Use the **monitor-collector** parameter to move to monitor-collector configuration mode.

   While in this mode you identify an OCOM Mediation Engine collector by IP address and port number.

   ```
   ACMEPACKET(comm-monitor)# monitor-collector
   ACMEPACKET(monitor-collector)#
   ```

7. Use the **address** and **port** parameters to specify the IP address and port number monitored by an OCOM Mediation Engine for incoming IPFIX traffic.

   Enter an IPv4 address and a port number with values either 4739 (unsecured) or 4740 (secured). The default value for the port is 4739.

   ```
   ACMEPACKET(monitor-collector)# address 172.30.101.239
   ACMEPACKET(monitor-collector)# port 4739
   ACMEPACKET(monitor-collector)#
   ```

8. Use the **network-interface** parameter to specify the network interface that supports the TCP connection between the Oracle Communications Enterprise Session Border Controller to the OCOM Mediation Engine.

   To specify the wancom0 management interface:

   ```
   ACMEPACKET(comm-monitor)# network-interface wancom0:0
   ACMEPACKET(comm-monitor)#
   ```

   To specify a media interface:

   ```
   ACMEPACKET(comm-monitor)# network-interface m01
   ACMEPACKET(comm-monitor)#
   ```

   Note: If configuring with a media interface, that interface must belong to a configured realm.

9. Use **done** and **exit** to return to comm-monitor configuration mode.
10. Use **done**, **exit**, and **verify-config** to complete configuration.
11. Repeat Steps 1 through 10 to configure additional as required.

TSCF Rekey Profile Configuration

Rekeying is a cryptographic technique that enhances security by enforcing the negotiation of existing keys on an ongoing secure connection. Rekeying can be either time-based, in which case new keys are negotiated at the expiration of a timer, or traffic-based, in which case new keys are negotiated when a threshold byte count is exceeded.

Use the following procedure to configure an optional tscf-rekey-profile. Later, you will assign the profile to a specific TSCF interface. If you do not intend to enforce re-keying, this procedure can be safely ignored.

1. From superuser mode, use the following command sequence to access tscf-rekey-profile configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# tscf
ACMEPACKET(tscf)# tscf-rekey-profile
ACMEPACKET(tscf-rekey-profile)#
```

2. Use the **name** parameter to provide a unique identifier for this tscf-rekey-profile.

   ```
   ACMEPACKET(tscf-rekey-profile)# name tscfRekey01
   ACMEPACKET(tscf-rekey-profile)#
   ```

3. Use the **initiator** parameter to identify the rekey initiator.

   Supported values are **client** (default) | **server** (the Session Director)

   ```
   ACMEPACKET(tscf-rekey-profile)# initiator client
   ACMEPACKET(tscf-rekey-profile)#
   ```

4. Use the **max-rekey-time** parameter to specify the maximum interval (in minutes) between re-keying operations.

   Supported values are **0** (default) | **30** - **1440** (minutes)

   The default value, **0**, specifies that time-based rekeying is not enforced; other integer values specify that time-based re-keying must be initiated by the tunnel endpoint designated by the **initiator** parameter.

   ```
   ACMEPACKET(tscf-rekey-profile)# max-rekey-time 30
   ACMEPACKET(tscf-rekey-profile)#
   ```

5. Use the **max-rekey-data** parameter to specify the maximum traffic exchange (measured in Kb) between rekeying operations.

   The default value, **0**, specifies that traffic-based rekeying is not enforced; other integer values specify that traffic-based re-keying must be initiated by the tunnel endpoint designated by the **initiator** parameter.

   ```
   ACMEPACKET(tscf-rekey-profile)# max-rekey-data 0
   ACMEPACKET(tscf-rekey-profile)#
   ```

6. Use **done**, **exit**, and **verify-config** to complete tscf-rekey-profile configuration.
7. Repeat Steps 1 through 6 to configure additional tscf-rekey-profiles as required.

TLS Profile Configuration

Use the following procedure to configure a tls-profile that identifies the cryptographic resources, specifically certificates and protocols, required for the establishment of a secure/encrypted connection between the Oracle Communications Enterprise Session Border Controller and the Oracle Communications Operations Monitor (OCOM) Mediation Engine.

1. From superuser mode, use the following command sequence to access tls-profile configuration mode.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# security
   ACMEPACKET(security)# tls-profile
   ACMEPACKET(tls-profile)#
   ```

2. Use the **name** parameter to provide a unique identifier for this tls-profile.

   ```
   ACMEPACKET(tls-profile)# name commMonitor
   ACMEPACKET(tls-profile)#
   ```

3. Use the required **end-entity-certificate** parameter to specify the name of the certificate-record configuration that identifies the credential (specifically, an X509.v3 certificate) offered by the Oracle Communications Enterprise Session Border Controller in support of its asserted identity.

   ```
   ACMEPACKET(tls-profile)# end-entity-certificate commMonitor509
   ACMEPACKET(tls-profile)#
   ```

4. Use the required **`trusted-ca-certificates`** parameter to compile a list or one or more certificate-record configuration elements referencing trusted Certification Authority (CA) certificates used to authenticate the offered certificate. These referenced certificates are conveyed to the OCOM Monitor Mediation Engine as part of the TLS exchange.

   Provide a comma separated list of existing CA **`certificate-record`** configuration elements.

   ```
   ACMEPACKET(tls-profile)# trusted-ca-certificates verisignClass3-
   a,verisignClass3-b,baltimore,thawtePremium,acme-CA
   ACMEPACKET(tls-profile)#
   ```

5. Retain the default value, **`all`**, for the **`cipher-list`** parameter.

6. Use the **`verify-depth`** parameter to specify the maximum number of chained certificates that will be processed while authenticating end-entity certificate received from the OCOM Mediation Engine.

   Provide an integer within the range 1 through 10 (the default).

   The Oracle Communications Enterprise Session Border Controller supports the processing of certificate chains (consisting of an end-entity certificate and some number of CA certificates) when X.509v3 certificate-based authentication is used. The following process validates a received TLS certificate chain.

   - Check the validity dates (Not Before and Not After fields) of the end certificate. If either date is invalid, authentication fails; otherwise, continue chain validation
   - Check the maximum length of the certificate chain (specified by verify-depth). If the current chain exceeds this value, authentication fails; otherwise, continue chain validation.
   - Verify that the Issuer field of the current certificate is identical to the Subject field of the next certificate in the chain. If values are not identical, authentication fails; otherwise, continue chain validation.
   - Check the validity dates (Not Before and Not After fields) of the next certificate. If either date is invalid, authentication fails; otherwise, continue chain validation.
   - Check the X509v3 Extensions field to verify that the current certificate identifies a CA. If not so, authentication fails; otherwise, continue chain validation.
   - Extract the Public Key from the current CA certificate. Use it to decode the Signature field of the prior certificate in the chain. The decoded Signature field yields an MD5 hash value for the contents of that certificate (minus the Signature field).
   - Compute the same MD5 hash. If the results are not identical, authentication fails; otherwise, continue chain validation.
   - If the hashes are identical, determine if the CA identified by the current certificate is a trust anchor by referring to the trusted-ca-certificates attribute of the associated TLS-profile configuration object. If the CA is trusted, authentication succeeds. If not, return to Step 2.

   ```
   ACMEPACKET(tls-profile)# verify-depth 8
   ACMEPACKET(tls-profile)#
   ```

7. Use the **`mutual-authenticate`** parameter to **`enable`** or **`disable`** (the default) mutual authentication.

   Protocol requirements mandate that the server present its certificate to the client application. Optionally, the server can implement mutual authentication by requesting a certificate from the client application, and authenticating the certificate offered by the client.

   Upon receiving a server certificate request, the client application must respond with a certificate; failure to do so results in authentication failure.

   ```
   ACMEPACKET(tls-profile)# mutual-authenticate disabled
   ACMEPACKET(tls-profile)#
   ```

8. Retain the default value, **`compatibility`**, for the **`tls-version`** parameter.

9. Retain default values for all other parameters.

10. Use **`done`**, **`exit`**, and **`verify-config`** to complete tls-profile configuration.

11. Repeat Steps 1 through 10 to configure additional tls-profiles as required.

Configure Interim QoS Update - ACLI

You can configure the Acme Packet 3900, the Acme Packet 4600, and the Acme Packet 6300 to sample voice quality information in 10 second increments by enabling Interim QoS Update.

- Confirm that Quality of Service (QoS) is enabled.
- Confirm that an Operations Monitor is configured.

Use the **system-config** object to enable Interim QoS update.

1. Access the **system-config** configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# system
   ORACLE(system)# system-config
   ORACLE(system-config)#
   ```
2. Type select, and press ENTER.
3. Type comm-monitor, and press ENTER.
4. Type select, and press ENTER.
5. Type interim-qos-update enable, and press ENTER.
6. Save and activate the configuration.

3

Packet Trace

The Oracle Communications Enterprise Session Border Controller's packet trace tool provides the user with the ability to capture traffic from the Oracle Communications Enterprise Session Border Controller itself.

The user invokes the tool from the ACLI, manually specifying:

- How to capture (local vs remote)
- What to capture
- Capture start and stop

There are two capture modes, one that saves traffic locally and one that mirrors traffic to a user-specified target. Software only deployments support local capture only. Proprietary Acme hardware deployments support both local and remote capture.

Local capture supports PCAP filters to specify the type of traffic to capture. Remote capture supports its own syntax to identify the traffic to mirror.

Local packet capture is dependent on access control configuration, not capturing any denied traffic. Remote capture mirrors traffic regardless of access control configuration.

Installed NIUs impact remote packet capture. Fragmented packets that ingress HIFNs or Cavium NIUs include only the outer header within the fragments. As a result, these traces do not appear to be using IPIP encapsulation. This differs from fragmented packets that ingress the Quad port GiGE and Copper PHY NIUs. These traces include inner and outer headers within the fragments.

Do not run packet-trace simultaneously with other Oracle Communications Enterprise Session Border Controller replication features, such as LI, SRS, SIP Monitoring and Trace, and Call Recording. These features may interfere with each other, corrupting each's results.

Packet Trace Remote

Packet trace remote enables the Oracle Communications Enterprise Session Border Controller to mirror traffic between two endpoints, or between itself and a specific endpoint to a user-specified target. To accomplish this, the Oracle Communications Enterprise Session Border Controller replicates the packets sent and received, encapsulates them according to RFC 2003, and sends them to a user-configured target. At the target, the user would capture and analyze the packets.

Currently, the Oracle Communications Enterprise Session Border Controller supports:

- One configurable trace server (on which you capture/analyze the traffic)

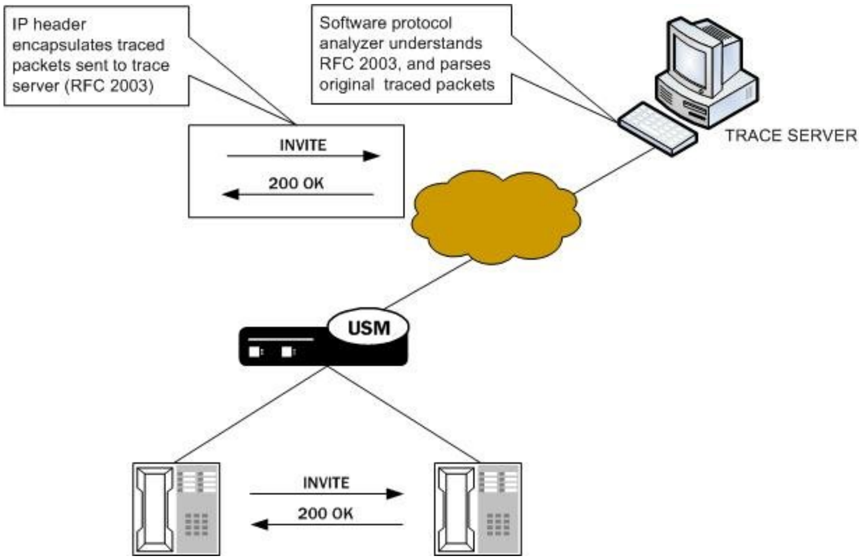
- Sixteen concurrent endpoint traces

To use this feature, the user configures a **`capture-receiver`** on the Oracle Communications Enterprise Session Border Controller so that it knows where to send the mirrored packets. Once the **`capture-receiver`** is configured, the user issues the **`packet-trace`** command to start, stop and specify filters for traces.

You establish a packet trace filter with the following information:

- Network interface—The name of the network interface on the Oracle Communications Enterprise Session Border Controller from which you want to trace packets. The user can enter this value as a name or as a name and subport identifier value (name:subportid)
- IP address—IP address of the endpoint to or from which the target traffic goes.
- Local port number—Optional parameter; Layer 4 port number on which the Oracle Communications Enterprise Session Border Controller receives and from which it sends; if no port is specified or if it is set to 0, then all ports will be traced
- Remote port number—Optional parameter; Layer 4 port number to which the Oracle Communications Enterprise Session Border Controller sends and from which it receives; if no port is specified or if it is set to 0, then all ports will be traced.

The Oracle Communications Enterprise Session Border Controller then encapsulates the original packets in accordance with RFC 2003 (*IP Encapsulation within IP*); it adds the requisite headers, and the payload contains the original packet trace with the Layer 2 header removed. Since software protocol analyzers understand RFC 2003, they can easily parse the original traced packets.


IP header encapsulates traced packets sent to trace server (RFC 2003)
Software protocol analyzer understands RFC 2003, and parses original traced packets
TRACE SERVER
INVITE
200 OK
USM
INVITE
200 OK

It is possible that—for large frames—when the Oracle Communications Enterprise Session Border Controller performs the steps to comply with RFC 2003 by adding the requisite header, the resulting packet might exceed Ethernet maximum transmission unit (MTU). This could result in packets being dropped by external network devices, but widespread support for jumbo frames should mitigate this possibility.

If the Oracle Communications Enterprise Session Border Controller either receives or transmits IP fragments during a packet trace, it only traces the first fragment. The first fragment is likely to be a maximum-sized Ethernet frame.

The Oracle Communications Enterprise Session Border Controller continues to conduct the packet trace and send the replicated information to the trace server until you instruct it to stop. You stop a packet trace with the ACLI **`packet-trace remote stop`** command. With this command, you can stop either an individual packet trace or all packet traces that the Oracle Communications Enterprise Session Border Controller is currently conducting.

Packet Trace Local

Packet Trace Local enables the Oracle Communications Enterprise Session Border Controller to capture traffic between two endpoints, or between itself and a specific endpoint. To accomplish this, the Oracle Communications Enterprise Session Border Controller replicates the packets sent and received and saves them to disk in .PCAP format.

The default packet trace filter uses the specified interface to capture both ingress and egress traffic. To specify captured traffic, the user can append the command with a PCAP filter enclosed in quotes. PCAP filter syntax is widely published.

While capturing, the system displays the number of packets it has captured and prevents the user from entering any other ACLI commands from that session. The user terminates the capture by pressing Ctrl+C.

By default, the system saves the .PCAP file in /opt/traces, naming it with the applicable interface name as well as the date and time of the capture. Alternatively, the user can specify file name using the system supports the PCAP filter flags -w.

The system rotates the PCAPs created in this directory by size. The last 25 files are kept and are rotated when they reach 100 MB. If there are capture files in the /opt/traces directory when this command is run, the system prompts the user to remove them before running new captures. If preferred, the user can decline this file deletion.

Packet Trace Scenarios

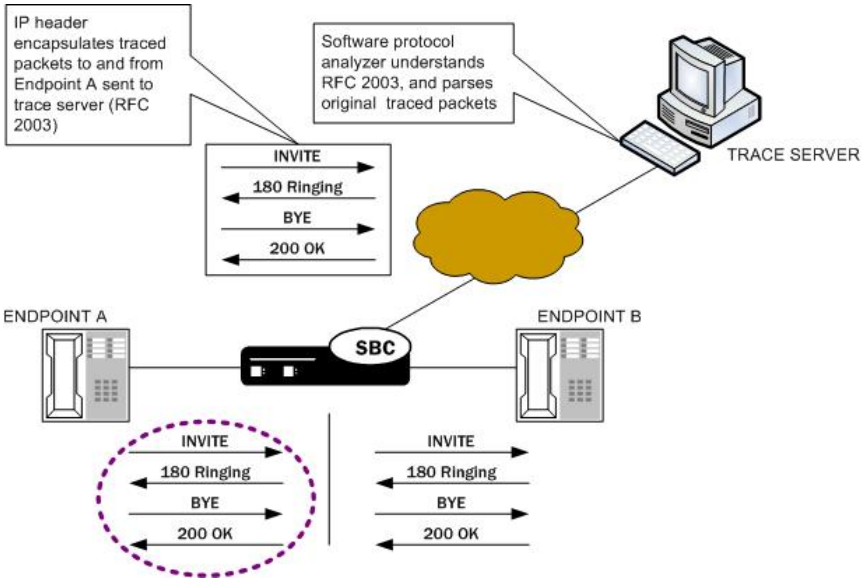
This section describes three possible ways that you might use the packet trace feature. You can examine communications sent to and from one endpoint, sent between two endpoints, or sent between ingress and/or egress Oracle Communications Enterprise Session Border Controller interfaces to endpoints.

Packet Trace for One Endpoint

When you use the packet-trace remote <state> command, the Oracle Communications Enterprise Session Border Controller sets up packet tracing for one endpoint. The Oracle Communications Enterprise Session Border Controller collects and replicates the packets to and from one endpoint. To enable this kind of trace, you set up one packet trace using the **packet-trace** command.

The commands you carry out for **packet-trace remote** would take the following form:

```
ORACLE# packet-trace remote start F01:0 <IP address of Endpoint A>
```


IP header encapsulates traced packets to and from Endpoint A sent to trace server (RFC 2003)
Software protocol analyzer understands RFC 2003, and parses original traced packets
TRACE SERVER
INVITE
180 Ringing
BYE
200 OK
ENDPOINT A
SBC
ENDPOINT B
INVITE
180 Ringing
BYE
200 OK
INVITE
180 Ringing
BYE
200 OK

The commands you carry out for **packet-trace local** would take the following form:

```
ORACLE# packet-trace local F01:0 <"host IP address of Endpoint A">
```

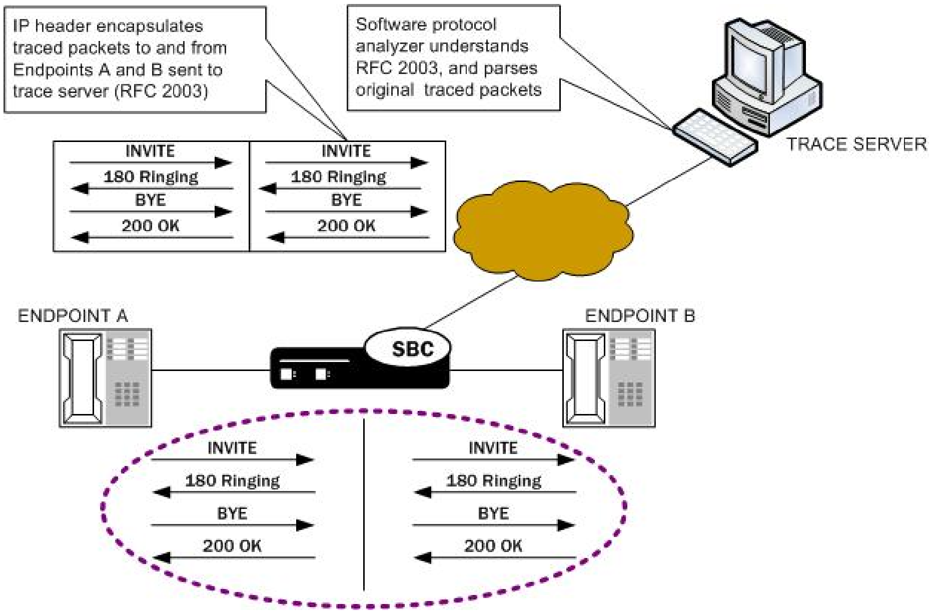
Packet Trace for Both Call Legs

If you want to trace both sides (both call legs), then you must set up individual traces for each endpoint—meaning that you would initiate two packet traces. The results of the trace will give you the communications both call legs for the communication exchanged between the endpoints you specify.

If you initiate a packet trace for both endpoints that captures both signaling and media, the signaling will be captured as usual. However, RTP will only be traced for the ingress call leg. This is because the Oracle Communications Enterprise Session Border Controller performs NAT on the RTP, which means it cannot be captured on the egress call leg.

The commands you carry out for **packet-trace remote** would take the following form:

```
ORACLE# packet-trace remote start F01:0 <IP address of Endpoint A>
ORACLE# packet-trace remote start F02:0 <IP address of Endpoint B>
```


IP header encapsulates traced packets to and from Endpoints A and B sent to trace server (RFC 2003)
Software protocol analyzer understands RFC 2003, and parses original traced packets
TRACE SERVER
INVITE
180 Ringing
BYE
200 OK
INVITE
180 Ringing
BYE
200 OK
ENDPOINT A
ENDPOINT B
SBC
INVITE
180 Ringing
BYE
200 OK
INVITE
180 Ringing
BYE
200 OK

The commands you carry out for **packet-trace local** would take the following form:

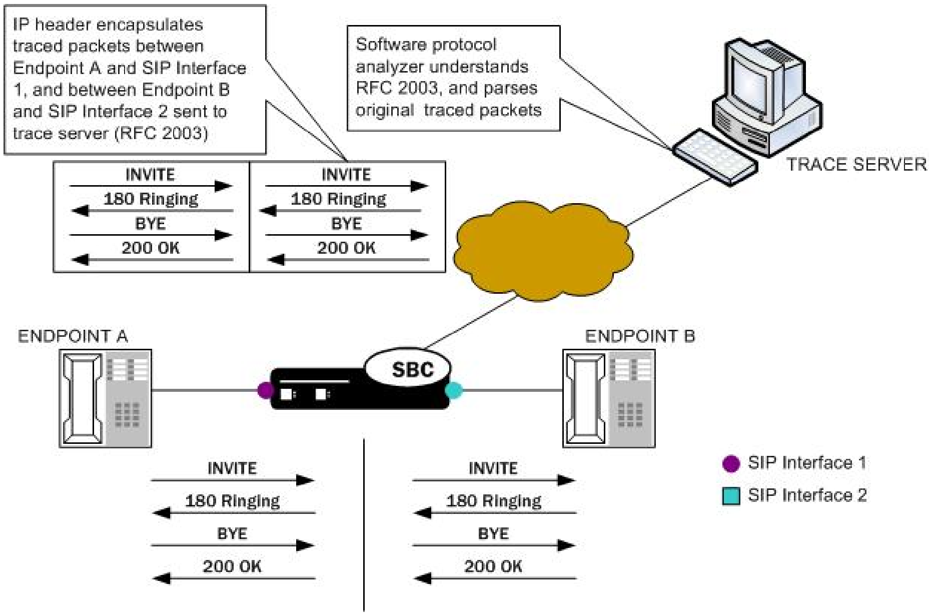
```
ORACLE# packet-trace local F01:0 <"host IP address of Endpoint A">
ORACLE# packet-trace local F02:0 <"host IP address of Endpoint B">
```

Packet Trace for a Signaling Address

You can perform a packet trace for addresses internal to the Oracle Communications Enterprise Session Border Controller; this can be the address, for example, of a SIP interface. Using signaling interface addresses puts the emphasis on the Oracle Communications Enterprise Session Border Controller rather than on the endpoints by allowing you to view traffic from specified interfaces.

The commands you carry out for **packet-trace remote** would take the following form:

```
ORACLE# packet-trace remote start F01:0 <IP address of Oracle Communications
Enterprise Session Border Controller interface1>
ORACLE# packet-trace remote start F02:0 <IP address of Oracle Communications
Enterprise Session Border Controller interface2>
```


IP header encapsulates traced packets between Endpoint A and SIP Interface 1, and between Endpoint B and SIP Interface 2 sent to trace server (RFC 2003)
Software protocol analyzer understands RFC 2003, and parses original traced packets
TRACE SERVER
INVITE
180 Ringing
BYE
200 OK
INVITE
180 Ringing
BYE
200 OK
ENDPOINT A
SBC
ENDPOINT B
INVITE
180 Ringing
BYE
200 OK
INVITE
180 Ringing
BYE
200 OK
● SIP Interface 1
□ SIP Interface 2

The commands you carry out for **`packet-trace local`** would take the following form:

```
ORACLE# packet-trace local F01:0 <"host IP address of Oracle Communications
Enterprise Session Border Controller interface1">
ORACLE# packet-trace local F02:0 <"host IP address of Oracle Communications
Enterprise Session Border Controller interface2">
```

Note: The system does not support egress RTP capture with Transcoding NIU

Running Packet Trace

There are four steps you take when you use the **`packet-trace remote`** feature. For **`packet-trace local`**, there are only two.

- Configuring the Oracle Communications Enterprise Session Border Controller with the trace server information so that the Oracle Communications Enterprise Session Border Controller knows where to send replicated data. (**`packet-trace remote`** only)
- Setting up the capture filter ip proto 4 in your software protocol analyzer if you only want to see the results of the Oracle Communications Enterprise Session Border Controller packet trace(s). (**`packet-trace remote`** only)
- Starting a packet trace.
- Stopping a packet trace.

This section provides information about how to perform all tasks.

Configuring a Trace Server

Trace servers only apply to **`packet-trace remote`**. You need to configure a trace server on the Oracle Communications Enterprise Session Border Controller; this is the device to which the Oracle Communications Enterprise Session Border Controller sends replicated data. The Oracle Communications Enterprise Session Border Controller supports one trace server.

To configure a trace server on your Oracle Communications Enterprise Session Border Controller:

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

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **`system`** and press Enter.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Enter **capture-receiver** and press Enter.

   ```
   ACMEPACKET(system)# capture-receiver
   ACMEPACKET(capture receiver)#
   ```

4. **state**—Type **enabled** so that you can use the trace server to which you want to send the mirrored packets for calls you are packet tracing. The default is **disabled**. The valid values are:

   - enabled | disabled

     Disable capture receivers you are not actively using for traces to prevent potential service outages caused by the capture's system resource utilization.

5. **address**—Enter the IP address of the trace server; there is no default.
6. **network-interface**—Enter the name and subport of the Oracle Communications Enterprise Session Border Controller network interface from which the Oracle Communications Enterprise Session Border Controller is to send mirrored packets. Your entry needs to take the form name:subport. The default is `:0`.
7. Save and activate your configuration.

Starting a Remote Packet Trace

You use the start a remote packet trace by entering the appropriate ACLI command with these pieces of information:

- Network interface (name:subport ID combination)
- IP address to be traced; if you do not enter local and/or remote ports when you start the trace, the Oracle Communications Enterprise Session Border Controller will trace all ports
- (Optional) Local UDP/TCP port on which the Oracle Communications Enterprise Session Border Controller sends and receives traffic to be traced
- (Optional) Remote UDP/TCP port to which the Oracle Communications Enterprise Session Border Controller sends traffic, and from which it receives traffic to be traced; you cannot enter the remote port without specifying a local port

  To start a packet trace with local and remote ports specified:

  Enter the ACLI **packet-trace remote** command followed by a Space, and the word **start**. After another Space, type in the name and subport ID for the network interface followed by a Space, the IP address to be traced followed by a Space, the local port number followed by a Space, and then optionally the remote port number. Then press Enter.

  ```
  ACMEPACKET# packet-trace remote start core:0 192.168.10.99 5060 5060
  Trace started for 192.168.10.99
  ```

Stopping a Remote Packet Trace

You stop a remote packet trace by entering the appropriate ACLI command with these pieces of information:

- Network interface (name:subport ID combination)
- IP address to be traced
- (Optional) Local UDP/TCP port on which the Oracle Communications Enterprise Session Border Controller sends and receives traffic to be traced
- (Optional) Remote UDP/TCP port to which the Oracle Communications Enterprise Session Border Controller sends traffic, and from which it receives traffic to be traced

If the packet trace you want to stop has no entries for local and/or remote ports, then you do not have to specify them.

1. To stop a packet trace with local and remote ports specified, enter the ACLI **packet-trace remote** command followed by a Space, and the word **stop**. After another Space, type in the name and subport ID for the network interface followed by a Space, the IP address to be traced followed by a Space, the local port number followed by a Space, and then optionally the remote port number. Then press Enter.

   ```
   ACMEPACKET# packet-trace remote stop core:0 192.168.10.99 5060 5060
   ```

2. To stop all packet traces on the Oracle Communications Enterprise Session Border Controller, enter the ACLI **packet-trace remote** command followed by a Space, and the word **stop**. After another Space, type the word **all** and press Enter.

   ```
   ACMEPACKET# packet-trace remote stop all
   ```

Starting a Local Packet Trace

You use the start a packet trace by entering the appropriate ACLI command with these pieces of information:

- Network interface (name:subport ID combination)
- (Optional) Enter a tcpdump command line within quotes

Note that the system supports local packet trace on all platforms.

Enter the ACLI **packet-trace local** command followed by a Space. Type in the name and subport ID for the network interface followed by a Space, the IP address to be traced followed by a Space, the local port number followed by a Space, and then optionally the remote port number. Then press Enter.

```
ACMEPACKET# packet-trace local s0p0 "host 192.168.12.12"
Files found in trace directory. Remove [y/n]?: y
File: /opt/traces/s0p0_0_00001_20150723095442.pcap
Packets: 5 Packets dropped: 0
```

The ACLI session does not accept use input while the **packet-trace local** command is running.

Stopping a Local Packet Trace

Type Ctrl-C to stop a local packet trace. This also re-enables the command line session.

Packet Trace Over VNF Systems

The Oracle Communications Enterprise Session Border Controller's packet trace tool provides the user with the ability to capture traffic from the Oracle Communications Enterprise Session Border Controller itself. The packet capture command is documented elsewhere, but the syntax and operation for VNF systems is not the same.

There are two capture modes across the product line, one that saves traffic locally and one that mirrors traffic to a user-specified target. Software only deployments support local capture only.

The user invokes the tool from the ACLI, manually specifying:

- How to capture (local only for VNF)
- What to capture
- Capture start and stop

Local capture supports PCAP filters to specify the signaling traffic you want to capture. The default packet trace filter uses the specified interface to capture both ingress and egress traffic. The user can then use the **packet-trace** command's syntax to filter based on target IP ad well as local and remote port. To further specify captured traffic, the user can also append the command with a PCAP filter enclosed in quotes. PCAP filter syntax is widely published.

The user can run only a single capture on a given interface. However, the user can run multiple captures simultaneously, as long as they are on separate interfaces.

Local packet capture is dependent on access control configuration, not capturing any denied traffic.

Note: Do not run packet-trace simultaneously with other Oracle Communications Enterprise Session Border Controller replication features, such as LI, SRS, SIP Monitoring and Trace, and Call Recording. These features may interfere with each other, corrupting each's results.

Packet Trace Local enables the Oracle Communications Enterprise Session Border Controller to capture traffic between two endpoints, or between itself and a specific endpoint. To accomplish this, the Oracle Communications Enterprise Session Border Controller replicates the packets sent and received and saves them to disk in .PCAP format.

By default, the system saves the .PCAP file in /opt/traces, naming it with the applicable interface name as well as the date and time of the capture. Alternatively, the user can specify file name using the system supports the PCAP filter flags -w.

The system rotates the PCAPs created in this directory by size. The last 25 files are kept and are rotated when they reach 100 MB. If there are capture files in the /opt/traces directory when this command is run, the system prompts the user to remove them before running new captures. If preferred, the user can decline this file deletion.

Starting a Local Packet Trace on VNF Systems

You use the start a packet trace by entering the appropriate ACLI command with these pieces of information:

- Network interface (name:subport ID combination)
- (Optional) IP address to be traced; if you do not enter local and/or remote ports when you start the trace, the Oracle Communications Enterprise Session Border Controller traces all open sockets.
- (Optional) Local UDP/TCP port on which the Oracle Communications Enterprise Session Border Controller sends and receives traffic to be traced.
- (Optional) Remote UDP/TCP port to which the Oracle Communications Enterprise Session Border Controller sends traffic, and from which it receives traffic to be traced; you cannot enter the remote port without specifying a local port.
- (Optional) Enter a tcpdump command line within quotes.

Note that the system supports local packet trace on all platforms. To start a packet trace with local and remote ports specified:

1. Enter the ACLI **`packet-trace local`** command followed by a Space, and the parameter **`start`**. After another space, type in the name and subport ID for the network interface followed by a Space.

   The syntax below includes the IP address to be traced, the local port number, then the remote port number separated by spaces.
2. Press Enter.

```
ORACLE# packet-trace local start core:0 192.168.10.99 5060 5060
Trace started for 192.168.10.99
```

Stopping a Local Packet Trace on VNF Systems

You stop a local packet trace by entering the appropriate ACLI command with these pieces of information:

- Network interface (name:subport ID combination)
- (Optional) IP address to be traced
- (Optional) Local UDP/TCP port on which the Oracle Communications Enterprise Session Border Controller sends and receives traffic to be traced
- (Optional) Remote UDP/TCP port to which the Oracle Communications Enterprise Session Border Controller sends traffic, and from which it receives traffic to be traced

If the packet trace you want to stop has no entries for local and/or remote ports, then you do not have to specify them.

1. To stop a packet trace with local and remote ports specified, enter the ACLI **packet-trace local** command followed by a Space, and the word **stop**. After another Space, type in the name and subport ID for the network interface followed by a Space, the IP address to be traced followed by a Space, the local port number followed by a Space, and then optionally the remote port number. Then press Enter.

   ```
   ORACLE# packet-trace local stop core:0 192.168.10.99 5060 5060
   ```

2. To stop all packet traces on the Oracle Communications Enterprise Session Border Controller, enter the ACLI **packet-trace local** command followed by a Space, and the word **stop**. After another Space, type the word **all** and press Enter.

   ```
   ORACLE# packet-trace local stop all
   ```

4

Persistent Protocol Tracing

This section explains how to configure persistent protocol tracing to capture specific SIP protocol message logs and persistently send them off the Oracle Communications Enterprise Session Border Controller, even after rebooting the system. This feature is not applicable to log for H.323 or IWF.

About Persistent Protocol Tracing

You can configure sending protocol message logs off of the Oracle Communications Enterprise Session Border Controller, and have that persist after a reboot. You no longer have to manually issue the notify command each time you reboot.

To support persistent protocol tracing, you configure the following system-config parameters:

- **`call-trace`**—Enable/disable protocol message tracing (currently only sipmsg.log and alg.log) regardless of the process-log-level setting. If the process-log-level is set to trace or debug, call-trace will not disable.
- **`internal-trace`**—Enable/disable internal ACP message tracing for all processes, regardless of process-log-level setting. This applies to all *.log (internal ACP message exchange) files other than sipmsg.log and alg.log. If the process-log-level is set to trace or debug, call-trace will not disable.
- **`log-filter`**—Determine what combination of protocol traces and logs are sent to the log server defined by the process-log-ip parameter value. You can also fork the traces and logs, meaning that you keep trace and log information in local storage as well as sending it to the server. You can set this parameter to any of the following values: none, traces, traces-fork, logs, logs, all, or all-fork.

  The Oracle Communications Enterprise Session Border Controller uses the value of this parameter in conjunction with the process-log-ip and process-log-port values to determine what information to send. If you have configured the proc-log-ip and proc-log-port parameters, choosing traces sends just the trace information (provided they are turned on), logs sends only process logs (log.*), and all sends everything (which is the default).

About the Logs

When you configure persistent protocol tracing, you affect the following types of logs.

Note: Enabling logs can have an impact on Oracle Communications Enterprise Session Border Controller performance.

Process Logs

Events are logged to a process log flow from tasks and are specific to a single process running on the Oracle Communications Enterprise Session Border Controller. By default they are placed into individual files associated with each process with the following name format:

log.<taskname>

By setting the new log-filter parameter, you can have the logs sent to a remote log server (if configured). If you set log-filter to logs or all, the logs are sent to the log server. Otherwise, the logs are still captured at the level the process-log-level parameter is set to, but the results are stored on the Oracle Communications Enterprise Session Border Controller's local storage.

Communication Logs

These are the communication logs between processes and system management. The logs are usually named <name>.log, with <name> being the process name. For example, sipd.log.

This class of log is configured by the new internal-trace parameter.

Protocol Trace Logs

The only protocol trace logs included at this time are sipmsg.log for SIP. The H.323 system tracing is not included. All of the logs enabled with the call–trace parameter are sent to remote log servers, if you also set the log-filter parameter to logs or all.

Persistent Protocol Tracing Configuration

Before you configure persistent protocol tracing, ensure you have configured the process logs by setting the system configuration's **`process-log-ip`** parameter.

To configure persistent protocol tracing:

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

   ```
   ORACLE# configure terminal
   ```

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

   ```
   ORACLE(configure)# system
   ```

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

   ```
   ORACLE(system)# system-config
   ORACLE(system-config)#
   ```

4. **`call-trace`**—Set to **`enabled`** to enable protocol message tracing for sipmsg.log for SIP. The default is **`disabled`**. The valid values are:

   - enabled | disabled

5. **`internal-trace`**—Set to **`enabled`** to enable internal ACP message tracing for all processes. The default is **`disabled`**. The valid values are:

   - enabled | disabled

6. **`log-filter`**—Choose the appropriate setting for how you want to send and/or store trace information and process logs. The valid values are:

   - **`none`**—No information will be sent or stored.
   - traces—Sends the trace information to both the log server; includes <name>.log files that contain information about the Oracle Communications Enterprise Session Border Controller's internal communication processes (<name> is the name of the internal process)

- **`traces-fork`**—Sends the trace information to both the log server and also keeps it in local storage; includes <name>.log files that contain information about the Oracle Communications Enterprise Session Border Controller’s internal communication processes (<name> is the name of the internal process)
- **`logs`**—Sends the process logs to both the log server; includes log.* files, which are Oracle Communications Enterprise Session Border Controller process logs
- **`logs-fork`**—Sends the process logs to both the log server and also keeps it in local storage; includes log.* files, which are Oracle Communications Enterprise Session Border Controller process logs
- **`all`**—Sends all logs to the log servers that you configure
- **`all-fork`**—Sends all logs to the log servers that you configure, and it also keeps the logs in local storage

7. Save and activate your configuration.

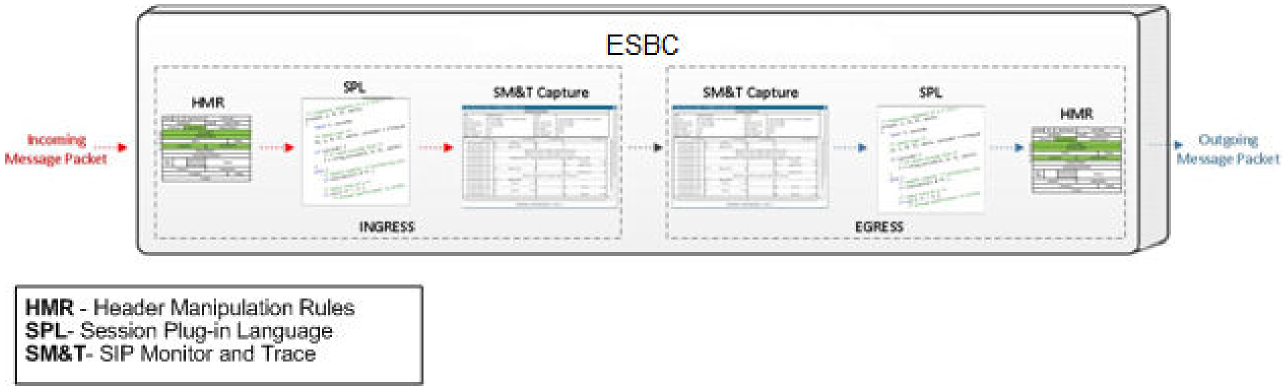
5

SIP Monitor and Trace

Introduction to SIP Monitor and Trace

SIP Monitor and Trace provides the ability to set filters on the Oracle Communications Enterprise Session Border Controller (ESBC) for filtering SIP session data, and displaying the results in a Web-based Graphical User Interface (GUI). You can use the data to troubleshoot the ESBC.

The SIP Monitor and Trace feature allows the ESBC to monitor SIP sessions. The ESBC captures SIP messages, applies the Header Manipulation Rules (HMR) configured on the ESBC, and applies the Session Plug-in Language (SPL) to that message. When the message is sent out, the ESBC applies the SPL, then applies the HMR, and then sends out the captured SIP message.



You configure the monitoring process to filter the active session data from original ingress or final egress SIP session messages. The filters are based on the Acme Packet Command Line Interface (ACLI)-configured filters matching criteria or dynamic events that occur, and are used for the purpose of troubleshooting SIP sessions on the network.

As the system monitors active sessions, it captures data using the following filters:

- `Static filters`—Filters you specify that filter the data on ingress and egress requests and responses in the SIP session dialogs.

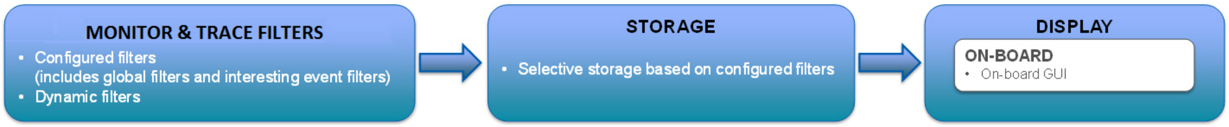
  You configure these filters from the ACLI as part of the ESBC configuration. The configured filters save to the current configuration after you save and activate the configuration.
- `Dynamic filters`—Filters you specify that match information in the ingress and egress SIP messages according to the filters you dynamically specified.

You configure these filters from the ACLI, but there is no change to the current configuration. The filters take effect immediately and do not require the Save and Activate commands. Oracle recommends using dynamic filters when you want to set specific filters without changing the current configuration.

For more information about configuring static filters and dynamic filters, see *Filters to Configure* and *Dynamic Filters*

When you enable a filter configuration, the system matches the values in the configured filters to the headers of messages before applying any changes. When the system finds no match in the headers during monitoring, the system uses the filter defaults in the system configuration to perform the filtering. The system logs the filter results along with any additional call details and displays the results in the GUI.

The following illustration shows the SIP Monitor and Trace flow process.


MONITOR & TRACE FILTERS
• Configured filters (includes global filters and interesting event filters)
• Dynamic filters
STORAGE
• Selective storage based on configured filters
DISPLAY
ON-BOARD
• On-board GUI

The ESBC supports the following numbers of SIP monitor and trace sessions for all platforms.

- On systems with less than 4Gb Ram—2000 sessions.
- On systems with more than 4Gb Ram—4000 sessions.

Configure the Web Server From the ACLI

You must configure and enable the Web server for Oracle Communications Enterprise Session Border Controller (ESBC) operations before you can use the Web GUI.

If you previously ran the Set Initial Configuration wizard from the Web GUI, confirm whether or not the Web GUI is already enabled.

The following procedure provides instructions to configure and enable the Web server through the ACLI.

Note: The Web GUI supports only IPv4.

To configure the Web server:

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

```
ACMEPACKET# configure terminal
```

2. Type **`system`** and press Enter to access the system-related objects.

```
ACMEPACKET(configure)# system
ACMEPACKET(system)#
```

3. Type **`web-server-config`** and press Enter to access the event monitoring-related attributes.

```
ACMEPACKET(system)# web-server-config
ACMEPACKET(web-server-config)#
```

`state`—Enter whether or not to enable the Web GUI. Default is enabled. Valid values are:

- enabled
- disabled

```
ACMEPACKET(web-server-config)# state enabled
```

`inactivity-timeout`—Enter the amount of time, in minutes, that the Web GUI must have remained inactive before it ends Web session. For example, if the timeout value is set as 5, the Web session disconnects after 5 minutes of inactivity. Default is **`5`**. Valid values are **`0`** to **`20`**.

```
ACMEPACKET(web-server-config)# inactivity-timeout 5
```

Note: The following http-state, http-port, https-state, and https-port parameters may have already been set through the Web GUI installation wizard on the ESBC. You can edit these parameters using the ACLI.

`http-state`—Enter whether or not to enable HTTP for accessing the Web server. Default is enabled. Valid values are:

- enabled
- disabled

```
ACMEPACKET(web-server-config)# http-state enabled
```

`http-port`—Enter the HTTP port to use to connect to the Web server. Default is 80. Valid values are **`1`** to **`65535`**.

```
ACMEPACKET(web-server-config)# http-port 80
```

`https-state`—Enter whether or not to enable HTTPS (secure connection) for accessing the Web server. Default is disabled. Valid values are:

- enabled (default)
- disabled

```
ACMEPACKET(web-server-config)# https-state enabled
```

`https-port`—Enter the HTTPS port to use to connect to the Web server. Default is 443. Valid values are **`1`** to **`65535`**.

```
ACMEPACKET(web-server-config)# https-port 443
```

`tls-profile`—Enter the Transport Layer Security (TLS) Protocol profile name to use with HTTPS. Default is blank. Valid values are **`alpha-numeric characters`**.

```
ACMEPACKET(web-server-config)# tls-profile tlsSM&T
```

Note: If you specify a tls-profile, and HTTP is enabled, the ESBC checks against the TLS profile table for a match. If there is no match, the applicable errors display during the verification of the configuration. To create a TLS profile, see Chapter 22, Configuring a TLS Profile.

4. Enter **`exit`** to exit the Web server configuration.

```
ACMEPACKET(web-server-config)# exit
```

5. Enter **`exit`** to exit the system configuration.

```
ACMEPACKET(system)# exit
```

6. Enter **`exit`** to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

7. Enter **`save-config`** to save the configuration.

```
ACMEPACKET# save-config
```

8. Enter **`activate-config`** to activate as the current configuration.

```
ACMEPACKET# activate-config
```

Filter Objects

The Oracle Communications Enterprise Session Border Controller (ESBC) provides configuration objects you can set on the ESBC to customize filters for SIP Monitor and Trace. The system can monitor and filter specific SIP session data and display it to the GUI. The filter objects you can configure include:

Filters	Description
filter-config	Object that allows you to create custom filters to use for SIP Monitor and Trace. You can then configure session agents (SA) and/or realms to use these filters, or set sip-monitoring to use the filters on a global basis. For more information, see *Creating Custom Filters* .
sip-monitoring	Object that allows you to configure SIP Monitor and Trace features. Note: You must configure the sip-monitoring object to enable filtering. A session agent and/or realm must also be configured, or you must set filtering on a global basis, for Monitor and Trace to occur.
state	Attribute that enables/disables SIP Monitor and Trace. For more information, see *Enabling Disabling SIP Monitoring & Tracing* .
monitoring-filters	Attribute that allows you to specify the name of the custom filter(s) to use on a global basis. This value is based on the filter(s) created in "filter-config". You can also specify an * (asterisk) as a value for this attribute, which monitors all session data on the ESBC. For more information, see *Using Filters to Monitor on a Global-Basis* .
interesting-events	Object that allows you to configure the following attributes: type - Sets the interesting events to monitor (short-session, local-rejection) trigger-threshold - Sets the number of interesting events that must occur within the time set by the trigger-window value. If the number of events reaches the trigger-threshold during the trigger-window time, monitoring is started. trigger-timeout - Sets the amount of time, in seconds, that the trigger is active once monitoring has started. If no interesting events occur in the time frame set for this value, monitoring never starts. For example, if trigger-timeout is set to 40, and no interesting events occur in 40 seconds, then monitoring never starts. Note: Interesting Events are always enabled on a global-basis on the ESBC. For more information, see Configuring Interesting Events .
trigger-window	Attribute to specify the amount of time, in seconds, for the window of time that the trigger-threshold value must reach before monitoring begins. For example, if type is set to short-session, trigger-threshold is set to 2, and trigger-window is set to 60, monitoring begins when the ESBC has discovered 2 short-session events in a 60 second window. For more information, see *Configuring a Trigger Window* .

The following paragraphs provide information and procedures for configuring these features.

Creating Custom Filters

You can create single or multiple custom, session filters on the Oracle Communications Enterprise Session Border Controller for Monitor and Trace purposes. These filters allow incoming and outgoing session data to be filtered with specific information and then displayed to the GUI. You can use the custom filter(s) during monitoring on a global basis, or when monitoring session agent (SAs) and/or realms.

You create custom filters using the **`filter-config`** object at the path **Configure terminal** > **session-router** > **filter-config** in the ACLI.

The following table identifies the attributes you can configure for each filter.

Filters	Description
filter-config	Object that allows you to create a custom filter(s) to be used for Monitor and Trace on the Net-Net ESD.
name	Name of the custom filter. Note: You specify this filter name when configuring global monitoring, SA monitoring, and/or realm monitoring.
address	IP address to be filtered. Depending on the value you specify for this attribute, filtering matches the IP address or IP address and netmask, in the message header. For example: 1.1.1.1 is <IP address> 1.1.1.1/24 is <IP address>/<Netmask>
user	Phone number or user-part to be filtered. Depending on the value you specify for this attribute, filtering matches the phone number string or the user-part with the following header information if it exists in the message: From URI, To URI, Request URI, P-Preferred URI, P-Asserted Identity, P-Associated URI, P-Called Party URI.

You can define a single or multiple filters with specific names and then specify the filter name(s) to use for global monitoring, session agent monitoring, and/or realm monitoring.

Creating a Custom Filter

Use the following procedure to create a custom filter on the Net-Net ESD.

To configure a filter(s):

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 **`filter-config`** and press Enter to access the filter configuration-related attributes.

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

`name`—Enter a name to assign to this filter. Valid values are alpha-numeric characters. Default is blank.

```
ACMEPACKET(filter-config)# name FILTER1
```

Note: You can use this filter name when configuring monitoring on a global-basis, or when monitoring session-agents and/or realms.

`address`—Enter the IP address to apply to this filter. You can specify netmask if required. IP Address must be entered in dotted decimal format (0.0.0.0). Default is 0.0.0.0.

```
ACMEPACKET(filter-config)# address 1.1.1.1    (filters on IP address)
ACMEPACKET(filter-config)# address 1.1.1.1/24 (filters on IP address and
netmask)
```

user—Enter a phone number or user-part to apply to this filter. Valid values are numeric characters. Default is blank.

```
ACMEPACKET(filter-config)# user 5551212
```

You must specify either the phone number OR user part for the user attribute. If you want both the phone number AND user part to be filtered, you must create separate filters to set each value.

4. Enter **done** to save the filter.

```
ACMEPACKET(filter-config)# done
```

5. Enter **exit** to exit the filter configuration.

```
ACMEPACKET(filter-config)# exit
```

6. Enter **exit** to exit the session-router configuration.

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

7. Enter **exit** to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

8. Enter **save-config** to save the filter configuration.

```
ACMEPACKET# save-config
```

9. Enter **activate-config** to activate the filter configuration.

```
ACMEPACKET# activate-config
```

Multiple Custom Filter Examples

The following examples show three custom filters (FILTER1, FILTER2, and FILTER3) created for SIP Monitor and Trace on the Net-Net ESD.

- Filter 1

```
ACMEPACKET(filter-config)# name FILTER1
ACMEPACKET(filter-config)# address 1.1.1.1
ACMEPACKET(filter-config)# user 5551212
```

- Filter 2

```
ACMEPACKET(filter-config)# name FILTER2
ACMEPACKET(filter-config)# address 3.3.3.3/24
ACMEPACKET(filter-config)# user 1781
```

- Filter 3

```
ACMEPACKET(filter-config)# name FILTER3
ACMEPACKET(filter-config)# user sip
```

You can specify the Net-Net ESD monitoring process to use FILTER1, FILTER2, and/or FILTER3 for global monitoring, or for monitoring SAs and/or realms. However, before you apply the custom filters, you can enable/disable SIP monitoring on the Net-Net ESD.

To enable/disable SIP monitoring, see *Enabling Disabling SIP Monitoring & Tracing*. To use a custom filter(s) on a global basis, see *Using Filters to Monitor on a Global-Basis*. To use a custom filter(s) when monitoring SAs, see *Using Filters when Monitoring Session Agents*. To use a custom filter(s) when monitoring realms, see *Using Filters when Monitoring Realms*.

Enabling Disabling SIP Monitoring & Tracing

You can enable or disable the Oracle Communications Enterprise Session Border Controller (ESBC) to perform SIP monitoring using the state parameter at the path **Configure terminal** > **session-router** > **sip-monitoring** in the ACLI.

Use the following procedure to enable/disable SIP monitoring on the ESBC.

Note: You must enable the sip-monitoring object for monitoring and filtering to occur on the ESBC. With sip-monitoring enabled, you can configure a filter(s) on a global basis, as well as for a session agent and/or a realm. You can also initiate dynamic filter commands.

To enable/disable sip-monitoring:

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 **`sip-monitoring`** and press Enter to access the SIP monitoring-related attributes.

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

   `state`—Enter whether or not to enable the sip monitoring on the ESBC. Default is enabled. Valid values are:

   - enabled (default)
   - disabled

4. Enter **`done`** to save the setting.

```
ACMEPACKET(sip-monitoring)# done
```

5. Enter **`exit`** to exit the sip-monitoring configuration.

```
ACMEPACKET(sip-monitoring)# exit
```

6. Enter **`exit`** to exit the session-router configuration.

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

7. Enter **`exit`** to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

8. Enter **`save-config`** to save the filters.

```
ACMEPACKET# save-config
```

9. Enter **`activate-config`** to activate the filters in the current configuration.

```
ACMEPACKET# activate-config
```

10. Configure global filters, or assign filters to a session agent and/or realm. For more information, see the following:

   - *Using Filters to Monitor on a Global-Basis*
   - *Using Filters when Monitoring Session Agents*
   - *Using Filters when Monitoring Realms*

   With sip-monitoring enabled, you can also initiate dynamic filter commands if required. For more information about dynamic filter commands, see *Dynamic Filter Commands*.

Using Filters to Monitor on a Global-Basis

The Oracle Communications Enterprise Session Border Controller (ESBC) allows you to filter SIP session data on a global-basis using the monitoring-filters object at the path **Configure terminal** > **session-router** > **sip-monitoring** > **monitoring-filters** in the ACLI. You can apply a single or multiple custom filter for global monitoring. For more information about creating a custom filter, see *Creating a Custom Filter*.

Note: For SIP Monitor and Trace to trigger interesting-events, a filter value must be configured for the monitoring-filters object.

To configure filtering on a global basis:

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 **sip-monitoring** and press Enter to access the SIP monitoring-related attributes.

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

4. Type **select** and press Enter to select the sip-monitoring objects.

```
ACMEPACKET(sip-monitoring)# select
ACMEPACKET(sip-monitoring)#
```

monitoring-filters—Enter the custom filter name(s) you want to use when monitoring on a global-basis. You can enter multiple filter names in a comma-separated list (with no spaces) if required. To add to an existing filter list, use the "+" before the filter name you are adding. Use a "-" to remove filter names. Enter an * (asterisk) to filter all session data.

```
ACMEPACKET(sip-monitoring)# monitoring-filters FILTER1,FILTER2
ACMEPACKET(sip-monitoring)# monitoring-filters FILTER1,FILTER2 +FILTER3
ACMEPACKET(sip-monitoring)# monitoring-filters FILTER1,FILTER2 -FILTER3
ACMEPACKET(sip-monitoring)# monitoring-filters *
```

Note: If you enter the * with a filter name, the filter name is ignored and the ESBC uses the * to filter all session data.

5. Enter **done** to save the configuration.

```
ACMEPACKET(sip-monitoring)# done
```

6. Enter **exit** to exit the sip-monitoring configuration.

```
ACMEPACKET(sip-monitoring)# exit
```

7. Enter **done** to save the sip-monitoring configuration.

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

8. Enter **exit** to exit the session-router configuration.

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

9. Enter **exit** to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

10. Enter **save-config** to save the configuration.

```
ACMEPACKET# save-config
```

11. Enter **activate-config** to activate the configuration.

```
ACMEPACKET# activate-config
```

Using Filters when Monitoring Session Agents

You can configure the Oracle Communications Enterprise Session Border Controller (ESBC) to perform filtering of SIP session data for session agent (SA) configurations. You must specify the hostname of the SA and the filter to use to perform the filtering, at the path **Configure terminal** > **session-router** > **session-agent**in the ACLI. For more information about creating a custom filter, see *Creating a Custom Filter*.

To configure filtering for 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 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 **select** and press Enter.

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

hostname—Specify the hostname of the session agent to which you want to apply the custom filter(s).

```
ACMEPACKET(session-agent)# hostname SA1
```

monitoring-filters—Enter the custom filter name(s) you want to use when monitoring on a global-basis. You can enter multiple filter names in a comma-separated list (with no spaces) if required. To add to an existing filter list, use the "+" before the filter name you are adding. Use a "-" to remove filter names. Enter an * (asterisk) to filter all SIP session data.

```
ACMEPACKET(session-agent)# monitoring-filters FILTER1,FILTER2
ACMEPACKET(session-agent)# monitoring-filters FILTER1,FILTER2 +FILTER3
ACMEPACKET(session-agent)# monitoring-filters FILTER1,FILTER2 -FILTER3
ACMEPACKET(session-agent)# monitoring-filters *
```

Note: If you enter the * with a filter name, the filter name is ignored and the ESBC uses the * to filter all session data.

5. Enter **done** to save the configuration.

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

6. Enter **exit** to exit the session-agent configuration.

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

7. Enter **done** to save the configuration.

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

8. Enter **exit** to exit the session-router configuration.

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

9. Enter **exit** to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

10. Enter **save-config** to save the configuration.

```
ACMEPACKET# save-config
```

11. Enter **activate-config** to activate the configuration.

```
ACMEPACKET# activate-config
```

Using Filters when Monitoring Realms

You can configure the Oracle Communications Enterprise Session Border Controller (ESBC) to perform filtering of SIP session data for realm configurations. You must specify the realm identifier and the filter to use to perform the filtering, at the path Configure terminal->media-manager->realm-config in the ACLI. For more information about creating a custom filter, see *Creating a Custom Filter*.

To configure filtering for a realm:

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

```
ACMEPACKET# configure terminal
```

2. Type **media-manager** and press Enter to access the media manager-related objects.

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

3. Type **realm-config** and press Enter to access the realm configuration-related attributes.

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

4. Type **select** and press Enter.

```
ACMEPACKET(realm-config)# select
ACMEPACKET(realm-config)#
```

identifier—Specify the identifier of the realm to which you want to apply the custom filter(s).

```
ACMEPACKET(realm-config)# identifier REALM1
```

monitoring-filters—Enter the custom filter name(s) you want to use when monitoring on a global-basis. You can enter multiple filter names in a comma-separated list (with no spaces) if required. To add to an existing filter list, use the "+" before the filter name you are adding. Use a "-" to remove filter names. Enter an * (asterisk) to filter all SIP session data.

```
ACMEPACKET(realm-config)# monitoring-filters FILTER1,FILTER2
ACMEPACKET(realm-configg)# monitoring-filters FILTER1,FILTER2 +FILTER3
ACMEPACKET(realm-config)# monitoring-filters FILTER1,FILTER2 -FILTER3
ACMEPACKET(realm-config)# monitoring-filters *
```

Note: If you enter the * with a filter name, the filter name is ignored and the ESBC uses the * to filter all session data.

5. Enter **done** to save the configuration.

```
ACMEPACKET(realm-config)# done
```

6. Enter **exit** to exit the realm-config configuration.

```
ACMEPACKET(realm-config)# exit
```

7. Enter **done** to save the configuration.

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

8. Enter **exit** to exit the media-manager configuration.

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

9. Enter **exit** to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

10. Enter **save-config** to save the configuration.

```
ACMEPACKET# save-config
```

11. Enter **activate-config** to activate the configuration.

```
ACMEPACKET# activate-config
```

Global SA and Realm Filter Examples

The following are examples of global, session agent, and realm filters configured on the Oracle Communications Enterprise Session Border Controller. These examples assume that FILTER1, FILTER2, and FILTER3 have been pre-configured as custom filters.

Global Filter

```
ACMEPACKET(sip-monitoring)# monitoring-filters FILTER1,FILTER3
```

This filter captures the SIP session data based on the filter settings in FILTER1 and FILTER3 only, for all sessions on the Net-Net ESD.

Session Agent Filters

```
ACMEPACKET(session-agent)# hostname SA1
ACMEPACKET(session-agent)# monitoring-filters FILTER2
ACMEPACKET(session-agent)# hostname SA2
ACMEPACKET(session-agent)# monitoring-filters FILTER2,FILTER3
```

These filters capture the SIP session data for SA1 only, based on the filter settings in FILTER2, and the SIP session data for SA2 only, based on the filter settings in FILTER2 and FILTER3.

Realm Filters

```
ACMEPACKET(realm-config)# identifier REALM1
ACMEPACKET(realm-config)# monitoring-filters *
ACMEPACKET(realm-config)# identifier REALM2
ACMEPACKET(realm-config)# monitoring-filters FILTER1
```

These filters capture all SIP session data for REALM1, and the SIP session data for REALM2 only, based on the filter settings in FILTER1.

Note: If you leave a monitoring-filter field blank, no monitoring takes place for that object.

Interesting Events

Interesting events on the Oracle Communications Enterprise Session Border Controller (ESBC) are those events that are considered “interesting” for the purpose of troubleshooting SIP sessions in your network. You can specify the type of interesting event you want to filter using the object, interesting-events at the path, **Configure terminal** > **session-router** > **sip-monitoring** > **interesting-events** in the ACLI.

Currently, there are two types of interesting events that the ESBC can monitor:

- **`short-session`** (short session events on the ESBC)
- **`local-rejection`** (local rejection events on the ESBC)

You can use the following trigger attributes to specify time provisioning for the interesting events:

- **`trigger-threshold`**
- **`trigger-timeout`**

Note: You can also set a trigger-window object to support these trigger attributes. For more information, see *Configuring a Trigger Window*.

The following table identifies the attributes you can set for the interesting-events object.

Filter	Description
interesting-events	Allows you to configure trigger attributes that apply to the filters you set on the ESBC. You can configure the following interesting-event attributes: Note: Interesting Events are always enabled on a global-basis on the ESBC.
type	Sets the interesting events to monitor (short-session, local-rejection)
trigger-threshold	Sets the number of interesting events that must occur within the time set by the trigger-window value. If the number of events reaches the trigger-threshold during the trigger-window time, monitoring is started.
trigger-timeout	Sets the amount of time, in seconds, that the trigger is active once monitoring has started. If no interesting events occur in the time frame set for this value, monitoring never starts. For example, if trigger-timeout is set to 40, and no interesting events occur in 40 seconds, then monitoring never starts.

The ESBC considers short session and local rejection interesting events. A session is viewed as a short session if the length of time, in seconds, is equal to or below the short-session-duration value configured at the path **Configure terminal** > **session-router** > **session-router-config** > **short-session-duration**. A local rejection can occur when sessions are locally rejected at the ESBC for any reason (for example, Session Agent (SA) unavailable, no route found, SIP signaling error, etc.)

If a short session or local rejection event occurs, the ESBC uses the values configured for the trigger attributes to determine when to start filtering the SIP session data.

If a short session event occurs when the ESBC is NOT monitoring, the event information is taken from the last BYE that occurred in the session; therefore, only some parts of the call flow may display in the GUI. If a local rejection event occurs when the ESBC is NOT monitoring, it displays only the information in the last rejected transaction.

Use the following procedure to configure interesting events for SIP Monitor and Trace on the ESBC.

Interesting Events Configuration

To configure interesting events:

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-router** again and press Enter to access the session router configuration-related attributes.

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

short-session-duration—Enter the maximum session duration, in seconds, to be considered a short session. Default is 0 (disabled). Valid values are 0 to 999999999.

```
ACMEPACKET(session-router-config)# short-session-duration 30
```

4. Enter **done** to save the filters.

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

5. Enter **exit** to exit the interesting-events configuration.

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

6. Type **sip-monitoring** and press Enter to access the SIP monitoring-related attributes.

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

7. Type **select** and press Enter.

```
ACMEPACKET(sip-monitoring)# select
ACMEPACKET(sip-monitoring)#
```

8. Type **interesting-events** and press Enter to access the interesting events-related attributes.

```
ACMEPACKET(sip-monitoring)# interesting-events
ACMEPACKET(interesting-events)#
```

type—Enter the type of interesting event you for which you want to filter. Default is blank and disables this filter. Valid values are:

- short-session
- local-rejection

```
ACMEPACKET(interesting-events)# type short-session
```

trigger-threshold — (optional) Enter the number of interesting events that must occur within the time set by the trigger window value. If the number of events reaches the trigger-threshold during the trigger-window time, monitoring is started.Default is 0 (disabled). Valid values are 0 to 999999999.

```
ACMEPACKET(interesting-events)# trigger-threshold 50
```

trigger-timeout —Sets the amount of time, in seconds, that the trigger is active once monitoring has started. If no interesting events occur in the time frame set for this value, monitoring never starts. Default is 0 (trigger always on). Valid values are 0 to 999999999.

```
ACMEPACKET(interesting-events)# trigger-timeout 30
```

9. Enter **done** to save the filters.

```
ACMEPACKET(interesting-events)# done
```

10. Enter **exit** to exit the interesting-events configuration.

```
ACMEPACKET(interesting-events)# exit
```

11. Enter **exit** to exit the sip-monitoring configuration.

```
ACMEPACKET(sip-monitoring)# exit
```

12. Enter **done** to save the configuration.

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

13. Enter **exit** to exit the session-router configuration.

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

14. Enter **exit** to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

15. Enter **save-config** to save the filters.

```
ACMEPACKET# save-config
```

16. Enter **activate-config** to activate the filters in the current configuration.

```
ACMEPACKET# activate-config
```

Configuring a Trigger Window

The trigger-window attribute specifies the amount of time, in seconds, for the window of time that the trigger-threshold value must reach before monitoring begins. For example, if “interesting-event” type is set to short-session, “trigger-threshold” is set to 2, and trigger-window is set to 60, monitoring begins when the Net-Net ESD has discovered 2 short-session events in a 60 second window.

Use the following procedure to configure a trigger window.

To configure a trigger window:

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 **sip-monitoring** and press Enter to access the SIP monitoring-related attributes.

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

4. Type **select** and press Enter.

```
ACMEPACKET(sip-monitoring)# select
ACMEPACKET(sip-monitoring)#
```

trigger-window—Enter the amount of time, in seconds, for the window of time that the trigger-threshold value must reach before monitoring begins. Default is 30. Valid values are 0 to 999999999. Zero (0) disables this the trigger-window parameter.

```
ACMEPACKET(sip-monitoring)# trigger-window 50
```

5. Enter **done** to save the filters.

```
ACMEPACKET(sip-monitoring)# done
```

6. Enter **exit** to exit the sip-monitoring configuration.

```
ACMEPACKET(sip-monitoring)# exit
```

7. Enter **done** to save the configuration.

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

8. Enter **exit** to exit the session-router configuration.

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

9. Enter **exit** to exit the configure mode.

```
ACMEPACKET(configure)# exit
```

10. Enter **save-config** to save the filters.

```
ACMEPACKET# save-config
```

11. Enter **activate-config** to activate the filters in the current configuration.

```
ACMEPACKET# activate-config
```

Example

The following is an example filter configuration, filtering interesting events with a trigger window on the Oracle Communications Enterprise Session Border Controller. These parameters perform filtering on a global basis.

Monitoring Enabled on a Global Basis

```
ACMEPACKET(sip-monitoring)# state enabled
```

Short-Session Configured

```
ACMEPACKET(interesting-events)# type short-session
ACMEPACKET(interesting-events)# trigger-threshold 2
ACMEPACKET(interesting-events)# trigger-timeout 60
```

Local-Rejection Configured

```
ACMEPACKET(interesting-events)# type local-rejection
ACMEPACKET(interesting-events)# trigger-threshold 1
ACMEPACKET(interesting-events)# trigger-timeout 0
```

Trigger-Window Configured

```
ACMEPACKET(sip-monitoring)# trigger-window 120
```

The configuration above has global SIP monitoring enabled and is set to capture interesting events that are short-session and local-rejection events.

Per the triggers for the short-session configuration, if 2 (trigger-threshold) short-session events occur in a window of 120 seconds (trigger-window), then monitoring is started. If no short-session events occur after 60 seconds (trigger-timeout), no monitoring is started.

Per the triggers for the local-rejection configuration, if more that 1 (trigger-threshold) local-rejection event occurs in a window of 120 seconds (trigger-window), then monitoring is started. The value of 0 (trigger-timeout) indicates that monitoring is always enabled for this event.

Dynamic Filters

The SIP Monitor and Trace feature provides a time-saving feature of adding filters dynamically, and turning the filters ON and OFF as required. The filtering process performs on a dynamic basis dependant on the filters you specify.

Dynamic Filter Commands

You can use the ACLI to initiate the following dynamic filtering commands:

- **capture start**—starts the filters you specify in the filter syntax
- **capture stop**—stops the filters you specify in the filter syntax

Note: Initiating these commands does not change the values set in the ACLI-configured filters on the Oracle Communications Enterprise Session Border Controller (ESBC). The ESBC uses the dynamic filters until you initiate a stop command.

The syntax for the dynamic filter commands are:

```
capture start <main filter> <subfilter(s)>
```

```
capture stop <main filter> <subfilter(s)>
```

You MUST enter a <main filter> and a <subfilter(s)> when initiating the “capture start” and capture stop commands.

The following table identifies the values you can use for each attribute in the command syntax.

Syntax Attribute	Values
<main filter>	global - monitors and captures all realm <realm name> - monitors and captures everything matching realm session-agent <session-agent name> - monitors and captures everything matching session agent. int-ev <short-session | local-rejection> - monitors and captures everything matching a short- session and/or local-rejection.
[<subfilter(s)>]	* - monitors and captures all sessions. user <Phone Number or User Part URI> - monitors and captures everything that matches this phone number or user part. addr-prefix <IP address or IP address and netmask> - monitors and captures everything that matches this address or address prefix.

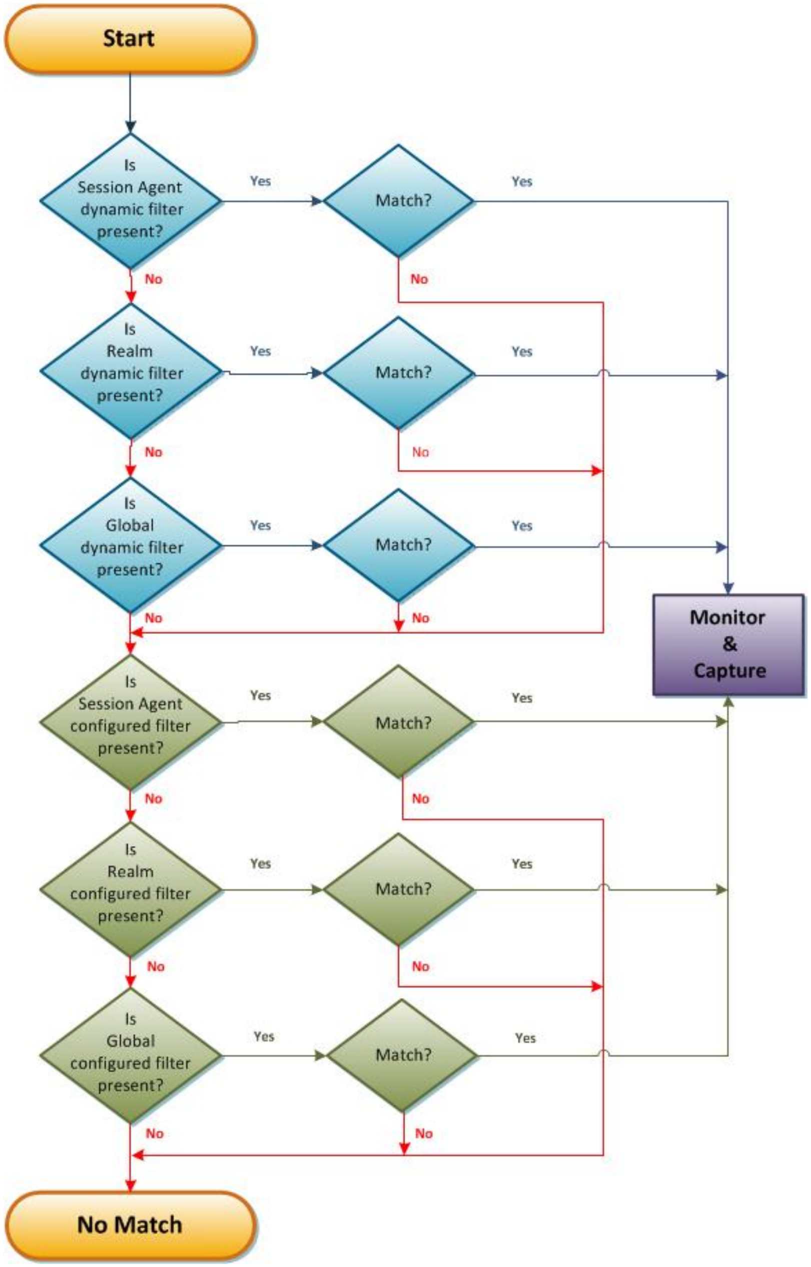
Examples

The following table provides examples for using the dynamic filter commands.

Example	Description
capture start global *	Captures all session data.
capture start global user USER1	Captures all session data for USER1.
capture start global addr-prefix 1.1.1.1	Captures all session data for IP address 1.1.1.1.
capture start global addr-prefix 1.1.1.1/24	Captures all session data for IP address 1.1.1.1 using netmask of 24.
capture start session-agent 172.1.1.1 addr-prefix 10.10.10.10	Captures session data for SA 172.1.1.1 at IP address 10.10.10.10.
capture start int-ev local-rejection	Captures session data for interesting events that occur that are of type local-rejection.
capture start int-ev short session	

Example	Description
	Captures session data for interesting events that occur that are of type short-session.

The following flow chart shows the dynamic filter process.


Start
Is Session Agent dynamic filter present?
Yes
No
Match?
Is Realm dynamic filter present?
Is Global dynamic filter present?
Monitor & Capture
Is Session Agent configured filter present?
Is Realm configured filter present?
Is Global configured filter present?
No Match

Note: Dynamic filters are only removed after a reboot/switchover of the Oracle Communications Enterprise Session Border Controller.

Issuing another dynamic command may or may not affect previous dynamic commands that were already initiated. If you issue a dynamic command with a <main filter> object, and then issue another command with the same <main filter> object, the new command tasks precedence. If you issue a dynamic command with a different <main filter> object, then the Oracle Communications Enterprise Session Border Controller uses both <main filter> commands to monitor traffic.

For example, if you enter the following dynamic command:

```
ORACLE# capture start realm1 user 123
```

and then enter:

```
ORACLE# capture start realm2 user 123
```

The Oracle Communications Enterprise Session Border Controller monitors realm1 AND realm2 with user 123.

To stop dynamic filter commands, you can initiate the capture stop <main filter> command. For example:

```
ORACLE# capture stop realm1 user 123
```

To stop configured filters, you must manually remove them from the ACLI configuration.

Clearing all Dynamic Filters

You can clear all dynamic filters using the following command:

- **reset monitoring dynamic-commands**—clears all dynamic filters previously initiated

The ESBC maintains a record of all dynamically initiated active filters. When you initiate this reset command, the ESBC searches through all of the filters and resets all the dynamic filters for each main filter (realm, session-agent, session-group, interesting event).

Example

The following command is an example of using the reset command to clear all dynamic capture filters.

```
ACMEPACKET# reset monitoring dynamic-commands
```

The following message displays: Reset all dynamically created monitoring capture commands....

Clearing Event Monitoring Records

You can clear all records stored in the event monitoring in-memory database using the following command:

- **reset monitoring records**—clears all event monitoring records from the in-memory database.

Use the following procedure to clear all event monitoring records.

To clear event monitoring records:

1. At the prompt, type **reset monitoring records**, and press Enter.

   ```
   ACMEPACKET# reset monitoring records
   ```

   The following prompt displays:

   ```
   All in-memory event monitoring records will be deleted [y/n]?:
   ```

2. Type **y** and press Enter.

   ```
   All in-memory event monitoring records will be deleted [y/n]?: y
   ```

   The following message displays: Deleting the in-memory event records.

   If you enter **n** for Step 2, the following message displays: Cancelling the reset.

   No event monitoring records are deleted.

Format of Exported Text Files

This section provides a sample and format of each type of exported file from the Web-based GUI. Sample information in these files are provided as a reference for your convenience.

Exported file examples include:

- Session Summary exported file (text format)
- Session Details exported file (text format)
- Ladder Diagram exported file (HTML format)

Note: Oracle recommends you open an exported text file using an application that provides advanced text formatting to make it easier to read.

Exporting Files

The Web-based GUI allows you to export Monitor and Trace information to a text file from the Sessions, Registrations, Subscriptions and Notable Events Reports, as well as from a specific ladder diagram, or from a page containing the results of a search. The data exports to a file that you can open and view as required.

You can export any of the following to a file:

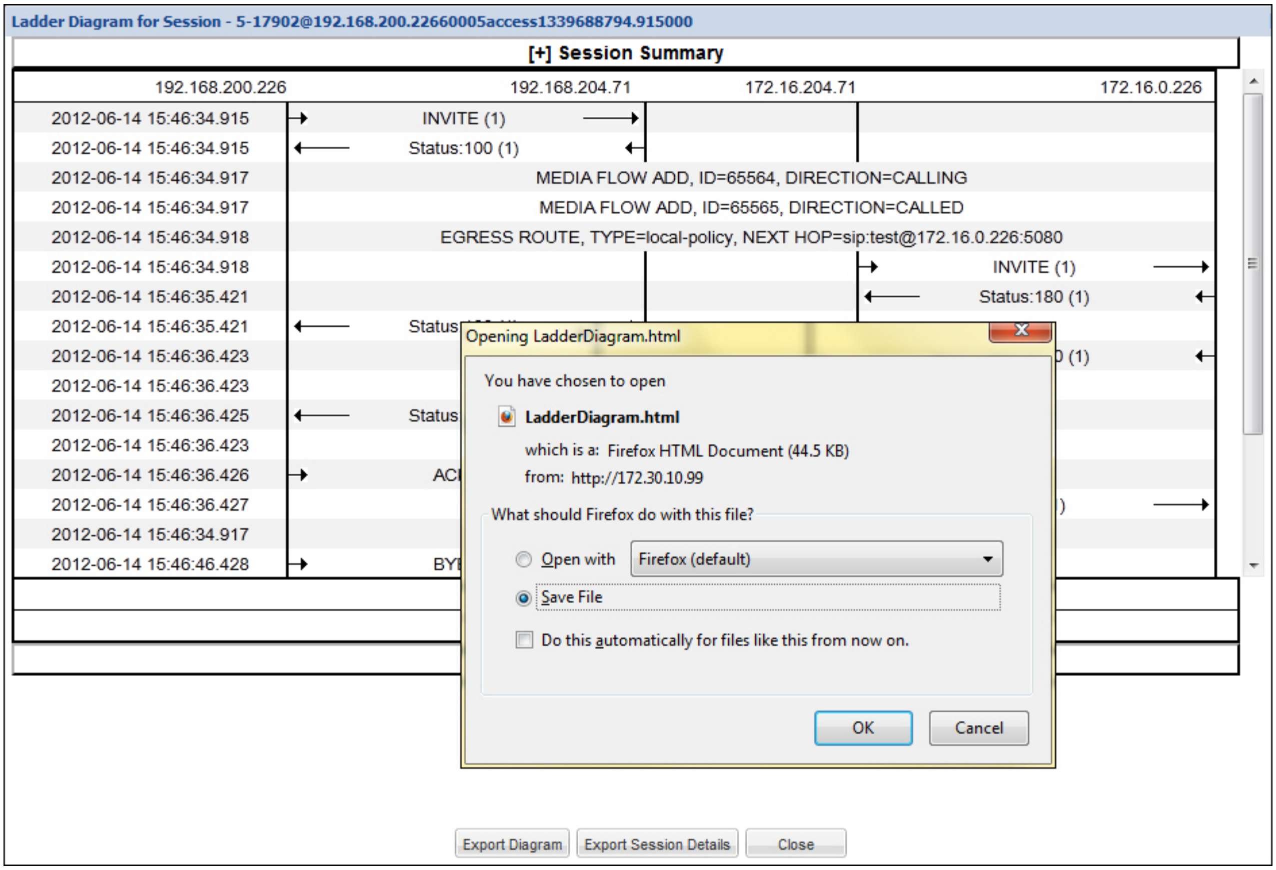
From the Sessions, Registrations, Subscriptions, and Notable Events Reports:

- **`Export session details`** - Exports the SIP messages and media events associated with the selected session, to a text file.
- **`Export summary`** - Exports all logged session summary records, to a text file. (Exports ALL call session summary records or the records that matched a search criteria).

From the Ladder Diagram:

- **`Export diagram`** - Exports all of the information in the Ladder Diagram to an HTML file (Session Summary, SIP Message Details, and QoS statistics).
- **`Export session details`** - Exports detailed information about the SIP messages and media events in the Ladder Diagram associated with the selected session, to a text file.

The following example shows the export of a Ladder Diagram to a file called LadderDiagram.html.



Ladder Diagram for Session - 5-17902@192.168.200.22660005access1339688794.915000

[+] Session Summary

```
                          192.168.200.226   192.168.204.71   172.16.204.71   172.16.0.226
2012-06-14 15:46:34.915   INVITE (1)
2012-06-14 15:46:34.915   Status:100 (1)
2012-06-14 15:46:34.917   MEDIA FLOW ADD, ID=65564, DIRECTION=CALLING
2012-06-14 15:46:34.917   MEDIA FLOW ADD, ID=65565, DIRECTION=CALLED
2012-06-14 15:46:34.918   EGRESS ROUTE, TYPE=local-policy, NEXT HOP=sip:test@172.16.0.226:5080
2012-06-14 15:46:34.918   INVITE (1)
2012-06-14 15:46:35.421   Status:180 (1)
2012-06-14 15:46:35.421   Status
2012-06-14 15:46:36.423   0 (1)
2012-06-14 15:46:36.423
2012-06-14 15:46:36.425   Status
2012-06-14 15:46:36.423
2012-06-14 15:46:36.426   ACI
2012-06-14 15:46:36.427   )
2012-06-14 15:46:34.917
2012-06-14 15:46:46.428   BYI
```

Opening LadderDiagram.html

You have chosen to open

LadderDiagram.html

which is a: Firefox HTML Document (44.5 KB)

from: http://172.30.10.99

What should Firefox do with this file?

○ Open with Firefox (default)

● Save File

□ Do this automatically for files like this from now on.

Session Summary Exported Text File

The following is an example of a Session Summary exported text file from the Web-based GUI.

Example

```
----------Session Summary----------

Startup Time: 2011-09-20 12:58:44.375

State: TERMINATED-200
Duration: 5
From URI: sipp &lt;sip:sipp@172.16.34.16:5060&gt;;tag=1
To URI: sut &lt;sip:service@172.16.34.225:5060&gt;;tag=13451
Ingress Src Address: 172.16.34.16
Igress Src Port: 5060
Igress Dest Address: 172.16.34.225
Igress Dest Port: 5060
Egress Source Address: 192.168.34.225
Egress Source Port: 5060
Egress Destination Address: 192.168.34.17
Egress Destination Port: 5060
Igress Realm: access
Egress Realm: backbone
Igress NetworkIf: access
Egress NetworkIf: backbone

----------Session Summary----------
Startup Time: 2011-09-20 12:58:05.340
State: TERMINATED-200
Duration: 5
From URI: sipp &lt;sip:sipp@172.16.34.16:5060&gt;;tag=1
To URI: sut &lt;sip:service@172.16.34.225:5060&gt;;tag=13450
Ingress Src Address: 172.16.34.16
Igress Src Port: 5060
Igress Dest Address: 172.16.34.225
Igress Dest Port: 5060
Egress Source Address: 192.168.34.225
Egress Source Port: 5060
Egress Destination Address: 192.168.34.17
Egress Destination Port: 5060
Igress Realm: access
Egress Realm: backbone
Igress NetworkIf: access
Egress NetworkIf: backbone
```

Session Details Exported Text File

The following is an example of the a Session Details exported text file from the Web-based GUI.

Example

```
Session Details:

----------------------------------------
Nov 3 08:50:56.852 On [2:0]172.16.34.225:5060 received from 172.16.34.16:5060

INVITE sip:service@172.16.34.225:5060 SIP/2.0
Via: SIP/2.0/UDP 172.16.34.16:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>
Call-ID: 1-668@172.16.34.16
CSeq: 1 INVITE
```

```
Contact: sip:sipp@172.16.34.16:5060
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 135

v=0
o=user1 53655765 2353687637 IN IP4 172.16.34.16
s=-
c=IN IP4 172.16.34.16
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000

----------------------------------------
Nov 3 08:50:56.855 On [2:0]172.16.34.225:5060 sent to 172.16.34.16:5060

SIP/2.0 100 Trying
Via: SIP/2.0/UDP 172.16.34.16:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>
Call-ID: 1-668@172.16.34.16
CSeq: 1 INVITE

----MBCD Evt
Nov 3 08:50:56.862 On 127.0.0.1:2945 sent to 127.0.0.1:2944
mbcdEvent=FLOW ADD
FlowCollapsed=enabled
FlowDirection=CALLING
FlowID=65541
MediaFormat=0
MediaReleased=disabled
MediaType=audio/PCMU/
OtherFlowID=0
TOSBits=0
InputSourcev4Addr=
InputSourcePort=0
InputDestv4Addr=172.16.34.225
InputDestPort=10004
OutputSourcev4Addr=192.168.34.225
OutputDestv4Addr=
OutputDestPort=0
InputRealm=access
OutputRealm=backbone
----MBCD Evt
Nov 3 08:50:56.862 On 127.0.0.1:2945 received from 127.0.0.1:2944

mbcdEvent=FLOW ADD
FlowCollapsed=enabled
FlowDirection=CALLED
FlowID=65542
MediaFormat=0
MediaReleased=disabled
MediaType=audio/PCMU/
OtherFlowID=0
TOSBits=0
InputSourcev4Addr=
InputSourcePort=0
InputDestv4Addr=192.168.34.225
InputDestPort=20004
OutputSourcev4Addr=172.16.34.225
OutputDestv4Addr=172.16.34.16
OutputDestPort=6000
InputRealm=backbone
```

```
OutputRealm=access

----------------------------------------
Nov 3 08:50:56.865 On [1:0]192.168.34.225:5060 sent to 192.168.34.17:5060

INVITE sip:service@192.168.34.17:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.34.225:5060;branch=z9hG4bK4od0io20183g8ssv32f1.1
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>
Call-ID: 1-668@172.16.34.16
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.34.225:5060;transport=udp>
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140

v=0
o=user1 53655765 2353687637 IN IP4 192.168.34.225
s=-
c=IN IP4 192.168.34.225
t=0 0
m=audio 20004 RTP/AVP 0
a=rtpmap:0 PCMU/8000

----------------------------------------
Nov 3 08:50:56.868 On [1:0]192.168.34.225:5060 received from
192.168.34.17:5060

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.34.225:5060;branch=z9hG4bK4od0io20183g8ssv32f1.1
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 1 INVITE
Contact: <sip:192.168.34.17:5060;transport=UDP>
Content-Length: 0

----------------------------------------
Nov 3 08:50:56.872 On [2:0]172.16.34.225:5060 sent to 172.16.34.16:5060

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 172.16.34.16:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 1 INVITE
Contact: <sip:service@172.16.34.225:5060;transport=udp>
Content-Length: 0

----------------------------------------
Nov 3 08:50:56.872 On [1:0]192.168.34.225:5060 received from
192.168.34.17:5060
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.34.225:5060;branch=z9hG4bK4od0io20183g8ssv32f1.1
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 1 INVITE
Contact: <sip:192.168.34.17:5060;transport=UDP>
Content-Type: application/sdp
Content-Length: 137
v=0
o=user1 53655765 2353687637 IN IP4 192.168.34.17
```

```
s=-
c=IN IP4 192.168.34.17
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000

----MBCD Evt
Nov 3 08:50:56.878 On 127.0.0.1:2945 sent to 127.0.0.1:2944

mbcdEvent=FLOW MODIFY
FlowCollapsed=enabled
FlowDirection=CALLING
FlowID=65541
MediaFormat=0
MediaReleased=disabled
MediaType=audio/PCMU/
OtherFlowID=0
TOSBits=0
InputSourcev4Addr=
InputSourcePort=0
InputDestv4Addr=172.16.34.225
InputDestPort=10004
OutputSourcev4Addr=192.168.34.225
OutputDestv4Addr=192.168.34.17
OutputDestPort=6000
InputRealm=access
OutputRealm=backbone

----------------------------------------
Nov 3 08:50:56.881 On [2:0]172.16.34.225:5060 sent to 172.16.34.16:5060

SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.16.34.16:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 1 INVITE
Contact: <sip:service@172.16.34.225:5060;transport=udp>
Content-Type: application/sdp
Content-Length: 138
v=0
o=user1 53655765 2353687637 IN IP4 172.16.34.225
s=-
c=IN IP4 172.16.34.225
t=0 0
m=audio 10004 RTP/AVP 0
a=rtpmap:0 PCMU/8000

----------------------------------------
Nov 3 08:50:56.883 On [2:0]172.16.34.225:5060 received from 172.16.34.16:5060

ACK sip:service@172.16.34.225:5060 SIP/2.0
Via: SIP/2.0/UDP 172.16.34.16:5060;branch=z9hG4bK-1-5
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 1 ACK
Contact: sip:sipp@172.16.34.16:5060
Max-Forwards: 70
Subject: Performance Test
Content-Length: 0

----------------------------------------
Nov 3 08:50:56.887 On [1:0]192.168.34.225:5060 sent to 192.168.34.17:5060
```

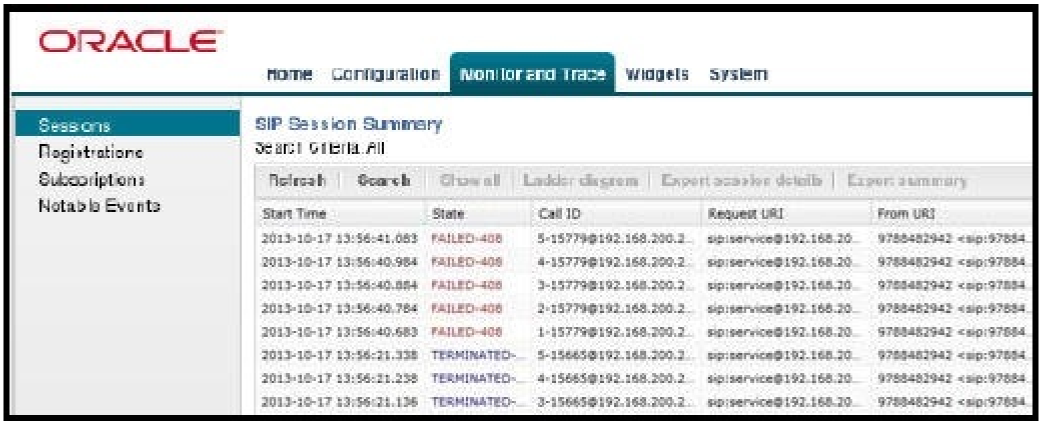
```
ACK sip:192.168.34.17:5060;transport=UDP SIP/2.0
Via: SIP/2.0/UDP 192.168.34.225:5060;branch=z9hG4bK48k3k1301ot00ssvf1v0.1
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 1 ACK
Contact: <sip:sipp@192.168.34.225:5060;transport=udp>
Max-Forwards: 69
Subject: Performance Test
Content-Length: 0

----------------------------------------
Nov 3 08:51:01.883 On [2:0]172.16.34.225:5060 received from 172.16.34.16:5060
BYE sip:service@172.16.34.225:5060 SIP/2.0
Via: SIP/2.0/UDP 172.16.34.16:5060;branch=z9hG4bK-1-7
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 2 BYE
Contact: sip:sipp@172.16.34.16:5060
Max-Forwards: 70
Subject: Performance Test
Content-Length: 0

----------------------------------------
Nov 3 08:51:01.887 On [1:0]192.168.34.225:5060 sent to 192.168.34.17:5060

BYE sip:192.168.34.17:5060;transport=UDP SIP/2.0
Via: SIP/2.0/UDP 192.168.34.225:5060;branch=z9hG4bK5oq46d301gv0dus227f1.1
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 2 BYE
Contact: <sip:sipp@192.168.34.225:5060;transport=udp>
Max-Forwards: 69
Subject: Performance Test
Content-Length: 0

----------------------------------------
Nov 3 08:51:01.889 On [1:0]192.168.34.225:5060 received from
192.168.34.17:5060

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.34.225:5060;branch=z9hG4bK5oq46d301gv0dus227f1.1
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 2 BYE
Contact: <sip:192.168.34.17:5060;transport=UDP>
Content-Length: 0

----------------------------------------
Nov 3 08:51:01.892 On [2:0]172.16.34.225:5060 sent to 172.16.34.16:5060
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.16.34.16:5060;branch=z9hG4bK-1-7
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 2 BYE
Contact: <sip:service@172.16.34.225:5060;transport=udp>
Content-Length: 0

----MBCD Evt
```

```
Nov 3 08:51:01.891 On 127.0.0.1:2945 sent to 127.0.0.1:2944

mbcdEvent=FLOW DELETE
FlowCollapsed=enabled
FlowDirection=CALLING
FlowID=65541
MediaFormat=0
MediaReleased=disabled
MediaType=audio/PCMU/
OtherFlowID=0
TOSBits=0
InputSourcev4Addr=
InputSourcePort=0
InputDestv4Addr=172.16.34.225
InputDestPort=10004
OutputSourcev4Addr=192.168.34.225
OutputDestv4Addr=192.168.34.17
OutputDestPort=6000
InputRealm=access
OutputRealm=backbone

----MBCD Evt
mbcdEvent=FLOW DELETE
FlowCollapsed=enabled
FlowDirection=CALLED
FlowID=65542
MediaFormat=0
MediaReleased=disabled
MediaType=audio/PCMU/
OtherFlowID=65541
TOSBits=0
InputSourcev4Addr=
InputSourcePort=0
InputDestv4Addr=192.168.34.225
InputDestPort=20004
OutputSourcev4Addr=172.16.34.225
OutputDestv4Addr=172.16.34.16
OutputDestPort=6000
InputRealm=backbone
OutputRealm=access

----------Session Summary----------
Startup Time: 2012-01-25 10:28:30.394

State: TERMINATED-200
Duration: 5
From URI: sipp &lt;sip:sipp@172.16.34.16:5060&gt;;tag=1
To URI: sut &lt;sip:service@172.16.34.225:5060&gt;;tag=2578
Ingress Src Address: 172.16.34.16
Igress Src Port: 5060
Igress Dest Address: 172.16.34.225
Igress Dest Port: 5060
Egress Source Address: 192.168.34.225
Egress Source Port: 5060
Egress Destination Address: 192.168.34.17
Egress Destination Port: 5060
Igress Realm: access
Egress Realm: backbone
Igress NetworkIf: access
Egress NetworkIf: backbone
```

Ladder Diagram Exported HTML File

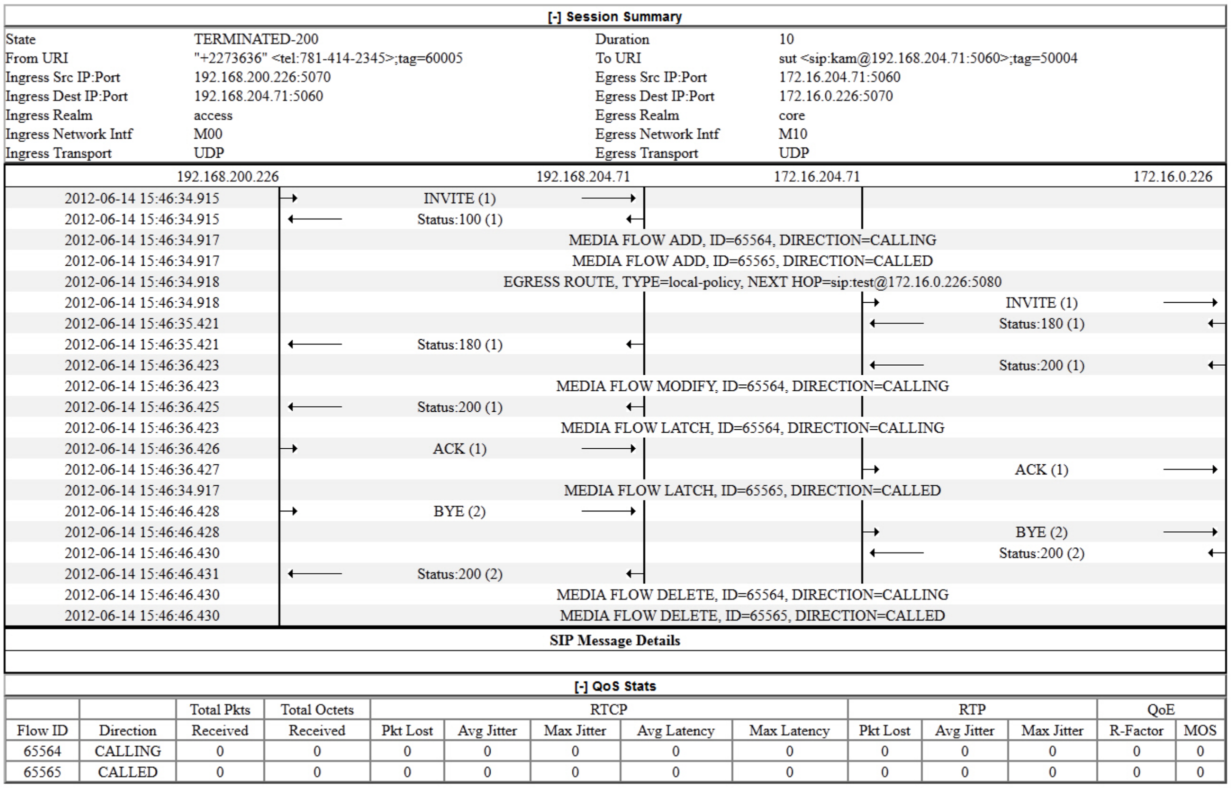
The following is an example of a Ladder Diagram for a session, exported to an HTML file from the Web-based GUI.

Example



SIP Session Summary
Search Criteria All

Start Time	State	Call ID	Request URI	From URI
2013-10-17 13:56:41.083	FAILED-408	5-15779@192.168.200.2...	sip:service@192.168.20...	9788482942 <sip:97884...
2013-10-17 13:56:40.984	FAILED-408	4-15779@192.168.200.2...	sip:service@192.168.20...	9788482942 <sip:97884...
2013-10-17 13:56:40.884	FAILED-408	3-15779@192.168.200.2...	sip:service@192.168.20...	9788482942 <sip:97884...
2013-10-17 13:56:40.784	FAILED-408	2-15779@192.168.200.2...	sip:service@192.168.20...	9788482942 <sip:97884...
2013-10-17 13:56:40.683	FAILED-408	1-15779@192.168.200.2...	sip:service@192.168.20...	9788482942 <sip:97884...
2013-10-17 13:56:21.338	TERMINATED-...	5-15665@192.168.200.2...	sip:service@192.168.20...	9788482942 <sip:97884...
2013-10-17 13:56:21.238	TERMINATED-...	4-15665@192.168.200.2...	sip:service@192.168.20...	9788482942 <sip:97884...
2013-10-17 13:56:21.136	TERMINATED-...	3-15665@192.168.200.2...	sip:service@192.168.20...	9788482942 <sip:97884...



[-] Session Summary

State	TERMINATED-200	Duration	10
From URI	"+2273636" <tel:781-414-2345>;tag=60005	To URI	sut <sip:kam@192.168.204.71:5060>;tag=50004
Ingress Src IP:Port	192.168.200.226:5070	Egress Src IP:Port	172.16.204.71:5060
Ingress Dest IP:Port	192.168.204.71:5060	Egress Dest IP:Port	172.16.0.226:5070
Ingress Realm	access	Egress Realm	core
Ingress Network Intf	M00	Egress Network Intf	M10
Ingress Transport	UDP	Egress Transport	UDP

	192.168.200.226 → 192.168.204.71	172.16.204.71 → 172.16.0.226
2012-06-14 15:46:34.915	→ INVITE (1) →	
2012-06-14 15:46:34.915	← Status:100 (1) ←	
2012-06-14 15:46:34.917	MEDIA FLOW ADD, ID=65564, DIRECTION=CALLING	
2012-06-14 15:46:34.917	MEDIA FLOW ADD, ID=65565, DIRECTION=CALLED	
2012-06-14 15:46:34.918	EGRESS ROUTE, TYPE=local-policy, NEXT HOP=sip:test@172.16.0.226:5080	
2012-06-14 15:46:34.918		→ INVITE (1) →
2012-06-14 15:46:35.421		← Status:180 (1) ←
2012-06-14 15:46:35.421	← Status:180 (1) ←	
2012-06-14 15:46:36.423		← Status:200 (1) ←
2012-06-14 15:46:36.423	MEDIA FLOW MODIFY, ID=65564, DIRECTION=CALLING	
2012-06-14 15:46:36.425	← Status:200 (1) ←	
2012-06-14 15:46:36.423	MEDIA FLOW LATCH, ID=65564, DIRECTION=CALLING	
2012-06-14 15:46:36.426	→ ACK (1) →	
2012-06-14 15:46:36.427		→ ACK (1) →
2012-06-14 15:46:34.917	MEDIA FLOW LATCH, ID=65565, DIRECTION=CALLED	
2012-06-14 15:46:46.428	→ BYE (2) →	
2012-06-14 15:46:46.428		→ BYE (2) →
2012-06-14 15:46:46.430		← Status:200 (2) ←
2012-06-14 15:46:46.431	← Status:200 (2) ←	
2012-06-14 15:46:46.430	MEDIA FLOW DELETE, ID=65564, DIRECTION=CALLING	
2012-06-14 15:46:46.430	MEDIA FLOW DELETE, ID=65565, DIRECTION=CALLED	

SIP Message Details

[-] QoS Stats

		Total Pkts	Total Octets	RTCP					RTP			QoE	
Flow ID	Direction	Received	Received	Pkt Lost	Avg Jitter	Max Jitter	Avg Latency	Max Latency	Pkt Lost	Avg Jitter	Max Jitter	R-Factor	MOS
65564	CALLING	0	0	0	0	0	0	0	0	0	0	0	0
65565	CALLED	0	0	0	0	0	0	0	0	0	0	0	0