Oracle® Communications Session Border Controller

Call Monitoring Guide

Release S-Cz9.3.0
F92217-06
June 2026

ORACLE

Oracle Communications Session Border Controller Call Monitoring Guide, Release S-Cz9.3.0

F92217-06

Copyright © 2024, 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

ADD THE FIRST PARAGRAPH HERE.

ADD THE NEXT PARAGRAPH HERE.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Contents

About this Guide

My Oracle Support ii

Revision History

1 Monitoring Warning

2 Selective Call Recording SIPREC

SIPREC for Active Recording 1
Configuring SIPREC 2
- Session Recording Server (SRS) 2
- Enforcing Parity Check on Media Port Numbers with the SRS 3
- Session Recording Group 3
  - Load Balancing 4
  - Session Recording Group within Logical Remote Entities 4
- Selective Recording 5
- SIPREC Support for SRTP 5
- High Availability (HA) Support 5
- SIPREC Configuration Procedure 6
  - Session-recording-server Attribute 6
  - Session-recording-group Attribute (for HA only) 8
  - Realm-config Attribute 9
  - Session-agent Attribute 10
  - Sip-interface Attribute 11
- P-Asserted Identity and Diversion Headers in SIPREC Metadata 13
- Metadata Contents 14
- Show Commands for Recording Sessions 14
  - Show rec 15
  - Show rec redundancy 15
- Inserting SIP Headers into SIPREC Metadata 16
  - Sample Metadata 17

Copyright © 2024, 2026, Oracle and/or its affiliates.

Configure SIP Headers for SIPREC Metadata 18
SIPREC Recording Session Refresh 19
Timer_B 19
OPTIONS Request/Response 20
Recording Session Refresh Configuration 20
Codec Negotiation 21
SIPREC Call Flows 21
Selective Recording 22
Normal Call (recording required) 22
Sample SDP and Metadata 23
Normal Call (recording not required) 25
REFER Pass-Through Call (REFER handled by User Agent) 26
REFER Call (REFER handled by Oracle Communications Session Border Controller) 28
SRS Indicates Busy in Call (recording not required) 29
Call Transfer Scenario (recording required) 31

3 Operations Monitor Deployments

IPFIX 1
Operations Monitor Configuration 2
Configure the Operations Monitor 2
TSCF Rekey Profile Configuration 4
Configure TLS Profile 5
Anonymize Personal Data in Messaging Sent to the Operations Monitor 7
Enable Anonymization of Information Sent to OM 7
Operations Monitor Statistics 8

4 Packet Trace

Packet Trace Remote 2
Packet Trace Local 3
Packet Trace Scenarios 4
Packet Trace for One Endpoint 4
Packet Trace for Both Call Legs 5
Packet Trace for a Signaling Address 6
Running Packet Trace 7
Configuring a Trace Server 8
Starting a Remote Packet Trace 8
Stopping a Remote Packet Trace 9
Starting a Local Packet Trace on Non-DPDK Platforms 9
Stopping a Local Packet Trace on Non-DPDK Platforms 10

Copyright © 2024, 2026, Oracle and/or its affiliates.

Starting a Local Packet Trace on DPDK Systems 10
Stopping a Local Packet Trace on DPDK Systems 11

5 Persistent Protocol Tracing

About Persistent Protocol Tracing 1
About the Logs 1
Process Logs 2
Communication Logs 2
Protocol Trace Logs 2
Persistent Protocol Tracing Configuration 2

Copyright © 2024, 2026, Oracle and/or its affiliates.

About this Guide

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

Documentation Set

The following table describes the documentation set for this release.

Document Name	Document Description
Acme Packet 3900 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 3900.
Acme Packet 4600 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 4600.
Acme Packet 4900 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 3950 and Acme Packet 4900.
Acme Packet 6100 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6100.
Acme Packet 6350 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6350.
Release Notes	Contains information about the current documentation set release, including new features and management changes.
Known Issues & Caveats	Contains known issues and caveats
Configuration Guide	Contains information about the administration and software configuration of the Service Provider Session Border Controller (SBC).
ACLI Reference Guide	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Maintenance and Troubleshooting Guide	Contains information about SBC logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.
MIB Guide	Contains information about Management Information Base (MIBs), Oracle Communication'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.
Accounting Guide	Contains information about the SBC's accounting support, including details about RADIUS and Diameter accounting.
HDR Guide	Contains information about the SBC's Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.

Copyright © 2024, 2026, Oracle and/or its affiliates.

Document Name	Document Description
Admin Security Guide	Contains information about the SBC's support for its Administrative Security license.
Security Guide	Contains information about security considerations and best practices from a network and application security perspective for the SBC family of products.
Platform Preparation and Installation Guide	Contains information about upgrading system images and any pre-boot system provisioning.
Call Traffic Monitoring Guide	Contains information about traffic monitoring and packet traces as collected on the system. This guide also includes WebGUI configuration used for the SIP Monitor and Trace application.
HMR Guide	Contains information about configuring and using Header Manipulation Rules to manage service traffic.
REST API	Contains information about the supported REST APIs and how to use the REST API interface.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support (CAS) can assist you with My Oracle Support registration.

Call the CAS main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select 2 for New Service Request.
2. Select 3 for Hardware, Networking, and Solaris Operating System Support.
3. Select one of the following options:
   - For technical issues such as creating a new Service Request (SR), select 1.
   - For non-technical issues such as registration or assistance with My Oracle Support, select 2.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Emergency Response

In the event of a critical service situation, emergency response is offered by the Customer Access Support (CAS) main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. The emergency response provides immediate coverage, automatic escalation, and other features to ensure that the critical situation is resolved as rapidly as possible.

Copyright © 2024, 2026, Oracle and/or its affiliates.

A critical situation is defined as a problem with the installed equipment that severely affects service, traffic, or maintenance capabilities, and requires immediate corrective action. Critical situations affect service and/or system operation resulting in one or several of these situations:

- A total system failure that results in loss of all transaction processing capability
- Significant reduction in system capacity or traffic handling capability
- Loss of the system's ability to perform automatic system reconfiguration
- Inability to restart a processor or the system
- Corruption of system databases that requires service affecting corrective actions
- Loss of access for maintenance or recovery operations
- Loss of the system ability to provide any required critical or major trouble notification

Any other problem severely affecting service, capacity/traffic, billing, and maintenance capabilities may be defined as critical by prior discussion and agreement with Oracle.

Locate Product Documentation on the Oracle Help Center Site

Oracle Communications customer documentation is available on the web at the Oracle Help Center (OHC) site, http://docs.oracle.com. You do not have to register to access these documents. Viewing these files requires Adobe Acrobat Reader, which can be downloaded at http://www.adobe.com.

1. Access the Oracle Help Center site at http://docs.oracle.com.
2. Click **Industries**.
3. Under the Oracle Communications sub-header, click the **Oracle Communications documentation** link.
   The Communications Documentation page appears. Most products covered by these documentation sets appear under the headings "Network Session Delivery and Control Infrastructure" or "Platforms."
4. Click on your Product and then Release Number.
   A list of the entire documentation set for the selected product and release appears.
5. To download a file to your location, right-click the **PDF** link, select **Save target as** (or similar command based on your browser), and save to a local folder.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Copyright © 2024, 2026, Oracle and/or its affiliates.

Revision History

This section provides a revision history for this document.

Date	Description
March 2024	• Initial release.
December 2024	• Updates Session-recording-group Attribute (for HA only) to correct the simultaneous-recording-servers parameter's valid and default values.
August 2025	• Corrects strategy parameters for Session-recording-group Attribute (for HA only). • Adds examples of packet-trace with VLANs. • Clarifies round-robin and SRS behavior. • Clarifies how to configure Operations Monitor to capture QoS on both sides of a call.
November 2025	• Remove session-agent as requirement for SIPREC.
January 2026	• Removes Early Media Call (recording not required) as the SBC does not support early media recording in SIPREC.
June 2026	• Adds note in SIPREC for Active Recording section about not supporting RBT on SRS.

Copyright © 2024, 2026, Oracle and/or its affiliates.

1

Monitoring Warning

Only one monitoring service can be run at a time.

The SBC replication features may interfere with each other, corrupting each ones results. Do not run more than one of the following monitoring services:

- Call Recording (SIPREC)

  > **Note**
  >
  > SIPREC can be used alongside comm-monitor. However, it should not be used along with any other trace tools.

- Operations Monitor
- Packet Trace
- call-trace
- SIP Monitoring and Trace (available only for the Enterprise Session Border Controller)

Copyright © 2024, 2026, Oracle and/or its affiliates.

2

Selective Call Recording SIPREC

The SIPREC protocol is used to interact between a Session Recording Client (SRC) (the role performed by the Oracle Communications Session Border Controller) and a Session Recording Server (SRS) (a third-party call recorder or Oracle Communications Interactive Session Recorder's Record and Store Server (RSS)). Selective Call Recording 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 switchovers, 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 needed, must be accomplished by the recording server. The recording server performs the filtering and the selection of which sessions it should record.

SIPREC supports sending transcoded and SRTP calls.

SIPREC for Active Recording

SIPREC supports active recording, where the Oracle Communications Session Border Controller (SBC) acting as the Session Recording Client (SRC), purposefully streams media to the Oracle Communications Interactive Session Recorder's RSS (or 3rd party call recorder) acting as the Session Recording Server (SRS). The SRC and SRS act as SIP User Agents (UA). 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 SBC acting as the SRC, is the source for the recorded media. The SBC 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 SBC 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 by way of SIP signaling to the SBC. If the call is not to be recorded, the SRS signals termination of the recording session.

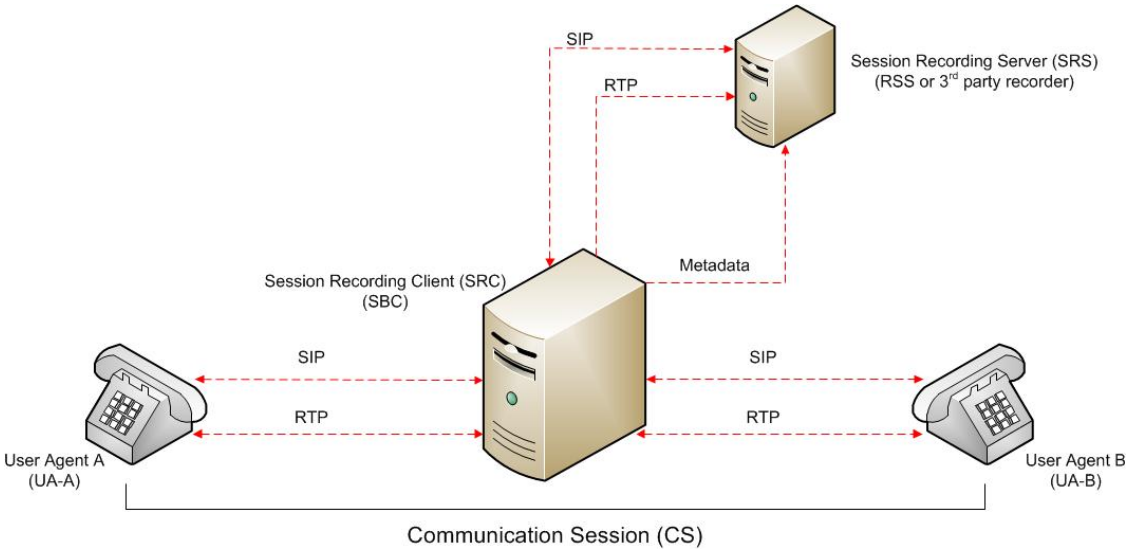
> **Note**
>
> The SBC does not support ringback tone (RBT) recording on SRS.

The SBC 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 SBC. The SRS is responsible for maintaining call history, etc. The SBC creates and logs call detail records (CDRs) in the current manner, the 3rd party SRS vendor may collate this information if desired.

Copyright © 2024, 2026, Oracle and/or its affiliates.

> **Note**
>
> SIPREC metadata is not included in CDRs.

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 SBC) and an SRS.



Configuring SIPREC

This section defines the information required to configure SIPREC on the Oracle Communications 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** element in the SBC ACLI allows you to enable or disable the SRS. This object is the session recording server that receives replicated media and records signaling. Additional parameters for SRS may be configured under **session-agent**, **realm-config**, or **sip-interface**. Attributes in the **session-agent** element take precedence over attributes in **realm-config** element, which in turn take precedence over attributes in the **sip-interface** element.

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.

If you have an HMR manipulation applied to a session-agent, your manipulations will not be applied to messages sent to the SRS server. To capture the modified messages, you can apply your HMR manipulations to a realm or SIP interface that includes both your session agent and your 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

Copyright © 2024, 2026, Oracle and/or its affiliates.

communication. Although the SBC supports large UDP packets, Oracle recommends the **sip-interface** associated with the SRS realm, be provisioned with a TCP port.

Enforcing Parity Check on Media Port Numbers with the SRS

You can configure the SBC to enforce media port number parity on flows between the SBC and the SRS, as discussed in RFC 4566. By default, the SBC does not consider port number parity when assigning or recognizing RTP and RTCP flows in SDP session descriptions. This can result in signaling issues, including one-way audio recording, when the recording server and the SBC establish flows that have a port number conflict.

By default, the SBC does not enforce port number parity for RTP and RTCP flows to the SRS, acting as the UAS callee on the east side interface. This parity defines the use of an even port for RTP and the subsequent odd port for RTCP. The **force-parity** parameter causes the SBC to behave as follows when receiving port number is the SDP m-line from the SRS:

- Disabled - The SBC accepts both odd and even ports on the m-line. The SBC sends both rtp and rtcp to this same port.
- Enabled - The SBC rejects odd ports on the m-line, and accepts even ports. Upon receiving an even port, the SBC sends rtp to the even port and rtcp to (rtp+1), which is an odd numbered port.

You enable the **force-parity** parameter on the applicable **session-recording-server**.

```
ACMEPACKET(session-recording-server)# force-parity enabled
```

Session Recording Group

The Oracle Communications 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 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 Session Border Controller load balancing in session recording groups, see Load Balancing). When multiple SRSs are in a session recording group, the Oracle Communications 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 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 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.

Copyright © 2024, 2026, Oracle and/or its affiliates.

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 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 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 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 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 Session Border Controller to attempt the next possible SRS. When simultaneous-recording-servers is greater than 1, the Oracle Communications 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 SBC 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 SBC 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 SBC replicates media to both destinations. However, if the same set of SRSs has the exact same identifier, the
SBC sends media to one and not both SRSs.

Copyright © 2024, 2026, Oracle and/or its affiliates.

Selective Recording

SIPREC defines a number of use cases for which the SBC 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 SBC 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 (SBC) re-uses the recording session ID in the metadata instead of creating a new ID for each recording.

SIPREC Support for SRTP

With the exception noted in the following table, the SBC supports SIPREC on all media flows with any combination of SRTP-RTP call legs on ingress and egress for all Acme Packet platforms. The SBC also supports SRTP on the interface between the SBC and the SIPREC server.

Caller A	Caller B	SRS	Supported or Not Supported
RTP	RTP	RTP	Supported
RTP	SRTP	RTP	Supported
SRTP	RTP	RTP	Supported
SRTP	SRTP	RTP	Supported
RTP	RTP	SRTP	Supported
RTP	SRTP	SRTP	Supported
SRTP	RTP	SRTP	Supported
SRTP	SRTP	SRTP	Supported

- The supported combinations apply to transcoded and non-transcoded calls.
- The supported combinations apply to recording and requires either the disabled mode or the enabled mode.
- The SDES profile that you use for in the media-security-policy configuration must include both the AES_CM_128_HMAC_SHA1_80 and AES_CM_128_HMAC_SHA1_32 ciphers in the crypto-list. Apply this media security policy to each realm where you want SRTP traffic.

High Availability (HA) Support

An Oracle Communications Session Border Controller using SIPREC supports HA in the network. The Oracle Communications Session Border Controller replicates all metadata states

Copyright © 2024, 2026, Oracle and/or its affiliates.

between the active and standby Oracle Communications 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 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 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 Session Border Controller has the session recording license enabled on the Oracle Communications 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 only)

And at least one of the following attributes:

- realm-config
- sip-interface

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

Session-recording-server Attribute

To configure the session-recording-server attribute:

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

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# session-recording-server
   ORACLE(session-recording-server)#
   ```

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

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

3. **(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>
   ```


Copyright © 2024, 2026, Oracle and/or its affiliates.

4. **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.

5. **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
```

6. **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
```

7. **port**—Enter the port number to contact the session recording server. Valid values are 1024 to 65535. Default is 5060.
8. **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
```


Copyright © 2024, 2026, Oracle and/or its affiliates.

9. **force-parity**—Enable to enforce port number parity for flows between the system and the session recording server. Default is disabled.

```
ACMEPACKET(session-recording-server)# force-parity enabled
```

10. Enter **done** to save the session recording configuration.

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

Session-recording-group Attribute (for HA only)

For environments that required high availability (HA) requirements, configure the **session-recording-group** attribute.

To configure the session-recording-group attribute and enable HA:

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. **name**—Enter a unique name for the session recording group that is a collection of one or more session recording servers. This name can be referenced when configuring the realm-config or sip-interface elements. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-group)# name <SRG Group Name>
```

> **Note**
>
> The name of the session recording group must be prefixed with SRG.

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

```
ACMEPACKET(session-recording-group)# description <Recording Group Name>
```

4. **session-recording-servers**—Enter the names of the session recording servers that belong to this session recording group. Valid values are alpha-numeric characters. Default is no value specified.

```
ACMEPACKET(session-recording-group)# session-recording-servers SRS1,SRS2
```

> **Note**
>
> You must enter multiple servers as values for the session-recording-servers attribute.

5. **strategy**—Enter the load balancing strategy that the session reporting client (SBC) uses when sending recordings to the session reporting server. Valid values are:

Copyright © 2024, 2026, Oracle and/or its affiliates.

- **Round-robin** (default) - In a round robin, the SBC adds only 1 SRS, even if there are multiple SRS configured in SRG. If the SRS is not available, the recording will fail, even if another SRS is available in SRG.
- **hunt** - The SBC 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 SBC 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 SBC to attempt the next possible SRS. When simultaneous-recording-servers is greater than 1, the SBC attempts to establish n recording devices in a hunting fashion.

```
ORACLE(session-recording-group)# strategy round-robin
```

6. **simultaneous-recording-servers**—Enter the number of simultaneous SIP dialogs that the session reporting client (SBC) establishes to the session reporting servers in the session reporting group per communication session. Valid values are **1** to **10**. Default is **1**.

```
ACMEPACKET(session-recording-group)# simultaneous-recording-servers 2
```

7. Enter **done** to save the session recording group configuration.
8. Enter **quit** to exit configuration mode.
9. Save and activate 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 a space-separated list containing up to four names of session-recording-servers, or session-recording-groups, or a combination of both exisiting in the realm associated with the session reporting client (SBC). For each entry, use '**+**' to add, '**-**' to remove, and omit to replace the list. For example, `realm-config# session-recording-server "srs1 srs2 srs4 SRG:srg1"`. Enter a **?** to view a list of all acceptble name formats. Press the **TAB** key after the session-recording-server to view a list of configured SRSs and SRGs. For each entry, use **+** to add, **-** to remove, and omit to replace the list.

   Do not use the format `+srs1 -srs1`. In this case, srs1 is still retained even after using -srs1in the command.

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

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


Copyright © 2024, 2026, Oracle and/or its affiliates.

> **Note**
>
> The value for this attribute is the name you specified in the session-recording-server attribute. 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 SBC 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 default value is 0 (off/ignored).

   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.

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

Session-agent Attribute

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

1.

2. **session-recording-server** — Enter a space-separated list containing up to four names of session-recording-servers, or session-recording-groups, or a combination of both to apply to the session recording client (Oracle Communications Session Border Controller). For each entry, use '**+**' to add, '-' to remove, omit to replace list. For example, `(session-agent)# session-recording-server "+srs1 +srs2 -srs4 +SRG:srg1"`. Enter a **?** to view a list of all acceptble name formats. Press the **TAB** key after the session-recording-server to view a list of configured SRSs and SRGs. Valid values are alpha-numeric characters. Default is no value specified.

   Do not use the format `+srs1 -srs1`. In this case, srs1 is still retained even after using -srs1 in the command.

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

   or

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


Copyright © 2024, 2026, Oracle and/or its affiliates.

> **Note**
>
> The value for this attribute is the name you specified the session-recording-server attribute. If specifying a session-recording-group, you must precede the group name with **SRG:**.

3. **session-recording-required** — Enter whether or not you want a call to be accepted by the Oracle Communications 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.

4. Enter **quit** to return to the top-level of the ACLI.

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

5. Save and activate the configuration.

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.

5. **session-recording-server** — Enter a space-separated list containing up to four names of session-recording-servers, or session-recording-groups, or a combination of both to apply to the SIP interface on the session recording client (Oracle Communications Session Border Controller). For each entry, use '**+**' to add, '**-**' to remove, and omit to replace the list. For example, `(sip-interface)# session-recording-server "+srs1 -srs2 +srs4`

Copyright © 2024, 2026, Oracle and/or its affiliates.

SRG:srg1". Enter a **?** to view a list of all acceptble name formats. Press the **TAB** key after the session-recording-server to view a list of configured SRSs and SRGs.Valid values are alpha-numeric characters. Default is no value specified.

Do not use the format +srs1 -srs1. In this case, srs1 is still retained even after using -srs1in the command.

```
ORACLE(sip-interface)# sessio-recording-server SRG:<"session recording
server names or session-recording group name>
```

> **Note**
>
> The value for this attribute is the name you specified in the session-recording-server attribute.

6. **session-recording-required** — Enter whether or not you want a call to be accepted by the Oracle Communications 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.

7. Enter **exit** to exit the SIP interface configuration.

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

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

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

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

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

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

    ```
    ORACLE# save-config
    ```

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

    ```
    ORACLE# activate-config
    ```


Copyright © 2024, 2026, Oracle and/or its affiliates.

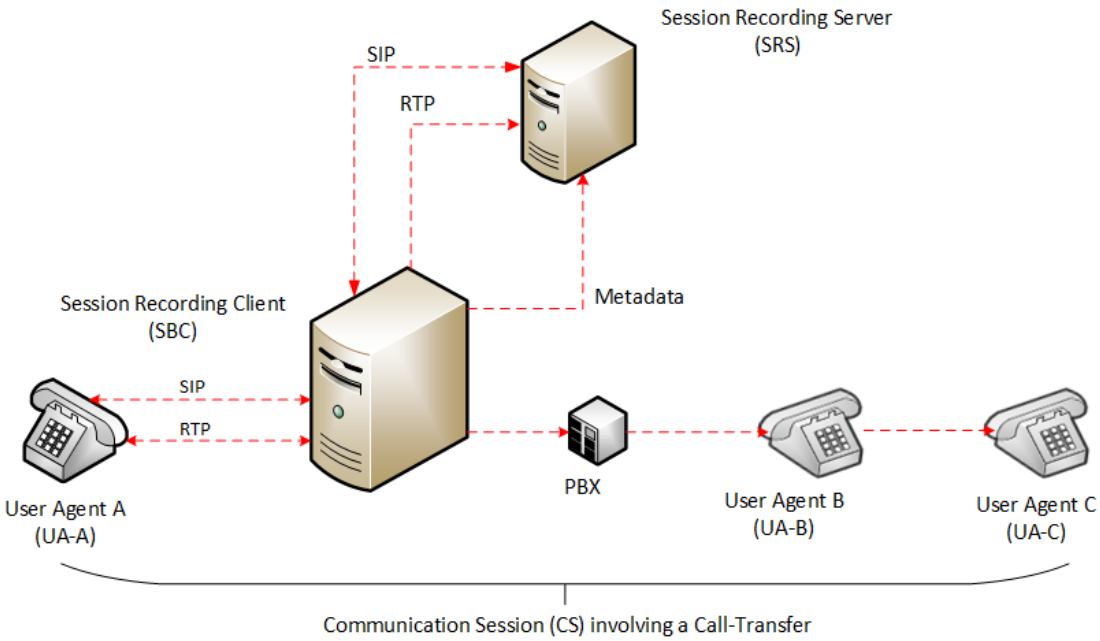
P-Asserted Identity and Diversion Headers in SIPREC Metadata

The Oracle Communications Session Border Controller supports some call transfer scenarios in which the contents of the P-Asserted-Identity, Diversion and History-info headers must be included in the SIPREC metadata for in-dialog requests (re-INVITE and UPDATE) as well as initial requests.

During a Communication Session (CS) between user-agents, Oracle Communications Session Border Controllerupdates the SRS, for in-dialog requests like Re-INVITE and UPDATE, in the Recording Session (RS). In call transfer scenarios, the participant information available in the headers is updated in the extension data under SIPREC Metadata. When the option **disable-re-invite-on-update** is configured in the session-agent, sip-interface, or real-config configuration elements, the SBC restricts the Re-INVITE to RS for the in-dialog UPDATE in CS.

You will have to configure the system with the SipHeaderExtensionMetadata.spl (SBC Processing Language) plugin for the **disable-re-invite-on-update** option to work. For more information, see Inserting SIP headers into SIPREC Metadata.

Consider a call transfer scenario between User Agent A (UA-A) and User Agent C (UA-C). The session will be recorded by Oracle Communications Session Border Controlleracting as an SRC and an SRS. The SRS does not record the session with User Agent B. In order to record the session between UA-A and UA-C it is vital to correctly identify the User Agents in the session. The system uses the headers in the metadata to reflect the identities of the participating agents.



The header information such as P-Asserted Identity, Diversion and History-info for INVITE/Re-INVITE and UPDATE cases will be updated in the extension data of participant xml element.

You can configure the option **disable-re-invite-on-update** to stop the Re-INVITE requests towards the Recording Session (RS) for any in-dialog UPDATE in Communication Session

Copyright © 2024, 2026, Oracle and/or its affiliates.

(CS). This option can be configured under a specified **sip-interface** , **realm-config** , or a **session-agent** .

```
ORACLE#(session-agent) options +disable-re-invite-on-update
```

Session-agent takes precedence over **realm-config**, and **realm-config** takes precedence over sip-interface.

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 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 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

Copyright © 2024, 2026, Oracle and/or its affiliates.

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 Session Border Controller to the present time). The following example shows the use of this command.

1. Log into the Oracle Communications 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 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 Session Border Controller to the present time) on both the primary and redundant Oracle Communications Session Border Controller. The following example shows the use of this command.

Copyright © 2024, 2026, Oracle and/or its affiliates.

1. Log into the Oracle Communications 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 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.

Inserting SIP Headers into SIPREC Metadata

The SIPREC Extension Data Enhancements SPL provides additional header information in the originating SIP messages metadata sent to the Interactive Session Recorder. With this SPL,

Copyright © 2024, 2026, Oracle and/or its affiliates.

you can introduce more options for recording policy decisions when using the SIPREC feature of the Oracle Communications Session Border Controller (SBC). The enhanced metadata also allows for the realm-id to be used as an indicator of the recording account. The SPL also provides configurable values that collect additional header information to store in the metadata.

When the SPL is configured, the SIPREC Extension Data Enhancements SPL is only triggered upon INVITE/UPDATE requests, and stores the additional header information in the metadata that is sent to the Interactive Session Recorder (ISR). Metadata is a XML MIME attachment that describes recording details to the ISR.

By default, the **Extension-Headers** SPL option collects only the Request-URI in a received INVITE. You can store additional header information by configuring the SPL with additional attributes in the **spl-options** under the global **spl-config**. The values must be in a comma separated list enclosed in double quotation marks. For example:

```
Extension-Headers="P-Asserted-Identity,Diversion"
```

This configuration of the **Extension-Headers** option adds the originating Request-URI along with all P-Asserted-Identity and Diversion-Headers into the participant section of the metadata.

You can configure the **LRE-Identifier** SPL option to add an identifier of the logical remote entity (LRE) that triggered the recording to the <apkt:realm> element of the extension metadata. When configured with a value added, the value appears in place of the identifier. When configured without a value, the identifier of the logical remote entity is used. For example, session-agent will be the hostname, realm-config will be the realm, and sip-interface will be the realm name.

> **Note**
>
> Both options are required for the SPL to function properly.

Sample Metadata

The sample below shows metadata with new extension data added by the SIPREC Extension Data Enhancements SPL:

```
<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording">
  <datamode>complete</datamode>
  <session id="BYiC7uSZQGN3VQdzWI1HWw==">
    <associate-time>2012-06-26T13:44:13</associate-time>
  </session>
  <participant id="hq18GJs3TtJdhjPsfPNV8A=="
session="BYiC7uSZQGN3VQdzWI1HWw==">
    <nameID aor="sip:sipp@192.168.10.1">
      <name>sipp</name>
    </nameID>
    <send>aD50KX+LTvxNzASg+/GQTg==</send>
    <associate-time>2012-06-26T13:44:13</associate-time>
    <extensiondata xmlns:apkt="http://acmepacket.com/siprec/extensiondata">
      <apkt:callingParty>true</apkt:callingParty>
      <apkt:request-uri>sip:service@192.168.101.13:5060  </apkt:request-uri>
      <apkt:in-realm>net192</apkt:in-realm>
      <apkt:header label="P-Asserted-Identity">
```


Copyright © 2024, 2026, Oracle and/or its affiliates.

```
        <value>sip:mike@example.com</value>
        <value>sip:bob@example.org</value>
      </apkt:header>
      <apkt:header label="Diversion">
        <value>&lt;sip:jojo@example1.com&gt;;happy=days;green=envy</value>
        <value>&lt;sip:bebe@example2.net&gt;;green=monster;go=carts</value>
        <value>&lt;tel:+8675309;night=mare&gt;;gear=head;green=monitor</value>
      </apkt:header>
    </extensiondata>
  </participant>
  <participant id="Ki6WEUi4TPRUPLtEaEhA7Q=="
session="BYiC7uSZQGN3VQdzWI1HWw==">
    <nameID aor="sip:service@192.168.101.13">
      <name>sut</name>
    </nameID>
    <send>f9NDVhyMTul+ePlM2SceQA==</send>
    <associate-time>2012-06-26T13:44:13</associate-time>
    <extensiondata xmlns:apkt="http://acmepacket.com/siprec/extensiondata">
      <apkt:callingParty>false</apkt:callingParty>
    </extensiondata>
  </participant>
  <stream id="aD50KX+LTvxNzASg+/GQTg==" session="BYiC7uSZQGN3VQdzWI1HWw==">
    <label>65804</label>
    <mode>separate</mode>
    <associate-time>2012-06-26T13:44:13</associate-time>
  </stream>
  <stream id="f9NDVhyMTul+ePlM2SceQA==" session="BYiC7uSZQGN3VQdzWI1HWw==">
    <label>65805</label>
    <mode>separate</mode>
    <associate-time>2012-06-26T13:44:13</associate-time>
  </stream>
</recording>
```

Configure SIP Headers for SIPREC Metadata

To get more detailed information about a recorded session, you can add more SIP headers within the SIPREC metadata by way of the **Extension-Headers** option. The default behavior stores only the Request-URI and realm-id.

You must configure the **Extension-Headers** option at the global level under **spl-config** because the session-agent, realm-config, and sip-interface configurations do not recognize the option. The first time you configure one or more extension headers, you need only to save and activate the configuration for the system to recognize the extension headers. When you modify the existing SPL extension header list you need to save and activate the configuration, and reboot the system for the changes to take effect. Real Time Configuration (RTC) does not apply to extension header options.

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

```
ORACLE# configure terminal
ORACLE(configure)# system
ORACLE(system)# spl-config
ORACLE(spl-config)#
```


Copyright © 2024, 2026, Oracle and/or its affiliates.

2. Type **spl-options +Extension-Headers="<value>"** , where <value> is the additional header information to store, and press Enter.

   ```
   ORACLE(spl-config)# spl-options +Extension-Headers="P-Asserted-
   Identity,Diversion"
   ```

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

SIPREC Recording Session Refresh

The Oracle Communications Session Border Controller provides for periodic in-dialog OPTIONS request/response exchanges to test the availability of the recording session dialog.

As shown below, and previously explained, establishment of a SIPREC session involves the creation of three distinct dialogs: a recording session dialog between the SIPREC client and SIPREC server, a SIP dialog between the calling party (A) and the SBC (acting as a B2BUA), and another SIP dialog between the SBC (acting as a B2BUA) and the called party (B). The two SIP dialogs are viewed conceptually as a single communications session dialog.

Both the call recording and the communication session dialogs are triggered by an INVITE at the SIPREC client. Upon reception, the recording client (the SBC) buffers the original INVITE, and sends a copy if it, along with call-related meta-data to the recording server. A 200 OK response from the server establishes the recording session dialog that will carry replicated call content; signalling information is conveyed by the metadata in the recording dialog. After establishing the recording session, the client transmits the buffered INVITE to the called party to begin establishing the communications session dialog.

In the absence of a 200 OK response, the recording client is required to reject the offered INVITE with a 503 (Service Unavailable) SIP error code and an accompanying Reason header indicating that the recording session could not be established.

After establishing the recording session dialog, the client does not monitor its connection state. Its only opportunity to verify the state occurs after the termination of the communications session dialog, which triggers a client-originated BYE to the recording server. Reception of a 200 OK/ACK confirms the persistence of the recording session, while the lack of acknowledgement indicates the failure of the recording session at some unspecified point in time.

This failure to identify a prematurely terminated recording session has caused service providers and some regulatory agencies to require timely recognition of a failed call recording dialog.

Release S-CZ7.2.0, and later releases, address this requirement by providing for the exchange of periodic in-dialog OPTIONS request/response exchanges to test the availability of the recording session dialog.

Timer_B

Section 17.1.1.2 of RFC 3261, SIP: Session Initiation Protocol describes a TIMER_B, the SIP INVITE transaction timer, which specifies the maximum interval between an INVITE request and response. The RFC suggests a default value of 32 seconds. Depending on the installed release version, two ACLI commands, `trans-expire` and `initial-inv-trans-expire`, provide user control over the TIMER_B value at both the SIP global level and the SIP interface level. Use these commands to adjust TIMER_B values if required.

Copyright © 2024, 2026, Oracle and/or its affiliates.

OPTIONS Request/Response

Availability of the recording session is tested by an in-dialog SIP OPTIONS request/response exchange initiated by the call recording client and completed by the call recording server. If the OPTIONS exchange is enabled, two timers are set when the recording session dialog is established: a refresh-timer that specifies the interval between OPTIONS requests sent by the call recording client, and a response-timer that specifies the maximum interval between the OPTIONS request and the OPTIONS response. The response-timer is set to the smaller of the configured TIMER_B or refresh-timer values.

Expiration of the refresh-timer results in the transmission of an OPTIONS request to the call recording server. The server, in turn, must reply with a 200 OK OPTIONS response prior to the expiration of the response-timer. In the event of a positive response, the call recording client restarts the refresh-timer, and re-issues the OPTIONS request when the timer next expires.

In the event that the call recording server response is non-positive, that is either

1. not received prior to the expiration of the response timer, or
2. a non-2xx OPTIONS response

the SBC terminates the session recording dialog and records the termination in the event log.

Typically, a SIPREC environment contains multiple recorders configured to record a communications session dialog, consequently the communications session should be terminated only when all recording dialogs in the call recording server have terminated. At that point the communications session dialog is torn down with a 503 (Service Unavailable) error report.

Recording Session Refresh Configuration

Use the following procedure to enable a SIP OPTIONS request/response mechanism used to detect a prematurely terminated call recording session dialog.

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

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# session-recording-server
ORACLE(session-recording-server)#
```

2. Type **select** to choose and configure an existing object

```
ORACLE(session-recording-server)# select
<recordingServerName>:1: name=SRS02 desc=SRS Atlanta
2: name=name desc="SRS First Server:
3: name=srs01 desc=Miami

selection: 2
```

3. **session-recording-required**—ensure that this parameter is enabled.
   This is a prerequisite for the provision of recording session refresh services.
4. **refresh-interval**—use this parameter to both enable the SIP OPTIONS request/response mechanism, and to assign a value to the refresh-timer, which measures the maximum allowed interval (in seconds) between the OPTIONS request sent by the call-recording client and the OPTIONS response returned by the call-recording server.

Copyright © 2024, 2026, Oracle and/or its affiliates.

By default, refresh-interval is set to 0, which disables detection of a failed recording session dialog.
Assignment of any non-zero value enables detection and sets the allowable interval between OPTIONS requests and responses.
Consult local policy and regulatory requirements when specifying the refresh-interval value.

```
ACMEPACKET(session-router)# refresh-interval 12
ACMEPACKET(session-recording-server)#
```

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

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 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)
- REFER Pass-Through Call (REFER handled by User Agent)
- REFER Call (REFER handled by the Oracle Communications Session Border Controller)
- SRS Indicates Busy in Call (recording not required)
- Call Transfer scenario

Copyright © 2024, 2026, Oracle and/or its affiliates.

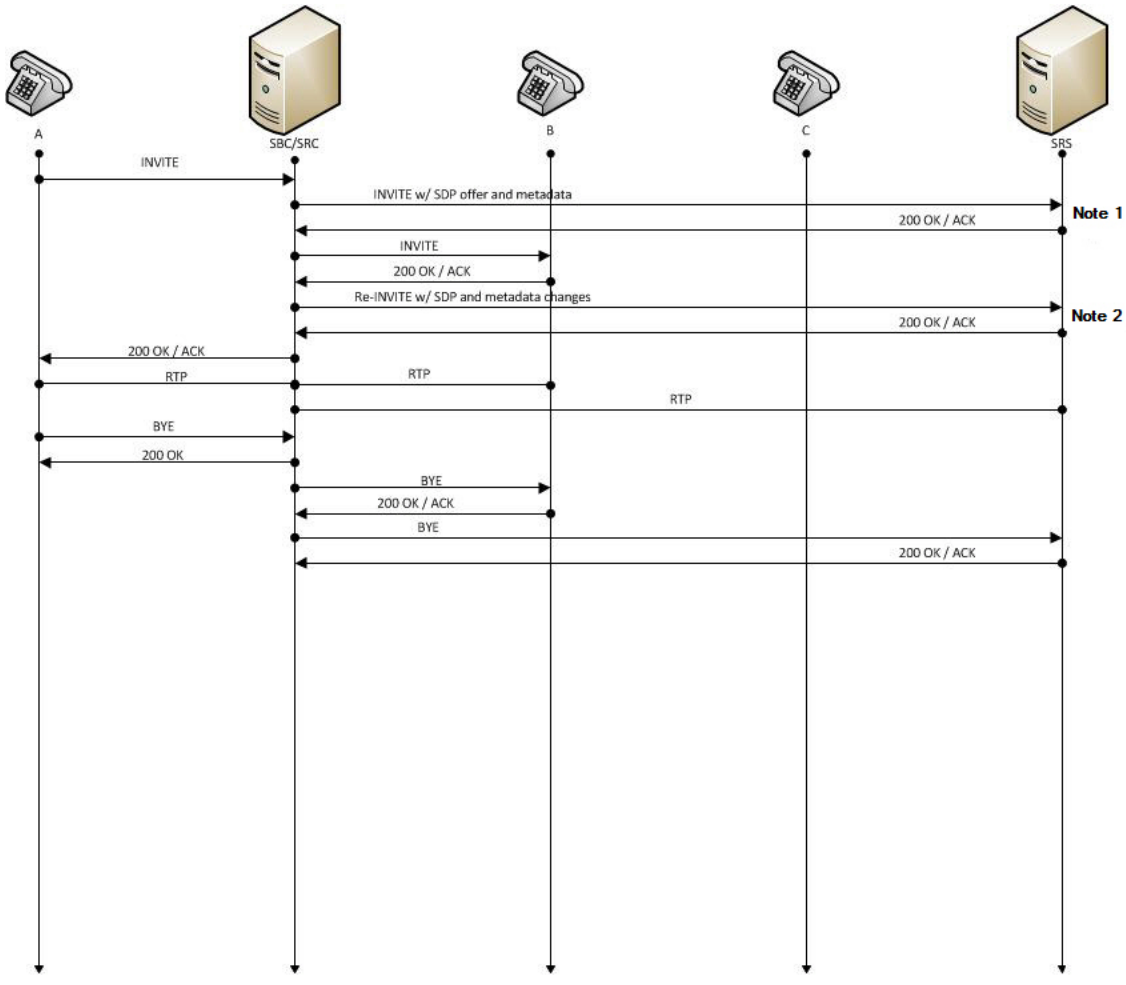
> **Note**
>
> REFER is a SIP method indicating that the recipient (identified by the Request-URI) should contact a third party using the contact information provided in the request.

The SBC does not support early media recording in SIPREC.

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

- UA-A sends INVITE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards INVITE with SDP and metadata to SRS.

Copyright © 2024, 2026, Oracle and/or its affiliates.

- SRS responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller sends INVITE to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller sends re-INVITE with SDP and metadata
  changes to SRS.
- SRS responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards OK response to UA-A.
- RTP stream initiated between UA-A and Oracle Communications Session Border Controller.
- RTP stream initiated between Oracle Communications Session Border Controller and UA-B.
- RTP stream initiated between Oracle Communications Session Border Controller and SRS.
- UA-A sends BYE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller responds with OK to UA-A.
- Oracle Communications Session Border Controller sends BYE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller responds with OK to UA-A.
- Oracle Communications Session Border Controller sends BYE to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller sends BYE to SRS.
- SRS responds with OK to Oracle Communications 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>
```


Copyright © 2024, 2026, Oracle and/or its affiliates.

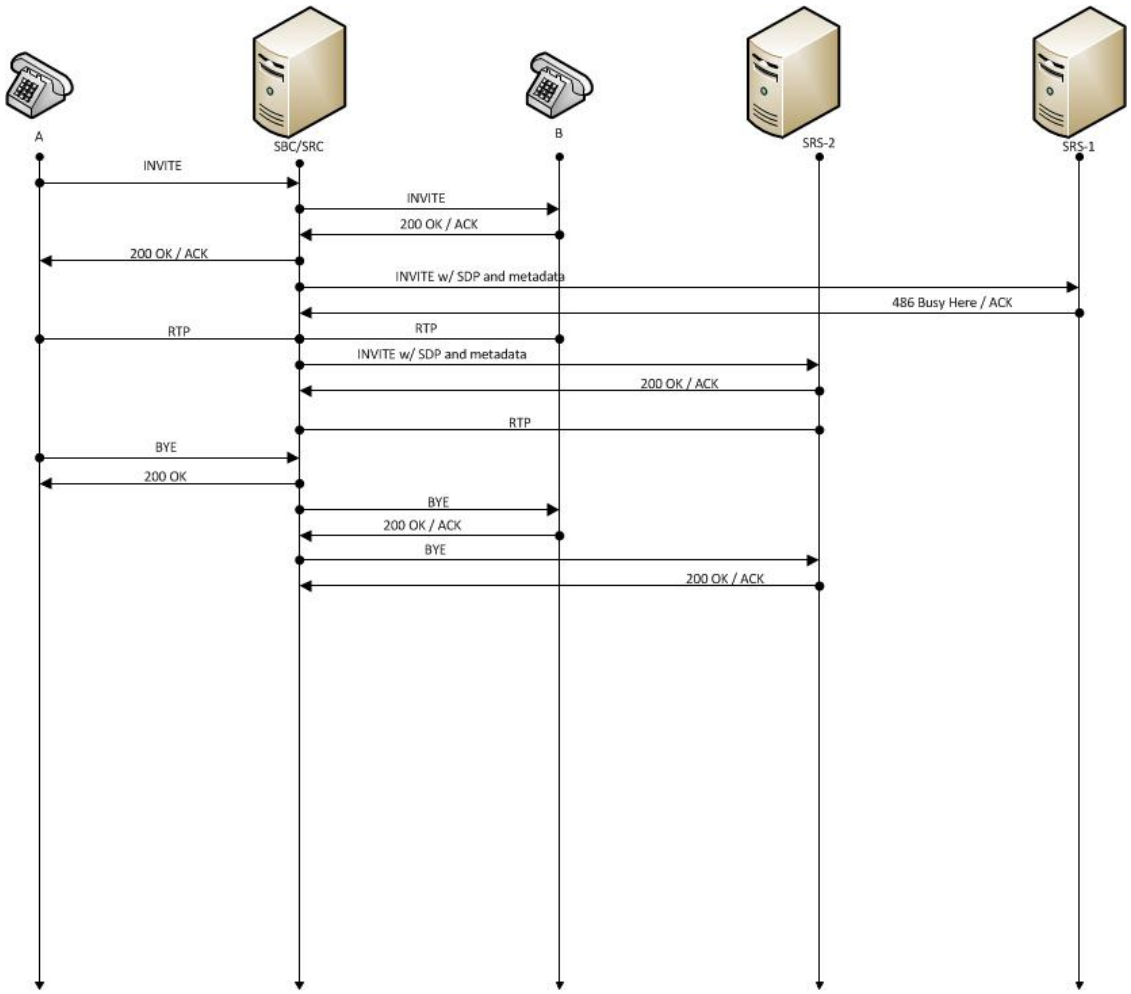
```
        <participant id="urn:uuid:10ac9063-76b7-40bb-4587-08ba290d7327"
session="urn:uuid:79b2fcd8-5c7f-455c-783f-db334e5d57d0">
                <aor>sip:sipp@168.192.24.40</aor>
                <name>sipp </name>
                <send>urn:uuid:07868c77-ef8e-4d6f-6dd5-a02ff53a1329</send>
                <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>
```


Copyright © 2024, 2026, Oracle and/or its affiliates.

```
        </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"

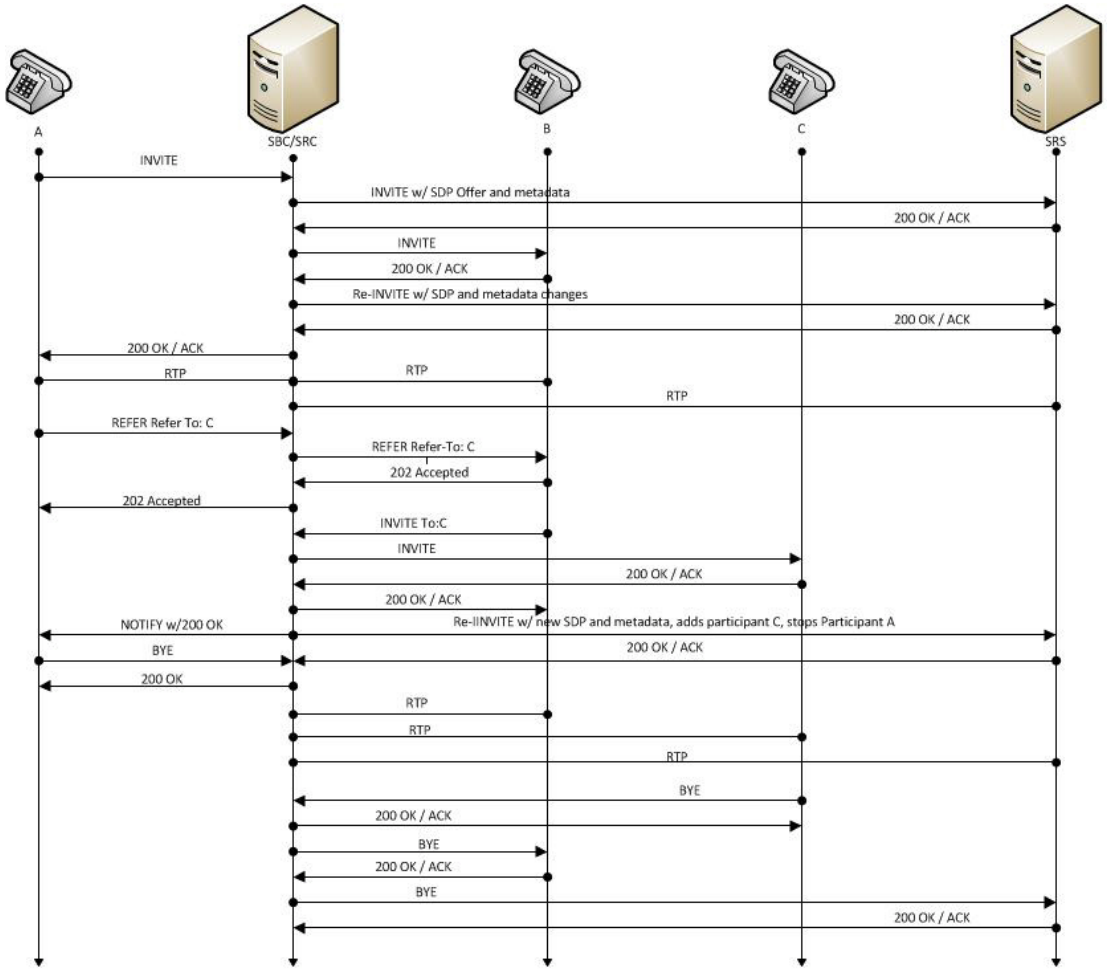
- UA-A sends INVITE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards INVITE to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards OK response to UA-A.
- Oracle Communications Session Border Controller sends INVITE with SDP and metadata to SRS.
- SRS responds with OK to Oracle Communications Session Border Controller.
- RTP stream initiated between UA-A, Oracle Communications Session Border Controller, and UA-B.

Copyright © 2024, 2026, Oracle and/or its affiliates.

- RTP stream initiated between Oracle Communications Session Border Controller and SRS.
- UA-A sends BYE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller responds with OK to UA-A.
- Oracle Communications Session Border Controller sends BYE to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller sends BYE to SRS.
- SRS responds with OK to Oracle Communications 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

- UA-A sends INVITE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards INVITE with SDP Offer and metadata to SRS.

Copyright © 2024, 2026, Oracle and/or its affiliates.

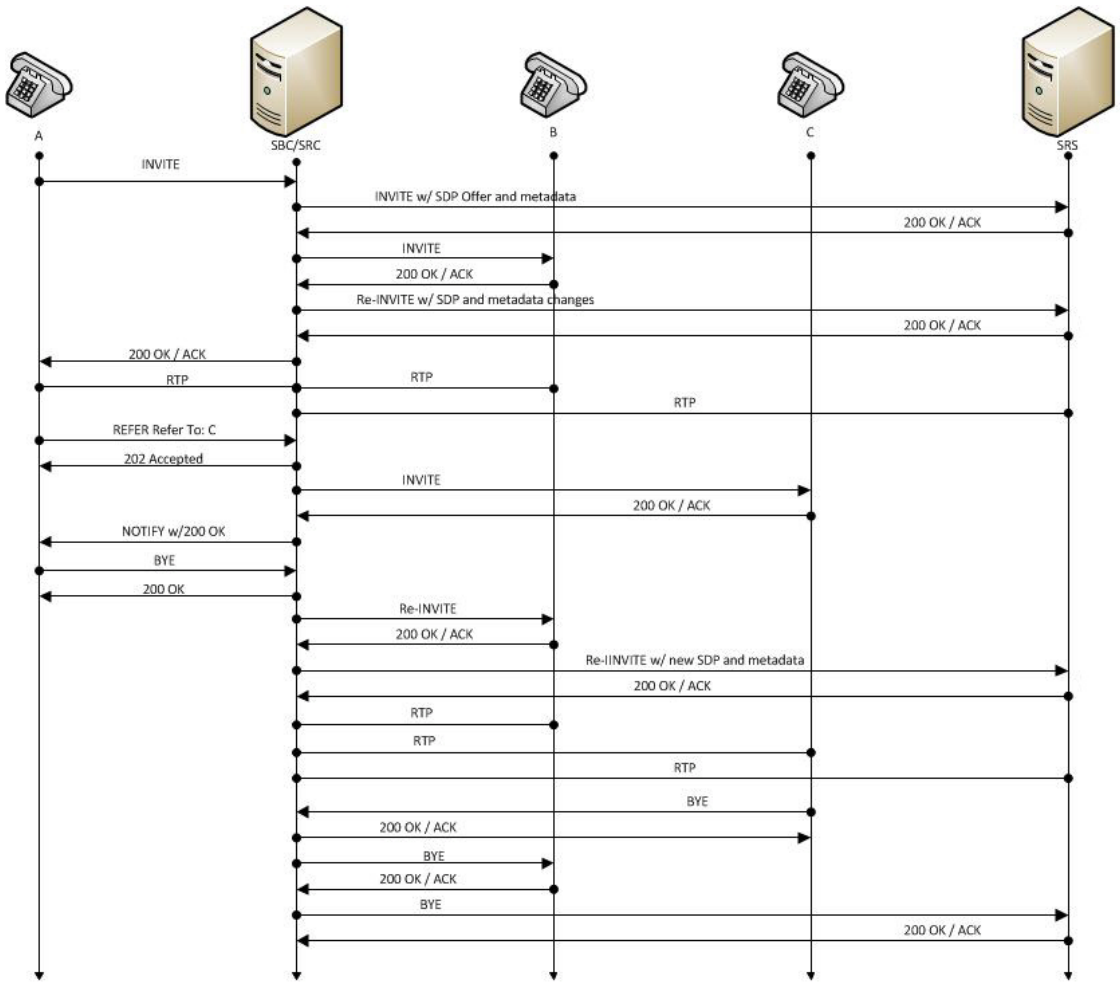
- SRS responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller sends INVITE to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller sends re-INVITE with SDP and metadata changes to SRS.
- SRS responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards OK response to UA-A.
- RTP stream initiated between UA-A and Oracle Communications Session Border Controller.
- RTP stream initiated between Oracle Communications Session Border Controller and UA-B.
- RTP stream initiated between Oracle Communications Session Border Controller and SRS.
- UA-A sends REFER-TO: C to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards REFER-TO: C to UA-B.
- UA-B responds with 202 ACCEPTED to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards 202 ACCEPTED to UA-A.
- UA-B sends INVITE TO: C to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller sends INVITE to UA-C.
- UA-C responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards OK response to UA-B.
- Oracle Communications Session Border Controller sends NOTIFY with OK reponse to UA-A.
- Oracle Communications Session Border Controller sends re-INVITE to SRS with new SDP and metadata, adds participant C, stops participant A .
- SRS responds with OK to Oracle Communications Session Border Controller.
- UA-A sends BYE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller responds with OK to UA-A.
- Oracle Communications Session Border Controller responds with OK to UA-A.
- RTP stream initiated between Oracle Communications Session Border Controller and UA-B.
- RTP stream initiated between Oracle Communications Session Border Controller and UA-C.
- RTP stream initiated between Oracle Communications Session Border Controller and SRS.
- UA-C sends BYE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller responds with OK to UA-C.
- Oracle Communications Session Border Controller sends BYE to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller sends BYE to SRS

Copyright © 2024, 2026, Oracle and/or its affiliates.

- SRS responds with OK to Oracle Communications Session Border Controller.

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

The following illustration shows a call using selective recording and the Session Border Controller (Oracle Communications 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

1. UA-A sends INVITE to Oracle Communications Session Border Controller.
2. Oracle Communications Session Border Controller forwards INVITE with SDP Offer and metadata to SRS.
3. SRS responds with OK to Oracle Communications Session Border Controller.
4. Oracle Communications Session Border Controller sends INVITE to UA-B.
5. UA-B responds with OK to Oracle Communications Session Border Controller.
6. Oracle Communications Session Border Controller sends re-INVITE with SDP and metadata changes to SRS.
7. SRS responds with OK to Oracle Communications Session Border Controller.

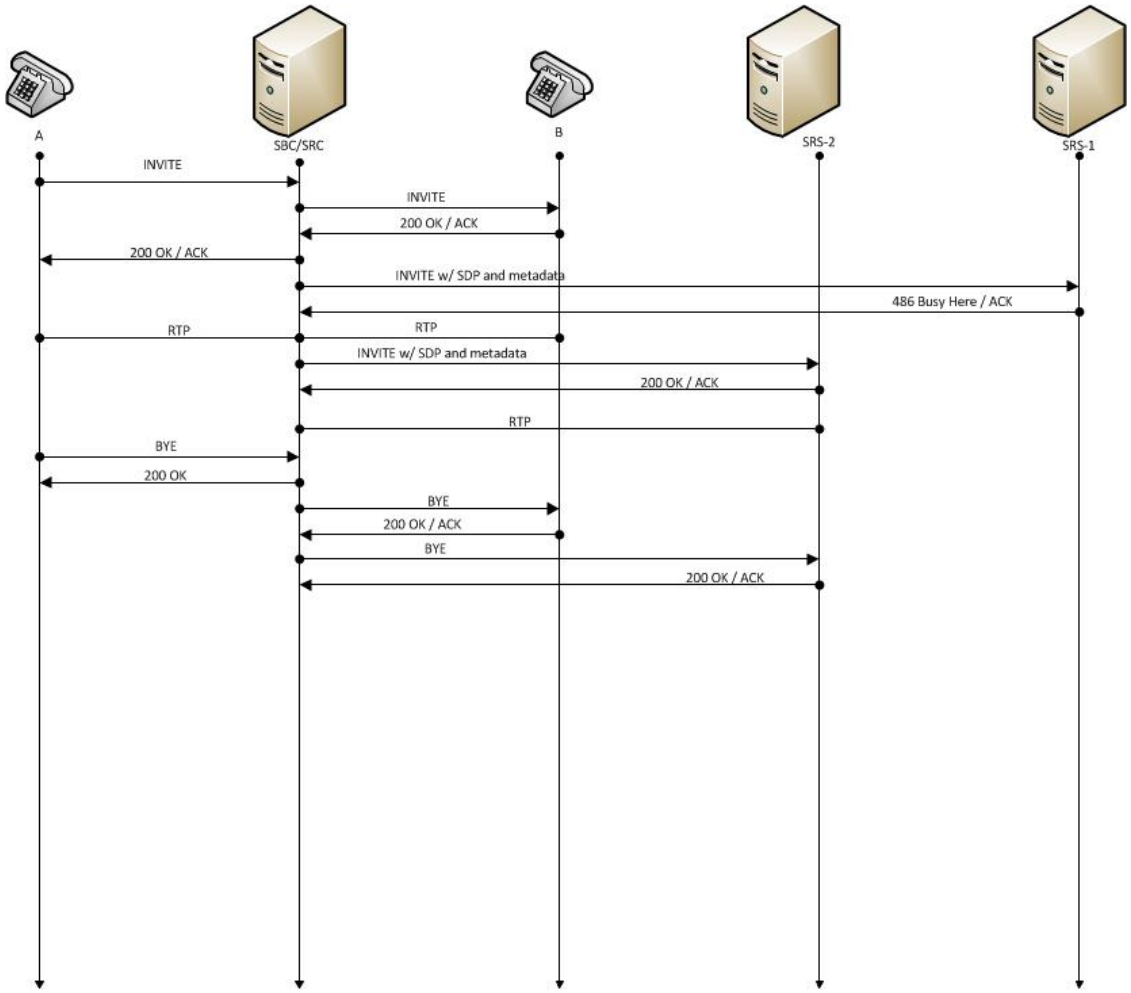
Copyright © 2024, 2026, Oracle and/or its affiliates.

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

Copyright © 2024, 2026, Oracle and/or its affiliates.



SIP INVITE, "recording not required"

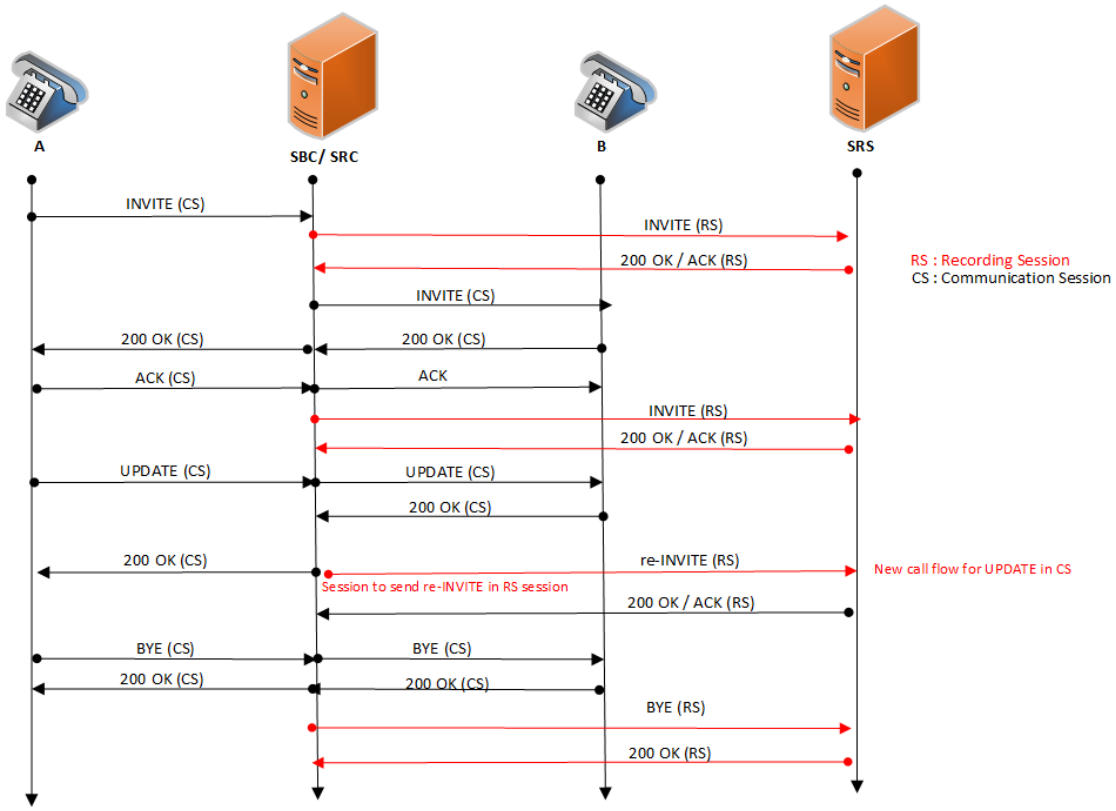
- UA-A sends INVITE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards INVITE to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards OK response to UA-A.
- Oracle Communications Session Border Controller sends INVITE to SRS1 with SDP and metadata.
- SRS1 responds to Oracle Communications Session Border Controller with 436 BUSY HERE.
- RTP stream initiated between UA-A andOracle Communications Session Border Controller.
- RTP stream initiated between Oracle Communications Session Border Controller and UA-B.
- Oracle Communications Session Border Controller sends INVITE to SRS2 with SDP and metadata.
- SRS2 responds with OK to Oracle Communications Session Border Controller.
- RTP stream initiated between Oracle Communications Session Border Controller and SRS2.

Copyright © 2024, 2026, Oracle and/or its affiliates.

- UA-A sends BYE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller responds with OK to UA-A.
- Oracle Communications Session Border Controller sends BYE to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller sends BYE to SRS2.
- SRS2 responds with OK to Oracle Communications Session Border Controller.

Call Transfer Scenario (recording required)

The following illustration shows the Re-INVITE from Oracle Communications Session Border Controller to SRS on receiving in-dialog-requests INVITE/ UPDATE/ Re-INVITE during call session. Recording is required in this call flow.



Call Flow (when **session-recording-server** is enabled)

- UA-A sends INVITE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller updates the INVITE message to the Metadata in SRS.
- SRS responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller sends an INVITE to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards OK response to UA-A.

Copyright © 2024, 2026, Oracle and/or its affiliates.

- Oracle Communications Session Border Controller sends INVITE to SRS in Recording Session.
- SRS responds with OK to Oracle Communications Session Border Controller.
- UA-A sends an UPDATE message to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards UPDATE message to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards OK to UA-A..
- Oracle Communications Session Border Controller sends a re-INVITE to SRS in Recording Session.
- SRS responds with OK to Oracle Communications Session Border Controller.
- UA-A sends BYE to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards BYE to UA-B.
- UA-B responds with OK to Oracle Communications Session Border Controller.
- Oracle Communications Session Border Controller forwards OK to UA-B.
- Oracle Communications Session Border Controller sends BYE to SRS in Recording Session.
- SRS responds with OK to Oracle Communications Session Border Controller in Recording Session.

Copyright © 2024, 2026, Oracle and/or its affiliates.

3
Operations Monitor Deployments

The Operations Monitor (OM) Mediation Engine is a platform that collects SIP, DIAMETER, DNS and ENUM protocol message traffic received from OM Probes. You can configure the Oracle Communications Session Border Controller (SBC) to run an on board Probe. Probes can also run on Common Off The Shelf hardware collecting packets, for example, from span and monitor ports on Ethernet switches. A Probe takes the protocol packets, prepends a receive time stamp and other information, encapsulates the packets, and passes them to the OM 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. This traffic can be both IPv4 and IPv6.

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 time stamp, port and vlan information, and IP:port information in ASCII format. Message Logging is performed after all decryption, meaning that SIP and TLS traffic can be monitored. Because remote message logging sends the protocol messages over UDP, there is no guarantee or confirmation of delivery.

The SBC provides support for a user-configurable capability that enables the system to function as an OM Probe. Acting as a Probe, or as an exporter, the SBC can:

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

IPFIX

The Oracle Communications Session Border Controller (SBC) uses the IPFIX suite of standards to export protocol message traffic and related data to the Operations Monitor (OM) 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 SBC and the OM Mediation Engine share ten predefined templates that

Copyright © 2024, 2026, Oracle and/or its affiliates.

facilitate protocol message exchange, and subsequent processing and analysis by the OM Engine.

The predefined templates include:

- incoming SIP and DNS over UDP
- incoming SIP over TCP
- incoming SIP over SCTP
- incoming DNS over UDP (entire IP and UDP header not included)
- outgoing SIP and 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)

Operations Monitor Configuration

The Operations Monitor configuration process includes the following steps.

1. Configure of one or more Oracle Communications Session Border Controller-Operations Monitor exporter-collector pairs.
2. Assign a TLS profile to an exporter-collector pair if transport protocol field is left blank or TLS is selected.

Configure the Operations Monitor

Use the following procedure to configure the Operations Monitor.

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 Session Border Controller (SBC), acting as a exporter of protocol message traffic and related data, and an Operations Monitor 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 SBC, in its role as an information exporter.

Copyright © 2024, 2026, Oracle and/or its affiliates.

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

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

4. Optional—If the network interface specified in Step 8 is a media interface, you can 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)#
   ```

   > **Note**
   >
   > If you want to report on RTP (voice) QoS for both directions of a call, you must enable **qos-enable** on both the ingress and egress realms.

6. Use the **interim-qos-update** parameter to enable or disable 10 second interim QoS update.

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

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

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

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

8. Use the **address** and **port** parameters to specify the IP address and port number monitored by an Operations Monitor 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)#
   ```

9. Use the **network-interface** parameter to specify the network interface that supports the TCP connection between the SBC to the Operations Monitor Mediation Engine.

Copyright © 2024, 2026, Oracle and/or its affiliates.

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.

10. Use **done** and **exit** to return to comm-monitor configuration mode.
11. Use **done**, **exit**, and **verify-config** to complete configuration.
12. 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 SBC)

   ```
   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.

Copyright © 2024, 2026, Oracle and/or its affiliates.

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.

Configure TLS Profile

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 Session Border Controller and the Operations Monitor (OM) 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 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 OM Monitor Mediation Engine as part of the TLS exchange.

Copyright © 2024, 2026, Oracle and/or its affiliates.

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 OM Mediation Engine.

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

   The Oracle Communications 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.

Copyright © 2024, 2026, Oracle and/or its affiliates.

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. Set **tls-version** to **compatibility**.
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.

Anonymize Personal Data in Messaging Sent to the Operations Monitor

When you allow people to examine SIP INVITE or SIP MESSAGE messages in the Operations Monitor (OM), you might want to hide certain sensitive information from their view for security and confidentiality reasons. For example, you might want to hide the **SUBJECT** header in the message and in the CPIM body, as well as the MIME content of the CPIM body. Oracle's solution is to provide an option to anonymize such information for display in the OM.

When you enable the **anonymize-invite** option, the system makes a copy of the inbound SIP INVITE and allows the original to continue on its way. In the copy, the system parses the body of the INVITE and replaces the **SUBJECT** header and MIME content with a hyphen (-). No other message content is affected, and the full functionality of the OM remains available. When the troubleshooter views the SIP INVITE message, OM displays the anonymized copy of the SIP INVITE.

You can also enable the **anonymize-message** option, which performs the same functions to the SIP MESSAGE, defined in RFC 3428, to support the transfer of Instant Messages. When enabled, this option hides the **SUBJECT** header as well as the CPIM subject and MIME content, replacing them with a hyphen (-) before sending them to OM.

The default setting for both options is disabled. Use the options parameter in the comm-monitor configuration to enable them.

Enable Anonymization of Information Sent to OM

When you want to hide certain sensitive information in a SIP **INVITE** message that the Operations Monitor (OM) can display, you can configure the Oracle Communications Session Border Controller (SBC) to anonymize the **SUBJECT** header in the message and in the CPIM body, as well as the MIME content of the CPIM body with the **anonymize-invite** option.

> **Note**
>
> The anonymize-invite option for CommMonitor is not RTC.

You can enable the same functionality for the SIP **MESSAGE** method using the **anonymize-message** option. You can enable both options on the same **comm-monitor**, if desired using the options' plus-sign (+) syntax.

The default setting for these anonymize options is disabled. Use the options parameter in the comm-monitor configuration to enable them.

Copyright © 2024, 2026, Oracle and/or its affiliates.

1. Access the **comm-monitor** configuration element.

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

2. Select the comm-monitor instance that you want to enable for anonymization.
3. Set the **anonymize-invite** option, referring to the syntax below, and press ENTER.

```
ORACLE(comm-monitor)#options + anonymize-invite
```

   To perform the same functionality on the SIP **MESSAGE** method, use the same syntax as above replacing the option with **anonymize-message**, and press ENTER.
4. Save and exit the configuration.

Operations Monitor Statistics

The Oracle Communications Session Border Controller (SBC) collects statistics about the operations of its communications monitor probe, which provides protocol traffic information to the Operations Monitor. You can display information about the connections to the Operations Monitor servers using the **show comm-monitor** command.

The **show comm-monitor** command displays the following types of aggregated statistics for all clients:

- Client State
- Socket Statistics
- Other Aggregate Statistics

The SBC **show comm-monitor** command also shows the Lifetime statistics. The command runs without or with arguments. For example,

- **show comm-monitor**—Shows client connection states and aggregate stats
- **show comm-monitor by-client <IP-Addr>**—Shows client stats
- **show comm-monitor errors**—Shows errors captured by the Operations Monitor
- **show comm-monitor internal**—Shows SBC traffic-specific counters

The following example shows **show comm-monitor** output without arguments:

```
ORACLE# show comm-monitor
Client                      State         PROTOCOL
================================================
192.168.42.10:4739      Out-of-Service       TCP
192.168.42.11:4739      Connecting           TCP
192.168.42.12:4739      In-Service           TCP

15:06:10-35 (recent)
CommMonitor Socket Statistics                      --- Lifetime ---
                                        Recent      Total     PerMax
=================================================================
Socket Message Dropped                    0          0          0
```


Copyright © 2024, 2026, Oracle and/or its affiliates.

```
Socket Send Error                        0          0          0
Socket Not Ready                         0          0          0
Socket Timeouts                          0          0          0
Socket Disconnects                       0          0          0
Socket Reconnects                        0          0          0
Buffer Allocation Error                  0          0          0

CommMonitor Statistics                                --- Lifetime ---
                                      Recent      Total     PerMax
=================================================================
Handshake Msg Sent                       0          0          0
Handshake Msg ACK                        0          0          0
Handshake Msg NAK                        0          0          0
Keep Alive                               0          0          0
SIP UDP Recv Msg Sent                    0          0          0
SIP UDP Send Msg Sent                    0          0          0
SIP TCP Recv Msg Sent                    0          0          0
SIP TCP Send Msg Sent                    0          0          0
SIP SCTP Recv Msg Sent                   0          0          0
SIP SCTP Send Msg Sent                   0          0          0
ENUM Recv Msg Sent                       0          0          0
ENUM Send Msg Sent                       0          0          0
```

You can set all comm-monitor statistics to zero using the **reset comm-monitor** command.

Copyright © 2024, 2026, Oracle and/or its affiliates.

4 Packet Trace

The Oracle Communications Session Border Controller (SBC) packet trace tool provides the ability to capture traffic from the SBC.

> ⚠ **Caution**
>
> packet-trace is a troubleshooting tool for use only with Oracle Support guidance. Oracle recommends using packet-trace only in lab environments and not under heavy load.

Invoke the packet trace manually from the ACLI by specifying the following.

- Capture method (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.

- 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.
- The system does not capture RTP through local packet capture.
- The system does not support running packet trace on a standby node.

Do not run packet-trace simultaneously with other SBC replication features, such as SRS, SIP Monitoring and Trace, and Call Recording. These features may interfere with each other, corrupting each ones results.

The default packet trace filter uses the specified interface to capture both ingress and egress traffic. To specify captured traffic, you can append the command with a PCAP filter enclosed in quotes. PCAP filter syntax is widely published (See Oracle Linux man pages). You can determine the version of libpcap with the `show platform components` command.

Refer to Wireshark, tcpdump and Berkley Packet Filter (BPF) syntax and example resources as guidance for your capture filters:

https://wiki.wireshark.org/CaptureFilters

https://www.tcpdump.org/manpages/pcap-filter.7.html

http://biot.com/capstats/bpf.html

Copyright © 2024, 2026, Oracle and/or its affiliates.

Note

When operating on a VNF, The SBC requires that you prepend the VLAN key to a capture filter to run packet-trace on VLAN interface. These commands take the following form.
ORACLE# packet-trace local start <network interface> < vlan [vlan_id]&& capture filter>

Examples include:

```
ORACLE# packet-trace local start M00:100 "vlan && port 5060"
```

or

```
ORACLE# packet-trace local start M00:100 "vlan 100 && port 5060"
```

Packet Trace Remote

Packet trace remote enables the Oracle Communications Session Border Controller (SBC) to mirror traffic between two endpoints, or between itself and a specific endpoint to a user-specified target. To accomplish this, the SBC replicates the packets sent and received, encapsulates them according to RFC 2003, and sends them to a user-configured target. At the target, you capture and analyze the packets. The syntax for remote packet-trace is the same across platforms.

You can use the SBC remote capture feature to analyze:

- SIP signaling traffic—Subsequent media is also captured
- H323 signaling traffic—Subsequent media is also captured
- MSRP traffic—This includes the TCP handshake to set up connections and support MSRP traffic.
- IPv4 and IPv6 traffic
- Remote packet trace is capable of capturing IPv4 traffic when configured with an IPv6 address and vice-versa.

Currently, the SBC supports:

- One configurable trace server (on which you capture and analyze the traffic)
- Fifteen concurrent endpoint traces

To use this feature, the user configures a **capture-receiver** on the SBC 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.

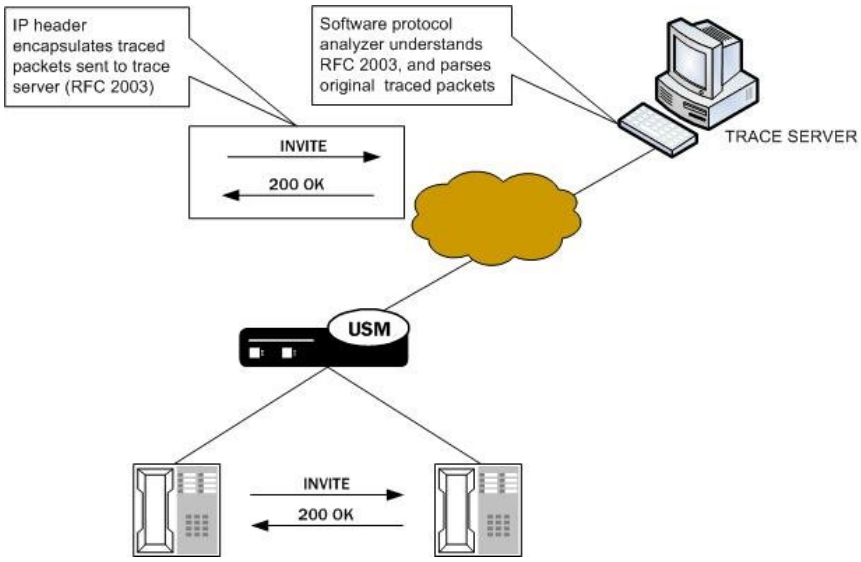
You establish a packet trace filter with the following information:

- Network interface—The name of the network interface on the SBC 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.

Copyright © 2024, 2026, Oracle and/or its affiliates.

- Local port number—Optional parameter; Layer 4 port number on which the SBC 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 SBC 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 SBC 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.



For large frames that are close to Maximum Transmission Unit (MTU) size, it is possible that when the SBC performs the steps to comply with RFC 2003 by adding the requisite header that the resulting packet might exceed Ethernet MTU size. If required, the SBC will create multiple fragments as needed before sending the packet output. If the SBC either receives or transmits IP fragments during a packet trace, the SBC performs reassembly and then performs the steps to comply with RFC 2003 by adding the requisite header, and then creates multiple fragments as needed before sending the packet output.

Packet Trace Remote works as follows:

- Packet capture mode begins only after all fragments are received and assembled.
- When the complete packet is available the system adds the Outer IP header and applies more IP fragment logic with the payload as a complete packet including the Inner header.
- The first packet contains the Inner and Outer header. Subsequent packets contain only the Outer IP header.

The SBC 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 SBC is conducting.

Packet Trace Local

Packet Trace Local enables the Oracle Communications Session Border Controller to capture traffic between two endpoints, or between itself and a specific endpoint. To accomplish this, the

Copyright © 2024, 2026, Oracle and/or its affiliates.

Oracle Communications 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. The command syntax differs based on platform. To specify captured traffic, you 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 captured and prevents you from entering any other ACLI commands from that session. On Virtual Network Function (VNF) and virtual SBC systems, you stop captures using the command line syntax with the **stop** argument. On all other platforms, you terminate 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, you can specify file name using the system supports the PCAP filter flags -w.

The system rotates the PCAP files 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 you to remove them before running new captures. If preferred, you can decline this file deletion.

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

> **Note**
>
> Although local packet trace captures and re-assembles fragmented packets, it does not recognize and show fragmentation of the capture.

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 Session Border Controller interfaces to endpoints.

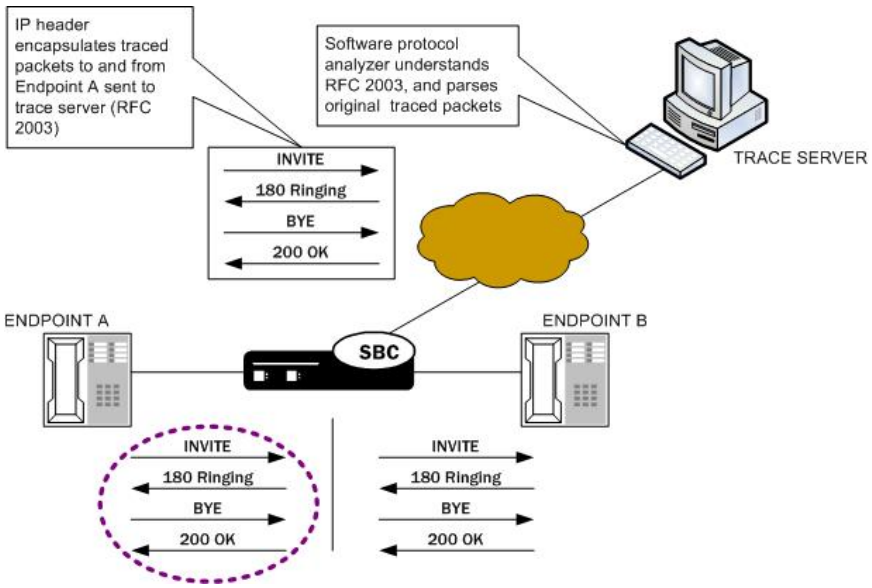
Packet Trace for One Endpoint

When you use the **packet-trace remote start** command, the Oracle Communications Session Border Controller sets up packet tracing for one endpoint. The SBC 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 <interface> <IP address>
```


Copyright © 2024, 2026, Oracle and/or its affiliates.



The syntax for **packet-trace local** commands on platforms that use the DPDK datapath take the following form:

```
packet-trace local start <interface> "<filter>"
```

The syntax for **packet-trace local** commands on all other platforms take the following form:

```
packet-trace local <interface> "<filter>"
```

For example, to start capturing traffic for a specific IP address on a DPDK datapath platform:

```
packet-trace local start s0p0 "host 10.2.2.20"
```

Or to start capturing traffic for a specific IP address when you have configured VLANs on a DPDK datapath platform:

```
packet-trace local start M10:2990 "vlan and host 10.2.2.20"
```

Or to start capturing traffic for a specific IP address when you have configured VLANs on a platform that does not support DPDK datapath:

```
packet-trace local F01:0 "vlan and host 10.2.2.20"
```

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.

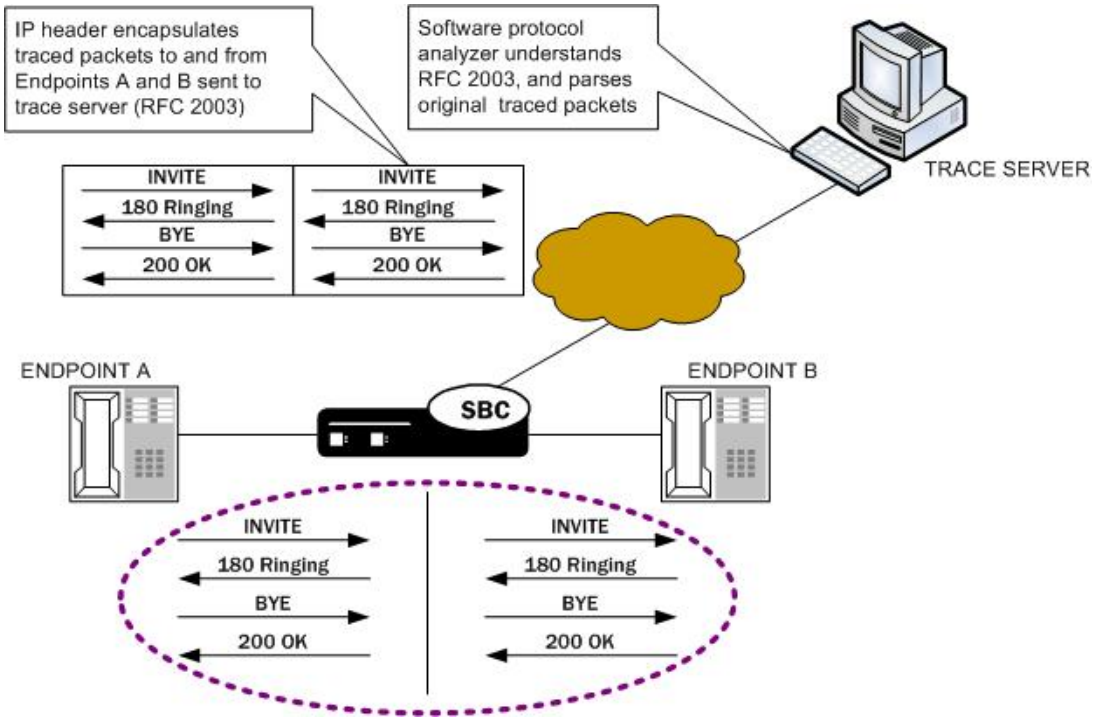
If you initiate a packet trace for both endpoints that captures both signaling and media, the signaling will be captured as usual. However, RTP will only be traced for the ingress call leg.

Copyright © 2024, 2026, Oracle and/or its affiliates.

This is because the Oracle Communications 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>
```



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>"
```

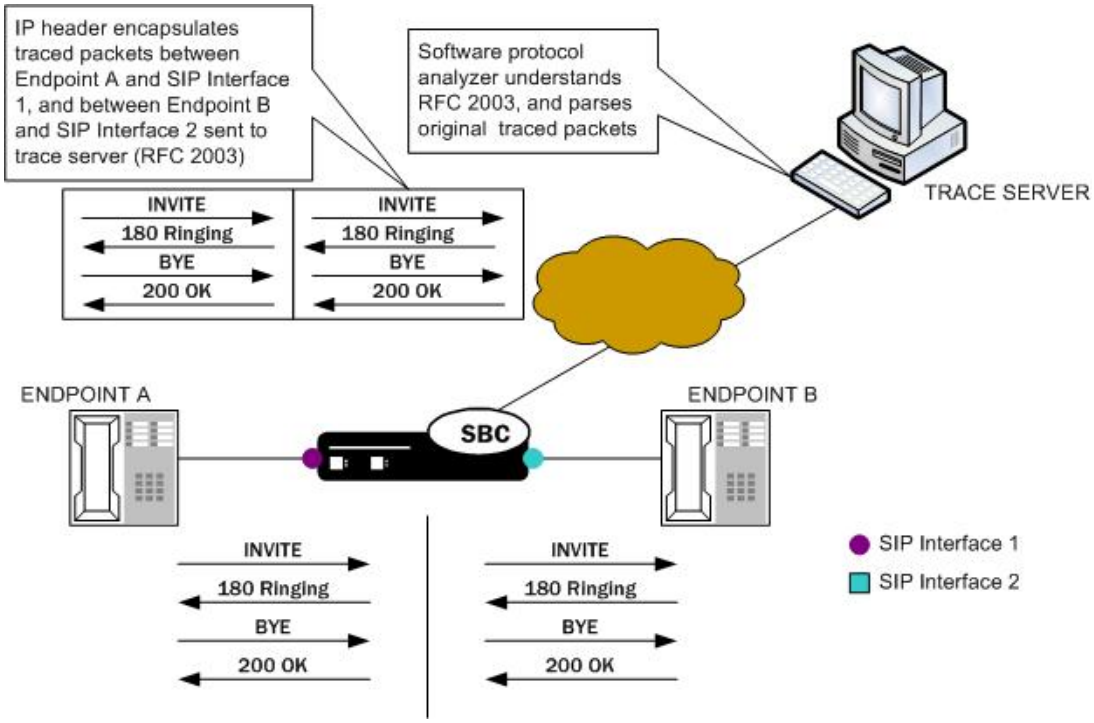
Packet Trace for a Signaling Address

You can perform a packet trace for addresses internal to the SBC; this can be the address, for example, of a SIP interface. Using signaling interface addresses puts the emphasis on the SBC 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 SBC interface1>
ORACLE# packet-trace remote start F02:0 <IP address of SBC interface2>
```


Copyright © 2024, 2026, Oracle and/or its affiliates.



The commands you carry out for **packet-trace local** on platforms that use the DPDK datapath take the following form:

```
ORACLE# packet-trace local start F01:0 <"host IP address of SBC interface1">
ORACLE# packet-trace local start F02:0 <"host IP address of SBC interface2">
```

The commands you carry out for **packet-trace local** on all other platforms take the following form:

```
ORACLE# packet-trace local F01:0 <"host IP address of SBC interface1">
ORACLE# packet-trace local F02:0 <"host IP address of SBC interface2">
```

> **Note**
>
> The system does not support egress RTP capture with Transcoding modules.

Running Packet Trace

There are three operations you can perform when you use **packet-trace remote**. For **packet-trace local**, there are only two.

- **packet-trace remote** and **packet-trace local**—Configure the SBC with the trace server information so that the SBC knows where to send replicated data.
- **packet-trace local**—Start a packet trace.
- **packet-trace local**—Stop a packet trace.

Copyright © 2024, 2026, Oracle and/or its affiliates.

> **⚠ Caution**
>
> Do not run packet-trace simultaneously with other replication features, such as SRS, SIP Monitoring and Trace, and Call Recording. These features may interfere with each other and lead to undefined behavior.

Configuring a Trace Server

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

To configure a trace server on your Oracle Communications 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 Session Border Controller network interface from which the Oracle Communications 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

The syntax for the remote packet trace is:

```
packet-trace remote <start | stop> <network interface> <remote IP> [ <local
port> <remote port> ]
```

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

Copyright © 2024, 2026, Oracle and/or its affiliates.

- 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 SBC will trace all ports
- (Optional) Local UDP/TCP port on which the SBC sends and receives traffic to be traced
- (Optional) Remote UDP/TCP port to which the SBC 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 on all platforms 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 Session Border Controller sends and receives traffic to be traced
- (Optional) Remote UDP/TCP port to which the Oracle Communications 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 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 on Non-DPDK Platforms

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

- Network interface (name:subport ID combination)

Copyright © 2024, 2026, Oracle and/or its affiliates.

- (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. Type in any desired capture filter surrounded by quotes, 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 on Non-DPDK Platforms

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

Starting a Local Packet Trace on DPDK Systems

Local packet-trace syntax differs on the platforms using the DPDK datapath. The primary differences include using the **start** argument and the ability to continue to use the ACLI while running packet-capture. Additional considerations include:

- You can run only a single capture on a given interface. However, you can run multiple captures simultaneously, as long as they are on separate interfaces.
- You must manually stop a local packet capture prior to restarting it. The command syntax does not notify you of this requirement prior to capture re-start. Running the command to either "stop all" or "stop the specific capture" allows you to successfully restart your capture.

You use the start a local 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

Copyright © 2024, 2026, Oracle and/or its affiliates.

> **Note**
>
> When operating on a VNF, The Oracle Communications Session Border Controller requires that you prepend the VLAN key to a capture filter to run packet-trace on VLAN interface. These commands take the following form.
> ORACLE# packet-trace local start <network interface> < vlan [vlan_id]&& capture filter>
>
> Examples include:
>
> ```
> ORACLE# packet-trace local start M00:100 "vlan && port 5060"
> ```
>
> or
>
> ```
> ORACLE# packet-trace local start M00:100 "vlan 100 && port 5060"
> ```

- Enter the ACLI **packet-trace local** command followed by a Space, and the word **start** followed by a space. Type in the name and subport ID for the network interface followed by a Space. Type in any desired capture filter surrounded by quotes, then press Enter.

```
ACMEPACKET# packet-trace local start s0p0 "vlan && host 192.168.12.12"
Files found in trace directory. Remove [y/n]?: y
ACMEPACKET#
```

Stopping a Local Packet Trace on DPDK Systems

You stop a specific local 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

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 s0p0 "host 192.168.12.12"
   ```

2. To stop all packet traces on the Oracle Communications 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
   ```


Copyright © 2024, 2026, Oracle and/or its affiliates.

5

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 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 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 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).

> **Note**
>
> Set the **log-filter** to **all-fork** for the system to include TCP and TLS traces in logs.

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 Session Border Controller performance.

Copyright © 2024, 2026, Oracle and/or its affiliates.

Process Logs

Events are logged to a process log flow from tasks and are specific to a single process running on the Oracle Communications 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 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

Copyright © 2024, 2026, Oracle and/or its affiliates.

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 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 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 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 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.

Copyright © 2024, 2026, Oracle and/or its affiliates.