Oracle® Communications

Unified Session Manager Essentials Guide

S-CZ8.2.5

August 2019

Oracle Communications Unified Session Manager Essentials Guide, S-CZ8.2.5

Contents

5 ENUM Based Oracle Communications Unified Session Manager

6 Local Subscriber Tables

7 Third Party Registration

8 References and Debugging

About This Guide

This Essentials Guide provides information about:

- Basic concepts that apply to the key features and abilities of your Oracle Communications Unified Session Manager (OCUSM)
- Information about how to load the OCUSM system software image you want to use and establish basic operating parameters
- System-level functionality for the OCUSM
- Configuration of key components of the OCUSM
- Direction to OCSBC documentation for configuration of cross-product components and features that apply to the OCUSM

Supported Platforms

Release Version S-CZ8.2.5 includes both the Oracle Core Session Manager (CSM) and Unified Session Manager (USM) products. The Oracle USM is supported on the Acme Packet series platforms. The Oracle CSM is supplied as virtual machine software or as a software-only delivery suitable for operation on server hardware. Refer to sales documentation updates for information further specifying hardware support.

Related Documentation

Version S-CZ8.2.5 software relies on version SCZ820 documentation for some documentation. The following table lists the members that comprise the documentation set for this release:

Hardware documentation is relevant only to the Oracle USM. Refer to your hardware vendor's documentation for information required for Oracle CSM operation.

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 6100 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6100.
Acme Packet 6300 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6300.
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.
ACLI Configuration Guide	Contains information about the administration and software configuration of the Service Provider Oracle Communications Unified Session Manager.
ACLI Reference Guide	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.

Document Name	Document Description
Maintenance and Troubleshooting Guide	Contains information about Oracle Communications Unified Session Manager logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.
MIB Reference Guide	Contains information about Management Information Base (MIBs), 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 Oracle Communications Unified Session Manager's accounting support, including details about RADIUS and Diameter accounting.
HDR Resource Guide	Contains information about the Oracle Communications Unified Session Manager's Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Administrative Security Essentials	Contains information about the Oracle Communications Unified Session Manager's support for its Administrative Security license.
SBC Family Security Guide	Contains information about security considerations and best practices from a network and application security perspective for the Oracle Communications Unified Session Manager family of products.
Installation and Platform Preparation 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 Resource Guide	Contains information about configuring and using Header Manipulation Rules to manage service traffic.
REST API Guide	Contains information about the supported REST APIs and how to use the REST API interface.

Revision History

Date	Description
August 2019	• Initial Release

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.

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.

1

Introduction to Oracle Communications Unified Session Manager S-CZ8.2.5

This Oracle Communications Unified Session Manager (OCUSM) Release Introduction chapter provides the following information about this product:

- Supported platforms and hardware requirements
- An overview of the new features available in this release
- An overview of previously-available features that are new to the GA of this major release
- A summary of changes the OCUSM interfaces including the ACLI, MIB Support, and accounting interfaces.
- A summary of behavioral changes
- A summary of known issues and caveats

Review the S-CZ8.2.0 OCSBC Release Notes for further information on this S-CZ8.2.5 release of the OCUSM. There is overlap between these products, including functionality, features, compatibility, known issues and caveats. The S-CZ8.2.0 OCSBC Release Notes provides complimentary detail to this *Essentials Guide* and documents the aspects of the products that are common to both.

New Features

The S-Cz8.2.5 release of the Oracle Communications Unified Session Manager (OCUSM) supports the following new features and enhancements.

The OCUSM 825 Release adds OCSBC 820 functionality. Please refer to OCSBC 8.2.0 Release Documentation for the product details. It may be possible that OCUSM use cases differ from OCSBC use cases. Therefore, it is recommended to validate the applicable scenarios in the lab environment before a field deployment.

ODI Preservation

As the Oracle Communications Unified Session Manager (OCUSM) works through dialogs with Application Servers (AS), it saves and uses the Original Dialog Indicator (ODI) parameter to manage a call's service subscription sequence. By default, the OCUSM deletes the in-memory Service Profile, including the ODI, on receiving a final response to a transaction with an AS. If you enable the **preserve-odi** parameter in the **sip-config** however, the OCUSM maintains the in-memory service profiles, and each associated ODI, until it receives the **BYE** from the AS that ends the dialog between them. This is a global configuration, causing the OCUSM to maintain all ODIs for the duration of the dialog.

See the SIP Call section in this Essentials Guide.

Making Key Information sent to OCOM Anonymous

The OCUSM allows you to hide the information presented in the **SUBJECT** headers and MIME body in **INVITE** and/or **MESSAGE** methods before sending the packets for analysis to

Oracle Communications Operation Manager (OCOM). You do this by configuring the **anonymize-invite** and/or the **anonymize-message** options in the **comm-monitor** element. This provides an extra layer of security for Chat and SIP sessions.

See the explanation and configuration procedure on this feature in the References and Debugging chapter of this Essentials Guide.

Platform Support

The Oracle Communications Unified Session Manager (OCUSM) S-CZ8.2.5 software supports the following platforms.

Acme Packet Platforms

The following platforms are supported by the S-CZ8.2.5 version of the OCUSM:

- Acme Packet 4600
- Acme Packet 6300

Hypervisors are Not Supported

The OCUSM does not operate on Hypervisors. Upon installation, the software autodetects the platform and sets the product to OCUSM when deployed over Acme Packet hardware.

Image Files and Boot Files

This software version, when deployed on Acme Packet hardware, does not require that you run the **setup product** command. Instead, it autodetects the Acme Packet platform and sets up as the OCUSM. There are no other valid product setup options.

For Acme Packet Platforms

Use the following files for new installations and upgrades on Acme Packet platforms.

- Image file: `nnSCZ825.bz`
- Bootloader file: `nnSCZ825.boot`

Upgrade Information

This section provides key information about upgrading to this software version.

Supported Upgrade Paths

The following in-service (hitless) upgrade and rollback paths are supported by both the OCCSM and the OCUSM:

- S-CZ7.3.5 to S-CZ8.2.5

When upgrading to this release from a release older than the previous release, read all intermediate Release Notes documents for notification of incremental changes.

Upgrade Checklist

Before upgrading the Oracle Communications Unified Session Manager software:

1. Obtain the name and location of the target software image file from either Oracle Software Delivery Cloud, https://edelivery.oracle.com/, or My Oracle Support, https://support.oracle.com, as applicable.
2. Provision platforms with the Oracle Communications Unified Session Manager image file in the boot parameters.
3. Run the **check-upgrade-readiness** command and examine its output for any recommendations or requirements prior to upgrade.
4. Verify the integrity of your configuration using the ACLI **verify-config** command.
5. Back up a well-working configuration. Name the file descriptively so you can fall back to this configuration easily.
6. Refer to the Oracle Communications Unified Session Manager Release Notes for any caveats involving software upgrades.

Upgrade and Downgrade Caveats

The following items provide key information about upgrading and downgrading with this software version.

Reset the rsa_ssh.key

After you upgrade from S-CZ7.3.5 to S-CZ8.2.5, you must manually reset the rsa_ssh.key when the host OpenSSH client version is 7.6 or newer. Applies to all platforms.

1. Delete the old ssh_rsa.key in the /code/ssh directory in the shell environment.
2. Reboot the OCUSM, using reboot from the ACLI prompt.

Reset Local Passwords for Downgrades

Oracle delivers increased encryption strength for internal password hash storage for the S-CZ8.2.5 release. This affects downgrades to the S-CZ7.3.5 releases because the enhanced password hash algorithm is not compatible with those earlier OCUSM software versions. If you change any local account passwords after upgrading to S-CZ8.2.5, local authentication does not work and the system locks. Unlocking the system requires a factory reset. Oracle recommends that you do not change any local account passwords after upgrading to S-CZ8.2.5 from a prior release, until you are sure that you will not need to downgrade. If you do not change any local account passwords after upgrading to S-CZ8.2.5, downgrading is not affected.

> **Caution:**
>
> If you change the local passwords after you upgrade to S-CZ8.2.5, and then later want to downgrade to a previous release, reset the local user passwords with the following procedure before you downgrade because the system locks you out until all passwords are cleared. If you get locked out, you must contact Oracle support to clear the passwords.

Perform the following procedure on the standby OCUSM first, and then force a switchover. Repeat steps 1-10 on the newly active OCUSM. During the procedure, the OCUSM powers down and you must be present to manually power up the OCUSM.

> **Caution:**
>
> Be aware that the following procedure erases all of your local user passwords, as well as the log files and CDRs located in the /opt directory of the OCUSM.

1. Log on to the console of the standby OCUSM in Superuser mode, type `halt sysprep` on the command line, and press ENTER.
   The system displays the following warning:

   ```
   ******************************************
   WARNING: All system-specific data will be permanently
   erased and unrecoverable.

   Are you sure [y/n]
   ```

2. Type `y`, and press ENTER.
3. Type your Admin password, and press ENTER.
   The system erases your local passwords, log files, and CDRs and powers down.
4. Power up the standby OCUSM.
5. During boot up, press the space bar when prompted to stop auto-boot so that you can enter the new boot file name.
   The system displays the boot parameters.
6. For the Boot File parameter, type the boot file name for the software version to which you want to downgrade next to the existing version. For example,`nnECZ800.bz`.
7. At the system prompt, type `@`, and press ENTER.
   The standby reboots.
8. After the standby reboots, do the following:
   a. Type `acme`, and press ENTER.
   b. Type `packet`, and press ENTER.
9. Type and confirm the password that you want for the User account.
10. Type and confirm the password that you want for the Superuser account.
11. Perform a **notify berpd force** on the standby to force a switchover.
12. Repeat steps 1-10 on the newly active OCUSM.

Maintain DSA-Based HDR and CDR Push Behavior

To maintain your existing DSA key-based CDR and HDR push behavior after upgrading from S-CZ7.3.5M2px to S-CZ8.2.5, perform the following procedure:

1. Navigate to the **security**, **ssh-config**, **hostkey-algorithms** configuration element and manually enter the DSA keys you want to use.
2. Save and activate your configuration.
3. Execute the **reboot** command from the ACLI prompt.

Self-Provisioned Entitlements

This release uses the following self-provisioned entitlements and license keys to enable features.

This table lists the features you enable with the **setup entitlements** command.

Feature	Type
Unified Session Manager Base	boolean
Session Capacity	Integer
Accounting	boolean
BFD	boolean
IPv4 - IPv6 Interworking	boolean
Load Balancing	boolean
Policy Server	boolean
Quality of Service	boolean
Admin Security	boolean
ANSSI R226 Compliance	boolean
Endpoint Capacity (Registration Cache)	Integer
IMS-AKA Endpoints	Integer
IPSec Trunking Sessions	Integer
MSRP B2BUA Sessions	Integer
SRTP Sessions	Integer
Transcode Codec AMR Capacity	Integer
Transcode Codec AMRWB Capacity	Integer
Transcode Codec EVRC Capacity	Integer
Transcode Codec EVRCB Capacity	Integer
Transcode Codec EVS Capacity	Integer
Transcode Codec OPUS Capacity	Integer
Transcode Codec SILK Capacity	Integer
TSCF Tunnels	Integer

Note:

Despite its presence in this entitlement list, this version of the OCUSM does not support TSCF tunnels.

You enable the following features by installing a license key at the **system, license** configuration element. Request license keys at the License Codes website at http://www.oracle.com/us/support/licensecodes/acme-packet/index.html.

Feature	Type
Lawful Intercept	boolean

System Capacities

System capacities vary across the range of platforms that support the Oracle Communications Unified Session Manager. To query the current system capacities for the platform you are using, execute the **show platform limits** command.

Coproduct Support

The following products and features run in concert with the Oracle Communications Unified Session Manager for their respective solutions. Contact your Sales representative for further support and requirement details.

Oracle Communications Operations Manager

Oracle Communications Operations Manager (OCOM) versions 4.0 and later support this GA release of the OCUSM.

Oracle Communications Session Delivery Manager

Oracle Communications Session Deliver Manager (OCSDM) versions 8.2.0 and later supports this GA release of the OCUSM.

> **Note:**
>
> The ability to enable and disable the new ODI preservation feature is planned to be supported in SDM Rel 8.2.1 (current GA release is 8.2).

Documentation Changes

The following information lists and describes the changes made to the Oracle Communications Unified Session Manager (OCUSM) documentation set for S-CZ8.2.5.

Standalone USM Documentation Set

For version S-CZ7.2.5 and S-CZ7.3.5 software, the OCUSM had a documentation set that could be considered unique to that product. Starting with version S-CZ8.2.5, product documentation is reverted to the original Essentials model, which provides a *OCUSM Essentials Guide* as unique, and refers to the OCSBC Documentation Set for product components, features and procedures that are generic across all three product sets. The documentation set, listed in the front matter of this document, provides configuration information across all session control products.

The Essentials Guides for OCUSM includes a **Basics** chapter. This **Basics** chapter provides you with direction on required and optional configuration documentation you need from the S-CZ8.2.0 OCSBC documentation set to configure and operate the S-CZ8.2.5 OCUSM.

Behavioral Changes

The following information documents the behavioral changes to the Oracle Communications Unified Session Manager (OCUSM) in this software release.

TLS1.0

TLS1.0 is no longer advertised by default during session negotiation when the **tls-version** parameter is set to **compatibility**. To advertise TLS1.0 during session negotiation, navigate to the **security-config** element and set the **options** parameter to **+sslmin=tls1.0**. Note that the current default is TLSv1.2.

```
ORACLE(security-config)# options +sslmin=tls1.0
```

Patches Included in This Release

The following information assures you that when upgrading, the S-CZ8.2.5 release includes defect fixes from neighboring patch releases.

Baseline

S-Cz8.2.0p3 is the patch baseline, which is the most recent build from which Oracle created S-Cz8.2.5.

Neighboring Patches Also Included

- S-Cz7.3.5m2p11

2

Oracle Communications Unified Session Manager Basics

This chapter introduces key features and capabilities of the Oracle Communications Unified Session Manager. As an integrated IP Multimedia Sub-system (IMS) Call Session Control Function (CSCF), the Oracle Communications Unified Session Manager performs CSCF roles on a single platform. This configuration can simplify IMS deployment and operation scenarios. See product specification or Oracle consultative services to understand applicable deployments. This document does not provide general instruction on IMS; review the many generally-available IMS resources if you need introduction to its complex mechanisms.

The Oracle Communications Unified Session Manager operates within the context of an IMS or over-the-top (OTT) deployment. Within IMS, the device performs standard registration and call handling functions in compliance with 3GPP requirements. The presence of multiple CSCF functions on a single platform can simplify IMS operations. For those who prefer an OTT deployment, the Oracle Communications Unified Session Manager supports registration and call management via an ENUM or local subscriber database.

The Oracle Communications Unified Session Manager functions within IMS environments using Home Subscriber Server (HSS) deployments as subscriber database. Operation with IMS HSS resources is described in the Diameter Based Oracle USM chapter. For OTT, operation using ENUM resources is described in the ENUM Based Oracle Communications Unified Session Manager chapter and operation with local subscriber tables is described in the Local Subscriber Tables chapter.

By utilizing Oracle's Session Border Controller (SBC) product features, the Oracle Communications Unified Session Manager is also capable of performing a wide range of edge functions, separated along the following roles:

- Access functions
- Interconnect functions

This functionality and configuration is covered in the Oracle Communications Unified Session Manager Supporting the IMS Edge chapter.

Oracle Communications Unified Session Manager and IMS

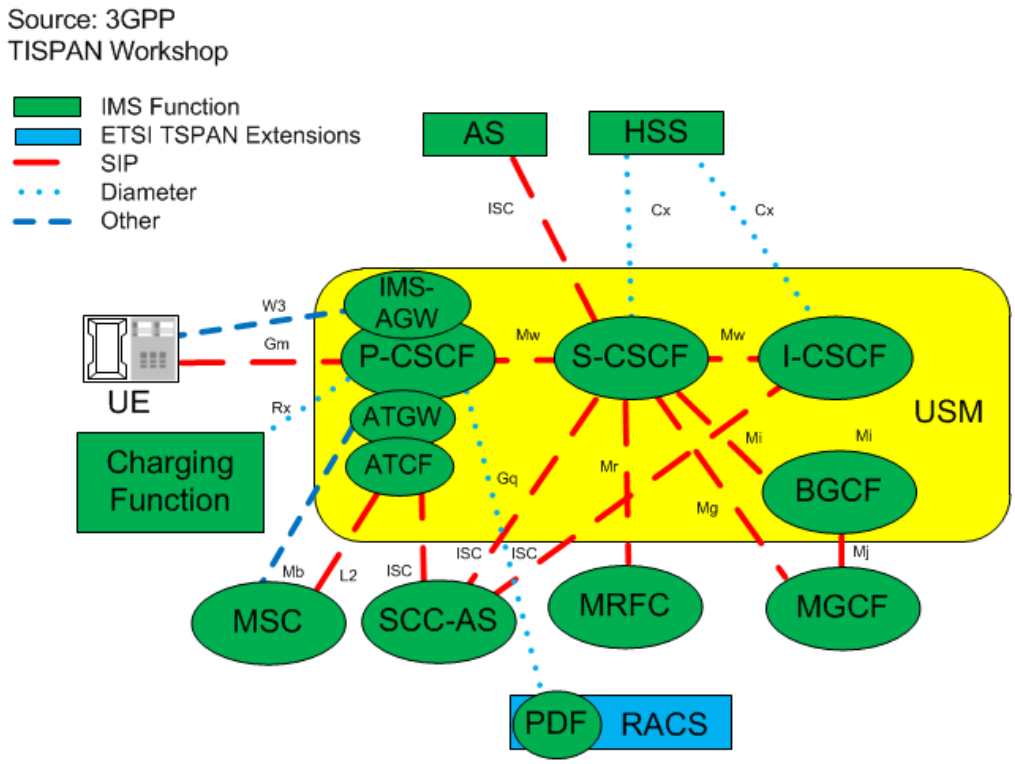
The ETSI TISPAN NGN defines several subsystems that make up the NGN architecture. The model for the target NGN architecture is depicted below.

The Oracle Communications Unified Session Manager is designed to function as an integrated:

- Proxy-Call Session Control Function (P-CSCF)
- Interrogating-Call Session Control Function (I-CSCF)
- Serving-Call Session Control Function (S-CSCF)

In addition, the Oracle Communications Unified Session Manager performs many related functions. As such, the Oracle Communications Unified Session Manager is typically deployed in support of LTE and other IMS-based networks.

The functions performed by the Oracle Communications Unified Session Manager are best understood as functions of standard IMS elements. The diagram below depicts the interconnection of these elements across an IMS architecture.



High level definitions of these functions include:

- P-CSCF—The first point of contact within the IMS system for the endpoint. The P-CSCF determines whether and how to pass the traffic to the appropriate S-CSCF.
- I-CSCF—IMS passes traffic to the I-CSCF if the target S-CSCF is unknown.
- S-CSCF—Interaction with the Home Subscriber Server (HSS) determines whether and how to provide service to the endpoint.
- BGCF—The breakout gateway control function provides signaling transit to network domains external to the IMS.
- IMS-AGW—Access gateway complimenting the P-CSCF function to broaden the range of devices that can access the IMS.
- Ancilliary Access Functions—As part of 3GPP Release 10, new interfaces and protocols have been defined to improve mobility across LTE and 2G/3G networks and address the latency concern from previous architectures. The enhancements to the SRVCC system defines two logical entities located in the access network:
  - ATCF—The (Access Transfer Control Function) is a signalling controller function complementing the signalling and media control roles of the P-CSCF and IMS-AGW functions in the SBC.
  - ATGW—The Access Transfer Gateway is a media anchor point complementing the signalling and media control roles of the P-CSCF and IMS-AGW functions.

Refer to 3GPP specifications for complete element definitions and explanations of the functions they can or must perform. Key functions and considerations that introduce the Oracle Communications Unified Session Manager's operation are briefly discussed below.

As P-CSCF, the Oracle Communications Unified Session Manager adds traffic policing, UA functions for special SIP signaling handling, CDR generation, topology hiding, policy enforcement, and hosted NAT traversal to the 3GPP-defined P-CSCF functions. As such, the Oracle Communications Unified Session Manager can reside at the network's edge performing functions that may otherwise be performed by an SBC.

As I-CSCF, the Oracle Communications Unified Session Manager complies with 3GPP standards to perform the interrogating function and locate the proper S-CSCF for a given session.

As S-CSCF, the near the subscriber that facilitate rapid and predictable handover from LTE to circuit 2G/3G networks and update the VCC application server after the access transfer. They facilitate rapid and predictable handover from LTE to circuit 2G/3G networks and update the VCC application server after the access transfer. The combination of these new functions and improved call flow reduces the signalling hops required to handover the active voice call to the new access network complies with 3GPP standards and Oracle Communications Unified Session Manager to manage sessions. It interacts with the HSS to determine whether any given registration can reside locally, or be managed by another S-CSCF device. It also interacts with the HSS and other infrastructure components to provide applicable services within the context of a given session.

As ATCF and ATGW, Oracle Communications Unified Session Manager provides services near the subscriber that facilitate rapid and predictable handover from LTE to circuit 2G/3G networks and update the VCC application server after the access transfer. The combination of these new functions and improved call flow reduces the signalling hops required to handover the active voice call to the new access network

Oracle USM and OTT

The Oracle USM may be deployed as a SIP registrar in OTT environments. OTT environments may use ENUM resources as subscriber database. Alternatively, they may use local subscriber tables (LST). Both ENUM and LST deployment descriptions and configuration instructions are provided within this guide.

Multiple Control Functions on a Single Device

Most Oracle Communications Unified Session Manager deployments are expected to be homogenous. That is, the Oracle Communications Unified Session Manager would most commonly be operating with other Oracle Communications Unified Session Managers. In addition, because individual Oracle Communications Unified Session Managers can perform all CSCF functions, the network administrator can expect traffic patterns that are different from deployments that use discrete components to perform CSCF functions.

For example, it would be common for the subscriber database to allow the S-CSCF within a given Oracle Communications Unified Session Manager to take ownership of a UE seeking services via that same device's P-CSCF. In these cases, the user can expect the effective path and service route between a UE and an S-CSCF to be the same as the path between that UE and the P-CSCF.

IMS design anticipates multiple devices performing CSCF functions. Therefore, the Oracle Communications Unified Session Manager also supports operation across multiple Oracle Communications Unified Session Managers.

Elements of Oracle Communications Unified Session Manager Configuration

Oracle Communications Unified Session Manager consists of multiple configuration elements. This guide presents these elements, separating them along conceptual category with chapters roughly equating to configuration sequence. This section lists configuration elements, providing the reader with a consolidated picture of overall product configuration.

Oracle documents this product using an Essentials model, which results in a unique *OCCSM Essentials Guide* document, and refers to the OCSBC Documentation Set for additional, related components, features and procedures. The documentation set, listed in the front matter of this document, provides configuration information across all session control products. The OCUSM and OCCSM filter out configuration elements, sub-elements and parameters that do not apply to themselves, preventing you from performing invalid configuration procedures.

See the Base Configuration Elements Appendix for minimal configuration setting examples that establish an operable Oracle Communications Unified Session Manager.

USM Configuration Elements

Required initial device configuration elements, explained in the Getting Started chapter in the *ACLI Configuration Guide*, include:

- Boot Parameters
- Management Interfaces
- Device Password
- Default Gateway
- Product licensing/entitlement

Required network and SIP service configuration components, explained in multiple chapters in the *ACLI Configuration Guide*, include:

- Enable SIP-Config—System Configuration Chapter
- Service physical and network interface(s)—System Configuration Chapter
- SIP Interfaces—System Configuration Chapter
- SIP Ports—System Configuration Chapter
- Realms—Realms and Nested Realms Chapter
- ENUM—Routing with Local Policy Chapter

Required IMS configuration elements, explained in the Oracle Communications Unified Session Manager Supporting the IMS Core Chapter in this document, include:

- Subscriber Database
- SIP Registrar
- Authentication Profile
- ENUM for e.164 Translation

- Registration Event
- IMS Access Interface

Common Configuration Elements

Common configuration that may be needed for your deployment includes:

- Session Agents
- ENUM Routing
- Local Routing
- Initial Filter Criteria (iFC)
- 3rd Party Registration Service
- IMS Access Functions
- IMS Interconnect Functions
- Media manager
- Media Steering pools

Common secondary management element configuration includes:

- High Availability (HA)
- CDR Accounting
- SNMP Management

Other Configuration Elements

Configuration elements that are available, but may not be required for your deployment include:

- Assorted SIP Configuration
- Number Translation
- Admission Control and QoS
- DoS and other Security Configuration
- Diameter Policy Configuration
- Transcoding
- Local Routing Configuration
- Realm-based media policy and controls
- Traffic Monitoring

See the Appendix on Base USM Configuration Elements for a list of configuration setting examples that bring your system to a minimally operational state in an IMS environment. Change addressing and other infrastructure-dependent setting examples to match that of your environment.

High Availability

Oracle Communications Unified Session Managers are deployed in pairs to deliver continuous high availability (HA) for interactive communication services. The HA design guarantees that

no stable calls are dropped in the event of any single point failure. Furthermore, the Oracle Communications Unified Session Manager HA design provides for full media, registration, call and service state to be shared across an HA node. The solution uses a VRRP-like design, where the two systems share a virtual MAC address and virtual IPv4 address for seamless switchovers.

In the HA pair, one Oracle Communications Unified Session Manager is the primary system, and is used to process signaling and media traffic. The backup system remains fully synchronized with the primary system's session status. The primary system continuously monitors itself for connectivity and internal process health. If it detects service-disrupting conditions or degraded service levels, it will alert the backup Oracle Communications Unified Session Manager to become the active system.

3

Oracle USM Supporting the IMS Core

General Description

The Oracle Communications Unified Session Manager functions in an IMS core. It communicates with the HSS to obtain Authorization, Authentication, S-CSCF assignment, and ultimately routing instructions. To accomplish these functions, the Oracle Communications Unified Session Manager can perform the SIP registrar role in conjunction with an HSS.

Message Authentication for SIP Requests

The Oracle Communications Unified Session Manager authenticates requests by configuring the sip authentication profile configuration element. The name of this configuration element is either configured as a parameter in the sip registrar configuration element's authentication profile parameter or in the sip interface configuration element's sip-authentication-profile parameter. This means that the Oracle Communications Unified Session Manager can perform SIP digest authentication either globally, per domain of the Request URI or as received on a SIP interface.

After naming a sip authentication profile, the received methods that trigger digest authentication are configured in the methods parameter. You can also define which anonymous endpoints are subject to authentication based on the request method they send to the Oracle Communications Unified Session Manager by configuring in the anonymous-methods parameter. Consider the following three scenarios:

- By configuring the methods parameter with REGISTER and leaving the anonymous-methods parameter blank, the Oracle Communications Unified Session Manager authenticates only REGISTER request messages, all other requests are unauthenticated.
- By configuring the methods parameter with REGISTER and INVITE, and leaving the anonymous-methods parameter blank, the Oracle Communications Unified Session Manager authenticates all REGISTER and INVITE request messages from both registered and anonymous endpoints, all other requests are unauthenticated.
- By configuring the methods parameter with REGISTER and configuring the anonymous-methods parameter with INVITE, the Oracle Communications Unified Session Manager authenticates REGISTER request messages from all endpoints, while INVITES are only authenticated from anonymous endpoints.

User Authorization

In an IMS network, the Oracle Communications Unified Session Manager requests user authorization from an HSS when receiving a REGISTER message. An HSS is defined on the Oracle Communications Unified Session Manager by creating a home subscriber server configuration element that includes a name, ip address, port, and realm as its basic defining data.

UAR/UAA Transaction

Before requesting authentication information, the Oracle Communications Unified Session Manager sends a User Authorization Request (UAR) to the HSS for the registering endpoint to determine if this user is allowed to receive service. The Oracle Communications Unified Session Manager populates the UAR's AVPs as follows:

- Public-User-Identity—the SIP AOR of the registering endpoint
- Visited-Network-Identity—the value of the network-id parameter from the ingress sip-interface.
- Private-User-Identity—the username from the SIP authorization header, if it is present. If not, this value is the public User ID.
- User-Authorization-Type—always set to REGISTRATION_AND_CAPABILITIES (2)

The Oracle Communications Unified Session Manager expects the UAA to be either:

- DIAMETER_FIRST_REGISTRATION
- DIAMETER_SUBSEQUENT_REGISTRATION

Any of these responses result in the continued processing of the registering endpoint. Any other result code results in an error and a 403 returned to the registering UA (often referred to as a UE). The next step is the authentication and request for the H(A1) hash.

SIP Digest User Authentication

Authentication via MAR/MAA

To authenticate the registering user, the Oracle Communications Unified Session Manager needs a digest realm, QoP, and the H(A1) hash. It requests these from a server, usually the HSS, by sending it a Multimedia Auth Request (MAR) message. The MAR's AVPs are populated with:

- Public-User-Identity—the SIP AOR of the endpoint being registered (same as UAR)
- Private-User-Identity—the username from the SIP authorization header or the SIP AOR if the AOR for PUID parameter is enabled. (Same as UAR)
- SIP-Number-Auth-Items—always set to 1
- SIP-Auth-Data-Item -> SIP-Item-Number—always set to 1
- SIP-Auth-Data-Item -> SIP-Authentication-Scheme—always set to SIP_DIGEST
- Server-Name—the home-server-route parameter in the sip registrar configuration element. It is the URI (containing FQDN or IP address) used to identify and route to this Oracle Communications Unified Session Manager.

The Oracle Communications Unified Session Manager expects the MAA to include a SIP-Auth-Data-Item VSA, which includes digest realm, QoP and H(A1) information as defined in RFC2617. The information is cached for subsequent requests. Any result code received from the HSS other than DIAMETER_SUCCESS results in a 403 error response returned for the original request.

The MAR/MAA transaction is conducted with the server defined in the credential retrieval config parameter found in the sip-authentication profile configuration element. This parameter is populated with the name of a home-subscriber-server configuration element.

SIP Authentication Challenge

When the Oracle Communications Unified Session Manager receives a response from the HSS including the hash value for the user, it sends a SIP authentication challenge to the endpoint, if the endpoint did not provide any authentication headers in its initial contact the with Oracle Communications Unified Session Manager. If the endpoint is registering, the Oracle Communications Unified Session Manager replies with a 401 Unauthorized message with the following WWW-Authenticate header:

```
WWW-Authenticate: Digest realm="atlanta.com", domain="sip:boxesbybob.com",
qop="auth", nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="", stale=FALSE,
algorithm=MD5
```

If the endpoint initiates any other request to the Oracle Communications Unified Session Manager besides REGISTER, the Oracle Communications Unified Session Manager replies with a 407 Proxy Authentication Required message with the following Proxy-Authenticate header:

```
Proxy-Authenticate: Digest realm="atlanta.com", qop="auth",
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="", stale=FALSE, algorithm=MD5
```

Authentication Header Elements

- Domain—A quoted, space-separated list of URIs that defines the protection space. This is an optional parameter for the "WWW-Authenticate" header.
- Nonce—A unique string generated each time a 401/407 response is sent.
- Qop—A mandatory parameter that is populated with a value of "auth" indicating authentication.
- Opaque—A string of data, specified by the Oracle Communications Unified Session Manager which should be returned by the client unchanged in the Authorization header of subsequent requests with URIs in the same protection space.
- Stale—A flag indicating that the previous request from the client was rejected because the nonce value was stale. This is set to true by the SD when it receives an invalid nonce but a valid digest for that nonce.
- Algorithm—The Oracle Communications Unified Session Manager always sends a value of "MD5"

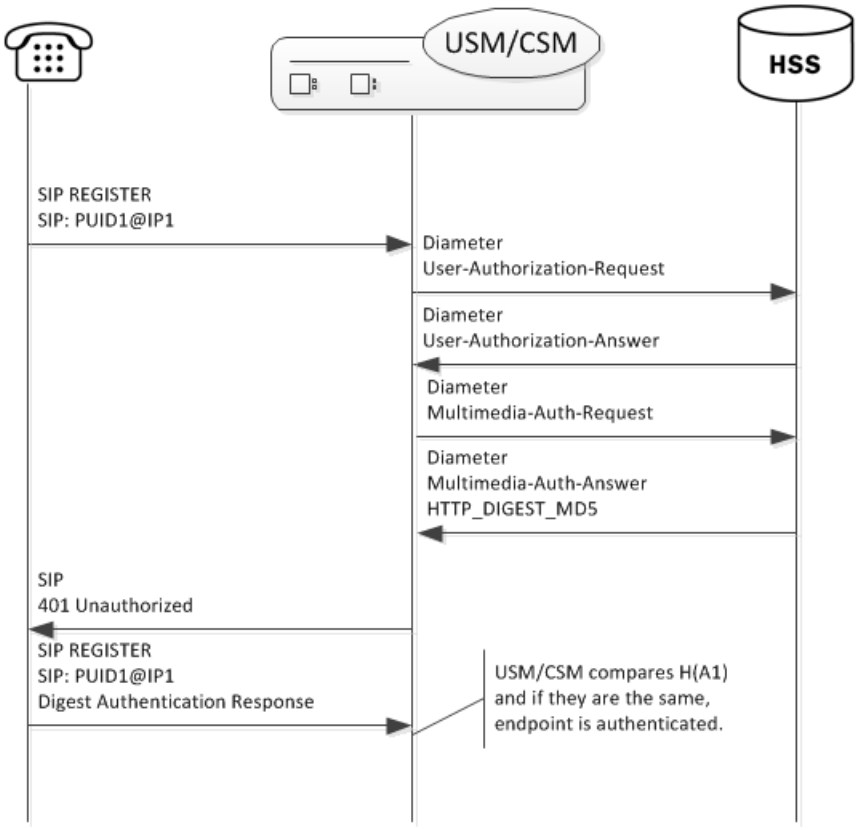
SIP Authentication Response

After receiving the 401/407 message from the Oracle Communications Unified Session Manager, the UA resubmits its original request with an Authorization: header including its own internally generated MD5 hash.

Oracle Communications Unified Session Manager Authentication Check

At this point, the Oracle Communications Unified Session Manager has received an MD5 hash from the HSS and an MD5 hash from the UA. The Oracle Communications Unified Session Manager compares the two values and if they are identical, the endpoint is successfully authenticated. Failure to match the two hash values results in a 403 or 503 sent to the authenticating endpoint.

The following image shows the User Authorization and Authentication process:



Note:

Diagram information states "USM/CSM" when the applicable content applies to both the Oracle USM and the Oracle CSM.

The Oracle Communications Unified Session Manager acts as a SIP Registrar and updates an HSS with the state of its registrants.

IMS-AKA Support

The Oracle Communications Unified Session Manager also supports IMS-AKA for secure authentication end-to-end between UAs in an LTE network and an IMS core. It supports IMS-AKA in compliance with 3GPP specifications TS 33-203 and TS 33-102.

The goal of IMS-AKA is to achieve mutual authentication between end station termination mechanisms, such as an IP Multimedia Services Identity Module (ISIM), and the Home Network (IMS Core). Achieving this goal requires procedures both inside and outside the core. Ultimately, IMS performs the following:

- Uses the IMPI to authenticate the home network as well as the UA;
- Manages authorization and authentication information between the HSS and the UA;
- Enables subsequent authentication via authentication vectors and sequence information at the ISIM and the HSS.

The Oracle Communications Unified Session Manager authenticates registrations only. This registration authentication process is similar to SIP Digest. The process accepts REGISTER requests from UAs, conducts authorization procedures via UAR/UAA exchanges and conducts authentication procedures via MAR/MAA exchanges and challenges with the UA.

Configuration and operational support are not the same on the Oracle USM and Oracle CSM. This is because the Oracle USM can perform the P-CSCF role as well as the I-CSCF and S-CSCF roles. Applicable configuration to support IMS-AKA on the P-CSCF access interface is documented in the Security chapter of the *Oracle Communications Session Border Controller ACLI Configuration Guide*. This configuration includes defining an IMS-AKA profile, enabling the **sip-interface** for IMS-AKA and configuring the **sip-port** to use the profile.

There is no configuration required for the S-CSCF role, but there is an optional configuration that specifies how may authentication vectors it can accept from the HSS. The S-CSCF stores these authentication vectors for use during subsequent authentications. Storing vectors limits the number of times the device needs to retrieve them from the HSS. The default number of authentication vectors is three.

Authentication Sequence - Registration

UAs get service from an IMS core after registering at least one IMPU. To become registered, the UA sends REGISTER requests to the IMS core, which then attempts to authenticate the UA.

The first device to receive the REGISTER at the core is a P-CSCF, such as the Oracle USM. For the Oracle USM, appropriate configuration determines that it uses IMS-AKA as the authentication mechanism on the access interface. For an Oracle CSM, the presence and state of the “integrity-protected” parameter in the Authorization header of a REGISTER triggers the use of IMS-AKA. If the value of this parameter is either “yes” or "no", IMS-AKA is invoked. If the parameter is not present, or it is set to any other value, the Oracle USM falls back to SIP Digest authentication.

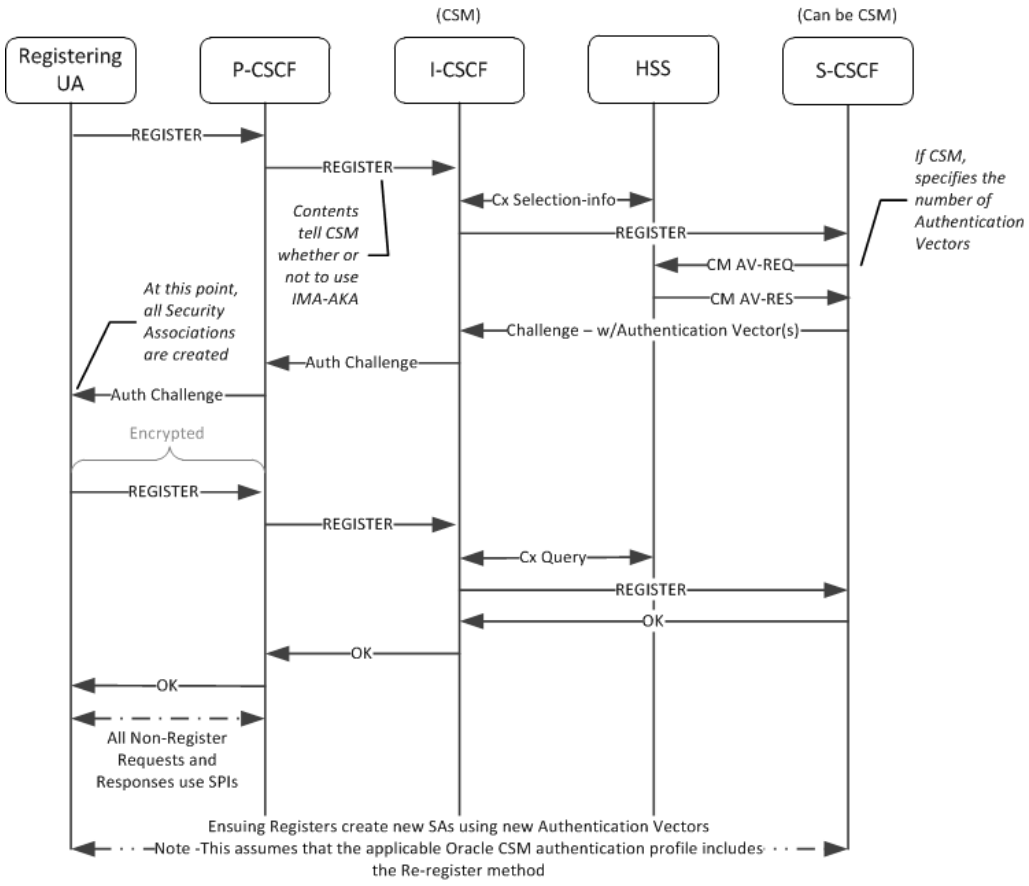
To proceed with IMS-AKA authentication, the P-CSCF engages in S-CSCF selection procedures via the I-CSCF to identify the target S-CSCF. Having identified the S-CSCF (your Oracle Communications Unified Session Manager), the I-CSCF forwards the REGISTER to it. The I-CSCF next engages in standard UAR and MAR procedures. For IMS-AKA deployments, the HSS follows procedures defined in TS 33-203 to create authentication vectors for the UA.

The HSS provides the vectors to the S-CSCF, which then proceeds with authentication procedures defined in TS 33-203.

After processing, the S-CSCF uses authentication vectors to challenge the UA. The UA uses the information in this challenge to, first, authenticate the Home Network. Having confirmed the network, the UA then prepares and sends its authentication information back towards the S-CSCF. The S-CSCF is then responsible for authenticating the UA. The S-CSCF sends a 200OK back to the UA upon successful authentication, allowing the UA to get service from the HN.

The Oracle Communications Unified Session Manager caches the AOR's registration and stores authentication vectors for subsequent authentications, thereby minimizing the work required by the HSS.

The overall sequence is depicted below.



Outside the Core

LTE networks include UAs that have an IP Multimedia Service Identity Module (ISIM) or equivalent. ISIMs are configured with a long-term key used to authenticate and calculate cipher keys, as well as IP Multimedia Private and Public Identities (IMPI and IMPU). The ISIM serves as the means of authenticating the home network to the UA. The UA, in turn, sends information based on it's ISIM configuration to the home network, which can then authenticate the UA.

Establishment of Security Associations (SAs) to and from the UA are the responsibility of the P-CSCF. The P-CSCF should also be capable of managing the processes when the UA is behind a NAT.

> **Note:**
>
> Within the context of IMS-AKA, only traffic between the P-CSCF and the UA is encrypted.

Authentication Success

When using IMS-AKA, successful registration of a UA consists of registering at least one IMPU and the IMPI authenticated within IMS. The UA begins this process by sending it REGISTER request to the P-CSCF properly specifying IMS-AKA authentication. IMS then performs standard procedures to identify the appropriate S-CSCF. Upon receipt of the REGISTER, the S-CSCF checks for the presence of an authentication vector. If it is present the S-CSCF issues the authentication challenge; if not, it requests authentication vector(s) from the HSS. Note that the Oracle Communications Unified Session Manager allows you to request multiple authentication vectors via configuration. The HSS provides the following components within an authentication vector:

- RAND—random number
- XRES—expected response
- CK—cipher key
- IK—integrity key
- AUTN—authentication token

The MAR provided to the S-CSCF differ from that of SIP digest authentication requests as follows:

- The SIP-Number-Auth-Items AVP specifies the number of authentication vectors, which is equal to the home-subscriber-server's `num-auth-vectors` setting.
- The SIP-Authentication-Scheme AVP specifies the authentication scheme, Digest-AKAv1-MD5.

At this point, the Oracle Communications Unified Session Manager can send the authentication challenge to the UA. If multiple authentication vectors were provided by the HSS, the Oracle Communications Unified Session Manager can independently authenticate the UA until the pool is exhausted. The S-CSCF stores the RAND it sends to the UA to resolve future synchronization errors, if any. No authentication vector can be used more than once. This is validated by the ISIM, using a sequence number (SQN).

When a P-CSCF receives an authentication challenge, it removes and stores the CK and the IK. The P-CSCF forward the rest of the information to the UA.

The UA is responsible for verifying the home network. Having received the AUTN from the P-CSCF, the UA derives MAC and SQN values. Verifying both of these, the UA next generates a response including a shared secret and the RAND received in the challenge. The UA also computes the CK and IK.

Upon receipt of this response, IMS provides the message to the S-CSCF, which determines that the XRES is correct. If so, it registers the IPMU and, via IMS sends the 200 OK back to the UA.

Authentication Failure

Either the UA or IMS can deny authentication via IMS-AKA. In the case of the UA, this is considered a network failure; in the case of IMS there would be a user authentication failure.

Network Authentication Failure

The UA determines that the HN has failed authentication, it sends a REGISTER request with an empty authorization header parameter and no authentication token for synchronization (AUTS). This indicates that the MAC parameter was invalid as determined by the UA. In this case, the S-CSCF sends a 403 Forbidden message back to the UA.

User Authentication Failure

IMS-AKA determines user authentication failure as either:

- IK incorrect—If the REGISTER includes a bad IK, the P-CSCF detects this and discards the packet at the IPSEC layer. In this case, the REGISTER never reaches the S-CSCF.
- XRES incorrect—In this case, the REGISTER reaches the S-CSCF. The S-CSCF detects the incorrect XRES, the S-CSCF sends a 4xxx Auth_Failure message back to the UA via IMS.

Synchronization

Synchronization refers to authentication procedures when the (REFRESH TIMING) is found to be stale. This is not an authentication failure.

The UA may send an AUTS in response to the challenge, indicating that the authentication vector sequence is "out-of-range". Upon receipt of the AUTS, the S-CSCF sends a new authorization vector request to the HSS. The HSS checks the AUTS and, if appropriate sends a new set of authentication vectors back the the S-CSCF. Next the S-CSCF sends 401 Unauthorized back to the UA. Assuming the UA still wants to register, this would trigger a new registration procedure.

Optional IMS-AKA Configuration

The following configuration enables the Oracle Communications Unified Session Manager to specify, on a per-HSS basis, the number of authentication vectors it can download per MAR. Making this setting is not required as it has a valid default entry (3).

home subscriber server

To configure the number of authentication vectors to download from a home subscriber server (HSS):

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 path.

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

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

```
ORACLE(session-router)# home-subscriber-server
ORACLE(home-subscriber-server)#
```

4. **Select**—If already configured, choose the home subscriber server for which you want to set the number of authentication vectors.
5. **num-auth-vector**— [1-10] 3 default - The number of authentication vectors downloaded from HSS per MAR. The range is from 1-10 with 3 as the default.
6. Type **done** when finished.

S-CSCF Selection Based on Capabilities

Within IMS environments, the I-CSCF identifies target S-CSCF's in response to SIP traffic for which the assigned S-CSCF is not known. Enhanced selection environments can include the HSS offering mandatory and optional capabilities for a user, and the I-CSCF selecting the best S-CSCF based on capabilities the S-CSCF is best suited to support (in addition to standard criteria). The user can configure the I-CSCF resident within Oracle CSM, Oracle USM and Oracle SLRM to support this capabilities-based S-CSCF selection. Resultant operation is compliant with ETSI TS 129 228 and ETSI TS 129 229.

S-CSCF selection based on capabilities utilizes AVP information exchanged with the HSS to identify required and preferred capabilities on a per-user basis. Capabilities themselves vary widely. Examples include administrator routing preferences for divergent service types. Capabilities are manually defined at the HSS for endpoints or groups of endpoints. The Oracle CSM, Oracle USM and Oracle SLRM user configures tables on the I-CSCF that map the S-CSCF's with the capabilities they support. Further configuration enables the I-CSCF to make the best S-CSCF selection, then forward appropriately.

Diameter messaging that can generate capabilities parsing for S-CSCF selection includes UAR/UAA and LIR/LIA traffic. Inclusion of the capabilities AVPs in the message sequence triggers this enhanced S-CSCF selection by the I-CSCF.

Configuration on the HSS and the I-CSCF must be compatible in deployments that use this feature. Configuration required on the Oracle device performing the I-CSCF function includes:

- servers-capabilities-list—A sip-registrar parameter that allows you to configure the registrar with a servers-capabilities-table.
- servers-capabilities-table—A multi-instance element that names the table and includes multiple servers-capability.
  - servers-capability—A multi-instance element within the servers-capabilities-table that includes a capability (capability value associated with users and supported by servers in the list) and a server-name-list that identifies the servers that support this capability.

The OCUSM verifies the **servers-capabilities-list** attribute with the **servers-capabilities-table** each time it loads the configuration. If the **servers-capabilities-table** with the name specified in the **servers-capabilities-list** does not exist , the system outputs the following message:

ERROR: sip-registrar [<object-name>] has invalid servers-capabilities-list entry [<entry-name>]

Server-Capabilities AVP

The Server-Capabilities AVP is a group AVP including the Mandatory-Capability AVP and Optional-Capability AVP. The number of Mandatory-Capability and Optional-Capability AVPs is not limited in a Server-Capabilities AVP. The AVP symbol notation, format and reference follows:

3GPP 32.299 states the following symbols are used in the message format definitions:

- <AVP> indicates a mandatory AVP with a fixed position in the message.
- {AVP} indicates a mandatory AVP in the message.
- [AVP] indicates an optional AVP in the message.
- *AVP indicates that multiple occurrences of an AVP is possible.

Format definitions include:

- Server-Capabilities ::= <AVP header: 603 10415>
- *{Mandatory-Capability}
- *[Optional-Capability]
- *[Server-Name] (not supported in this release)
- *[AVP] (not supported in this release)

AVP reference, including column definition and AVP table follows:

- AVP Name
- AVP Number
- Reference where the AVP was defined
- Type of data format used to express the AVP's data
- If a grouped AVP, the names of the AVPs in the group

AVP	Number	Reference	Type	Grouped
{ Server-Capabilities }	603	Base	Grouped	Mandatory-Capability Optional-Capability
{ Mandatory-Capability }	604	Base	Unsigned32	
[Optional-Capability]	605	Base	Unsigned32	

Selection Process without SLRM

The capabilities-oriented S-CSCF selection algorithm on the Oracle CSM and Oracle USM S-CSCF include selections based on mandatory and optional capabilities information received from HSS and the configured S-CSCF Capabilities Database.

The general approach to selection within this scenario include the following principles:

- Only S-CSCFs with all mandatory capabilities can be selected.

- The process gives priority to the S-CSCF with the most optional capabilities.
- The process gives priority to the local S-CSCF.
- The system attempts to spread assignments to remote S-CSCFs of the same priority.

The capabilities-oriented S-CSCF selection algorithm uses the following high-level steps within the I-CSCF function to arrive at a selection:

1. Determine that the capabilities algorithm is required:
   a. No server-name in the LIA or UAA.
   b. Capability list exists.
   c. Assigned S-CSCF flag is not set.
   d. Mandatory/Optional Capabilities received in UAA/LIA.
2. Identify potential S-CSCFs, which must support all mandatory capabilities:
   a. Ensure the S-CSCF capabilities database is configured.
   b. Build capable S-CSCF list. This list contains all S-CSCFs from the S-CSCF capabilities database that support the Mandatory capabilities.
   c. Ensure that the capable S-CSCF list is not empty. If the capable S-CSCF list is empty, return an error to the UE.
3. Ensure that the I-CSCF is not SLRM.
4. Complete capabilities selection process using optional capabilities as criteria:
   a. An S-CSCF has the most optional capabilities.
   (If so, forward.)
   b. The local S-CSCF can take on more users, has all mandatory capabilities, and has most optional capabilities.
   (If so, forward locally.)
   c. Use round robin to select the S-CSCF that has most optional capabilities.
   (If so, forward.)
5. Forward message:
   a. Forward to selected S-CSCF.
   b. Remove selected S-CSCF from capabilities list.
   c. If there is an error, for example, the SIP response requires a re-assignment, check the assigned flag.
   d. If the assigned flag is set, return to the top.
   If the assigned is not set, return to the step that checks whether the capable S-CSCF list is empty.
   e. If the capable S-CSCF list is empty, return an error to the UE.
   If the capable S-CSCF list is not empty yet, perform capabilities selection process using optional capabilities as criteria again.

Selection Process with an SLRM

The capabilities-oriented S-CSCF selection algorithm on the Oracle SLRM uses standard Oracle CSM selection criteria in addition to capabilities criteria. This criteria includes cluster configuration, S-CSCF resource utilization and SLRM synchronization.

The general approach to selection within this scenario include the following principles:

- Only Oracle CSMs with all mandatory capabilities can be selected.
- The process gives priority to the Oracle CSMs in the cluster with the most optional capabilities, and is best able to take on new users.

The capabilities-oriented S-CSCF selection algorithm uses the following high-level steps, including the SLRM's selection steps, within the I-CSCF function to arrive at a selection:

1. Determine that the capabilities algorithm is required:
   a. No server-name in the LIA or UAA.
   b. Capability list exists.
   c. Assigned S-CSCF flag is not set.
   d. Mandatory/Optional Capabilities received in UAA/LIA.
2. Execute capabilities selection:
   a. Ensure the S-CSCF capabilities database is configured.
   b. Build capable S-CSCF list. This list contains all S-CSCFs from the S-CSCF capability database that support the Mandatory capabilities.
   c. Ensure that the capable S-CSCF list is not empty. If the capable S-CSCF list is empty, return an error to the UE.
3. Execute SLRM's selection procedure, cycle through all Oracle CSMs in the cluster:
   a. Identify applicable cluster. Begin to cycle through cluster.
   b. Determine whether Oracle CSM is in capable list.
   c. Determine whether Oracle CSM is at 100% utilization.
   d. Determine whether the next Oracle CSM support more optional capabilities.
   e. Determine whether the selected Oracle CSM is synchronized.
   f. Determine whether the next Oracle CSM using fewer resources.
4. Complete capabilities selection process using optional capabilities as criteria:
   a. An S-CSCF has the most optional capabilities.
   (If so, forward message.)
   b. The local S-CSCF can take on more users and has all mandatory capabilities and most optional capabilities.
   (If so, forward message locally.)
   c. Use round robin to select the S-CSCF that has most optional capabilities.
   (If so, forward message.)
5. Forward message:
   a. Forward to selected S-CSCF.
   b. Remove selected S-CSCF from capabilities list.
   c. If there is an error, for example, the SIP response requires a re-assignment, check the assigned flag.
   d. If the assigned flag is set, return to the top.
   If the assigned is not set, return to the step that checks whether the capable S-CSCF list is empty.

e. If the capable S-CSCF list is empty, return an error to the UE.
If the capable S-CSCF list is not empty yet, perform SLRM's selection procedure again.

ACLI Instructions

Configuring the server-capabilities-table

A **server-capabilities-table** is a multi-instance element that allows the user to name a **servers-capability** object and apply it to a **registrar**. A **servers-capability** object is a **server-capabilities-table** sub-element that includes a **capability** and multiple server names, which support that capability.

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 path.

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

3. Type **server-capabilities-table** and press Enter to access the path.

   ```
   ORACLE(session-router)# server-capabilities-table
   ORACLE(server-capabilities-table)#
   ```

4. Enter a contiguous string to the **name** field. This name is the reference used in the registrar configuration to specify the use of this server capabilities table.

5. Type **servers-capability** and press Enter to access the path.

   ```
   ORACLE(server-capabilities-table)# servers-capability
   ORACLE(servers-capability)#
   ```

6. Enter a number to specify the capability **capability**. Valid entries range from 0 to 999999999.

7. Enter the names of the servers that belong to this **server-name-list**. Name format is the same as that used within the registrar's **home-server-route** field. The format is the URI (containing FQDN or IP address) used to identify a server to the HSS. Each entry in the list is enclosed with quotes and separated by comma.

8. Type **done** and **exit** twice to complete configuration of this **server-capabilities-table** configuration element.

Configuring the server-capabilities-list

To assign a server capabilities list to a sip-registrar:

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 path.

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

3. Type **sip-registrar** and press Enter to access the session router path.

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

4. Type **server-capabilities-list** and press Enter. Add a capability with associated servers.

```
ORACLE(sip-registrar)# server-capabilities-list my_capability_list1
ORACLE(sip-registrar)#
```

5. Type **done** and **exit** to complete configuration of this **sip-registrar** configuration element.

Oracle Communications Unified Session Manager as Registrar

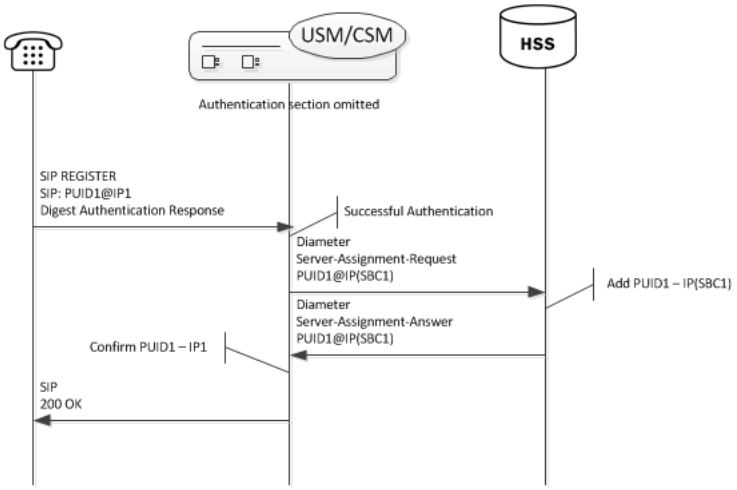
Creating a sip registrar configuration element enables the Oracle Communications Unified Session Manager to act as a SIP registrar. When registration functionality is enabled, the Oracle USM actually registers endpoints rather than only caching and forwarding registrations to another device. Oracle Communications Unified Session Manager registry services are enabled globally per domain, not on individual SIP interfaces or other remote logical entities.

On receiving a REGISTER message, the Oracle Communications Unified Session Manager checks if it is responsible for the domain contained in the Request-URI as defined by the domains parameter and finds the corresponding sip registrar configuration. This is a global parameter—all messages are checked against all sip registrar domains. Thus you could create one sip registrar configuration element to handle all .com domains and one sip registrar configuration element to handle all .org domains. The Oracle Communications Unified Session Manager begins registrar functions for all requests that match the configured domain per sip-registrar configuration element.

A UA is considered registered once a SAA assignment is received from the HSS, after which the Oracle Communications Unified Session Manager sends a 200 OK message back to the registering UA.

New Registration

The following image shows a simplified call flow for a registering user:



Registration Response with the Authentication-info Header

The Oracle Communications Unified Session Manager can include the authentication-info header, as described in RFC 2617, in its 200 OK response to REGISTERs when using SIP digest. The user enables this functionality using a **sip-registrar** option.

By default, the Oracle Communications Unified Session Manager supports registration with SIP digest authentication without using the authentication-info header. This is not compliant with TS 24.229. Enabling the **add-auth-info** option causes the Oracle Communications Unified Session Manager to calculate and insert the required authentication-info header fields in the 200 OK.

The Oracle Communications Unified Session Manager also presents this authentication header during third party registrations. The system includes the entire 200OK message in the third party registration request.

This authentication state is not shared across high availability nodes. The user can expect the Oracle Communications Unified Session Manager to request re-authentication by registering UEs after failover to a backup Oracle Communications Unified Session Manager.

Authentication-Info header field parameters sent by the Oracle Communications Unified Session Manager include:

- **qop**—Matches the **qop** sent by the UE
- **rspauth**—A response-digest calculated as described in RFC 2617
- **cnonce**—Matches the **cnonce** sent by the UE
- **nonce-count**—Matches the **nonce-count** sent by the UE

The **nextnonce** authentication-info header field parameter, which can request a new nonce for subsequent authentication responses from the UE, is not implemented on the Oracle Communications Unified Session Manager.

The ACLI syntax for enabling the **add-auth-info** option follows.

```
ORACLE(sip-registrar)#+options=add-auth-info enabled
```

The Oracle Communications Unified Session Manager provides NOTICE level log entries in **log.sipd** to indicate this option's status.

Handling Barred PUIDs

The Oracle Communications Unified Session Manager supports PUID barring functionality per 3GPP specification TS 24.229. As such, the system does not service any request method other than REGISTERs for SIP or Tel-URI PUIDs designated as barred by the HSS. The Oracle Communications Unified Session Manager also complies with the requirement that it allow Push Profile Requests (PPRs) to change a PUID from barred to non-barred (and vice versa) and issues a NOTIFY of the event to subscribers. No configuration is required.

A common use case for barring information is a cell phone registering with a temporary PUID (that is barred), along with a set of non-barred PUIDs in the P-Associated User (PAU) header. After registration, the cell phone should use only the non-barred PUIDs for all ensuing methods and its contacts.

An HSS should be configured with barring information for all PUIDs. During registration procedures, the HSS provides this information to the S-CSCF. PUID information in the User Data AVP of the Diameter SAA includes a tag indicating whether the PUID is barred. The Oracle Communications Unified Session Manager retains this information in the registration cache. To complete the registration, the Oracle Communications Unified Session Manager replies to the UE with a list of all non-barred PUIDs in the 200OK. For all the further procedures, the UE should use a PUID from the non-barred P-Associated-URI list. If the HSS does not identify a PUID's barring status, the Oracle Communications Unified Session Manager assumes it is not barred.

Typical Oracle Communications Unified Session Manager behaviors related to barring include:

- Responds to ensuing requests from barred PUIDs with (403) Forbidden.
- Responds to requests that have no PSU, but include barred PUIDs in their PAI header list with 403 (Forbidden).
- Responds to requests to or from wildcarded PUIDs that match barred PUIDs with 403 (Forbidden).
- Responds to registration attempts that have all barred implicit identities with 403 (Forbidden).
- Responds to requests for termination services wherein the served user (PSU/RURI) is barred with (404) Not Found.
- Recognizes barring status during third party registration procedures and does not attempt to register a barred PUID to an AS.
- Handles related subscription scenarios as follows:
  - When receiving a subscription from a barred subscriber, responds with 403 (Forbidden).
  - When receiving a subscription for a barred user, allows the SUBSCRIBE to proceed.
  - Does not include a barred identity in any NOTIFY.
    * When receiving a subscription for a user that has barred identities in its implicit set, issues NOTIFYs that only include non-barred identities.
    * Includes only non-barred PUIDs in NOTIFY messages generated by network-initiated re-registration and authorization requests.

> **Note:**
>
> The Oracle Communications Unified Session Manager does not support any PUID barring within the context of GRUU.

The user can verify PUID barring status using the **show reg sipd by-user <user> detailed** command. Example output is shown below.

```
ORACLE# show reg sipd by-user user detailed
Registration Cache (Detailed View)    Thu Jul 09 2015  15:16:08
User: sip:user_1@acme-ims.com
  Registered at:  2015-07-09-15:16:04    Surrogate User: false
  Emergency Registration? No
  ContactsPerAor Rejects 0
  ContactsPerAor OverWrites 0

  Contact Information:
    Contact:
      Name: sip:user_1@acme-ims.com
      Valid: true

...

    Associated URI(s):
      URI: sip:user_1@acme-ims.com
      Status: Barred
```

...

> **Note:**
>
> The Oracle Communications Unified Session Manager replicates barred status for PUIDs to standby systems.

Releasing Unregistered Users

When a call arrives at an Oracle Communications Unified Session Manager either to or from a user that is not registered at that Oracle Communications Unified Session Manager, it performs a location query with the HSS to determine if the unknown UE is registered at another S-CSCF. If there is no registration, the Oracle Communications Unified Session Manager takes ownership of the UE. The system stores information about these UEs in its registration cache, labelled "NEVER REGISTERED". Barring any further, related action within the infrastructure, the UE would remain homed to the Oracle Communications Unified Session Manager. Upon expiry of this feature's timer, the Oracle Communications Unified Session Manager sends an SAR to the HSS, providing an assignment type of ADMINISTRATIVE_DEREGISTRATION for the UE. This allows the UE to be a user at a different S-CSCF the next time it is a call sender or receiver. A common use case for this scenario is a roaming UE.

When the Oracle Communications Unified Session Manager issues the SAR, it also marks the UE as 'dirty' (in the process of being de-assigned) to accommodate the following operational scenarios:

- The UE attempts to register—The Oracle Communications Unified Session Manager rejects the register, replying with a 504 error message.
- The UE has existing calls—The Oracle Communications Unified Session Manager continues to support the call, based on a stored copy of the service profile.
- A new call arrives—The Oracle Communications Unified Session Manager rejects the call. The Oracle Communications Unified Session Manager replies with a '480, Temporarily Unavailable' error message if the UE is the callee; the Oracle Communications Unified Session Manager responds with a 504 if the UE is the caller.

The user can configure the **unreg-cache-expiry** parameter in seconds on a per-registrar basis. This syntax is shown below.

```
ORACLE(sip-registrar)# unreg-cache-expiry 120
```

The parameter accepts values in the range of 0 to 604800, with 0 specifying that the Oracle Communications Unified Session Manager does not cache unregistered users. A setting of 0 means the Oracle Communications Unified Session Manager takes ownership, downloads service profiles, and then releases the user after the call without caching.

Handling Public Service Identities (PSIs)

Public Service Identities (PSI) appear as unregistered users in the Oracle Communications Unified Session Manager. PSIs appear as either Distinct PSIs or Wildcarded PSIs. Similar to unregistered users, the Oracle Communications Unified Session Manager takes ownership of the PSI if it is unassigned and a call is made to or from it. By default, PSIs are not released. However, the user can configure the **psi-cache-expiry** option in seconds on a per-registrar basis

to cause the Oracle Communications Unified Session Manager to release PSIs. This syntax is shown below.

```
ORACLE(sip-registrar)# options psi-cache-expiry=120
```

Configurable Response to Timed-Out OPTIONS Messages

The Oracle Communications Unified Session Manager allows the user to configure a function by which they can cause the system to send a 408 as a response to an OPTIONS message sent to an un-responsive, registered called party. In addition, this function allows the user to specify when to send that 408.

By default, the Oracle Communications Unified Session Manager does not send messages to an originating node when OPTIONS transactions time out. This complies with RFC 4321.

When registered users do not respond to OPTIONS requests, the network never informs the calling party of the called party's status. Instead, the calling party waits for the standard 32-second retry timeout to expire. If the called party was previously reachable, the calling party treats it as reachable for the entire 32-second window.

The Oracle Communications Unified Session Manager includes a configuration option that:

- Starts a timer when the system forwards an applicable OPTIONS message and,
- Upon expiry of that timer, causes the system to send a 408 message to the calling party.

This option allows the network administrator to provide the calling party with this 408 response, and specify a shorter interval between request and response.

This feature works for:

- A called party that is registered via its P-CSCF, but not currently reachable.
- A called party that is reachable via an IBCF or BGCF.

This function has no impact on requests that result in a response, such as SIP 480, for un-registered subscribers.

For registered users with multiple contacts, the Oracle Communications Unified Session Manager uses a response from any contact as a trigger to stop the timer and not send a 408. The Oracle Communications Unified Session Manager cancels all remaining OPTIONS transactions when it receives a response from a contact. In addition, if the system used parallel forking to reach multiple contacts, it waits for the timer expiry before it sends the 200OK to the caller.

The option is available via S-CSCF processing and, as such, is available on both the Oracle USM and Oracle CSM products. There is, however, one operational difference between the Oracle USM and Oracle CSM. If the called party finally responds after this timer expires and the S-CSCF logic has sent the 408, the Oracle USM drops the response, whereas the Oracle CSM forwards it to the originating node.

The user sets the option globally in **sip-config** or on a **sip-interface**, with the **sip-interface** taking precedence. Values range from 1 to 32 seconds. Invalid ranges cause the system to use the maximum value of 32. The example below sets a sip-interface's timer to 4 seconds.

```
ORACLE(session-router)#sip-interface
ORACLE(sip-interface)#options +options-408-timeout=4
```

Option syntax on the **sip-config** and **sip-interface** configuration elements is the same.

The user must consider the infrastructure carefully. Setting the value too low can cause an inordinate number of invalid 408 responses.

Limiting REGISTER CDR Generation

The Oracle Communications Unified Session Manager allows the user to generate RADIUS CDRs for REGISTER events via configuration. Large networks, however, can generate an inordinate volume of CDRs. So the Oracle Communications Unified Session Manager also allows the user to reduce REGISTER CDR generation by filtering out some of the messages it sends.

When the user enables accounting with the generate-events parameter, the Oracle Communications Unified Session Manager can generate CDRs for the following register and/or local register events:

- Initial REGISTER
- REGISTER refresh
- REGISTER update
- de-REGISTER

Depending on the event, the system generates per-contact start, interim and/or stop CDRs. With no other configuration, the system generates the appropriate CDRs for all of these events.

The user can prevent the system from issuing some CDR via an **account-config** option that filters, as described below, and sets a timer that restarts the CDR suppression window. Use the syntax below to set this **register-cdr-interval** option with an expiry timer value of 43200 in minutes (30 days), and limit the number of generated CDRs as described below.

```
(account-config)#options +register-cdr-interval=43200
```

When configured with this option, the Oracle Communications Unified Session Manager limits the generation of CDRs for each user as follows:

1. Send a START CDR for first Register message (for first contact).
2. Don't send CDRs until the user specified time period expires. After it expires, when a Registration message causes a 'START' or 'INTERIM' CDR event to occur, send it. Then, re-set the time value. Applicable 'START' CDR events include:
   - Add new contact
   - Replace contact
   - Overwrite contact

The applicable 'INTERIM' CDR event is a Refresh Contact.

The **generate-event** parameter must also be set to **register**.

Limiting AOR Contacts

The Oracle Communications Unified Session Manager allows you to limit the number of contacts that apply to AORs. It also provides a configurable behavior allowing the system to either reject a new contact or overwrite an existing contact with the new one. The user specifies the maximum number of contacts and the operation mode on a per-registrar basis. Alternatively, the user can disable the feature. This feature is applicable to Cx and local database deployments.

The value for **max-contacts-per-aor** ranges from 0-256. A value of 0 disables the function. When **max-contacts-per-aor** is greater than zero, the Oracle Communications Unified Session Manager tracks the number of contacts registered per AOR. Settings for **max-contacts-per-aor-mode** include REJECT and OVERWRITE.

If you change the configured maximum while the system is operational, your setting only applies to new registrations. If there are more contacts than your newly configured maximum, the system removes older contacts. This ensures that the contacts are always within the configured maximum.

Both **max-contacts-per-aor** and **max-contacts-per-aor-mode** are RTC supported.

Maximum Contacts REJECT Mode

If the Oracle Communications Unified Session Manager receives a registration request that exceeds the maximum that you configured, it responds with a local response, a 403 Forbidden by default, and does not register the additional contact. The system only rejects registration requests that exceed the maximum. Existing contacts persist normally.

Maximum Contacts OVERWRITE Mode

If the number of contacts in the initial registration exceeds the maximum, the Oracle Communications Unified Session Manager selects only the highest priority contact based on q-values. If there are no q values, the Oracle Communications Unified Session Manager adds contacts in the order they appear in the REGISTER message until it reaches the maximum. The system then identifies the oldest contacts for overwriting using the last registered time stamp.

In all cases, the Oracle Communications Unified Session Manager follows this procedure to remove old contacts:

1. If reg-id/instance-id is present in the contact, the system simply updates the contact.
2. The system sends NOTIFY messages to the subscriber for whom the contact has been removed with a status of "terminated" and "de-activated" as the reason.
3. The system removes the contact from the registration cache.

HSS Server Assignment

As the Oracle Communications Unified Session Manager registers UAs, it requests to assign itself as the S-CSCF for the registering AoR. The Oracle Communications Unified Session Manager's S-CSCF identity is configured in the home-server-route parameter in sip-registrar configuration element. This is a entered as a SIP URI (containing FQDN or IP address) and is used to identify and route messages to this Oracle Communications Unified Session Manager on behalf of the registered user.

Server Assignment Messages

The Oracle Communications Unified Session Manager sends a Server Assignment Request (SAR) to the HSS requesting to confirm the SIP or SIPS URI of the SIP server that is currently serving the user. The SAR message also serves the purpose of requesting that the Diameter server send the user profile to the SIP server. The SAR's AVPs are populated as follows:

- Public-User-Identity—the SIP AOR of the endpoint being registered (same as UAR)
- Private-User-Identity—the username from the SIP authorization header, if it is present. If not, this value is the public User ID. (Same as UAR)

- Server-Name—the home server route parameter in the sip-registrar configuration element. It is the FQDN or IP address used to identify and route to this Oracle Communications Unified Session Manager sent as a URI.
- Server-Assignment-Type—the value of this attribute depends upon the registration state:
  - REGISTRATION (1)—for all new and refreshing registrations.
  - Set to TIMEOUT_DEREGISTRATION (4)—when the contact is unregistered due to expiration. This occurs if the force-unregistration option is configured in the sip config.
  - USER_DEREGISTRATION (5)—when the contact is unregistered by the user (contact parameter expires=0).
- User-Data-Already-Available—always set to USER_DATA_ALREADY_AVAILABLE (1)

Server-Assignment-Response

The Oracle Communications Unified Session Manager expects a DIAMETER_SUCCESS code in the SAA to indicate that the assignment was successful. Then a 200 OK response is returned to the registering user. Any other Diameter result code is an error and results in an error response for the original REGISTER request (by default 503) and the contacts to be invalidated in the registration cache.

Register Refresh

When a UA sends a register refresh, the Oracle Communications Unified Session Manager first confirms that the authentication exists for that UE's registration cache entry, and then is valid for the REGISTER refresh. (If a valid hash does not exist for that AoR, then the Oracle Communications Unified Session Manager sends an MAR to the HSS to retrieve authentication data once again).

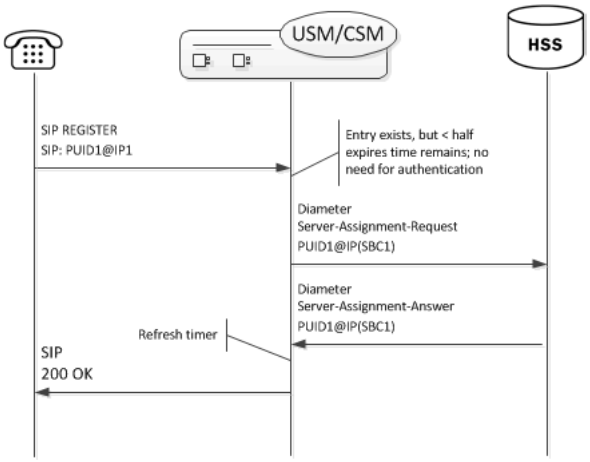
Next, the Oracle Communications Unified Session Manager determines if the it can perform a local REGSITER refresh or if the HSS needs to be updated. If any of the following 3 conditions exists for the re-registering UA, the Oracle Communications Unified Session Manager updates the HSS:

- The location update interval timer has expired—This value, configured in the sip registrar configuration element ensures that HSS database always has the correct Oracle Communications Unified Session Manager address by periodically sending SARs for each registered contact.
- The message's call-id changes while the f**orward-reg-callid-change** option in the sip config configuration element is set. This covers the case where the UA changes the Oracle Communications Unified Session Managers through which it attaches to the network.
- The REGISTER message's Cseq has skipped a number. This covers the case in which a user registered with Oracle Communications Unified Session Manager1, moves to Oracle Communications Unified Session Manager2 , and then returns to Oracle Communications Unified Session Manager1.

If the Oracle Communications Unified Session Manager updates the HSS database because of matching one of the above conditions, the access side expiration timer per contact is reset to the REGISTER message's Expires: header value, and returned in the 200 OK. This happens even in the case when the reREGISTER was received in the first half of the previous Expires period. In addition, the core-side location update interval timer are refreshed on both active and standby.

When the above three conditions are not met, the registration expiration proceeds normally.

If the timer has not exceeded half of its lifetime, a 200 OK is returned to the UA. If the timer has exceeded half of its lifetime, the Oracle Communications Unified Session Manager just refreshes the access-side expiration timer; the registration cache expiration timer for that AoR begins its count again.



Core-side SAR Lifetime

The Oracle Communications Unified Session Manager maintains a timer for user registrations per SAR on the core side as specified above. The core-side SAR lifetime timer is configured in the location update interval parameter in the sip registrar configuration element. This timer ensures that the HSS always has the correct Oracle Communications Unified Session Manager address, by sending SAR messages periodically.

Entry Unregistration

Because AoRs and not contacts are referenced by the HSS, an AoR is valid and should not be removed from HSS until all associated contacts have been removed or expired. If all the contacts are removed for an AoR by receiving REGISTER messages with Expires:0 header, then the SAR sent to the HSS includes Server-Assignment-Type of USER_DEREGISTRATION (5).

When the `force-unregister` option in the `sip config` is enabled, then the HSS is explicitly updated when all of the contacts for an AoR have expired. This event prompts the Oracle Communications Unified Session Manager to send a SAR to the HSS using the Server-Assignment-Type of TIMEOUT_DEREGISTRATION (4).

The HSS can send a Registration-Termination-Request to request removing a registration, which corresponds to entries in the Oracle Communications Unified Session Manager's registration cache. When an RTR is received, the following AVPs are expected:

- Private-User-Identity—Username of the user, which is being de-registered.
- Associated-Identities—The Private-Id's in the same subscription which need to be de-registered. (optional)
- Public-Identity—One or more public-Id's of the user being de-registered. (optional)

For the AoR specified by the Private-User-Identity AVP, all associated contacts are removed in the registration cache. The Oracle Communications Unified Session Manager sends a Registration Termination Answer to the HSS to indicate success.

Diameter Message Manipulations

The Oracle Communications Unified Session Manager can perform manipulations on all grouped and non-grouped AVPs. This is referred to as Diameter Manipulation Rules (DMR). A message manipulation is the ability to search for a predefined string within an AVP and then replace it with another value. This is similar to the Oracle Communications Unified Session Manager's header manipulation rules functionality.

A diameter manipulation configuration element is defined by a name parameter. You can optionally add a description field to the diameter manipulation. Within each diameter manipulation you can configure multiple diam manipulation rule subelements. The manipulation rule subelements are the configuration where AVPs are identified, searched, and in which the data is replaced.

The Oracle Communications Unified Session Manager supports diameter manipulation across the Cx interface, with the user configuring these manipulations to home subscriber server configurations.

> **Note:**
>
> The user can also apply diameter manipulations to external policy server configurations. These manipulations affect traffic between the Oracle Communications Unified Session Manager and the applicable policy server. The range of manipulation supported over the Rx interface is the same as that over the Cx interface.

See a full explanation on diameter manipulation, including configuration instructions, in the *External Policy Servers* chapter of this document.

User Registration based on Reg-ID and Instance-ID (RFC 5626)

Sometimes a user's device reregisters from a different network than its original registration. This event should be considered a location update rather that a completely new registration for the Contact. The Oracle Communications Unified Session Manager can perform this way by considering the endpoint's reg-id and instance-id parameters defined in RFC 5626.

The Oracle Communications Unified Session Manager identifies new REGISTER requests received on a different access network as a location update of the existing binding between the Contact and AoR. Without this feature, the Oracle Communications Unified Session Manager would create a new binding and leave the old binding untouched in the local registration cache/ ENUM database. This scenario is undesirable and leads to unnecessary load on various network elements including the Oracle Communications Unified Session Manager itself.

The following conditions must be matched to equate a newly registering contact as a location update:

For a received REGISTER:

- The message must not have more than 1 Contact header while 1 of those Contact headers includes a reg-id parameter. (failure to pass this condition prompts the Oracle Communications Unified Session Manager to reply to the requester with a 400 Bad Request).
- The Supported: header contains **outbound** value
- The Contact header contains a **reg-id** parameter
- The Contact header contains a **+sip.instance** parameter

After these steps are affirmed, the Oracle Communications Unified Session Manager determines if it is the First hop. If there is only one Via: header in the REGISTER, the Oracle Communications Unified Session Manager determines it is the first hop and continues to perform Outbound Registration Binding processing.

If there is more than 1 Via: header in the REGISTER message, the Oracle USM performs additional validation by checking that a Path: header corresponding to the last Via: includes an ob URI parameter, Outbound Registration Binding may continue.

If the Oracle Communications Unified Session Manager is neither the first hop nor finds an ob URI in Path headers, it replies to the UA's REGISTER with a 439 First Hop Lack Outbound Support reply.

reREGISTER Example

The user (AoR) bob@example.com registers from a device +sip.instance= <urn:uuid:0001> with a reg-id ="1", contact URI = sip:1.1.1.1:5060. A binding is be created for bob@example.com+<urn:uuid:0001>+reg-id=1 at sip:1.1.1.1.:5060.

Next, Bob@example.com sends a reREGISTER with the same instance-id but with a different reg-id = 2 and contact URI = sip:2.2.2.2:5060.

The previous binding is removed. A binding for the new contact URI and reg-id is created. bob@example.com+<urn:uuid:0001>+reg-id=2 at sip:2.2.2.2:5060

Outbound Registration Binding Processing

An outbound registration binding is created between the AoR, instance-id, reg-id, Contact URI, and other contact parameters. This binding also stores the Path: header.

Matching re-registrations update the local registration cache as expected. REGISTER messages are replied to including a Require: header containing the outbound option-tag.

If the Oracle Communications Unified Session Manager receives requests for the same AOR with some registrations with reg-id + instance-id and some without them, the Oracle Communications Unified Session Manager will store them both as separate Contacts for the AOR; The AoR+sip.instance+reg-id combination becomes the key to this entry.

Wildcarded PUID Support

The Oracle Communications Unified Session Manager supports the use of wildcarded Public User IDs (PUIDs), typically for registering multiple endpoints on a PBX with a single PUID. A wildcard is composed of a regular expression that, when used in a PUID prefix, represents multiple UEs. The group of UEs is referred to as an implicit registration set and share a single service profile. This support is typically implemented to reduce HSS resource requirements. The regular expressions themselves are in form of Perl Compatible Extended Regular Expressions (PCRE).

Each implicit registration set is associated with an explicitly registered distinct PUID. Typically, this distinct PUID is the PBX itself. The implicit registration set is dependent on the distinct PUID, including the distinct PUID's registration status.

There is no Oracle Communications Unified Session Manager configuration required.

Wildcarded PUID support is applicable to both I-CSCF and S-CSCF operation. In addition, all Oracle Communications Unified Session Managers in the applicable data paths must be in the same trust domain.

To allow the feature, the Oracle Communications Unified Session Manager supports:

- Wildcarded PUID AVP in the LIR, SAR and SAA
- User Profile AVP in the SAA
- P-Profile-Key across the Mw interface, as defined in RFC 5002

Note also that the HSS must support the wildcarded-public-Identify AVP.

ACLI Instructions

The following configuration enables the Oracle Communications Unified Session Manager to authorize and authenticate registering users. In addition it sets the Oracle Communications Unified Session Manager to request itself as the S-CSCF for the registering users.

home subscriber server

To configure a home subscriber server (HSS):

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 path.

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

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

   ```
   ORACLE(session-router)# home-subscriber-server
   ORACLE(home-subscriber-server)#
   ```

4. **name**—Enter the name for this home subscriber server configuration element to reference from other configuration elements.
5. **state**—Set this to **enabled** to use this configuration element.
6. **address**—Enter the IP address of this HSS. Both IPv4 and IPv6 addressing is supported.
7. **port**—Enter the port which to connect on of this HSS, the default value is 80.
8. **realm**—Enter the realm name where this HSS exists.
9. Type **done** when finished.

SIP Authentication Profile

To configure the SIP Authentication Profile:

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 path.

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

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

```
ORACLE(session-router)# sip-authentication-profile
ORACLE(sip-authentication-profile)#
```

You may now begin configuring the SIP Authentication Profile configuration element.

4. **name**—Enter the name of this SIP authentication profile that will be referenced from a SIP registrar (or a SIP interface) configuration element.
5. **methods**—Enter all the methods that should be authenticated. Enclose multiple methods in quotes and separated by commas.
6. **anonymous-methods**—Enter the methods from anonymous users that require authentication. Enclose multiple methods in quotes and separated by commas.
7. **digest-realm**—Leave this blank for Cx deployments.
8. **credential-retrieval-method**—Enter CX.
9. **credential-retrieval-config**—Enter the home-subscriber-server name used for retrieving authentication data.
10. Type **done** when finished.

SIP Interface

The full SIP interface should be configured according to your network needs. Please refer to the Oracle SBC ACLI Configuration Guide.

To configure a SIP Digest Authentication on a specific SIP Interface:

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 path.

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

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

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

4. Type **select** and choose the number of the pre-configured sip interface you want to configure.

```
ORACLE(sip-interface)# select
<realm-id>:
1: private 192.168.101.17:5060
2: public  172.16.101.17:5060
selection: 1
```

5. **registration-caching**—Set this parameter to **enabled.**
6. **ims-access**—Set this parameter to **enabled** for access interfaces, when applicable. Core interfaces should have this feature disabled.

7. **sip-authentication-profile**—Set this to the name of an existing sip-authentication profile if you wish to authenticate per SIP interface.
8. Type **done** when finished.

SIP Registrar

To configure the Oracle Communications Unified Session Manager to act as a SIP Registrar:

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 path.

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

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

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

4. **name**—Enter a name for this SIP registrar configuration element.
5. **state**—Set this to **enabled** to use this SIP registrar configuration element.
6. **domains**—Enter one or more domains that this configuration element will invoke SIP registration for. Wildcards are valid for this parameter. Multiple entries can be entered in quotes, separated by commas.
7. **subscriber-database-method**—Set this to **CX**.
8. **subscriber-database-config**—Enter the home-subscriber-server configuration element name that will handle REGISTER messages for this domain. The HSS configuration element includes the actual IP address of the server that SAR's are sent to.
9. **authentication-profile**—Enter a sip-authentication-profile configuration element's name. The sip authentication profile object referenced here will be looked up for a REGISTER message with a matching domain in the request URI. You may also leave this blank for the receiving SIP Interface to handle which messages require authentication if so configured.
10. **home-server-route**—Enter the identification for this Oracle Communications Unified Session Manager that will be sent as the Server-Name in MAR and SAR messages to the HSS. This value should be entered as a SIP URI.
11. **location-update-interval**—Keep or change from the default of 1400 minutes (1 day). This value is used as the timer lifetime for core-side HSS updates.
12. Type **done** when finished.

Maximum Number of Contacts

To configure a sip-registrar with a maximum of 10 contacts per AOR and a mode of overwrite:

1. From superuser mode, use the following command sequence to access sip-registrar element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# sip-registrar
   ORACLE(sip-registrar)# select
   ```

Select the registrar you want to configure.

2. Specify the number of contacts.

```
ORACLE(sip-registrar)# max-contacts-per-aor 10
ORACLE
```

3. Specify the contact mode to overwrite.

```
ORACLE(sip-registrar)# max-contacts-per-aor-mode overwrite
ORACLE
```

4. Type **done** and **exit** to complete configuration of this **sip-registrar** configuration element.

Response to Exceeding Maximum Contacts

To configure local response for the Oracle Communications Unified Session Manager to issue when max-contacts-per-aor is exceeded:

1. From superuser mode, use the following command sequence to access local-response and add an entry.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# local-response-map
```

2. Access the entries configuration.

```
ORACLE(local-response-map)# entries
```

3. Specify the local error you need to configure.

```
ORACLE(local-response-map-entry)# local-error contacts-per-aor-exceed
```

4. Specify the sip-reason for this error.

```
ORACLE(local-response-map-entry)# sip-reason forbidden
```

5. Specify the error code for this error.

```
ORACLE(local-response-map-entry)# sip-status 403
ORACLE(local-response-map-entry)# done
local-response-map-entry
        local-error                     contacts-per-aor-exceed
        sip-status                      403
        q850-cause                      0
        sip-reason                      forbidden
        q850-reason
        method
        register-response-expires
ORACLE(local-response-map-entry)# exit
```

SIP Registration Event Package Support

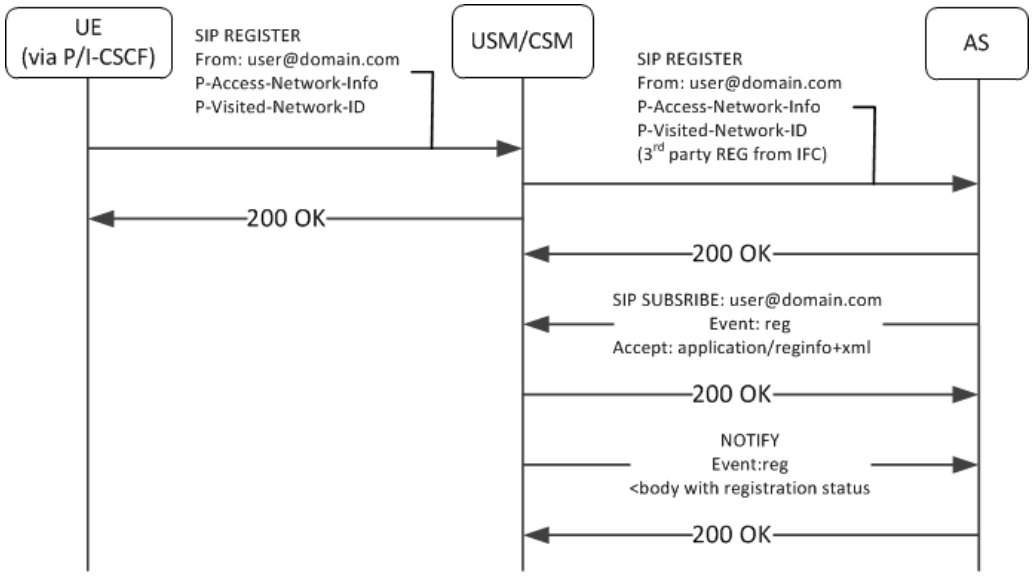
The Oracle Communications Unified Session Manager supports UA subscriptions to the registration event package, as defined in RFC3680. As such, it maintains contact with entities, often application servers, that need to know about UA registration events and provides those application servers with notifications when registration events occur.

Common usage for this functionality includes:

- Forcing re-authentication

- The provision of welcome notices to users who need information or instructions customized to their location

An operational example, shown below, begins with the Oracle Communications Unified Session Manager performing 3rd party registration on behalf of a UA to an AS, based on the iFC request from the UA. The AS, being an appropriately authorized UA itself, subscribes to NOTIFY messages on reg events for the initial UA. The Oracle Communications Unified Session Manager sends a 200OK to the AS, and then proceeds to forward NOTIFY messages about that UE's registration events to the AS.



This feature is relevant when the Oracle Communications Unified Session Manager is performing S-CSCF functions. You enable this feature on the Oracle Communications Unified Session Manager per registrar, by simply creating a profile and applying it to the applicable registrar.

SUBSCRIBE Processing

When the Oracle Communications Unified Session Manager has the reg-event notification function enabled for a registrar, it includes the allow-events header in its 200OK replies to successful REGISTERS. This lets UEs know that they can subscribe to registration event packages.

When the Oracle Communications Unified Session Manager receives reg-event subscription requests, it follows the sequence below to process SUBSCRIBE requests for reg events:

1. Determines validity of the request.

   Subscriptions cannot include more than one package name. If there is more than one package name in the request, the Oracle Communications Unified Session Manager replies with a 400 Bad Request message.

2. Determines if it can be a notifier, as follows:
   - The SUBSCRIBE must include EVENT=reg.
   - The requesting UA must be in the same domain as the registrar.

If both of the above are true, the Oracle Communications Unified Session Manager proceeds with the request.

3. Authorizes the request. The Oracle Communications Unified Session Manager only authorizes requests from UEs that come from the same realm and layer 2 connection on which it received the initial REGISTER.
   Furthermore, the Oracle Communications Unified Session Manager only authorizes the following UEs:
   - Public user identities from UEs that are subscribing to their own registration events.
   - Public user identities that this user owns. Examples include implicitly registered public user identities.
   - Entities that were included in the PATH header of the target UE's registration.
   - All ASs that are listed in the UE's iFC and that are part of the trust domain.

   If all of the above are true, the Oracle Communications Unified Session Manager proceeds with the request. If not, it sends 403 Forbidden to the requester.
4. Determines how it is functionally related to the UA. The Oracle Communications Unified Session Manager only processes subscriptions for users in its registration cache, replying with a 403 Forbidden if not. For cached users, the Oracle Communications Unified Session Manager forwards the request to the registrar if it is the P-CSCF. If it is the S-CSCF, it sends a 200 OK and begins to act as notifier.
5. Identifies the subscription duration, as follows, and sends the 200 OK to the UE:

   If there is no Expires header in the UE's 200OK message, the Oracle Communications Unified Session Manager applies it's own configured minimum or the default (600000 seconds), whichever is greater.

   If the SUBSCRIBE includes an Expires header, the Oracle Communications Unified Session Manager honors the request unless it is less than the configured minimum.

   If the SUBSCRIBE's Expires header is less than the minimum subscription time configured in the registration event profile, the Oracle Communications Unified Session Manager denies the subscription, sending a 423 Too Brief message.

   When the Oracle Communications Unified Session Manager encounters an Expires header set to 0, it terminates the subscription. This is referred to as unsubscribing.

SUBSCRIBE REFRESH Requests

Subscriptions must be refreshed to keep them from expiring. ASs accomplish this by sending SUBSCRIBE REFRESH messages to the Oracle Communications Unified Session Manager. Messages must be received from authorized subscribers and on the same realm and connection as the original SUBSCRIBE or the Oracle Communications Unified Session Manager rejects the refresh request.

Reg Event NOTIFY Messages

When configured, the Oracle Communications Unified Session Manager issues NOTIFY messages to subscribed ASs when significant registration events occur. NOTIFY messages sent by the Oracle Communications Unified Session Manager comply fully with RFC3680. Events that trigger NOTIFY messages include:

- Registered
- Registration refreshed

- Registration expired
- Registration deactivated
- UE unregistered

The Oracle Communications Unified Session Manager does not send NOTIFY messages for the following events:

- Registration created
- Registration shortened
- Registration probation
- Registration rejected

Additional detail about NOTIFY messages that is specific to the Oracle Communications Unified Session Manager includes:

- The Oracle Communications Unified Session Manager always sends full information on all contacts, and indicates such within the reginfo element. The Oracle Communications Unified Session Manager does not utilize the partial state described within RFC 3680.
- Wildcarded PUIDs are included, enclosed in the <wildcardedIdentity> tag within the <registration> element.
- The Oracle Communications Unified Session Manager does not include the following optional attributes within the contact element:
  - expires
  - retry-after
  - duration registered
  - display-name
- The Oracle Communications Unified Session Manager uses the optional unknown-param element within the contact element to convey UA capabilities and distribute reg-id, sip.instance and header filed attributes.

An example of the XML body of a NOTIFY message below documents the registration status for the AOR joe@example.com.

```
<reginfo xmlns="urn:ietf:params:xml:ns:reginfo"   xmlns:xsi=http://www.w3.org/
2001/XMLSchema-instance version="0" state="full">
  <registration aor="sip:joe@example.com" id="as9" state="active">
    <contact id="6" state="active" event="registered">
      <uri>sip:joe@pc887.example.com</uri>
    </contact>
      <contact id="7" state="terminated" event="expired">
      <uri>sip:joe@university.edu</uri>
    </contact>
  </registration>
</reginfo>
```

Use the show registration and show sipd subscription commands to display all information about each subscription.

Reducing NOTIFY Traffic

RFC 3265 stipulates that the Subscription server sends NOTIFY messages to all subscribers when a UA sends a registration refresh. This can generate excessive NOTIFY traffic. You,

however, can mitigate this by configuring the Oracle Communications Unified Session Manager to limit notification traffic. By specifying the number of seconds between NOTIFY messages, you prevent the Oracle Communications Unified Session Manager from sending notifications upon events that do not generate a change in the registration database.

Database changes that trigger notifications when this option is configured include:

- The Cseq number of the REGISTER message increases by more than 1
- The call-ID changes
- A contact parameter changes
- The number of contacts changes

Upon expiry of this timer, the Oracle Communications Unified Session Manager sends out a NOTIFY for every registration event subscription. Note also that the Oracle Communications Unified Session Manager does not send the cseq attribute in the CONTACT element when this interval is configured.

Configuring Registration Event Package

This section shows you how to create reg-event profiles and apply those profiles to sip-registrars. These profiles enable the monitoring of UA registration events and the delivery of state change notifications to each UA that subscribes to the package. The procedure includes:

- Create one or more registration-event profiles
- Apply each profile to the applicable sip-registrar
- Optionally specify the registration event notification interval timer

Registration Event Profile Configuration

To configure a registration event profile:

1. From superuser mode, use the following command sequence to access regevent-notification-profile command.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# regevent-notification-profile
ORACLE(registration-event-profile)#
```

2. To define the profile, simply name it and specify a timeout in seconds.

```
ORACLE(registration-event-profile)# name reg-event-profile1
ORACLE(registration-event-profile)# min-subscription-duration 2500
ORACLE(registration-event-profile)# done
ORACLE(registration-event-profile)# exit
```

3. Navigate to the registrar for which you want registration event package support.

```
ORACLE(session-router)# sip-registrar
ORACLE(sip-registrar)# regevent-notification-profile reg-event-profile1
ORACLE(sip-registrar)# done
ORACLE(sip-registrar)# exit
```

Optional NOTIFY Refresh Frequency

To specify optional NOTIFY refresh frequency:

1. From superuser mode, use the following command sequence to access registration-event-profile command within session router.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# regevent-notification-profile
ORACLE(registration-event-profile)#
```

2. To enable NOTIFY, set the send-notify-for-reg-refresh option to the time, in seconds,

```
ORACLE(registration-event-profile)# options notify-refresh-interval=1800
ORACLE(registration-event-profile)# done
ORACLE(registration-event-profile)# exit
```

Prepend the option with the + sign if you have multiple options configured that you want to retain.

```
ORACLE(registration-event-profile)# options +notify-refresh-interval=1800
```

Running the command without the + character causes the system to remove any previously configured options.

Message Routing

The Oracle Communications Unified Session Manager provides two major types of routing that use the routing precedence parameter in the sip registrar. Routing precedence can be set to either **registrar** (HSS) or **local policy**. Routing precedence is set to registrar by default. There are additional controls that the user may configure to refine message routing.

Registrar routing uses the configured subscriber database and registration cache to route the call. Local policy routing lets you configure routing decisions within the Oracle Communications Unified Session Manager's local policy routing functionality.

Within the context of local policy routing, the Oracle Communications Unified Session Manager chooses the next hop through the network for each SIP session based on information received from routing policies and constraints. Routing policies can be as simple as routing all traffic to a proxy or routing all traffic from one network to another. Routing policies can also be more detailed, using constraints to manage the volume and rate of traffic that can be routed to a specific network. For example, you can manage volume and rate of traffic to enable theOracle Communications Unified Session Manager to load balance and route around softswitch failures.

When a message arrives at the Oracle Communications Unified Session Manager, it determines whether it is coming from a session agent. If so, the Oracle Communications Unified Session Manager checks whether that session agent is authorized to make the call. Local policy is then checked to determine where to send the message.

Depending on whether the Oracle Communications Unified Session Manager is performing originating or terminating services for the call, described in the chapter on operations within the IMS core, it performs those services prior to routing to the endpoint.

If the Oracle Communications Unified Session Manager is unable to proceed with routing a request, it replies to the UA that sent the request with a 4xx response.

This chapter provides an overview of registrar routing for perspective, but focuses on local policy routing. Local policy routing is configuration intensive, allowing precise route specification. As a result, configuring local policy routing is a complex process requiring that the user understand the purpose and interaction of multiple configuration elements. This

chapter also provides descriptions and configuration instruction on additional routing controls, such as the use of multistage and UA capability routing.

Registrar Routing

When the routing precedence parameter is set to **registrar**, the Oracle Communications Unified Session Manager is using the HSS as a resource within the context of its routing decisions.

When an INVITE arrives, the Oracle Communications Unified Session Manager checks its own registration cache for a pre-existing matching contact in the INVITE. If it finds a match, it forwards the request to that location. If it does not find a match, it issues an Location Information Request (LIR) to the HSS. If the HSS's response, called an LIA, provides an assigned S-CSCF for that UA, the Oracle Communications Unified Session Manager proceeds as described below in the section LIR/LIA Transaction.

Note that you can configure the Oracle Communications Unified Session Manager to fallback to a local policy lookup if the lookup via the registrar fails. Configure this by adding the **fallback-to-localpolicy** option to the sip-registrar configuration element.

For situations where the database routing decision needs to be done in lieu of the default, you can set routing precedence to local-policy. Note that you can configure a routing entry that points to an HSS by setting a policy attribute with a next-hop of cx:<home-subscriber-server-name> within the local-policy.

LIR/LIA Transaction

An LIR includes the Public-User-Identity AVP, which contain a UA's actual PUID. The HSS responds with the assigned S-CSCF server (often a Oracle USM) for this PUID. The answer is the form of a Location Info Answer (LIA). The LIA includes the assigned S-CSCF in the Server Name AVP.

If the S-CSCF returned in the LIR is this Oracle Communications Unified Session Manager, then the Oracle USM performs unregistered termination services for this UA. (This situation indicates that the UA is currently unregistered.) Such services could include directing the call to voice mail. If the HSS returns an S-CSCF in the LIA that is not this Oracle Communications Unified Session Manager, it forwards the request to that S-CSCF.

Default Egress Realm

The sip registrar configuration element should be configured with a default egress realm id. This is the name of the realm config that defines the IMS control plane, through which all Oracle Communications Unified Session Managers, HSSs, and other network elements communicate and exchange SIP messaging. It is advisable to configure this parameter to ensure well defined reachability among Oracle Communications Unified Session Managers.

RX Interface Features

The Oracle Communications Unified Session Manager can run the Rx interface over a Diameter connection and act as a P-CSCF communicating with a PCRF. The Rx interface supports quality of service and policy management within applicable network infrastructures. See the Oracle SBC ACLI Configuration Guide for full descriptions of this functionality.

ACLI Instructions

Configuring the SIP Registrar's Routing Precedence

To configure a SIP registrar configuration element for message routing:

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 path.

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

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

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

4. Type **select** and choose the number of the pre-configured sip interface you want to configure.
5. **routing-precedence**— Set this to either **registrar** or **local-policy** depending on your deployment.
6. **egress-realm-id**—Enter the default egress realm for Oracle Communications Unified Session Manager messaging.
7. Type **done** when finished.

Home Subscriber Server

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 path.

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

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

   ```
   ORACLE(session-router)# home-subscriber-server
   ORACLE(home-subscriber-server)#
   ```

4. Begin configuring your HSS, or type **select** and choose the number of the pre-configured HSS you want to configure.
5. Type **done** when finished.

Tel-URI Resolution

The Oracle Communications Unified Session Manager can initiate number resolution procedures for requests that have tel-URI or SIP-URI (with user=phone) numbers in the R-URI. It does this by querying number resolutions services, including the local routing table(s) or ENUM server(s) to resolve the R-URI to a SIP URI. In addition, the original R-URI may not include a full E.164 number. As such, you can also configure the Oracle Communications Unified Session Manager to perform a number normalization procedure and ensure it presents a

full E.164 number for resolution. Upon successful resolution, the Oracle Communications Unified Session Manager proceeds with ensuing signaling procedures.

To configure the Oracle Communications Unified Session Manager to perform these lookups, you create applicable **local-routing-config** or **enum-config** elements and set an option within the **sip-registrar** that specifies a primary and, optionally, a secondary **local-routing-config** or **enum-config** that the **sip-registrar** uses for LRT or ENUM lookups. If there is no ENUM configuration on the **sip-registrar**, the Oracle Communications Unified Session Manager forwards applicable requests to a border gateway function via local policy.

Refer to the *Oracle Communications Session Border Controller ACLI Configuration Guide* , Session Routing and Load Balancing chapter for complete information on how to configure a **local-routing-config** and/or an **enum-config**.

Number Lookup Triggers

Use cases that are applicable to number lookups and the associated Oracle Communications Unified Session Manager procedures include:

- Request from the access side:
  1. The Oracle Communications Unified Session Manager performs originating services.
  2. If the R-URI is a tel-URI or SIP-URI (with user=phone), it requests e.164 resolution from the ENUM server(s), regardless of its presence in the registration cache.
- Request from core side including request for originating services:
  1. The Oracle Communications Unified Session Manager performs originating services.
  2. If the R-URI is a tel-URI or SIP-URI (with user=phone), it requests e.164 resolution from the ENUM server(s), regardless of its presence in the registration cache.
- Request from core side, for terminating services only:
  1. If the R-URI is a tel-URI or SIP-URI (with user=phone) and is not in the Oracle Communications Unified Session Manager cache, it performs an LIR.
  2. If the LIA reply indicates the tel-URI or SIP-URI (with user=phone) is not provisioned, the Oracle Communications Unified Session Manager requests e.164 resolution from the ENUM server(s).

Actions Based on Lookup Results

The Oracle Communications Unified Session Manager forwards to the resultant SIP-URI under the following conditions:

- The SIP-URI is in the Oracle Communications Unified Session Manager cache, in which case the Oracle Communications Unified Session Manager performs terminating services.
- The SIP-URI is not in the Oracle Communications Unified Session Manager cache, and the Oracle Communications Unified Session Manager is configured to service the returned domain.
  In this case, the Oracle Communications Unified Session Manager performs the following:
  1. The Oracle Communications Unified Session Manager issues an LIR for the SIP-URI.
  2. The Oracle Communications Unified Session Manager forwards the message to the correct S-CSCF.

- The SIP-URI is not in the Oracle Communications Unified Session Manager cache, and the Oracle Communications Unified Session Manager is not configured to service the returned domain.
  In this case, the Oracle Communications Unified Session Manager performs refers to local policy to forward the message via local policy.

PSTN Breakout Routing

The Oracle Communications Unified Session Manager complies with RFC 4694 for operation with request-URIs that include carrier identification code/route number/number portability database dip indicator (cic/rn/npdi) information and routes those requests according to the rn information. The routing process includes utilization of local policy configured to break the request out of the home network via gateways such as a BGCF.

The Oracle Communications Unified Session Manager does not validate any rn or cic information. Instead, it simply routes the request. Note that the Oracle Communications Unified Session Manager uses cic information instead of rn if both are present in the request. RFC 4694 compliant circumstances under which the Oracle Communications Unified Session Manager does not use rn, cic and npdi information include:

- Invalid routing information, including rn present, but npdi missing.
- Invalid routing information, including npdi present, but rn missing.
- Request uses a `sip-URI` presented without `user=phone`.

If the request includes originating services as well as cic/rn/npdi information, the Oracle Communications Unified Session Manager performs those services rather than break out. If, after completing originating services, the request still includes cic/rn/npdi information, the system performs this breakout.

Primary and Secondary ENUM Configuration

For the purpose of redundancy, the Oracle Communications Unified Session Manager allows you to configure these number lookups to use a backup resource in case the lookup at the primary fails. Such scenarios include losing contact with the primary ENUM/LRT server config (query time-out) and the entry is not found at the primary (LRT or ENUM).

To apply primary and secondary number lookup resources to a sip-registrar:

1. From superuser mode, use the following command sequence to access the sip-registrar element and select the registrar you want to configure.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# sip-registrar
   ORACLE(sip-registrar)# select
   ```

2. Specify the resources to use with the options command.

   Prepend the option with the + character if you have multiple options configured that you want to retain. Running the command without the + character causes the system to disable any previously configured options.

   To specify primary and secondary ENUM servers:

   ```
   ORACLE(sip-registrar)# options +e164-primary-config=enum:<enum-config name>
   ORACLE(sip-registrar)# options +e164-secondary-config=enum:<enum-config name>
   ORACLE(sip-registrar)# done
   ```

   To specify primary and secondary LRT resources:

```
ORACLE(sip-registrar)# options +e164-primary-config=lrt:<lrt-config name>
ORACLE(sip-registrar)# options +e164-secondary-config=lrt:<lrt-config name>
ORACLE(sip-registrar)# done
```

Bear in mind that an enum-config can reference multiple servers. When the Oracle Communications Unified Session Manager references an enum-config, queries follow the normal enum-config sequence, checking each referenced server in order. If the lookup is not successful at the primary, the Oracle Communications Unified Session Manager checks the servers in the registrar's e164-secondary-config.

In addition, each enum-config may refer to a different top-level-domain. This allows you to configure the Oracle Communications Unified Session Manager to successfully perform lookups within two domains.

HSS Initiated User Profile Changes

The Oracle Communications Unified Session Manager can receive Push Profile Request (PPR) messages from an HSS and update the state of the IMS User Profile and associated subscription information it has cached locally. The SIP digest authentication information can also be updated and reassociated with an AoR in case that has changed too. The Oracle Communications Unified Session Manager expects to receive the following AVPs in a PPR message.

- Private-User-Identity—the username, whose subscription/authentication data has changed.
- SIP-Auth-Data-Item—if present new authentication data is included here.
- User-Data—if present new User data is included here.
- Charging-Information—if present new charging information is included here.

The Oracle Communications Unified Session Manager replies to an HSS's PPR in a PPA message with the following AVPs:

- Result-Code—indicates Diameter base protocol error. Valid errors for in a PPA are:
  - DIAMETER_SUCCESS—The request succeeded.
  - DIAMETER_ERROR_NOT_SUPPORTED_USER_DATA—The request failed. The Oracle Communications Unified Session Manager informs HSS that the received user information contained information, which was not recognized or supported.
  - DIAMETER_ERROR_USER_UNKNOWN—The request failed because the Private Identity is not found in Oracle Communications Unified Session Manager.
  - DIAMETER_ERROR_TOO_MUCH_DATA—The request failed. The Oracle Communications Unified Session Manager informs to the HSS that it tried to push too much data into the Oracle Communications Unified Session Manager.
  - DIAMETER_UNABLE_TO_COMPLY—The request failed.
- Experimental-Result—indicates diameter application (3GPP/Cx) error if present.

Licensing and Database Registration Limits

The Oracle Communications Unified Session Manager limits the number of unexpired registration cache entries globally. The total number of system registrations is configured with the registration cache limit parameter in the sip config configuration element.

The Oracle Communications Unified Session Manager also limits the number of registration cache entries that were obtained from a User Subscriber Database; only REGISTERs that prompted the database query are counted here. As User Subscriber Database entries are added and removed, this counter is updated accordingly. Note that it is the actual number of SD-contacts that count against the license limit. Discrete database registration license values range from 20,000 through 500,000 in increments of 20,000.

When a registering contact is rejected because it will exceed one of these limits, the Oracle Communications Unified Session Manager sends a 503 message to the registering endpoint.

Refer to the Getting Started chapter for information about install license management.

Database Registration Limit Alarm

By default, a major alarm is enabled when 98% or more of the licensed number of Database Registrations are used. This alarm is cleared when the number of database registrations falls below 90%. You can configure minor and critical alarms when crossing configured thresholds and you can also reassign the major alarm. This is configured in by creating a **system-config**, and then **alarm-threshold** sub element with type of **database-registration**.

3GPP Compliance

P-Asserted-Id in Requests and Dialogs

When an AoR is successfully registered through the Oracle Communications Unified Session Manager, the list of implicitly registered public IDs is returned from the HSS. The set of implicitly registered public IDs includes the explicitly registered Public-id and may include wild-carded public-ids. If there are no implicitly registered public-ids, then the implicit set returned by HSS will at least contain the explicitly registered public-id.

Based on local configuration and network conditions, the registering UE may or may not be trusted.

Non-trusted UE

When a non-trusted UE sends an initial request for a dialog or a request for a standalone transaction to the Oracle Communications Unified Session Manager, the P-Asserted-Identity to be inserted in the outgoing request is formed as follows:

- If the request does not have a P-Preferred-Identity header field or none of the P-Preferred-Identity header fields included in the request match any of the registered public user identities, then the Oracle Communications Unified Session Manager inserts the default Public-Identity in the outgoing P-Asserted-Identity header.
- If the request includes one or more P-Preferred-Identity header fields which match the registered public user identities, then the Oracle Communications Unified Session Manager includes only the first P-Preferred-Identity header field.
- If the request includes one or more P-Asserted-Identity header fields which do not match the registered public user identities, then the Oracle Communications Unified Session Manager inserts the default Public-Identity in the outgoing P-Asserted-Identity header.

Trusted UE

When a trusted UE sends an initial request for a dialog or a request for a standalone transaction to the Oracle Communications Unified Session Manager, the P-Asserted-Identity to be inserted in the outgoing request is formed as follows:

- If the request does not have a P-Preferred-Identity header field or none of the P-Preferred-Identity header fields included in the request match any of the registered public user identities, then the Oracle Communications Unified Session Manager inserts the default Public-Identity in the outgoing P-Asserted-Identity header.
- If the request includes one or more P-Preferred-Identity header fields which match the registered public user identities, then the Oracle Communications Unified Session Manager inserts the first P-Preferred-Identity header field.
- If the request includes one or more P-Asserted-Identity header fields, then the Oracle Communications Unified Session Manager will use the first P-Asserted-Identity header field.
  - The contents of the From header field do not form any part of this decision process.
  - P-Preferred-Identity header fields will always be removed.

The Default Public Identity is the first appearing, non-barred identity in the set of implicitly registered Public User Identities.

To enable this behavior, add the **pai-comply-to-3gpp** option in the sip config configuration element.

P-Associated-URI in 200 OK

In a 200 OK response to a UE on a successful registration, the Oracle Communications Unified Session Manager includes a P-Associated-URI header. This header includes the list of registered, distinct public user identities and the associated set of implicitly registered distinct public user identities.

When there are no associated implicit public identities, only the explicitly registered Public User Identity is included.

Other Diameter Cx Configuration

Host and Realm AVP Configuration for Cx

You can configure the values sent in the origin-host, origin-realm and destination-host AVPs when the Oracle Communications Unified Session Manager communicates with a server over the Cx interface. Configure destination-host when you want to precisely specify the HSS with which these Cx exchanges take place.

The applicable configuration parameters are located in the home-subscriber-server configuration element. The parameters used to configured the AVPs are origin-realm, origin-host-identifier and destination-host-identifier. The AVPs are constructed as follows:

```
Origin Host AVP = <origin-host-identifier>.<origin-realm>
Origin Realm AVP = <origin-realm>
Destination Host AVP = <destination-host-identifier>.<destination-realm>
```

If the **origin-realm** is not configured, then the realm parameter in the home-subscriber-server configuration element will be used as the default. If **origin-host-identifier** is not configured, then the name parameter in the home-subscriber-server configuration element will be used as the default.

If these parameters are not configured, then the AVPs are constructed as follows:

```
Origin Host = <HSS Config name>.<HSS Config realm>.com
Origin Realm AVP = <HSS Config realm>
Destination Host = <HSS Config name>.<HSS Config realm>.com
```

ACLI Instructions

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 path.

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

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

   ```
   ORACLE(session-router)# home-subscriber-server
   ORACLE(home-subscriber-server)#
   ```

4. **origin-realm**—Set this to a string for use in constructing unique Origin Host and Origin Realm AVPs.
5. **origin-host-identifier**—Set this to a string for use in constructing a unique Origin Host AVP.
6. **destination-host-identifier**—Set this to a string for use in constructing a unique Destination Host AVP.
7. Save your work.

Initial Filter Criteria (IFC)

The Oracle Communications Unified Session Manager, acting as a S-CSCF, downloads a set of rules known as Initial Filter Criteria (IFC) from the HSS/AS. IFCs are downloaded over the Cx interface.

iFCs are a way for an S-CSCF to evaluate which ASs should be involved in the call flow for a given user agent (UA). iFCs are functionally defined by Boolean statements, whose component parts are expressed in XML; they reference the destination AS(s) where a desired service is provided.

IFC Evaluation

IFCs are evaluated as described in 3GPP TS 29.228. The Oracle Communications Unified Session Manager supports all tags found in the 3GPP initial filter criteria specifications. An IFC is evaluated until its end, after which the call flow continues as expected.

SIP Registration

When the Oracle Communications Unified Session Manager receives an authenticated REGISTER request from a served UA, it sends an SAR request to the HSS to obtain an SAA

which includes iFCs associated with the UE's subscription. Within the context of registration, the Oracle Communications Unified Session Manager also manages third party registration procedures in conjunction with iFC exchanges or manually via the ACLI. These procedures are described in the Third Party Registration chapter.

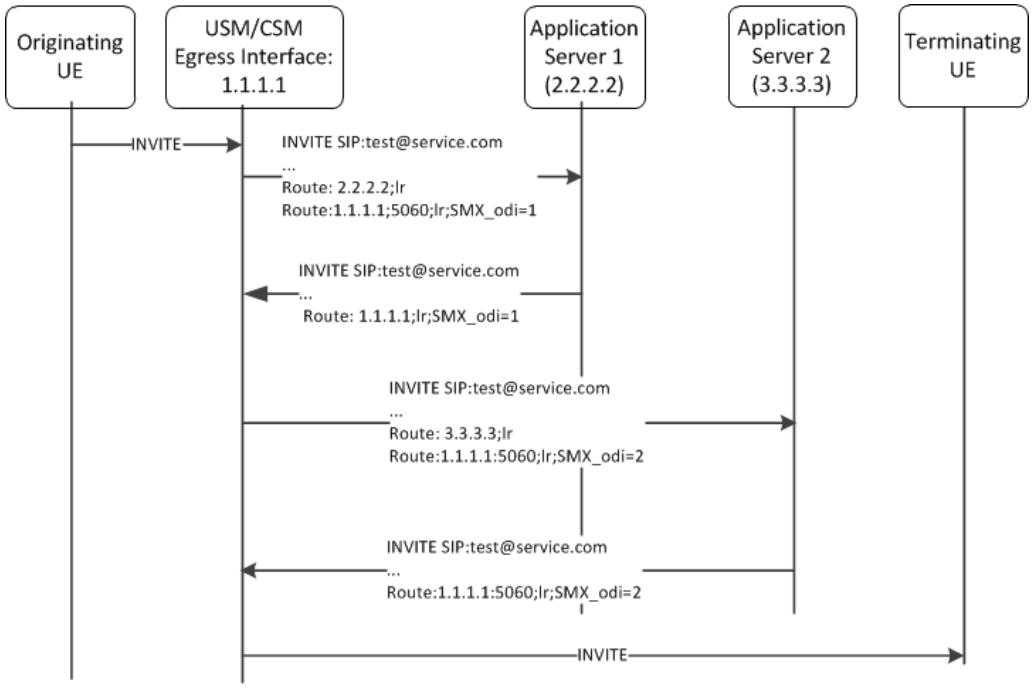
SIP Call

The Oracle Communications Unified Session Manager evaluates all IFC logic to determine that messages with matching characteristics are sent to the proper AS specified in the iFC evaluation using the IP Multimedia Service Control (ISC) interface. In this INVITE, the Oracle Communications Unified Session Manager adds two Route headers. The first (top) route header contains the target AS's URI. The second Route parameter is built using the IP address of the egress SIP interface and contains the ODI as a parameter. For example:

```
INVITE SIP:test@service.com
...
Route:2.2.2.2;lr
Route:1.1.1.1:5060;lr;smx_odi=1
```

If the AS routes the call back to the Oracle Communications Unified Session Manager, it is expected to include the ODI parameter that it received from the Oracle Communications Unified Session Manager, unchanged. The presence of the ODI parameter indicates that IFC evaluation needs to continue from where it left off for this call. If this continuation of IFC evaluation results in another AS URI, the Oracle Communications Unified Session Manager initiates a request towards that AS this time with a new ODI. In this way, the ODI is a state-signifier of Service Point Triggers.

The process continues until IFC evaluation is completed. Below is an example of an IFC evaluation completing after two iterations.



The iFC continues to be evaluated completely which may result in the INVITE being forwarded to additional ASs. At the conclusion of evaluating the iFC, the Oracle

Communications Unified Session Manager checks if the target of the initial request is registered to itself, or not. If the UA is not registered locally the Oracle Communications Unified Session Manager forwards the request by regular means into the network. If the target UA is registered locally, the Oracle Communications Unified Session Manager proceeds to obtain iFCs for the target and begin iFC evaluation for the terminating side of the call.

Preserving an Original Dialog Indicator

As the Oracle Communications Unified Session Manager (OCUSM) works through dialogs with Application Servers (AS), it saves and uses the Original Dialog Indicator (ODI) parameter to manage a call's service subscription sequence. By default, the OCUSM deletes the in-memory Service Profile, including the ODI, on receiving a final response to a transaction with an AS. If you enable the **preserve-odi** parameter in the **sip-config** however, the OCUSM maintains the in-memory service profiles, and each associated ODI, until it receives the **BYE** from the AS that ends the dialog between them. This is a global configuration, causing the OCUSM to maintain all ODIs for the duration of the dialog.

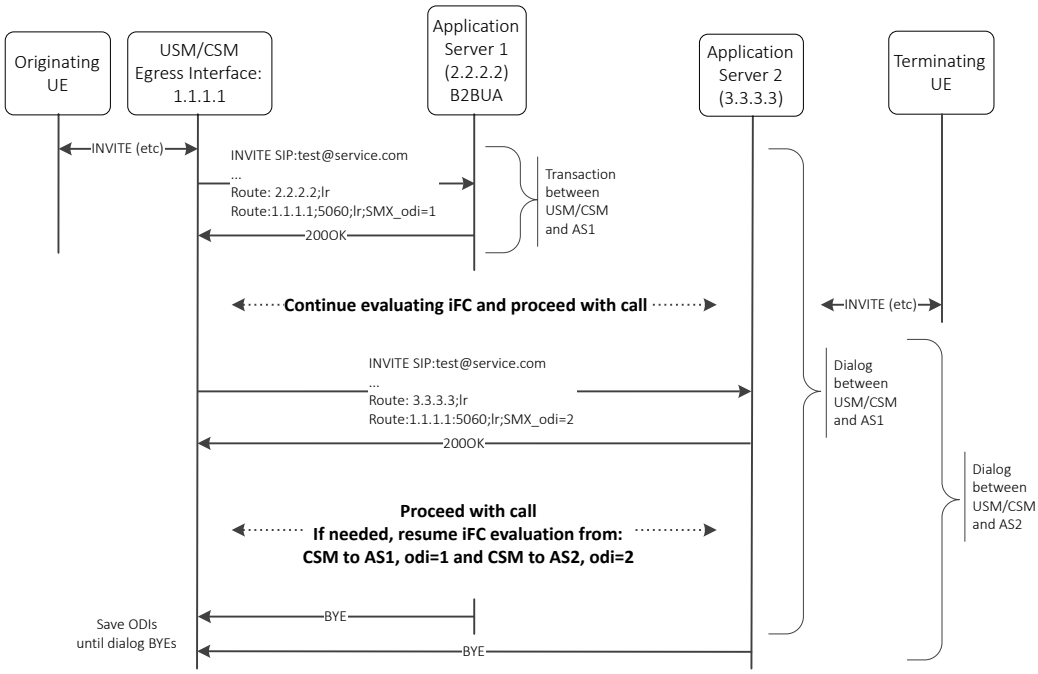
If the OCUSM discards the ODI when it receives the transaction's final message, it may experience problems, for example, with AS processes that are in B2BUA mode. In this mode, the AS may send a 200OK response to the request originator before forwarding the INVITE to the terminating side. If the OCUSM receives a message from the AS using a discarded ODI, the OCUSM restarts and repeats the entire iFC evaluation process for the call. Configuring ODI preservation provides value in applicable environments by avoiding this extraneous processing.

Enabling ODI preservation uses system resources to store this information. Balance this impact on resources with the value of deploying this behavior in your environment.

Key OCUSM behavioral detail with respect to the **preserve-odi** configuration includes:

- Upon receipt of an ODI, the OCUSM continues with iFC evaluation from the point where it left off for this ODI, using the same trigger-set and route-set as used within the initial iFC evaluation.
- If the originating user cancels a call, each AS also generates **CANCEL** message for each **INVITE**. The OCUSM clears ODIs from its memory when it processes the **CANCEL** for each original **INVITE**.
- If the HSS updates the user's service profile during iFC evaluation for a particular request message, for example, if the HSS sends a PPR to the OCUSM with a modified iFC, the OCUSM continues the current call with the existing in-memory Service Profile and uses the updated user-profile for subsequent calls.
- if an AS tries to send multiple request messages re-using an ODI, the OCUSM rejects those request messages and send 403 Forbidden response to the AS. The OCUSM only accepts the first request message from the AS using the same ODI.
- If the OCUSM does not receive the BYE, it retains the ODI until the session timers expire.
- An active OCUSM provides the service profile for each call, including all associated ODIs to its standby, enabling further service modification on a per-ODI basis despite a failover. This backup process requires a final message, a **200 OK** received from the AS. If a failover happens before the **200 OK**, the standby does not save the ODI.

The diagram below illustrates the OCUSM iFC evaluation behavior when you enable this parameter. By default, the OCUSM removes the ODI after the transaction between the OCUSM and AS1 is complete. With the **preserve-odi** parameter enabled, however, the OCUSM retains the ODI until it receives the **BYE** from AS1, which terminates the dialog between the OCUSM and AS1. If there are ODIs tracking service with other AS servers, shown below as AS2, the OCUSM retains those until the dialogs between itself and those servers terminates.



Given the diagram above, assume that the originating UE issues an invite towards the IMS core, which includes AS1 performing originating services and AS2 performing terminating services. The OCUSM initiates the iFC process with AS1, resulting in multiple transactions including the exchange of INVITEs using odi=1. When configured with **preserve-odi**, the OCUSM retains the service profile beyond the initial transaction. The OCUSM proceeds with terminating services via AS2, resulting in a similar set of transactions. In the meantime, the sequence has contacted the terminating UE. The sequence proceeds with completing the INVITE to 200OK interactions with AS1, AS2 and the terminating UE. The session proceeds until a UE issues a BYE to end the session. Subsequently, the process issues BYEs to terminate the dialogs with AS1 and AS2, at which time the OCUSM deletes those service profiles.

There are a large number of messages omitted from the diagram above for brevity and to highlight the dialogs between the OCUSM and the ASs.

Configuring ODI Preservation

Perform this sequence to enable ODI preservation on the OCUSM on a global basis.

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

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# sip-config
   ORACLE(sip-config)#
   ```

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

   ```
   ORACLE(sip-config)# select


   ORACLE(sip-config)#
   ```

3. **preserve-odi**—Set this parameter to **enabled**. The default value is disabled.

   ```
   ORACLE(sip-config)# preserve-odi enable
   ```

4. Save your work.

Evaluating Session Case in the P-Served-User Header

The P-served-user header field conveys the identity of the served user, the session case that applies to the particular communication session, and application invocation, as defined in RFC 5502 and TS 24.229. The Session Case (sescase) and Registration State (regstate) parameters are either populated by the UA originating the message or by the Oracle Communications Unified Session Manager after it determines if a request is originating or terminating, and registered or unregistered

The P-served-user header is created and added to an outgoing request if the next hop is trusted. A trusted next hop is an entity defined by a session agent whose trust-me parameter is enabled. Likewise, the P-served-user header is stripped if the request is forwarded to a next hop that is not known to be trusted.

When the Oracle Communications Unified Session Manager creates a P-served-User header, the user value in the originating case is the user value found in the P-Asserted-Identity header field. In the terminating case, the user value is taken from the Request URI.

Supported Sessioncase and Registration State

The following cases are supported for IFC evaluation. Conditions for classifying the calls as such are listed below.

Originating request - Registered User

When the Oracle Communications Unified Session Manager receives an Initial request, it is validated as an originating request from or on behalf of a registered user when the following conditions are met:

- When the **ignore-psu-sesscase** option is not set:
  - The request is a dialog creating request or a standalone request.
  - There is no "odi" parameter in the top route of the request.
  - The regstate and sescase parameters of the P-served-user indicate for this to be treated as originating request for a registered user.
- When the **ignore-psu-sesscase** option is set:
  - The request is a dialog creating request or a standalone request.
  - There is no "odi" parameter in the top route of the request.
  - There is an "orig" parameter in the top route of the request.
  - The served user is registered

Originating request - Unregistered User

When the Oracle Communications Unified Session Manager receives an Initial request, it is validated as an originating request from or on behalf of an unregistered user when the following conditions are met:

- When the **ignore-psu-sesscase** option is not set:
  - The request is a dialog creating request or a standalone request.
  - The served user is unregistered.

  - The request is from an AS or I-CSCF and the top route header contains the orig parameter OR
    The regstate and sescase of the P-served-user header indicates that the request is an originating request for an unregistered user.

- When the **ignore-psu-sesscase** option is set:
  - The request is a dialog creating request or a standalone request.
  - There is no "odi" parameter in the top route of the request.
  - There is an "orig" parameter in the top route of the request.
  - The served user is unregistered

Terminating Requests - Registered User

When the Oracle Communications Unified Session Manager receives an Initial request, it is validated as a terminating request towards a registered user when the following conditions are met:

- When the **ignore-psu-sesscase** option is not set:
  - The request is a dialog creating request or a standalone request.
  - There is no "orig" parameter in the top route of the request.
  - There is no "odi" parameter in the top route of the request.
  - The regstate and sescase parameters of the P-served-user indicate for this to be treated as terminating request for a registered user OR the request is finished with originating services if applicable and the request is destined to a user who is currently registered with the Oracle Communications Unified Session Manager.
  - If the Request-URI changes when visiting an application server, the Oracle Communications Unified Session Manager terminates the checking of filter criteria and routes the request based on the changed value of the Request-URI, per 3GPP Specification TS 23.218.
- When the **ignore-psu-sesscase** option is set:
  - The request is a dialog creating request or a standalone request.
  - There is no "odi" parameter in the top route of the request.
  - There is no "orig" parameter in the top route of the request.
  - The served user is registered

Terminating Requests - Unregistered User

See the IFC Support for Unregistered Users section in the Configuration Guide for this case.

- When the **ignore-psu-sesscase** option is not set:
  - If the Request-URI changes when visiting an application server, the Oracle Communications Unified Session Manager terminates the checking of filter criteria and routes the request based on the changed value of the Request-URI, per 3GPP Specification TS 23.218.
- When the **ignore-psu-sesscase** option is set:
  - The request is a dialog creating request or a standalone request.
  - There is no "odi" parameter in the top route of the request.

- The served user is not registered.

The request is a dialog creating request or a standalone request.

- There is no "orig" parameter in the top route of the request.
- There is no "odi" parameter in the top route of the request.
- The regstate and sescase parameters of the P-served-user indicate for this to be treated as terminating request for an unregistered user

Third Party Registration for an Implicit Registration Set

When using iFCs, the Oracle Communications Unified Session Manager performs third party registrations based on the iFC downloaded for each PUID. By default, the Oracle Communications Unified Session Manager performs third party registration for the service profiles of all PUID's in a user's implicit registration set. This is compliant with 3GPP specifications. The system includes any shared or default iFCs that apply to each PUID during this process. The system performs this function when it receives user-initiated de-registrations, but not when it receives RTRs. If desired, the user can configure the Oracle Communications Unified Session Manager to perform third party registration for only the REGISTERED PUID in the registration using a **sip-registrar** option.

> **Note:**
>
> The Oracle Communications Unified Session Manager does not attempt third party registration for any barred, tel or wildcard PUIDs.

The user can verify all third party registrations using the **show registration sipd by-user [user] detailed** command. Example output is shown below.

```
ORACLE# show registration sipd by-user 234 detailed

Registration Cache (Detailed View)    Wed Sep 16 2015  10:57:44

User: sip:234@acme-ims.com
  Registered at:  2015-09-16-10:57:40    Surrogate User: false
  Emergency Registration? No
  ContactsPerAor Rejects 0
  ContactsPerAor OverWrites 0

  Contact Information:
    Contact:
      Name: sip:234@acme-ims.com
      Valid: true
      Challenged: false
      Registered at: 2015-09-16-10:57:40
      Last Registered at: 2015-09-16-10:57:40
      Expire: 3596
      Local expire: 296
      Half: 1796

      Registrar IP: 0.0.0.0
      Transport: UDP
      Secure: false
      Local IP: 192.168.53.99:5060
```

```
      User Agent Info:
        Contact: sip:234@192.168.53.181:5060
        Realm: core
        IP: 192.168.53.181:5060

      SD Info:
        Contact: sip:234-tbcktcgo177fc@192.168.53.99:5060
Call-ID: 1-5853@192.168.53.181
      Path: <sip:234@192.168.53.181:5060;lr;p-acme-serving>

    Associated URI(s):
      URI: sip:234@acme-ims.com
      Status: Non-Barred
         Filter Criteria:
           Priority: 0
             Filter: ((method == REGISTER)) or
              ((method == INVITE))
             Application Server: sip:172.16.17.10:5060

      URI: sip:1@acme-ims.com
      Status: Non-Barred
         Filter Criteria:
           Priority: 0
             Filter: ((method == REGISTER)) or
              ((method == INVITE))
             Application Server: sip:172.16.17.10:5060
           Priority: 1
             Filter: ((method == INVITE)) or
              ((method == REGISTER))
             Application Server: sip:172.16.53.181:5065

      URI: tel:135
      Status: Barred
         Filter Criteria:
           Priority: 0
             Filter: ((method == INVITE)) or
              ((method == REGISTER))
             Application Server: sip:172.16.53.181:5065
           Priority: 1
             Filter: ((method == INVITE)) or
((method == REGISTER))
             Application Server: sip:172.16.53.181:5095


    Third Party Registration(s):
      Third Party Registration Host: 172.16.17.10
      Registration State: REGISTERED
      Last Registered at: Never
      Third Party Registration Host: 172.16.53.181
      Registration State: REGISTERED
      Last Registered at: Never
```

The user can check for third party registrations errors using the **show sipd third-party-reg all** command. Example output is shown below.

```
ORACLE# show sipd third-party-reg all
3rd Party Registrar             SA State   Requests     200OK  Timeouts    Errors
(D)111.11.17.10                   INSV         1         1         0         0
(D)111.11.53.181                  INSV         1         1         0         0
```

The user can disable the default behavior and perform third party registration only for the PUID in the REGISTER by configuration. Disabling this behavior can improve system performance by preventing the system from having to walk through large PUID sets for large numbers of ASs. The ACLI syntax for disabling this functionality using the **disable-thirdPartyReg-for-implicit-puid** setting follows.

```
ORACLE(sip-registrar)#options +disable-thirdPartyReg-for-implicit-puid
```

> **Note:**
>
> Prior to this version, the Oracle Communications Unified Session Manager's default behavior was the same as if the **disable-thirdPartyReg-for-implicit-puid** option was set in the SIP registrar. Users upgrading to this version of the Oracle Communications Unified Session Manager must set the **disable-thirdPartyReg-for-implicit-puid** option to retain the previous behavior.

TEL URI Replacement with SIP URI in R-URI to AS

When the USM receives a request containing a TEL URI from the Media Gateway Control Function (MGCF), it sends the TEL URI as an R-URI to the Application Server (AS) to perform services. However, in some implementations, the AS does not accept TEL URI and requires the trigger to be based on SIP URI. This feature, when enabled, causes the USM to replace the TEL R-URI with a SIP URI based on the first SIP user in the implicit set.

In the current implementation of the USM for terminating calls, when the USM receives an R-URI with SIP user=phone (for example, "sip:+359888528650@sip.mtel.bg; user=phone"), the USM replaces the SIP URI with a TEL URI and further uses the TEL URI (for example, "tel: +359888528650") for Location Information Requests (LIR) and Server Assignment Requests (SAR) when the user is not in the registration cache. The Server Assignment Answer (SAA) provides the Public Identity "sip:tel.359888528650@sip.mtel.bg" in the Service Profile as it's part of the implicit registration set and the USM stores it in its registration cache. Then, based on the Service Profile for the TEL URI, the USM triggers the AS using the R-URI "tel: +359888528650". However, in some implementations, for requests coming from the MGCF, the USM receives requests with a TEL URI which is sent as an R-URI to the AS while doing services, but the AS does not accept TEL URIs and requires the trigger to be based on a SIP URI.

To rectify this deficiency, this feature, when activated and when the USM is the assigned Serving Call Session Control Function (S-CSCF), causes the USM to replace the TEL R-URI with the first SIP URI in the implicit set for the TEL user; it then performs services based on the trigger for the user of the first implicit SIP URI. Once a TEL URI is changed to a SIP URI to perform services it will not be changed back to a TEL URI for the entire call flow. When this feature is enabled, the USM uses the first SIP user entry in the implicit set and performs services for the user irrespective of whether the user is in the registration cache.

TEL URI Replacement with SIP URI in R-URI to AS Configuration

Configuration changes occur in real time and do not require rebooting.

1. Access the **ifc-profile** configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# ifc-profile
   ORACLE(ifc-profile)#
   ```

2. Select the **ifc-profile** object to edit.

```
ORACLE(ifc-profile)# select
<name>:
1:  name=ifc_appserver

ORACLE(ifc-profile)#
```

3. **options** — Set the options parameter by typing **options** , a space, the option name **replace-tel-ruri-with-implicit-sip** with a plus sign in front of it, and then press Enter. You must prepend the new option with a plus sign to append the new option to the IFC profile's options list. If you type the option without the plus sign, you will overwrite any previously configured options.

   ORACLE(ifc-profile)# options +replace-tel-ruri-with-implicit-sip

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

Additional Options

- The Oracle Communications Unified Session Manager can populate the top Route: header with the sescase value for ASs that require it. In such a case, the parameter is created as either call=orig or call=term. This behavior is enabled by configuring the **add-sescase-to-route** option in the ifc-profile.
- When the dialog-transparency parameter in the sip-config is set to enabled and your network includes multiple ASs, you should add the **dialog-transparency-support** option in the ifc-profile.
- The Oracle Communications Unified Session Manager provides an alternative, configurable option that allows the user to specify the use of route header information to determine Served User and Session Case for out-of-the-blue (OOTB) calls. This method is 3GPP-compliant. By default, the Oracle Communications Unified Session Manager uses information from the P-Served-User (PSU) header. The user configures this behavior by enabling the ignore-psu-sesscase option in the ifc-profile.

IFC Support for Unregistered Users

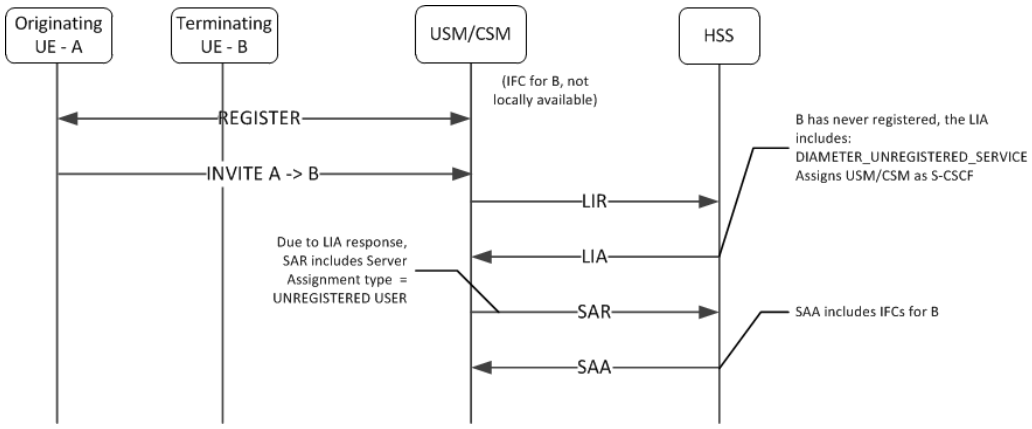
The Oracle Communications Unified Session Manager can download Initial Filter Criteria (IFC) from the HSS for unregistered users. This section displays applicable message sequence diagrams.

UE-terminating requests to an unregistered user

The Oracle Communications Unified Session Manager downloads and executes IFCs for the terminating end of calls. The following call flows indicate possible cases for the terminating unregistered user.

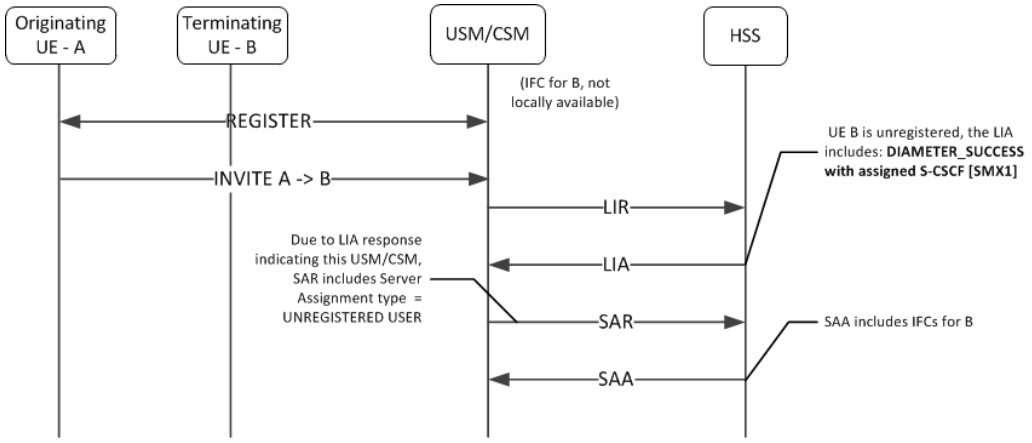
Terminating UA - Unregistered

UE has never registered.

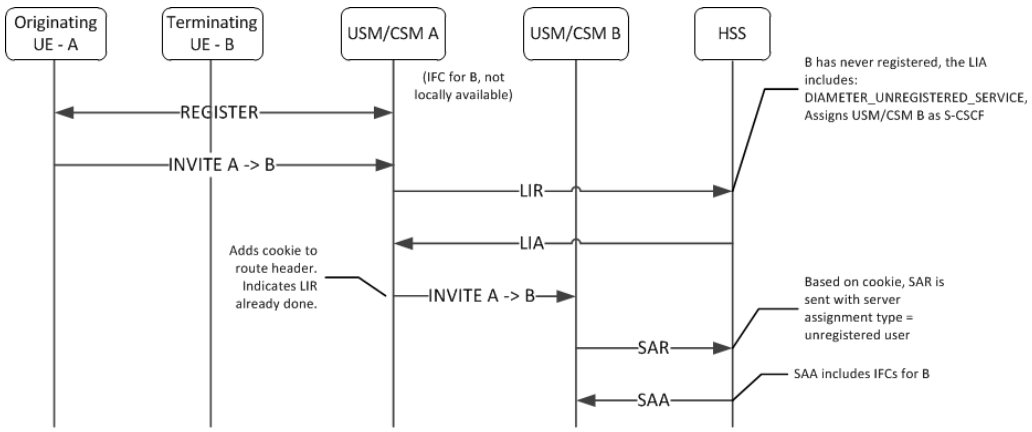


Terminating UA - Unregistered

UE originally registered as a consequence of an originating or terminating request or an S-CSCF has stored the user profile.



Terminating UA - Not Registered, Served by other Oracle Communications Unified Session Manager



UE Subsequent Registration

If the Oracle Communications Unified Session Manager has a cached IFC downloaded for an unregistered UA who later registers to that Oracle Communications Unified Session Manager, the cached IFC will be cleared and updated with the IFC downloaded by the registration process.

Caching the Downloaded IFC

When the Oracle Communications Unified Session Manager downloads IFCs for unregistered users, they are saved to a local cache. If the IFC cache fills up, an older cached IFC for a user is released.

Optimizing IFC Updates

The Oracle Communications Unified Session Manager aims to reduce the number of IFC updates traversing the network to save bandwidth and transactional overhead. Unless the unregistered UE's IFC entry has been deleted because of exhausting cache space, the following optimizations are performed:

- If IFCs are available locally, then an SAR/SAA operation to download IFCs will not be performed.
- If a previous IFC download operation did not return any IFCs, then subsequent calls to that unregistered user will not invoke the SAR/SAA messaging to download IFCs.

Push Profile Request (PPR) updates

The HSS can push service profile updates for private IDs. The Oracle Communications Unified Session Manager can process PPR updates for unregistered entities. If the user entry has been deleted because IFC cache space has been exhausted, the PPRs will not be processed.

ACLI Instructions

SIP Registrar

To create an IFC Profile:

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 path.

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

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

   ```
   ORACLE(session-router)# ifc-profile
   ORACLE(ifc-profile)#
   ```

4. **name**—Enter a name for this IFC profile.
5. **state**—Set this to enabled to use this ifc-profile.

6. **options**—Set the options parameter by typing options, a Space, the option name with a plus sign in front of it, and then press Enter.

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the options list, you must prepend the new option with a plus sign.

   The options included in this section are: **add-sescase-to-route** and **dialog-transparency-support.**

7. Type **done** when finished.

SIP Registrar

To enable IFC support in a SIP Registrar:

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 path.

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

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

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

4. Type **select** and choose the number of the pre-configured SIP registrar configuration element you want to configure.

   ```
   ORACLE(sip-registrar)# select
   name:
   1: registrar1
   selection:1
   ORACLE(sip-registrar)#
   ```

5. **ifc-profile**—Set this parameter to the name of the IFC profile you just created.
6. **serving-function**—Set this parameter to disabled when you want the Oracle Communications Unified Session Manager to act solely as an I-CSCF. When disabled, the Oracle Communications Unified Session Manager always forwards requests from unregistered users to the serving group.
   The default is enabled, which retains the S-CSCF function on this Oracle Communications Unified Session Manager.
7. **serving-group**—Set this parameter to a Session Agent Group (SAG) name. The Oracle Communications Unified Session Manager forwards requests from unregistered users to this group when the serving function parameter is disabled.
   Use of this parameter requires the prior configuration of a SAG that includes all prospective S-CSCFs. The name you give to that group is the name you specify as an argument to this parameter.
8. Type **done** when finished.

Shared and Default iFCs

The Oracle Communications Unified Session Manager supports Shared iFCs (SiFC), as defined by TS 29.229 and Default iFCs, which are an Oracle extension upon SiFCs. SiFCs provide an operator with the ability to create iFC templates and apply them to a large number of UEs. The

SiFC process optimizes the provisioning, storage and transfer of service profile information. The default IFC (DiFC) establishes a configuration wherein the iFC associations are available on the Oracle Communications Unified Session Manager itself. This establishes a backup scenario in case the HSS is not responsive.

To support the SiFC feature on the Oracle Communications Unified Session Manager, you create a profile that refers to a local, XML-formatted file. This file specifies the iFCs to be shared. You apply these profiles to registrars to specify where they are used.

When an SiFC configuration is in place, the Oracle Communications Unified Session Manager notifies the HSS that it supports SiFCs within the Supported-Features AVP in the SAR. The HSS replies to confirm that it supports SiFCs within the SAA. The SiFC feature must be enabled on the HSS.

Note that the form and function of the SiFC and DiFC files are compatible. You can use the same file for both SiFC and DiFC configuration, if desired.

SiFC Usage

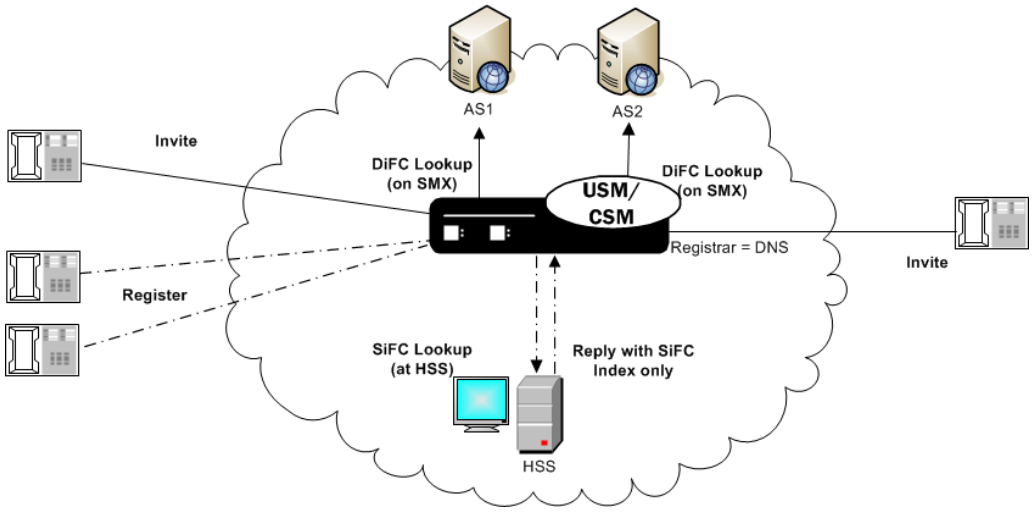
When an applicable end station registers, the Oracle Communications Unified Session Manager forwards the registration to the HSS normally. Given SiFC configuration however, the HSS sends a service-profile containing the SiFC identifiers to the Oracle Communications Unified Session Manager rather than the entire service definition. The Oracle Communications Unified Session Manager parses these identifiers and maps the user to the locally stored filter criteria.

The <IFCSet id =”x”> tags in the XML file on the Oracle Communications Unified Session Manager map to the HSS identifiers.

DiFC Usage

In contrast to SiFCs, the Oracle Communications Unified Session Manager fires DiFCs within the context of a session. During the session, the Oracle Communications Unified Session Manager associates the iFCs within the DiFC file with the user, as needed. DiFC usage is invoked during session initiation.

Note that DiFCs are database agnostic. You can use DiFCs for HSS, ENUM and local database configurations. An operational overview of SiFCs and DiFCs is shown below.



SiFC/DiFC File Example

An example of a Oracle Communications Unified Session Manager local SiFC/DiFC XML file, including a single iFC Set containing a single iFC, is presented below.

```
<?xml version="1.0" encoding="UTF-8"?>
<IFCSets>
  <IFCSet id=”0”>
    <InitialFilterCriteria>
      <Priority>0</Priority>
      <TriggerPoint>
        <ConditionTypeCNF>0</ConditionTypeCNF>
        <SPT>
          <ConditionNegated>0</ConditionNegated>
          <Group>0</Group>
          <Method>INVITE</Method>
          <Extension></Extension>
        </SPT>
      </TriggerPoint>
    <ApplicationServer>
      <ServerName>sip:172.16.101.26:5060</ServerName>
      <DefaultHandling>0</DefaultHandling>
    </ApplicationServer>
    <ProfilePartIndicator>0</ProfilePartIndicator>
  </InitialFilterCriteria>
</IFCSet>
</IFCSets>
```

Note that the Shared IFCSet contains the integer value property (id=”0”) that associates these filter criteria with users registered with the Oracle Communications Unified Session Manager. In the case of SiFC, it is the value that the HSS should send when referencing shared sets. In the case of DiFC, the integer is meaningless. The Oracle Communications Unified Session Manager loads and executes default iFCs in the order they appear within the XML file.

iFC Execution Order

Within the context of the 3GPP standards, the Oracle Communications Unified Session Manager evaluates explicitly downloaded iFCs first when determining where to look for a service. If the Oracle Communications Unified Session Manager cannot execute on the service based on explicitly downloaded iFCs, it refers to the SiFC, then the DiFC information to identify an AS that supports the service.

Refreshing SiFC and DiFC Files

Given the nature of local file usage, an ACLI command is available to allow the user to refresh SiFC and DiFC contexts in memory after the user has saved changes to the SiFC and DiFC files. Run the following command to deploy these changes:

ORACLE# **refresh ifc <ifc-profile name>**

Note also that the Oracle Communications Unified Session Manager validates the SiFC and DiFC files whenever you Activate your configuration.

SiFC and DiFC Configuration

To configure the Oracle Communications Unified Session Manager to use Shared and Default IFCs:

1. From superuser mode, use the following command sequence to access ifc-profile element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# ifc-profile
   ```

2. Define your profile.
3. **name**—Enter a name for this IFC profile.

   ```
   ORACLE(ifc-profile)# name acmeTelecomIFC
   ```

4. **state**—Set this to enabled to use this ifc-profile.

   ```
   ORACLE(ifc-profile)# state enabled
   ```

5. **default-ifc-filename**—Specify filename and, if not stored in the default directory /code/ifc, the applicable pathname.

   ```
   ORACLE(ifc-profile)# default-ifc-filename Afile.xml.gz
   ```

6. **shared-ifc-filename**—Specify filename and, if not stored in the default directory /code/ifc, the applicable pathname.

   ```
   ORACLE(ifc-profile)# shared-ifc-filename Bfile.xml.gz
   ```

7. **options**—Set the options parameter by typing options, a Space, the option name with a plus sign in front of it, and then press Enter.

   ```
   ORACLE(ifc-profile)# done
   ```

8. Apply the ifc-profile to your sip registrar.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# sip-registrar
   ```

   Select the registrar you want to configure and apply the profile.

   ```
   ORACLE(sip-registrar)# select
   ORACLE(sip-registrar)# ifc-profile acmeTelecomIFC
   ORACLE(sip-registrar)# done
   ```

Distinct and Wildcarded Public Service Identity (PSI) Support

The Oracle Communications Unified Session Manager supports the use of distinct Public Service Identity (PSI) and wildcarded PSIs, typically for specifying access to a service. There is no configuration required on the Oracle Communications Unified Session Manager to enable this support.

Administrators use individual PSI entries and/or wildcarded PSIs as service identifiers on an HSS. These identifiers provide the information needed to direct applicable messages to applicable application servers. Distinct PSIs can reside within individual PSI entries; wildcarded PSI entries are managed within iFC lists. Wildcarded PSI support is typically

implemented to reduce HSS resource requirements. By configuring a wildcarded PSI, administrators can use a single record within the iFC to manage multiple resources.

A wildcard is composed of an expression that, when used in a user part, provides for access to multiple service resources. The regular expressions themselves are in form of Perl Compatible Extended Regular Expressions (PCRE).

For example, consider the following two service resources:

- sip:chatroom-12@core.com
- sip:chatroom-64@core.com

These two service resources can be represented simultaneously at the HSS using the following syntax:

- sip:chatroom-!.*!@core.com

The Oracle Communications Unified Session Manager caches filter criteria information that uses this wildcard syntax. This avoids the need for SAR/SAA exchanges between the Oracle Communications Unified Session Manager and the HSS every time an entity requests the service. The Oracle Communications Unified Session Manager is equally capable of caching distinct PSIs, which similarly offloads the need for SAR/SAA exchanges during service resource location processes.

For most call flows, the Oracle Communications Unified Session Manager does not evaluate the expression for the purpose of finding a match. Instead, it keeps the syntax provided by the HSS in its cache and provides the wildcarded syntax in the applicable AVP.

To allow the feature, the Oracle Communications Unified Session Manager supports:

- Wildcarded public user identity AVP in the LIA, SAR and SAA
- User Profile AVP in the SAA
- P-Profile-Key across the Mw interface, as defined in RFC 5002

Configuring SIP Ping OPTIONS Support

You can configure the Oracle Communications Unified Session Manager to respond to SIP ping OPTIONS. This support is typically configured on an S-CSCF so it can respond to pings OPTIONS sent by a P-CSCF:

To configure an SIP Options Ping response support:

1. From superuser mode, use the following command sequence to access ping-response command on a sip-interface element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# sip-interface
   ORACLE(sip-interface)# sel
   ```

2. Enable the support with the ping-response command.

   ```
   ORACLE(http-config)# ping-response enabled
   ORACLE(http-config)# done
   ```

   ping-response—Enable ping-response to allow your device to respond to ping OPTIONS. For example, this feature is useful within hybrid deployment environments on a P-CSCF as a means of verifying the S-CSCF's availability. This configuration allows the S-CSCF to respond to SIP ping OPTIONS.

Redundancy and Load Balancing with HSS Servers

The Oracle Communications Unified Session Manager allows you to operate with multiple HSS servers, supporting:

- Redundancy - Continue normal operation despite HSS failure.
- Load Balancing - Divide the traffic load between HSS servers in a group of HSSs. Preference is based on the HSS list order configured on the Oracle Communications Unified Session Manager.

You configure HSS servers within HSS Groups to support this functionality. For redundancy, you create and assign HSS groups, and apply either the hunt or fail-over strategy to your HSS group. To implement load balancing, you configure the applicable HSS group with a the round-robin server allocation strategy. This functionality assumes the HSS infrastructure itself is configured for redundancy.

The Oracle Communications Unified Session Manager establishes and manages multiple Cx connections with each applicable HSS. This management is achieved by connection identifiers on the Oracle Communications Unified Session Manager that allow it to distinguish between connections. This provides the network with the flexibility of being able to use multiple paths to a given HSS regardless of AVP values.

About HSS Groups

You configure HSS groups based on your redundancy and failover design. You accomplish this by configuring your HSS groups with the applicable HSS servers. You then assign your group to a registrar. HSS group configuration does not preclude assigning an HSS in the group to a registrar individually.

HSS groups can contain individual HSSs. Members of an HSS group are prioritized by the server list; the first server in the list takes the highest priority; the last takes the lowest. You can manually disable an HSS group, if desired, which prevents the Oracle Communications Unified Session Manager from attempting to access any of the HSS servers via that group.

HSS group members do not need to reside in the same domain, network, or realm. The Oracle Communications Unified Session Manager can allocate traffic among member HSSs regardless of their location. It uses the allocation strategies you configure for the group to distribute traffic across the group members.

Group allocation strategies define how the Oracle Communications Unified Session Manager selects an HSS. For example, the hunt strategy selects HSSs in the order in which they are listed. Allocation strategies include the following:

Allocation Strategy	Description
failover	For HSS redundancy deployments, the failover strategy specifies that the Oracle Communications Unified Session Manager selects the next highest priority HSS server for all operations if the first HSS fails. The Oracle Communications Unified Session Manager does not resume operation with the initial HSS when it comes back into service.

Allocation Strategy	Description
hunt	For HSS redundancy deployments, the hunt strategy specifies that the Oracle Communications Unified Session Manager select HSSs in the order in which they are configured in the HSS group. If the first HSS is available, all traffic is sent to the first HSS. If the first HSS is unavailable, all traffic is sent to the second HSS. The system follows this process for all HSS servers in the group. When a higher priority HSS returns to service, all traffic is routed back to it.
roundrobin	This strategy targets HSS load balancing deployments. The Oracle Communications Unified Session Manager selects each HSS in the order in which it appears in the group list, routing diameter requests to each HSS in turn.

Paths taken by specific messaging is constrained by the purpose of that messaging, and refined by a group's allocation strategy. Applicable messaging includes UAR/UAA, MAR/MAA, SAR/SAA and LIR/LIA. For both failover and hunt strategies, all messaging is sent to the current active server. For the round-robin strategy, messaging is distributed to group members sequentially, using the member list order.

Connection Failure Detection

The Oracle Communications Unified Session Manager detects that a connection between itself and a given HSS has failed if either a diameter request fails or the diameter DWR/DWA handshake fails. If the HSS does not respond to five requests, the Oracle Communications Unified Session Manager marks that HSS as out of service.

The Oracle Communications Unified Session Manager forwards unacknowledged messages to subsequent HSSs based on strategy. It changes the destination host AVP of these messages and marks then with the T flag. The HSS recognizes the T flag as an indication that the request may be a duplicate, caused by a problem in the network.

Periodically, the Oracle Communications Unified Session Manager attempts to establish diameter connections with out of service HSS servers. When those connections succeed, the Oracle Communications Unified Session Manager marks the HSS as in-service and resumes using it within the context of the configured redundancy and load balancing strategy.

Configuring HSS Groups

To configure HSS groups:

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

   ```
   ORACLE# configure terminal
   ```

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

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

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

   ```
   ORACLE(session-router)# hss-group
   ORACLE(hss-group)#
   ```

4. **name**—Enter a unique name for the HSS group in Name format.

5. **state**—Enable or disable the HSS group on the Oracle Communications Unified Session Manager. The default value is **enabled**. Valid values are:
   - enabled | disabled
6. **origin-host-identifier**— Set this to a string for use in constructing a unique Origin Host AVP. This setting always takes precedence over the origin-host-identifier configured for the home-subscriber-server. This setting is required.
7. **strategy**—Indicate the HSS server allocation strategy you want to use. The strategy you chose selects the HSS servers that will be made available by this hss-group. The default value is **hunt**. The valid values are:
   - **hunt**—Selects HSS servers in the order in which they are listed. For example, if the first server is online, all traffic is sent to the first server. If the first server is offline, the second server is selected. If the first and second servers are offline, the third server is selected. When the Oracle Communications Unified Session Manager detects that a higher priority HSS is back in service, it routes all subsequent traffic to that HSS.
   - **roundrobin**—Selects each HSS server in the order in which they are listed in the destination list, selecting each server in turn, one per session.
   - **failover** — Selects the first server in the list until failure is detected. Subsequent signaling goes to the next server in the list.
8. **hss-configs**—Identify the HSS servers available for use by this hss-group. This list can contain as many HSS servers as is necessary. An hss-config list value must correspond to a valid hss-config.

   Display syntax for the hss-configs parameter by typing the question mark character after the parameter name on the ACLI.

```
ORACLE(hss-group)# hss-configs ?
<string> list of home-subscriber-server configs for this group
         for single-entry: hss1
         for multi-entry: (hss1 hss2)
         for adding an entry to an existing list: +hss3
         for deleting an entry from an existing list: -hss3
         for multiple entries add/remove from a list: +/-(hss1 hss2)
```

   The following example shows an HSS group using the hunt allocation strategy applied.

```
hss-group
        name                          group-test1
        state                         enabled
        origin-host-identifier
        strategy                      hunt
        hss-configs                   hss1, hss2
        last-modified-by              admin@console
        last-modified-date            2013-05-13 14:58:01
```

4

Oracle USM Supporting the IMS Edge

The ETSI TISPAN NGN defines several subsystems that make up the NGN architecture. The Oracle Communications Unified Session Manager is an integrated session control, policy enforcement and media management solution that incorporates functional components of the IP multimedia subsystem (IMS), the Resource and Admission Control Subsystem (RACS) and functions necessary for connecting with other IP networks/domains.

The functions of the Oracle Communications Session Border Controller within the NGN architecture are divided into access/interconnect and core functions. This chapter addresses the access and interconnect functions. Oracle USM core functions are addressed in the Oracle USM Supporting the IMS Core chapter.

Access Border Functions

The Oracle Communications Unified Session Manager is deployed as the access point between the core IMS network and UEs to deliver the functions defined in the TISPAN architecture as the P-CSCF, and A-BGF. These two functions can not be separated. The Oracle Communications Unified Session Manager performs the following functions as the Access Oracle Communications Unified Session Manager:

- P-CSCF functions
- Access Border Gateway Functions (A-BGF)
- Resource and Admission Control Functions (RACF)

P-CSCF Functions

The Oracle Communications Unified Session Manager performs the following functions in the role of P-CSCF:

- Forwards SIP REGISTER messages and maintains a cached mapping of the user info and the UE's Address of Record (AoR), including the far-end NAT address in the case of hosted NAT traversal (HNT).
- Performs local emergency session handling—Local routing policy is used by the Oracle Communications Unified Session Manager to identify emergency sessions and provide unique routing (e.g. can route to a dedicated S-CSCF function for emergency session handling).
- Operates as a UA (B2BUA) for generating independent SIP transactions for security purposes and handling of abnormal conditions.
- Offers current session timers which are used to monitor for media faults and abandoned calls.
- Generation of CDRs—The Oracle Communications Unified Session Manager generates real-time accounting records via RADIUS.
- Authorization of bearer resources and QoS management—With integrated BGF capabilities, the Oracle Communications Unified Session Manager allocates bearer

resources (NAPT flows) and applies QoS policies (including packet marking) based on local policies and/or policies acquired via interaction with the A-RACF (PDF).

- Interaction with the A-RACF (PDF) for session-based policy enforcement and admission control—The Oracle Communications Unified Session Manager PDF interface options include COPS and SOAP/XML.
- Traffic Policing—Traffic is policed at the session and media/transport layer. At the signaling layer, the Oracle Communications Unified Session Manager polices at a number of levels including:
  - Capacity—Total number of concurrent calls to/from each realm
  - Session set-up rate—Maximum rate of call attempts to/from each signaling element
  - Signaling message rate—Each endpoint's signaling message rate is monitored and policed
  - Signaling bandwidth—each endpoint's signaling bandwidth is policed individually

A-BGF Functions

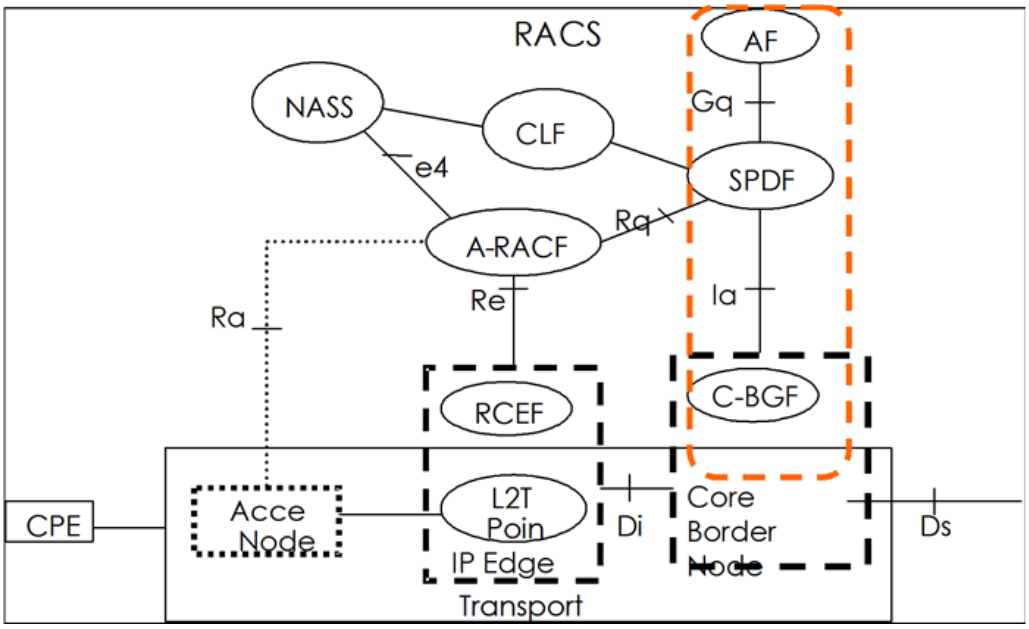
The Oracle Communications Unified Session Manager performs the following access-related BGF functions:

- Opening and closing gates/packet filtering—The Oracle Communications Unified Session Manager opens and closes gates (media pinholes) on a session-by-session basis. Packet filtering rules include full source and destination IP address and port number.
- Per-session DiffServ or ToS marking—Media flows destined for the IMS core network can be explicitly marked using ToS or DiffServ. Media packets can be marked by VPN, by codec (voice, video) or by E.164 phone number prefix.
- NAPT-PT and topology hiding—The Oracle Communications Unified Session Manager provides NAPT for all media flows associated with a session on a per session-basis. Double NATing, NATing both source and destination sides, is utilized to fully hide topology in each direction for RTP and RTCP. Local IP addresses and port resources are dynamically allocated from steering pools provisioned on the Oracle Communications Unified Session Manager.
- Hosted NAT traversal—The Oracle Communications Unified Session Manager supports HNT function that allows media flow traversal through the CPE firewall/NAT without upgrading the CPE equipment. The Oracle USM interacts with the endpoints to dynamically establish and maintain bindings in the CPE firewall/NAT that allow the signaled communications to pass through. The Oracle Communications Unified Session Manager's registration management and media relay functions make CPE-based NATs transparent to the service delivery elements.
- Traffic Policing—Traffic is policed at the session and media/transport layer. At the signaling layer, the Oracle Communications Unified Session Manager polices at a number of levels.
- Policing of Media (e.g. RTP & RTCP) traffic on a per-flow basis—CBR policing is applied to each flow based on offered and negotiated media codecs.

Resource and Admission Control (RACS) Functions

The figure below illustrates the mapping of Oracle Communications Unified Session Manager functions to the RACS functional model. In this model, the Oracle Communications Unified

Session Manager incorporates the Application Function (in the case of IMS this is the P-CSCF function), the SPDF (Service Policy Decision Function) and the Core Border Gateway function.



The Oracle Communications Unified Session Manager, acting as the SPDF, interfaces with the PDF (A-RACF policy decision function) for resource authorization and admission control on a call-by-call basis. COPS is the supported PDF interface.

IMS Interconnect Border Functions

The Oracle Communications Unified Session Manager is deployed at IP interconnect points between service providers to deliver the functions defined in the TISPAN architecture as the Interconnect Breakout Gateway Control Function (I-BGCF). The Oracle Communications Unified Session Manager performs the following functions as the interconnect border Oracle Communications Unified Session Manager:

- Interaction with I-BGF (including NAPT and firewall functions)
- Topology hiding-screening of signalling information

Oracle USM Access Interface Configuration

The Oracle USM supports two interface types. Oracle USM interfaces are core-side interfaces by default, and support IMS and OTT functions for the core. Access-side interfaces perform entirely different functions on the access-side of the Oracle USM. The user must identify access-side SIP interfaces and enable them for access and interconnect-related functions as presented below.

Further configuration may or may not be required for a specific function.

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

   ```
   ORACLE# configure terminal
   ```

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

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

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

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

4. Type **select** and the number of the pre-configured SIP interface you want to configure.

   ```
   ORACLE(sip-interface)# select 1
   ```

5. **ims-access**—Enable IMS access-side functionality on this SIP interface. The default value is **disabled**. Valid values are:

   - enabled | disabled

     ```
     ORACLE(sip-interface)# ims-access enabled
     ```

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

Wildcard PUI Introduction

The Oracle Communications Unified Session Manager supports wildcard Public User Identities (PUI). This capability is most often used to streamline processing of REGISTER and INVITE messages between a PBX and an IMS core.

The HSS (Home Subscriber Server, an IMS network-wide database) is pre-provisioned with the extension numbers associated with the PBX's base telephone number. When the PBX registers its own base telephone number with the Oracle Communications Unified Session Manager, that server downloads a wildcarded PUI -- a regular expression that describes all extension numbers associated with the registering PBX.

The Oracle Communications Unified Session Manager constructs a registration cache entry to implicitly treat subsequent INVITEs from PBXs extensions that match the wildcard as registered endpoints.

A wildcarded PUI consists of a delimited regular expression located either in the user info portion of the SIP URI or in the telephone-subscriber portion of the Tel URI. The regular expression takes the form of an Extended Regular Expressions (ERE) as defined in chapter 9 of The Single UNIX Specification (IEEE 1003.1-2004 Part 1). The exclamation mark (!) serves as the delimiter.

For example, the following PUIs will match to the wildcard ERE "sip:chatlist!.*!@example.com".

```
sip:chatlist1@example.com
sip:chatlist2@example.com
sip:chatlist42@example.com
sip:chatlistAbC@example.com
sip:chatlist!1@example.com
```

Wildcard PUI Message Flows

A basic registration exchange with support for PUIs enabled.

From UA to Oracle Communications Unified Session Manager

```
REGISTER sip:192.168.1.232:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.231:5060;branch=z9hG4bK-13794-1-0
```

```
From: <sip:7818888@hxu.com;tag=1
To: sut <sip:7818888@hxu.com>
Call-ID: 1-13794@192.168.1.231
CSeq: 1 REGISTER
Contact: <sip:7818888@192.168.1.231:5060>;expires=3600
Content-Length: 0
```

From Oracle Communications Unified Session Manager to UA

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.231:5060;branch=z9hG4bK-13794-1-0
From: <sip:7818888@hxu.com;tag=1
To: sut <sip:7818888@hxu.com>;tag=1
Call-ID: 1-13794@192.168.1.231
CSeq: 1 REGISTER
Contact: <sip:7818888@192.168.1.231:5060>;expires=3600
P-Associated-URI: <sip:781!.*!@hxu.com>
Content-Length: 0
```

Incoming Request INVITE from Core

For the incoming request from the core to the Oracle Communications Unified Session Manager, if reg-cache-route is enabled on ingress SIP Interface, the Oracle Communications Unified Session Manager checks the registered P-CSCF-contact-URI in the top Route P-CSCF-contact-URI, instead of with request-URI.

```
INVITE sip:7816666@192.168.200.232:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.200.231:5060;branch=z9hG4bK-13800-1-0
From: sipp <sip:3054252165@192.168.200.231:5060;user=phone>;tag=1
To: sut <sip:7816666@hxu.com>
Call-ID: 1-13800@192.168.200.231
Supported: 100rel,timer,resource-priority,replaces
CSeq: 1 INVITE
Contact: sip:7816666@192.168.200.231:5060
Max-Forwards: 70
Route: <sip:7818888-rrbgth3ot667c@192.168.200.232:5060;lr>
Content-Type: application/sdp
Content-Length:  141

v=0^M
o=user1 53655765 2353687637 IN IP4 192.168.200.231
s=-
c=IN IP4 192.168.200.231
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Outgoing Request INVITE

For outgoing Request from access side, the Register Cache entry will be found by reg-via-key, when options ‘reg-via-key’ and ‘reg-via-match’ are set, the Oracle Communications Unified Session Manager will have wildcarded PAU checking for allow-anonymous (or registered) verification if option wildcard-puid-match is set.

```
INVITE sip:service@192.168.1.232:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.231:5060;branch=z9hG4bK-11911-1-0
From: sipp <sip:7816666@hxu.com>;tag=1
To: sut <sip:service@192.168.1.232:5060>
Call-ID: 1-11911@192.168.1.231
Supported: 100rel,timer,resource-priority,replaces
CSeq: 1 INVITE
```

```
Contact: sip:781666@192.168.1.231:5060
Max-Forwards: 70
Content-Type: application/sdp
Content-Length:  137

v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.231
s=-
c=IN IP4 192.168.1.231
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

IMS Support for Private Header Extensions for 3GPP

As part of its RFC 3455 support, the Oracle Communications Unified Session Manager supports the following headers in its IMS implementation:

- P-Associated-URI
- P-Asserted-Identity
- P-Called-Party-ID
- P-Visited-Network-ID

The procedure to enable IMS access-side support is explained in the access interface configuration section. Any additional configuration required for specific header support is provided below.

P-Associated-URI Header

In the SIP registration process, the registrar often returns a set of associated URIs for a registering AoR. When the Oracle Communications Unified Session Manager receives the list of associated URIs, it stores them in the registration entry for the registering endpoint. The service provider allocates one or more associated URIs per user for his or her own usage. After an endpoint successfully registers, the P-Associated-URI header returned in a 200 OK message informs the UE of all URIs associated with the AoR.

When the Oracle Communications Unified Session Manager receives a request from a UE, the URI in the From header is matched against the registration cache for that endpoint. If the registering endpoint matches an associated-URI already in the registration table, the Service-Route associated with this endpoint is used to create the route for originating transactions associated with the endpoint and the Oracle Communications Unified Session Manager.

The inclusion or exclusion of the P-Associated-URI header is not dependent on the trust level of an ingress or egress realm.

P-Asserted-Identity Header

The Oracle Communications Unified Session Manager inserts a P-Asserted-Identity header into any initial request for a dialog or standalone transaction sourced by the UE.

The inclusion or exclusion of the P-Asserted-Identity header is dependent on the trust level of an egress realm.

P-Asserted-Identity Header Handling

1. The Oracle Communications Unified Session Manager inserts a P-Asserted-Identity header into all messages other than the REGISTER message.
2. When the P-Preferred-Identity header is present in an INVITE sourced by the UE, and the SIP URI contained in this header is also present in the UE's associated URI list, then this SIP URI is inserted in the P-Asserted-Identity header as the SIP message enters the core network.
3. When the P-Asserted-Identity header is present in an INVITE sourced by the UE, and the SIP URI contained in this header is also present in the UE's associated URI list, then the original P-Asserted-Identity header and SIP URI is passed unchanged into the core network.
4. When the From header is present in an INVITE sourced by the UE, and the SIP URI contained in this header appears in the UE's Associated URI list, then this SIP URI is inserted into the P-Asserted-Identity header as the SIP message enters the core network.
5. When the P-Asserted-Identity header is present in an INVITE sourced by the UE, and the SIP URI contained in this header is not present in the Associated URI list, the Oracle Communications Unified Session Manager acts like no P-Asserted-Identity was received from the UE.
6. When no P-Asserted-Identity can be derived from an INVITE sourced by the UE, the P-Asserted-Identity is based on the first URI in the Associated URI list.
7. The P-Asserted-Identity header will be removed from SIP messages sent and received from a UE if either the ingress or egress side is untrusted and the UE’s Privacy header’s contents is id.
8. If no P-Associated-URI exists for a registered endpoint, the Oracle Communications Unified Session Manager will use the configured default P-Asserted-Identity found on the sourcing session agent. This feature works with SIP session agents.
9. If the session agent that originates a message does not include a P-Asserted-Identity header or the request is not originated from the session agent, and the Oracle Communications Unified Session Manager has not received P-Associated-URI list from the registrar for a particular user, no P-Asserted-Identity will be created.
10. The P-Preferred-Identity header will never be passed to the Oracle Communications Unified Session Manager's S-CSCF function.

    If the above steps fail to insert a P-Asserted-Identity header, you can manually configure a value to be inserted into a P-Asserted-Identity header. The ims-access parameter must be enabled on the access SIP interface to use the P-Asserted-Identity header override.

P-Asserted-Identity Header Configuration

This section explains how to configure the P-Asserted-Identity header for a session agent using the ACLI or Oracle Communications Session Delivery Manager.

P-Asserted-Identity header handling is enabled with the **ims-access** command on the SIP Interface, as described in the previous section. A P-Asserted-Identity header can be manually configured for a session agent if the automatic logic, explained earlier in this section, fails.

To configure the P-Asserted-Identity header for a session agent:

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

```
ORACLE# configure terminal
```

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

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

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

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

4. Type **select** and the number of the pre-configured session agent you want to configure.

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

5. Type **p-asserted-id** <space> <URI to use when no P-Asserted-ID has been created> and press Enter. This completes the configuration.

```
ORACLE(session-agent)# p-asserted-id sip:name@acmepacket.com
```

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

P-Called-Party-ID Header

The Oracle Communications Unified Session Manager transparently passes the P-Called-Party-ID header between the S-CSCF and a UA.

P-Early-Media SIP Header Support

Version S-CZ7.2.0 provides support for the SIP P-Early-Media that can be used in SIP INVITES, PRACKS, and UPDATES to request and authorize the use of early media. It offers an alternative to policy-based early media support.

The P-Early-Media SIP header is defined in RFC 5009, Private Header (P-Header) Extension to the Session Initiation Protocol (SIP) for Authorization of Early Media. RFC 5009 defines the use of the P-Early-Media header within SIP messages in certain SIP networks to authorize the cut-through of backward (server-to-client) and/or forward (client-to-server) early media when permitted by the early media policies of the networks involved. The P-Early-Media header field is intended for use in a SIP network, such as a 3GPP IMS that has the following characteristics: its early media policy prohibits the exchange of early media between end users; it is interconnected with other SIP networks that have unknown, untrusted, or different policies regarding early media; and it has the capability to gate (enable/disable) the flow of early media to/from user equipment.

P-Early-Media SIP Header

The P-Early-Media SIP header is defined in RFC 5009, Private Header (P-Header) Extension to the Session Initiation Protocol (SIP) for Authorization of Early Media. RFC 5009 defines the use of the P-Early-Media header within SIP messages in certain SIP networks to authorize the cut-through of backward (server-to-client) and/or forward (client-to-server) early media when permitted by the early media policies of the networks involved. The P-Early-Media header field is intended for use in a SIP network, such as a 3GPP IMS that has the following characteristics: its early media policy prohibits the exchange of early media between end users; it is interconnected with other SIP networks that have unknown, untrusted, or different policies regarding early media; and it has the capability to gate (enable/disable) the flow of early media to/from user equipment.

A SIP network containing both PSTN gateways and SIP end devices, for example, can maintain such an early media policy by gating "off" any early media with a SIP end device acting as UAS, gating "on" early media with a SIP end device acting as UAC, and gating "on" early media at each PSTN gateway. This is what we have been doing for years with the SIP early media suppression feature, which allows determining who can send early media and in what direction.

Unfortunately, in SIP interconnection scenarios there is no means of assuring that the interconnected network is implementing a compatible early media policy, thus allowing the exchange of user data within early media under some circumstances. For example, if a network "A" allows all early media with user equipment as UAC and an interconnected network "B" allows all early media with user equipment as UAS, any session established between user equipment as UAC in "A" and user equipment as UAS in "B" will allow bidirectional user data exchange as early media.

The P-Early-Media header is used for the purpose of requesting and authorizing requests for backward and/or forward early media. It's sent from UAS to UAC to indicate authorization for early media.

P-Early-Media-Header Usage

The syntax of the P-Early-Media header field is as follows.

```
P-Early-Media = "P-Early-Media" HCOLON[ em-param *(COMMA em-param) ]
       em-param  = "sendrecv" / "sendonly" / "recvonly" / "inactive" /
"gated" /
      "supported" / token
```

The P-Early-Media header is used for requesting and authorizing requests for backward and/or forward early media. The P-Early-Media header field in an INVITE request contains the "supported" parameter. If P-CSCF is part of the trusted domain, then it must decide whether to insert or delete the P-Early-Media header field before forwarding the INVITE. The P-CSCF upon receiving the P-Early-Media header field in a message towards the UAC needs to verify that the early media request comes from an authorized source. If a P-Early-Media header field arrives from either an untrusted source, a source not allowed to send backward early media, or a source not allowed to receive forward early media, then it may remove the P-Early-Media header field or alter the direction parameter(s) of the P-Early-Media header field before forwarding the message, based on local policy.

The P-Early-Media header field with the "supported" parameter in an INVITE request indicates that the P-CSCF on the path recognizes the header field. The P-Early-Media header field includes one or more direction parameters where each has one of the values: "sendrecv", "sendonly", "recvonly", or "inactive", following the convention used for SDP stream directionality. Each parameter applies, in order, to the media lines in the corresponding SDP messages establishing session media. The parameter value "sendrecv" indicates a request for authorization of early media associated with the corresponding media line, both from the UAS towards the UAC and from the UAC towards the UAS. The value "sendonly" indicates request for authorization of early media from the sender to the receiver and not in the other direction. The value "recvonly" indicates a request for authorization of early media from the receiver, and not in the other direction. The value "inactive" indicates either a request that no early media associated with the corresponding media line be authorized, or a request for revocation of authorization of previously authorized early media. Each parameter applies, in order, to the media lines in the corresponding SDP lines. Unrecognized parameters are discarded and non-direction parameters are ignored. If there are more direction parameters than media lines, the excess are silently discarded. If there are fewer direction parameters than media lines, the value of the last direction parameter applies to all remaining media lines. The P-Early-Media header field in any message within a dialog towards the sender of the INVITE request can also include

the non-direction parameter "gated" to indicate that a network entity on the path towards the UAS is already gating the early media, according to the direction parameter(s).

As defined in 3GPP TS 24.229, IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 both the P-CSCF and IBCF may add, remove, or modify, the P-Early-Media header field within forwarded SIP requests and responses according to procedures in RFC 5009.

The P-CSCF can use the P-Early-Media header field for the gate control procedures, as described in 3GPP TS 29.214. Prior to Version S-CZ7.2.0, this capability was based on policy configuration, not examination of the P-Early-Media header.

In the current implementation, if the configuration option “early-media-allow” is set to none, the Oracle Communications Unified Session Manager will send the Flow-Status AVP in any AAR request set to disable until the final response.

Functional Design

Acceptance of and authorization for early media is accomplished with two new ACLI parameters -- **p-early-media-header** and **p-early-media-direction**, which are added to SIP interface configuration in Version S-CZ7.2.0 and later releases.

The **p-early-media-header** parameter will enable the feature when the value is set to either “add” or “modify”. The **p-early-media-header** and **p-early-media-direction** should be configured on egress interface of the incoming message. The values for parameter **p-early-media-direction** are “sendrecv, sendonly, recvonly, inactive”. It is a list and each configured value corresponds to the m-line in the SDP. If the number of configured values is more than the number of m-lines in the SDP, the excess configured values are ignored. If the number of configured value is less than the number of m-lines in the SDP, the last configured value is used for all the m-lines.

The following illustrations show the ingress and egress sip-interface configuration.

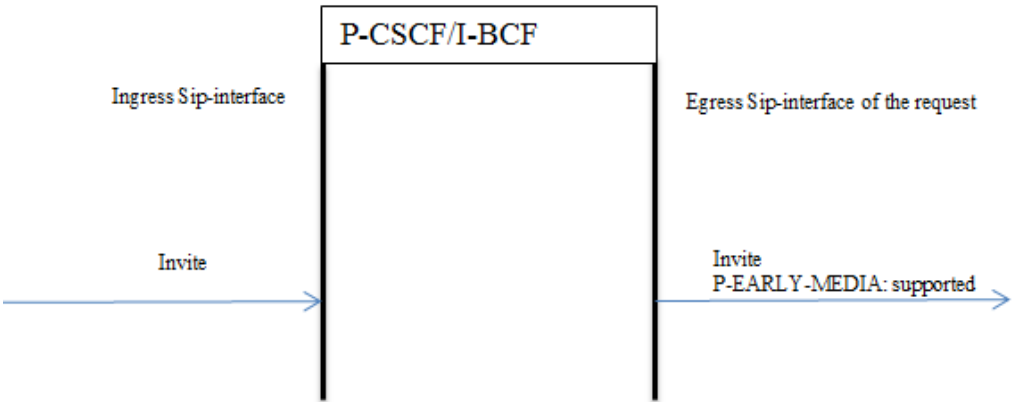
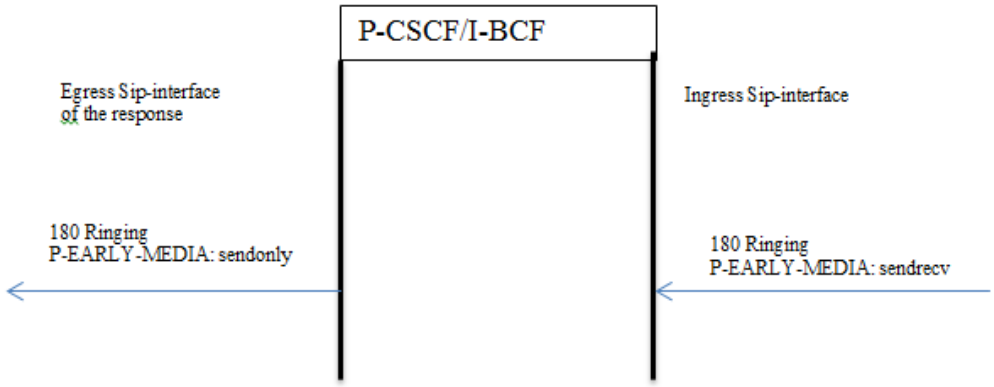


Figure 4-1 180 RINGING Specifying p-early-media Support



P-Early-Media headers in Re-Invites are ignored. If the SDP contains Content-Disposition: early-session the P-Early-Media header is ignored.

Endpoint is considered trusted or untrusted based on the configuration on the ingress sip-interface of the P-CSCF. Sip-interface has the configuration parameter “trust-mode”. If the “trust-mode” is set to “none” then nobody is trusted in sip-interface. By default the value is “all”. Possible values are <all, agents-only, realm-prefix, registered, none>.

For multiple dialogs due to forking, P-CSCF will identify the media associated with a dialog, and then setup early media flow for the selected media. The configuration elements restricted-latching in realm-config, and latching in media-router should be enabled.

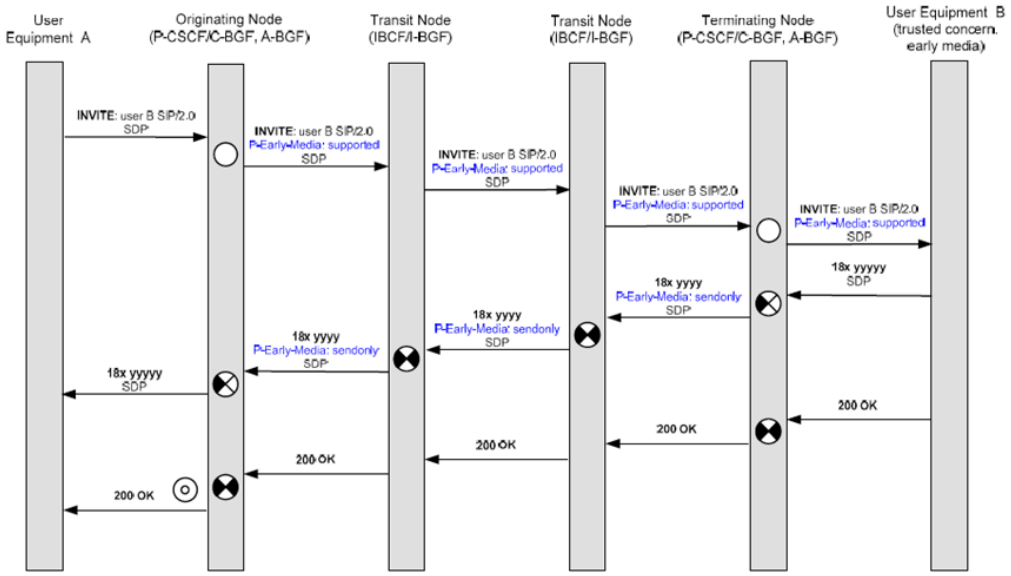
The following 3 tables detail early media implementation details.

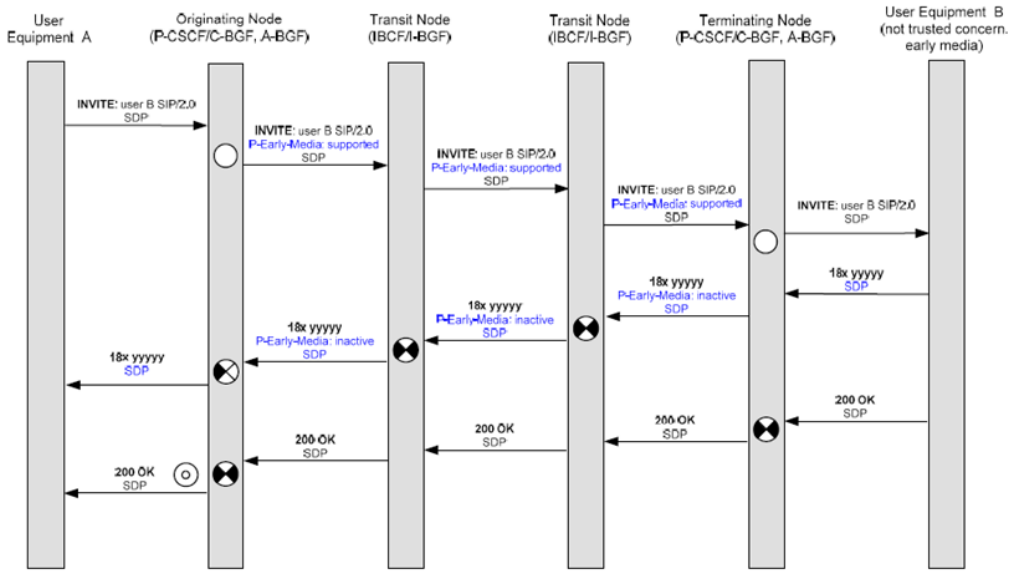
P-Early-Media Trusted to Trusted

This table illustrates the P-CSCF case when messages are received from trusted endpoints and forwarded to trusted endpoints.

Message Parameters configured on egress interface	**Request (Invite, UPDATE, PRACK) w/o header**	**Request (Invite, UPDATE, PRACK) with header with one or more direction parameters where each has one of the values: "sendrecv", "sendonly", "recvonly", or"inactive"**	**Response (18x, 200OK(UPDATE/ PRACK)) w/o header**	**Response (18x, 200OK(UPDATE/ PRACK) with header with one or more direction parameters where each has one of the values: "sendrecv", "sendonly", "recvonly", or"inactive"**
p-early-media-header-"add" p-early-media-direction -"sendonly"	Add header with "supported" value. Setup flows based on SDP value	Add "supported" value if not present. Setup the flows based on the value in the SDP value.	Setup flows based on local config PEM value. Add the PEM header based on local config value. Status Flow AVP in AAR message updated.	Setup the flows based on the value in the incoming PEM header.

Message Parameters configured on egress interface	Request (Invite, UPDATE, PRACK) w/o header	Request (Invite, UPDATE, PRACK) with header with one or more direction parameters where each has one of the values: "sendrecv", "sendonly", "recvonly", or"inactive"	Response (18x, 200OK(UPDATE/ PRACK)) w/o header	Response (18x, 200OK(UPDATE/ PRACK) with header with one or more direction parameters where each has one of the values: "sendrecv", "sendonly", "recvonly", or"inactive"
p-early-media-header-"modify" p-early-media-direction -"sendonly"	Add header with "supported" value. Setup flows based on SDP value	Add "supported" value if not present. Setup the flows based on the SDP value	Setup flows based on local config PEM value. Add the PEM header based on local config value. Status Flow AVP in AAR message updated.	Setup flows based on local config PEM value. Modify the PEM header based on local config value. Status Flow AVP in AAR message updated.
p-early-media-header-"add" No p-early-media-direction	Add header with "supported" value. Setup flows based on SDP value	Add "supported" value if not present. Setup the flows based on the SDP.	Setup flows based on local config PEM value. Add the PEM header based on default value. Status Flow AVP in AAR message updated.	Setup flows based on local config PEM value. Modify the PEM header based on default value. Status Flow AVP in AAR message updated.
p-early-media-header-"modify" No p-early-media-direction	Add header with "supported" value. Setup flows based on SDP value.	Add header with "supported" value if not present. Setup the flows based on the SDP.	config PEM value. Add the PEM header based on default value. Status Flow AVP in AAR message updated.	Setup flows based on local config PEM value. Modify the PEM header based on default value. Status Flow AVP in AAR message updated.





P-Early-Media Untrusted to Trusted

This table illustrates the P-CSCF case when messages are received from untrusted endpoints and forwarded to trusted endpoints.

Message Parameters configured on egress interface	Request (Invite, UPDATE, PRACK) w/o header	Request (Invite, UPDATE, PRACK) with header with one or more direction parameters where each has one of the values: "sendrecv", "sendonly", "recvonly", or"inactive"	Response (18x, 200OK(UPDATE/ PRACK)) w/o header	Response (18x, 200OK(UPDATE/ PRACK) with header with one or more direction parameters where each has one of the values: "sendrecv", "sendonly", "recvonly", or"inactive"
p-early-media-header-"add" p-early-media-direction -"sendonly"	Add header with "supported" value. Setup flows based on SDP value	Discard the header. Add the header with supported value. Setup flows based on SDP value.	Setup flows based on local config PEM value. Add the PEM header based on local config value. Status Flow AVP in AAR message updated.	Discard the header. Setup flows based on local config PEM value. Add the PEM header based on local config value. Status Flow AVP in AAR message updated.
p-early-media-header-"modify" p-early-media-direction -"sendonly"	Add header with "supported" value. Setup flows based on SDP value	Discard the header. Add the header with supported value. Setup flows based on SDP value.	Setup flows based on local config PEM value. Add the PEM header based on local config value. Status Flow AVP in AAR message updated.	based on local config PEM value. Add the PEM header based on local config value. Status Flow AVP in AAR message updated.
p-early-media-header-"add" No p-early-media-direction	Add header with "supported" value. Setup flows based on SDP value.	Discard the header. Setup flows based on SDP value.	Setup flows based on local config PEM value. Add the PEM header based on default value. Status Flow AVP in AAR message updated.	Discard the header. Setup flows based on local config PEM value. Add the PEM header based on default value. Status Flow AVP in AAR message updated.
p-early-media-header-"modify" No p-early-media-direction	Add header with "supported" value. Setup flows based on SDP value	Discard the header. Setup flows based on SDP value.	Setup flows based on local config PEM value. Add the PEM header based on default value. Status Flow AVP in AAR message updated.	Discard the header. Setup flows based on local config PEM value. Add the PEM header based on default value. Status Flow AVP in AAR message updated.

P-Early-Media Trusted to Untrusted

This table illustrates the P-CSCF case when messages are received from trusted endpoints and forwarded to untrusted endpoints.

Message Parameters configured on egress interface	Request (Invite, UPDATE, PRACK) w/o header	Request (Invite, UPDATE, PRACK) with header with one or more direction parameters where each has one of the values: "sendrecv", "sendonly", "recvonly", or"inactive"	Response (18x, 200OK(UPDATE/ PRACK)) w/o header.	Response (18x, 200OK(UPDATE/ PRACK) with header with one or more direction parameters where each has one of the values: "sendrecv", "sendonly", "recvonly", or"inactive"
p-early-media-header-"add" p-early-media-direction -"sendonly"	Setup flows based on SDP value.	Discard the header. Setup flows based on SDP value.	Setup flows based on SDP value.	Discard the header. Setup flows based on local config PEM value. Status Flow AVP in AAR message updated.
p-early-media-header-"modify" p-early-media-direction -"sendonly"	Setup flows based on SDP value.	Discard the header. Setup flows based on SDP value.	Setup flows based on SDP value.	Discard the header. Setup flows based on local config PEM value. Status Flow AVP in AAR message updated.
p-early-media-header-"add" No p-early-media-direction	Setup flows based on SDP value.	Discard the header. Setup flows based on SDP value.	Setup flows based on SDP value.	Discard the header. Setup flows based on default PEM value. Status Flow AVP in AAR message updated.
p-early-media-header-"modify" No p-early-media-direction	Setup flows based on SDP value.	Discard the header. Setup flows based on SDP value.	Setup flows based on SDP value.	Discard the header. Setup flows based on default PEM value. Status Flow AVP in AAR message updated.

P-Early-Media ACLI Configuration

Use the following procedure to configure P-Early-Media SIP header support.

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

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

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

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

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

3. Use the **p-early-media-header** parameter to enable P-Early-Media SIP header support. This parameter is disabled by default.
   - **disabled**—(the default value) disables support
   - **add**—enables support and allows the Oracle Communications Unified Session Manager to add the P-Early-Media header to SIP messages.
   - **modify**—enables support and allows the Oracle Communications Unified Session Manager to modify or strip the P-Early-Media header in SIP messages.

   ```
   ACMEPACKET(sip-interface)# p-early-media-header add
   ACMEPACKET(sip-interface)#
   ```
4. Use the **p-early-media-direction** parameter to specify the supported directionalities.
   - **sendrecv**—send and accept early media
   - **sendonly**—send early media
   - **recvonly**—receive early media
   - **inactive**—reject/cancel early media

   ```
   ACMEPACKET(sip-interface)# p-early-media-direction sendrecv,sendrecv
   ACMEPACKET(sip-interface)#
   ```
5. Type **done** to save your configuration.

P-Visited-Network-ID Header

The Oracle Communications Unified Session Manager's IMS support also includes the insertion of a P-Visited-Network-ID header into SIP messages when applicable. When a UE sends a dialog-initiating request (e.g., REGISTER or INVITE message) or a standalone request outside of a dialog (e.g., OPTIONS) to the Oracle Communications Unified Session Manager, it inserts the P-Visited-Network-ID header into the SIP message as it enters into the destination network.

The P-Visited-Network ID header will be stripped from SIP messages forwarded into untrusted networks as expected. The content of a P-Visited-Network-ID header is a text string that identifies the originating UE's home network. This string is user-configurable.

P-Visited-Network-ID Header Handling for SIP Interfaces Configuration

The actual P-Visited-Network-ID string must be configured on the access-side SIP interface. The Oracle Communications Unified Session Manager must consider the egress device trusted or it does not add that the P-Visited-Network-ID header to the forwarded request.

To configure the P-Visited-Network-ID string in a SIP interface:

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

   ```
   ORACLE# configure terminal
   ```
2. Type **session-router** and press Enter to access the session-level configuration elements.

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

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

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

4. Type **select** and the number of the pre-configured sip interface you want to configure.

```
ORACLE(sip-interface)# select 1
```

5. Type **network-id** <space> <network ID string> and press Enter. This completes the configuration of the P-Visited-Network-ID string for a given SIP interface.

```
ORACLE(sip-interface)# network-id examplenetworkid
```

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

Second P-Asserted-Identity Header for Emergency Calls

The Oracle Communications Unified Session Manager can add a second P-Asserted-Identity header when forwarding an emergency message into the network.

When the UE registers with an S-CSCF, the S-CSCF returns a set of associated URIs, the implicit registration set (IRS,) for the AoR in the 200 OK response. The Oracle Communications Unified Session Manager caches the IRS. The user identities that comprise the cached IRS are used for validation later.

As the Oracle Communications Unified Session Manager receive a UE's INVITE, the value in the P-Preferred-Identity header is validated against the public user identities in the cached IRS. If the inbound P-Preferred-Identity matches any items in the IRS, the Oracle Communications Unified Session Manager inserts that value into a P-Asserted-Identity header in the egress message. The P-Preferred- Identity header is stripped from the outbound message.

Inbound INVITE Contains:	Validate Against Implicit Registration Set	Outbound Invite Created With: (all P-Preferred-Identity headers are removed)
P-Preferred-Identity: value-1	value-1 present	P-Asserted-Identity: value-1

The Oracle Communications Unified Session Manager can create a second P-Asserted-Identity header by configuring the add-second-pai option in the SIP config. Once a combination of SIP URIs and/or TEL URIs in the inbound P-Preferred-Identity header are validated against the IRS, the Oracle Communications Unified Session Manager forwards the emergency INVITEs with two P-Asserted-Identity headers to the E-CSCF. The behavior is based upon the URI type received in the P-Preferred-Identity header and whether those values validate against the IRS list.

Inbound INVITE Contains:	Validate Against Implicit Registration Set	Outbound Invite Created With: (all P-Preferred-Identity headers are removed)
P-Preferred-Identity: SIP-URI-1	SIP-URI-1 present TEL-URI-1 present (TEL-URI-n present)	P-Asserted-Identity: SIP-URI-1 P-Asserted-Identity: TEL-URI-1
P-Preferred-Identity: TEL-URI-1	TEL-URI-1 present SIP-URI-1 present (SIP-URI-n present)	P-Asserted-Identity: TEL-URI-1 P-Asserted-Identity: SIP-URI-1

Inbound INVITE Contains:	Validate Against Implicit Registration Set	Outbound Invite Created With: (all P-Preferred-Identity headers are removed)
NO P-Preferred-Identity:	(SIP | TEL) URI -1 present (SIP | TEL) URI -n present	P-Asserted-Identity: 1st public identity from IRS
P-Preferred-Identity: SIP-URI-1	TEL-URI-1 present	P-Asserted-ID: SIP-URI-1
P-Preferred-Identity: TEL-URI-1	SIP-URI-1 present	P-Asserted-ID: TEL-URI-1
P-Preferred-Identity: SIP-URI-1	SIP-URI-1 present	P-Asserted-Identity: SIP-URI-1
P-Preferred-Identity: SIP-URI-2	SIP-URI-2 present TEL-URI-n present	P-Asserted-Identity: 1st TEL-URI from IRS
P-Preferred-Identity: TEL-URI-1	TEL-URI-1 present	P-Asserted-Identity: TEL-URI-1
P-Preferred-Identity: TEL-URI-2	TEL-URI-2 present SIP-URI-n present	P-Asserted-Identity: 1st SIP-URI from IRS

If the INVITE does not include a P-Preferred-Identity header and does not include a P-Asserted-Identity header, or the value in the original P-Preferred-Identity or P-Asserted-Identity header is not contained in the IRS, or the URI from the From: header is not the in the IRS, then default public user identity is inserted into a P-Asserted- Identity header in the egress message. (The default public user identity is the first on the list of URIs in the P-Associated-URI header.)

If the strict-3gpp-pai-compliance option is configured in the outbound SIP interface, the first P-Asserted-Identity header also includes the display name.

Two Incoming P-Asserted-Identity Headers

When the inbound INVITE contains 2 P-Asserted-Identity headers, the Oracle Communications Unified Session Manager ensures that both outbound P-Asserted-Identity headers contain public user identities from the IRS according to the following:

Inbound Invite Contains	Validate Against Implicit Registration Set:	Outbound Invite Created With:
P-Asserted-Identity: SIP-URI-1 P-Asserted-Identity: TEL-URI-1	SIP-URI-1 and TEL-URI-1 present	P-Asserted-Identity: SIP-URI-1 or TEL-URI-1 P-Asserted-Identity: 1st public identity from IRS other than value in 1st P-Asserted-Identity
P-Asserted-Identity: SIP-URI-1	SIP-URI-1 or SIP-URI-2 present	P-Asserted-Identity: SIP-URI-1
P-Asserted-Identity: SIP-URI-2	TEL-URI-1 present	P-Asserted-Identity: TEL-URI-1
P-Asserted-Identity: TEL-URI-1 P-Asserted-Identity: TEL-URI-2	TEL-URI-1 or TEL-URI-2 present SIP-URI-1 present	P-Asserted-Identity: TEL-URI-1 or TEL-URI-2 P-Asserted-Identity: SIP-URI-1

1. Navigate to the sip-config configuration element.

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

2. Type select to begin configuring this object.

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

3. options—Configure the add-second-pai option:

```
ACMEPACKET(sip-config)# options +add-second-pai
ACMEPACKET(sip-config)#
```

4. Save and activate your configuration.

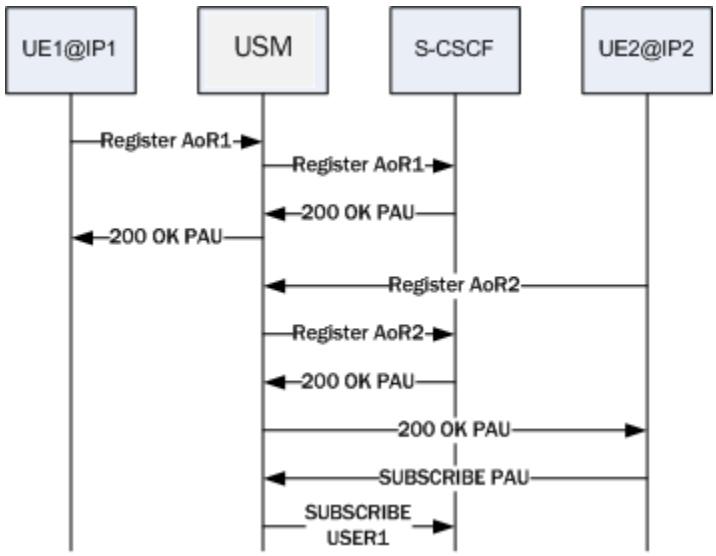
Temporary Public User Identities and Multi-SIM Scenarios

The Oracle Communications Unified Session Manager's SIP interface supports multiple registered users for the same P-Asserted-Uri (PAU), a useful ability for multi-SIM scenarios. The call flow for this type of scenario differs depending on whether or not you configure the SIP interface facing the UE with the **reg-via-key** and **reg-via-match** options.

In a multi-SIM scenario, the UE derives a temporary IMS public identity (IMPU); that UE then registers with the Oracle Communications Unified Session Manager using the IMPU as the address of record (AoR) from a unique IP. The S-CSCF returns a PAU in the 200 OK, which the Oracle Communications Unified Session Manager caches. The UE then derives another IMPU; it registers with the Oracle Communications Unified Session Manager using that IMPU as the AoR from another unique IP. The S-CSCF again returns the same PAU in the 200OK.

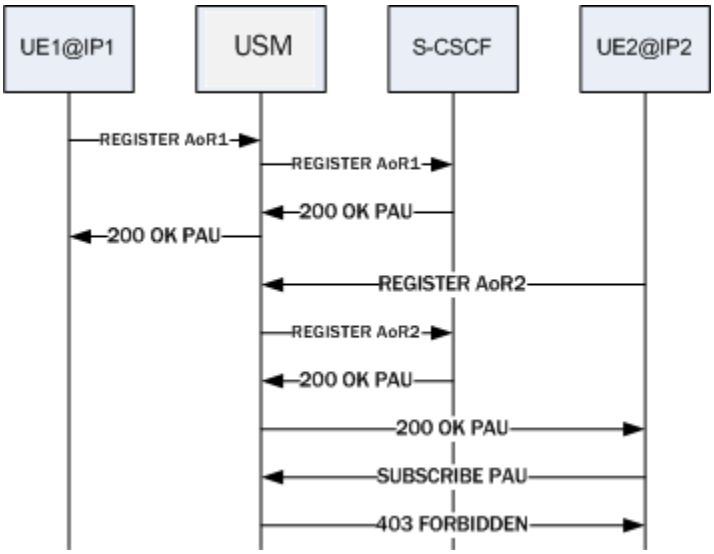
Old Behavior

Before the introduction of this change, the Oracle Communications Unified Session Manager (OCUSM) associated the PAU only with the first IMPU request. The OCUSM considered any request made from that PAU to be a request from the first user, regardless of the request's originating IP.



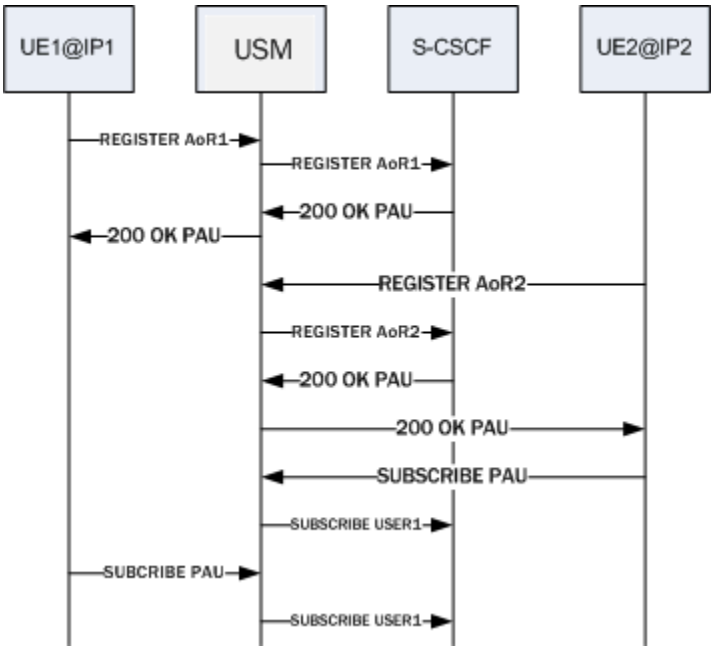
When the SIP interface facing the UE had the **reg-via-key** and **reg-via-match** options configured, the OCUSM rejected the request from the second user with the PAU as the From or

the PPI. Because it only associates the PAU with the first user, the OCUSM issued a 403 message.

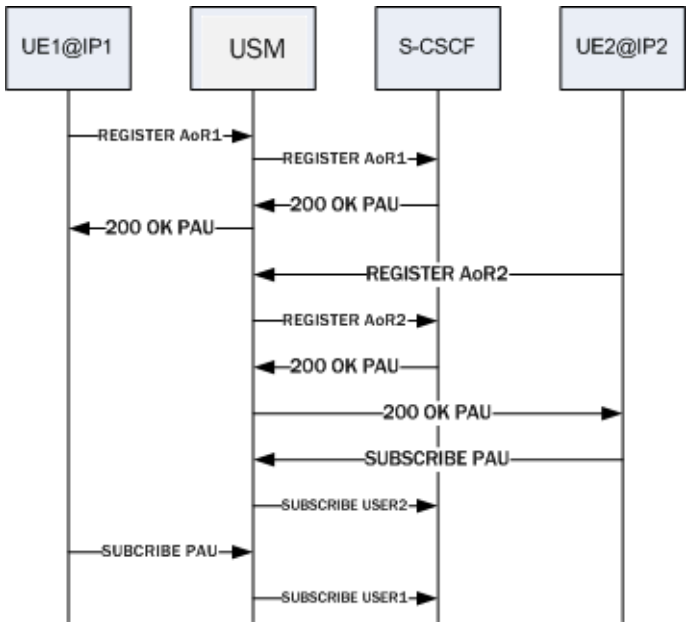


New Behavior

Your Oracle Communications Unified Session Manager (OCUSM) now associates the PAU with both the first and second IMPU. The **OCUSM** considers any request made from that PAU to be a request from the user at the top of its registration cache table irrespective of where the request originated (the IP).



When the SIP interface facing the UE has the **reg-via-key** and **reg-via-match** options configured, and the request from the user with the PAU as the From or with PPI, the **Oracle Communications Unified Session Manager**matches to the proper user based on the source IP.



Configuring SIP Interface with reg-via-key and reg-via-match

If you do not want the use the call scenario associated with the SIP interface options in the New Behavior section, you do not need to make any change to your configuration.

If you want your call scenarios to resemble the one associated with the SIP interface options in the New Behavior section, then you need to configure **reg-via-key** and **reg-via-match** options on the SIP interface facing the UE.

To configure a SIP interface with the reg-via-key and reg-via-match options:

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

   ```
   ORACLE# configure terminal
   ```

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

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

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

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

   If you are adding support for this feature to a pre-existing configuration, then you must select (using the ACLI **select** command) the configuration that you want to edit.

4. **options**—Set the options parameter by typing options, a Space, and then the option name. Then press Enter.

   ```
   ORACLE(sip-interface)# options +reg-via-key
   ORACLE(sip-interface)# options +reg-via-match
   ```

If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to this configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

5. Save your work.

IMS-AKA

The Oracle Communications Unified Session Manager supports IP Media Subsystem-Authentication and Key Agreement (IMS-AKA).

Defined in 3GPPr7 (specifications in TS 33.203 and call flows in TS 24.228), IMS-AKA can be used as a framework for authentication and for securing the signaling path between a UE and the Oracle Communications Unified Session Manager (when the Oracle Communications Unified Session Manager is acting as a P-CSCF or as a B2BUA) across the Gm interface.

In addition, the Oracle Communications Unified Session Manager's serving as an IMS-AKA termination point is valuable because it allows IMS-AKA use behind by multiple endpoints sitting behind a NAT device. IMS-AKA support also works when there are no NAT devices between endpoints and the Oracle Communications Unified Session Manager acting as a P-CSCF, and when the Oracle Communications Unified Session Manager sits behind a third-party P-CSCF. In addition, you can use IMS-AKA when the endpoint uses SIP UDP.

Requirements

IMS-AKA use assumes that you have installed the appropriate IPSec module on your Oracle Communications Unified Session Manager, or that it has come from Oraclewith those modules pre-installed. IMS-AKA will not work without this hardware.

IMS-AKA deployments require an activated **network-parameters** element configured with the options shown below.

```
options                         atcp-rxmt-count=2
                                atcp-rxmt-interval=2
                                atcp-syn-rxmt-interval=2
                                atcp-syn-rxmt-maxtime=6
                                atcp-idle-timer=3700
```

In addition, your configuration must have SIP registration caching enabled.

The refreshRegForward Option

The Oracle Communications Unified Session Manager provides a the user with a means of ignoring its registration refresh half-life timer, and send all applicable registration refreshes received via IMS-AKA to the core for authentication.

By default, the Oracle Communications Unified Session Manager uses its half-life function and attempts to manage registration refreshes prior to half-life expiry without forwarding the refresh to the core. The Oracle Communications Unified Session Manager sends registration refreshes that arrive after the half-life expiry to the core.

The user changes this behavior by setting the **refreshRegForward** in the applicable IMS-AKA profile to as follows.

```
ORACLE(ims-aka-profile)# options +refreshRegForward
```

When this option is set, the system forwards every refresh registration to the IMS core regardless of the half-life timer's status.

Monitoring

The ACLI **show sipd endpoint-ip** command is updated to show the IMS-AKA parameters corresponding to each endpoint. The display shows the algorithms used, the ports used, and the security parameter indexes (SPIs) used.

In addition, the **show sa stats** command now shows the security associations information for IMS-AKA.

ACLI Instructions and Examples

You enable IMS-AKA by configuring the following:

- An IMS-AKA profile
- Certain parameters in the global IPSec configuration
- Certain parameters in the SIP interface, and in the SIP interface's SIP port

Setting Up an IMS-AKA Profile

An IMS-AKA profile establishes the client and server ports to be protected, and it defines lists of encryption and authentication algorithms the profile supports. You can configure multiple IMS-AKA profiles, which are uniquely identified by their names.

You apply an IMS-AKA profile to a SIP port configuration using the name.

To configure an IMS-AKA profile:

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

   ```
   ORACLE# configure terminal
   ORACLE(configure)#
   ```

2. Type **security** and press Enter.

   ```
   ORACLE(configure)# security
   ORACLE(security)#
   ```

3. Type **ims-aka-profile** and press Enter.

   ```
   ORACLE(system)# ims-aka-profile
   ORACLE(ims-aka-profile)#
   ```

4. **name**—Enter the name you want to give this IMS-AKA profile. This is the value you will use to apply the profile to a SIP port configuration. This parameter is required, and it has no default value.

5. **protected-server-port**—Enter the port number of the protected server port, which is the port on which the Oracle Communications Unified Session Manager receives protected messages. The protected server port should not overlap with the port range defined in the steering ports configuration using the same IP address and the SIP interface. If there is overlap, the NAT table entry for the steering port used in a call will prevent SIP messages from reaching the system's host processor.

   This parameter defaults to 0, which disables the function associated with the parameter. The valid range for values is 1025 to 65535.

6. **protected-client-port**—Enter the port number of the protected client port, which is the port on which the Oracle Communications Unified Session Manager sends out protected messages. Like the protected server port, the protected client port should not overlap with the port range defined in the steering ports configuration using the same IP address and the SIP interface. If there is overlap, the NAT table entry for the steering port used in a call will prevent SIP messages from reaching the system's host processor.

   This parameter defaults to 0, which disables the function associated with the parameter. The valid range for values is 1025 to 65535.

7. **encr-alg-list**—Enter the list of encryption algorithms. You enter more than one value by separating the algorithms by <Spaces> and enclosing all values in quotations marks:

   ```
   ORACLE(ims-aka-profile)# encr-alg-list "aes-cbc null"
   ```

   This parameter defaults to the following three values: **aes-cbc**, **des-ede3-cbc**, and **null**.

8. **auth-alg-list**—Enter the list of authentication algorithms. You enter more than one value by separating the algorithms by <Spaces> and enclosing all values in quotations marks:

   ```
   ORACLE(ims-aka-profile)# auth-alg-list "hmac-sha-1-96 hmac-md5-96"
   ```

   This parameter defaults to hmac-sha-1-96.

Setting Up an IPSec Profile for IMS-AKA Use

Using the global IPSec configuration, you establish the parameters governing system-wide IPSec functions and behavior. This configuration also contains parameters required for IMS-AKA support. The IPSec global configuration is a single instance element, meaning there is one for the whole system.

To configure the global IPSec parameters that apply to IMS-AKA:

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

   ```
   ORACLE# configure terminal
   ORACLE(configure)#
   ```

2. Type **security** and press Enter.

   ```
   ORACLE(configure)# security
   ORACLE(security)#
   ```

3. Type **ipsec** and press Enter.

   ```
   ORACLE(system)# ipsec
   ORACLE(ipsec)#
   ```

4. Type **ipsec-global-config** and press Enter. If you are editing a pre-existing IPsec global configuration, then you need to select the configuration before attempting to edit it.

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

5. **red-ipsec-port**—Specify the port on which the Oracle Communications Unified Session Manager should listen for redundancy IPSec synchronization messages. The default is 1994, and valid values are in the range from 1025 to 65535.

6. **red-max-trans**—Enter the maximum number of redundancy transactions to retain on the active. The default is 10000, and valid values range up to a 999999999 maximum.

7. **red-sync-start-time**—Enter the time in milliseconds before the system starts to send redundancy synchronization requests. The default is 5000, and valid values range up to a 999999999 maximum.

8. **red-sync-comp-time**—Enter the time in milliseconds to define the timeout for subsequent synchronization requests once redundancy synchronization has completed. The default is 1000, and valid values range up to a 999999999 maximum.

Enabling IMS-AKA Support for a SIP Interface

To enable IMS-AKA for a SIP interface, you must set the **sec-agree-feature** parameter to enabled.

To enable IMS-AKA for a SIP interface:

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

```
ORACLE# configure terminal
ORACLE(configure)#
```

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

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

3. Type **sip-interface** and press Enter. If you are adding this feature to a pre-existing SIP interface, you need to select and edit that configuration.

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

4. **sec-agree-feature**—Change this parameter to **enabled** if you want to use IMS-AKA on this SIP interface. By default, this parameter is **disabled**.

Applying an IMS-AKA Profile to a SIP Port

The final step in setting up IMS-AKA support is to apply an IMS-AKA profile to a SIP port. Enter the **name** value from the IMS-AKA profile you want to apply in the SIP port's **ims-aka-profile** parameter.

To apply an IMS-AKA profile to a SIP port:

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

```
ORACLE# configure terminal
ORACLE(configure)#
```

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

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

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

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

4. Type **sip-interface** and press Enter. If you are adding this feature to a pre-existing SIP port, you need to select and edit that configuration.

```
ORACLE(session-interface)# sip-ports
ORACLE(sip-port)#
```

5. **ims-aka-profile**—Enter the **name** value for the IMS-AKA profile configuration you want applied to this SIP port. This parameter has no default.

6. Save and activate your configuration.

IPSec IMS-AKA

Compliance with the VoLTE specification (GSMA PRD IR.92) requires cluster member support for IPsec IMS-AKA (IP Multimedia Services Authentication and Key Agreement) as defined in 3GPP TS 24.299, *IP Multimedia Call Control Protocol Based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)*: Stage 3, and TS 33.203, *3G Security: Access Security for IP-based Services*.

Support for IMS-AKA requires no new additional configuration elements.

Sample IMS-AKA Configuration

The following formatted extract from **show running-config** ACLI output shows a sample IMS-AKA profile configuration.

```
ims-aka-profile
name                            dut2.test
protected-client-port           4060
protected-server-port           4060
encr-alg-list                   aes-cbc des-ede3-cbc null
auth-alg-list                   hmac-sha-1-96 hmac-md5-96
last-modified-by                admin@172.30.11.18
last-modified-date              2012-01-10 17:31:59
```

Sample Security Policy Configuration

The following formatted extracts from **show running-config** ACLI output shows three associated security policies.

The first policy, and the one with the highest priority, opens Port 5060 for SIP traffic.

```
security-policy
name                            pol1
network-interface               M10:0.6
priority                        0
local-ip-addr-match             3fff:c0ac::c0ac:ce12
remote-ip-addr-match            ::
local-port-match                5060
remote-port-match               0
trans-protocol-match            ALL
direction                       both
local-ip-mask                   ::
remote-ip-mask                  ::
action                          allow
ike-sainfo-name
outbound-sa-fine-grained-mask
local-ip-mask                   ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
remote-ip-mask                  ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
local-port-mask                 65535
remote-port-mask                65535
trans-protocol-mask             0
valid                           enabled
vlan-mask                       0xFFF
last-modified-by                admin@console
last-modified-date              2012-01-10 17:48:59
```

The second policy opens Port 4444 for CCP traffic.

```
security-policy
name                            pol2
network-interface               M10:0.6
priority                        2
local-ip-addr-match             3fff:b623::b623:ce02
remote-ip-addr-match            3fff:b623::b623:ce01
local-port-match                4444
remote-port-match               4444
trans-protocol-match            ALL
direction                       both
local-ip-mask                   ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
remote-ip-mask                  ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
action                          allow
ike-sainfo-name
outbound-sa-fine-grained-mask
local-ip-mask                   ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
remote-ip-mask                  ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
local-port-mask                 65535
remote-port-mask                65535
trans-protocol-mask             0
valid                           enabled
vlan-mask                       0xFFF
last-modified-by                admin@console
last-modified-date              2012-01-10 17:49:15
```

The third policy, the policy with the least priority, and, consequently, the last policy applied, requires IPsec on all ports.

```
security-policy
name                            pol3
network-interface               M10:0.6
priority                        10
local-ip-addr-match             3fff:c0ac::c0ac:ce12
remote-ip-addr-match            ::
local-port-match                0
remote-port-match               0
trans-protocol-match            ALL
direction                       both
local-ip-mask                   ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
remote-ip-mask                  ::
action                          ipsec
ike-sainfo-name
outbound-sa-fine-grained-mask
local-ip-mask                   ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
remote-ip-mask                  ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
local-port-mask                 65535
remote-port-mask                65535
trans-protocol-mask             0
valid                           enabled
vlan-mask                       0xFFF
last-modified-by                admin@console
last-modified-date              2012-01-10 17:50:42
```

Sec-Agree

Version S-CZ7.2.0 introduces support for RFC 3329, Security Mechanism Agreement for the Session Initiation Protocol, commonly referred to as Sec-Agree. The RFC defines three SIP headers, Security-Client, Security-Server, and Security-Verify that provide the ability for SIP UAs and other SIP entities (servers, proxies, and registrars) to negotiate next-hop security

mechanisms. Note that this initial implementation does not provide support for server-initiated security negotiation, nor does it support media-plane security. That is, support is limited to client-initiated negotiation during initial registration, and to signalling security.

Currently the P-CSCF functionality includes support for IMS-AKA feature for VoLTE deployments. In order to support RCS clients along with VoLTE P-CSCF functionality needs to be enhanced to support RFC 3329, Security Mechanism Agreement for the Session Initiation Protocol (commonly referred to as Sec-Agree), which includes support for TLS as security mechanism.

Sec-Agree defines three SIP headers, Security-Client, Security-Server and Security-Verify, to negotiate security agreements during initial REGISTER transactions. Header definitions are as follows:

```
security-client  = "Security-Client" HCOLON
                          sec-mechanism *(COMMA sec-mechanism)
security-server  = "Security-Server" HCOLON
                   sec-mechanism *(COMMA sec-mechanism)
security-verify  = "Security-Verify" HCOLON
                   sec-mechanism *(COMMA sec-mechanism)
sec-mechanism    = mechanism-name *(SEMI mech-parameters)
mechanism-name   = ( "digest" / "tls" / "ipsec-ike" /
                     "ipsec-man" / token )
mech-parameters  = ( preference / digest-algorithm /
                     digest-qop / digest-verify / extension )
preference       = "q" EQUAL qvalue
qvalue           = ( "0" [ "." 0*3DIGIT ] )/ ( "1" [ "." 0*3("0") ] )
digest-algorithm = "d-alg" EQUAL token
digest-qop       = "d-qop" EQUAL token
digest-verify    = "d-ver" EQUAL LDQUOT 32LHEX RDQUOT
extension        = generic-param
```

The Security-Client header contains one or more security mechanisms and associated parameters proposed by the initiating client. This initial implementation supports two security mechanisms: TLS and ipsec-3gpp.

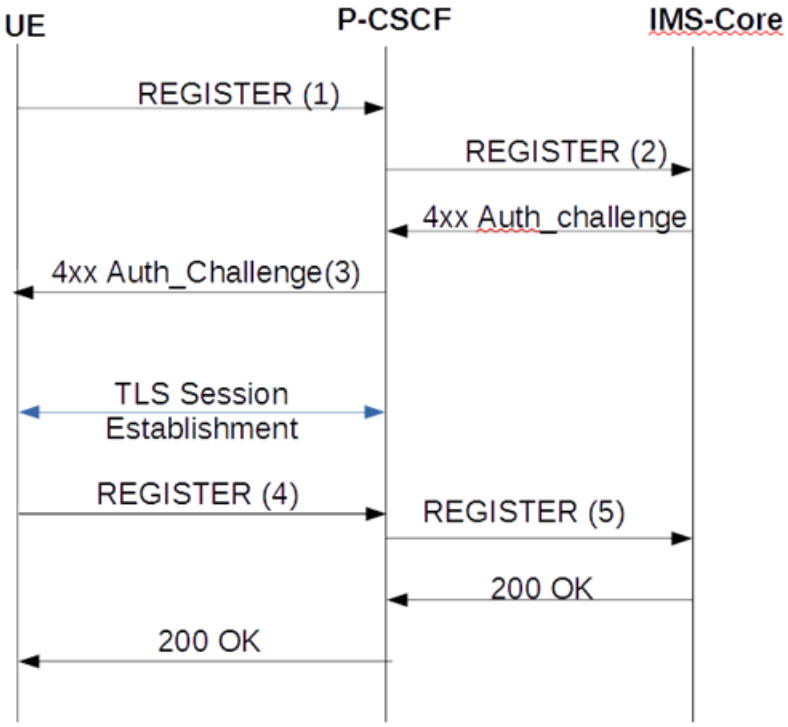
The Security-Server header contains the security mechanism chosen by the server from those mechanisms proposed by the client.

The Security-Verify header contains the contents of the Security-Server header.

Two additional header fields, Require and Proxy-Require, are also used in support of Sec-Agree negotiations. Both headers are required in client transmissions.

TLS Session Setup During Registration

This call flow depicts a TLS session setup during the registration procedure. Only relevant header fields are noted.



(1) REGISTER

```
Proxy-Require:sec-agree
Security-Client: ipsec-3gpp;alg=hmac-md5-96;ealg=aes-cbc;prot=esp;mod=trans;spi-
c=8765423;port-c=7524;spi-s=1234563;port-s=1358, ipsec-3gpp;alg= hmac-
sha-1-96;ealg=aes-cbc;prot=esp;mod=trans;spi-c=8765423;port-c=7524;spi-
s=1234563;port-s=1358, tls
```

(2) REGISTER

```
Authorization: Digest
uri="sip:ims.mnc007.mcc262.3gppnetwork.org",username="262073900320132@ims.mnc007.
mcc262.3gppnetwork.org",response="",realm="ims.mnc007.mcc262.3gppnetwork.org",non
ce=""
(No integrity-protected field will be present if TLS is selected as the security
mechanism)
```

(3) 401

```
Security-Server: tls
```

(4) REGISTER

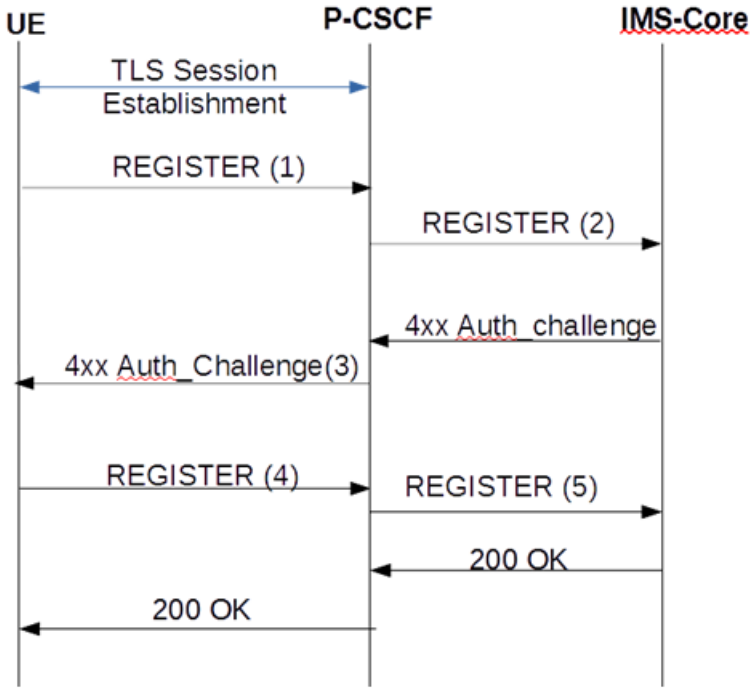
```
Proxy-Require: sec-agree
Security-Verify: tls
```

(5) REGISTER

```
Authorization: Digest
username="262073900320132@ims.mnc007.mcc262.3gppnetwork.org",realm="imstest1.tele
fonica.de",uri="sip:ims.mnc007.mcc262.3gppnetwork.org",qop=auth,nonce="Ms8l2xeF3l
4lb0VO8fK3KFMSKLv1sQAATdN2NpFUCgU=",nc=00000001,cnonce="3063397945",algorithm=AKA
v1-MD5,response="3779ff40a057f999a2f8288bbfafc10d", integrity-protected=tls-
pending
```

TLS Session Setup Prior to Registration

This call flow depicts a TLS session setup prior to the registration procedure.



(1) REGISTER

```
(No Security-Client or Proxy-Require header present)
```

(2) REGISTER

```
Authorization: Digest
uri="sip:ims.mnc007.mcc262.3gppnetwork.org",username="262073900320132@ims.mnc007.
mcc262.3gppnetwork.org",response="",realm="ims.mnc007.mcc262.3gppnetwork.org",non
ce=""
(No integrity-protected field will be present)
```

(3) 401

```
(No Security-Server header present)
```

(4) REGISTER

```
(No Security-Client, or Security-Verify or Proxy-Require header present)
```

(5) REGISTER

```
Authorization: Digest
username="262073900320132@ims.mnc007.mcc262.3gppnetwork.org",realm="imstest1.tele
fonica.de",uri="sip:ims.mnc007.mcc262.3gppnetwork.org",qop=auth,nonce="Ms8l2xeF3l
4lb0VO8fK3KFMSKLv1sQAATdN2NpFUCgU=",nc=00000001,cnonce="3063397945",algorithm=AKA
v1-MD5,response="3779ff40a057f999a2f8288bbfafc10d", integrity-protected=tls-
pending
```

Regardless of TLS Session setup procedure, if the newly added configurable item sec-agree feature is enabled, any messages on unprotected port will be rejected except REGISTER messages or messages related to emergency services.

For refresh registration, if the Sec_Agree occurred during Registration, it verifies for the presence or change of Security-Client & Security-Verify headers, if they differ it will be rejected with 4xx response and also Authorization header fields are verified irrespective of the above methods and if they differ with previous association it will be rejected with 403 (Forbidden) response. Also when the refresh REGISTER is being forward to the core, it will set the integrity-protected field to "tls-yes".

SEC-agree Configuration

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

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

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

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

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

3. **sec-agree-feature**—Set this parameter to enable or disable for Sec-Agree support. By default, support is disabled.
4. **sec-agree-pref**—Configure this parameter to specify security protocol preferences.
   - **ipsec3gpp** — support only IMS-AKA protocol
   - **tls** — support only TLS protocol
   - **ipsec3gpp-tls** — support both IMS-AKA and TLS, preferred protocol is IMS-AKA
   - **tls-ipsec3gpp** — support both TLS and IMS-AKA, preferred protocol is TLS
5. Type **done** to save your configuration.

IMS AKA over TCP

IMS-AKA registration is conducted over UDP or TCP protocol only. The Oracle Communications Unified Session Manager supports both transport protocols.

Within mobile IMS VoLTE/RCSe deployments, IP packets carrying SIP messages can be large due to IPv6 headers, IMS-AKA specific headers, extensive codec policies, and other 3GPP related headers. Because of this, IPv6 VoLTE signaling messages using IMS-AKA frequently exceed 1300 bytes and require TCP according to RFC3261 section 18.1.1.

IMS-AKA Secure Call Registration over TCP

To register and place a call into the network, a UE creates 3 TCP connections. The first insecure connection is established to the port (usually 5060) specified in the sip-port for the first registration request. You should create a sip-ports configuration element with port 5060 and an ims-aka-profile parameter that references an ims-aka-profile configuration element. The

ims-aka-profile configuration element initiates the process that creates secure connections. For example:

sip-ports

```
address                       sd-ip-address
port                          5060
transport-protocol            TCP
tls-profile
multi-home-addrs
allow-anonymous               registered
ims-aka-profile               profile
```

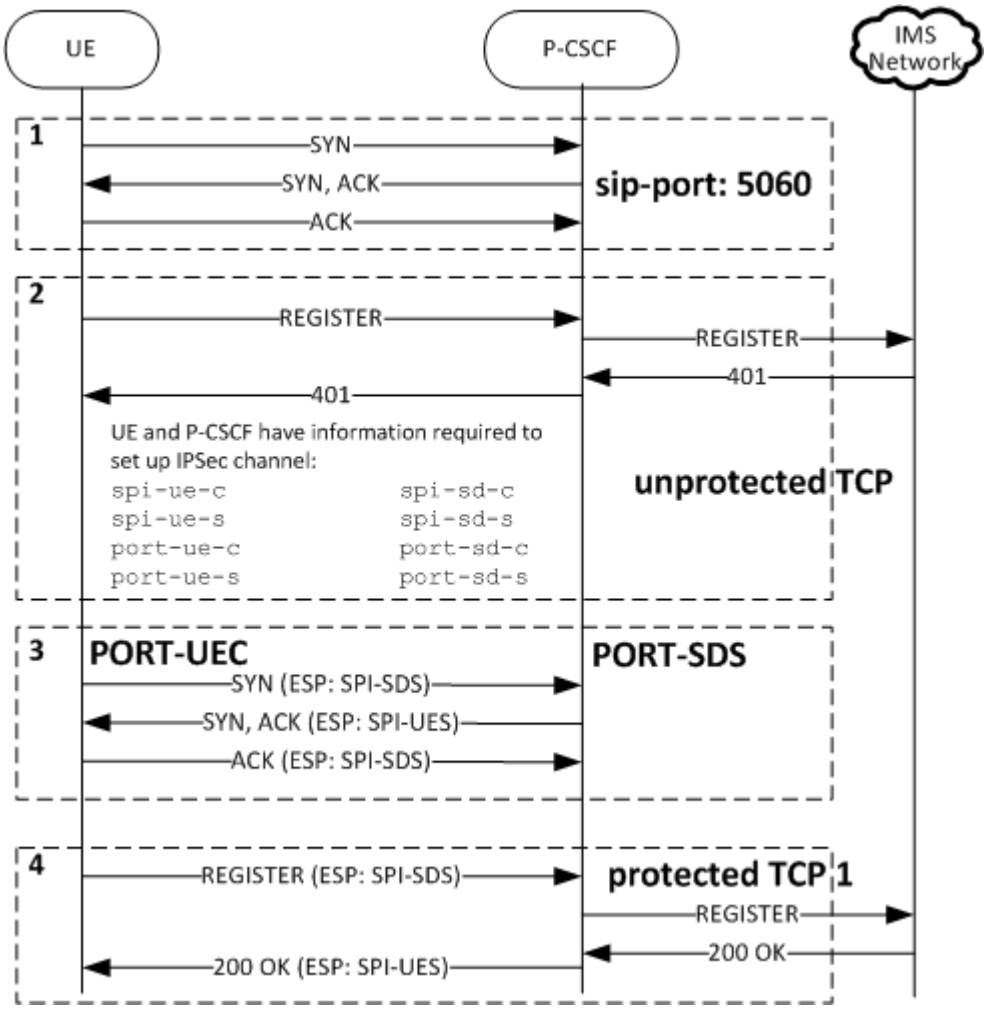
The ims-aka-profile configuration element defines the protected-server-port (PORT-SDS) and protected-client-port (PORT-SDC). The protected-server-port is opened for both inbound TCP and UDP traffic. For example:

ims-aka-profile

```
name                          profile
protected-client-port         PORT-SDC
protected-server-port         PORT-SDS
encr-alg-list                 aes-cbc des-ede3-cbc null
auth-alg-list                 hmac-sha-1-96
```

When the UE receives the 401Unauthorized challenge from the Oracle Communications Unified Session Manager acting as P-CSCF, both devices have the information to set up security association for two IPSec channels. The UE establishes the second TCP connection via IPSec channel from the UE's PORT-UEC to the P-CSCF's PORT-SDS, and the registration process continues.

Hereafter, the UE uses the IPSec channels from communication.

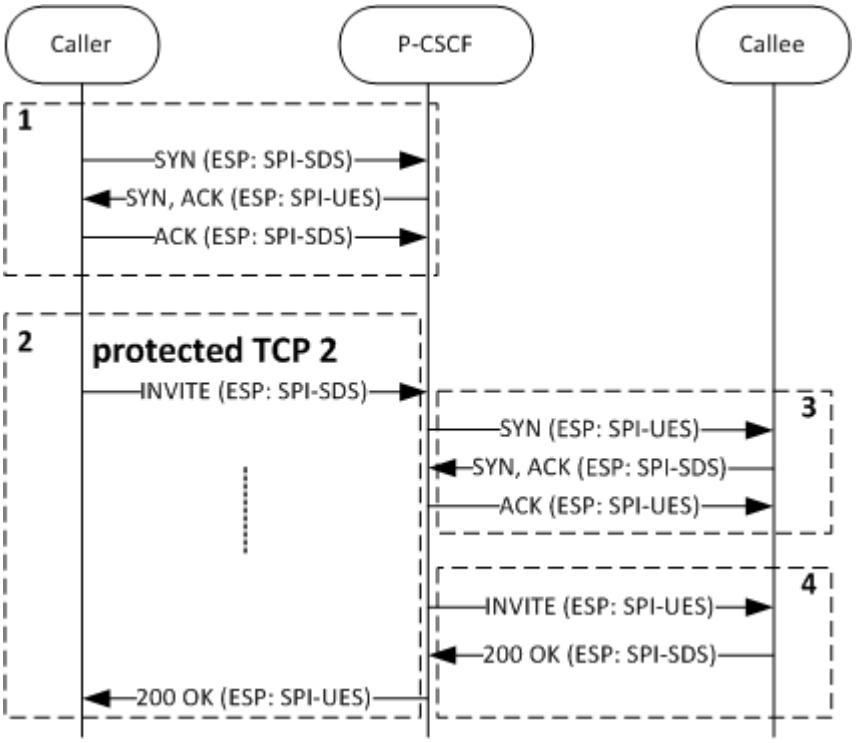


1. The UE and P-CSCF set up the TCP connection.
2. The UE sends an unauthenticated SIP Register message to the P-CSCF's unprotected server port (usually 5060). The Register message is forwarded to the UE's Home S-CSCF. The S-CSCF then replies with a the SIP 401 Authentication Required response back to P-CSCF. This message contains encryption keys and authentication information.

   The P-CSCF modifies the 401 message back to UE. At this point, both UE and P-CSCF should have all the information need to establish secure IPSec channels.
3. The UE and P-CSCF create a TCP connection over a secure channel from port-ue-c to port-sd-s.
4. The UE sends an authenticated REGISTER over the secure channel via the P-CSCF to the S-CSCF. If the authentication is valid, the P-CSCF will forward the 200 OK response from the S-CSCF to the UE. The 200 OK response will be sent in the same secure TCP connection.

IMS-AKA Call Establishment over TCP

For the UE to send a new request into the network and establish a call, a third TCP connection is created using the information configured/generated prior to creating the first secure connection.



1. If the TCP connection over a secure channel does not exist, the Caller will establish it.
2. The Caller initiates the call by sending SIP INVITE from its PORT-UEC to the P-CSCF's PORT-SDS.
3. The P-CSCF forwards the INVITE to the Caller. If not present, it will create a secure TCP connection from its PORT-SDC to the callee's PORT-UES.
4. The INVITE is then forwarded to the Callee securely.

SIP SUBSCRIBE and NOTIFY over TCP IMS-AKA

SUBSCRIBE and NOTIFY messages are exchanged between a UE and the P-CSCF in a manner similar to the previous INVITE example whereby the secure channel is first created and then the SIP messages are exchanged securely.

IMS-AKA Change Client Port

The Oracle Communications Unified Session Manager is now in compliance with 3GPP TS 33.203, *Access Security for IP-Based Services*. Previous releases did not comply with requirements specified in Section 7.4, Authenticated re-registration, which reads in part:

Every registration that includes a user authentication attempt produces new security associations. If the authentication is successful, then these new security associations shall replace the previous ones. This clause describes how the UE and P CSCF handle this replacement and which SAs to apply to which message.

When security associations are changed in an authenticated re-registration then the protected server ports at the UE (port_us) and the P-CSCF (port_ps) shall remain unchanged, while the protected client ports at the UE (port_uc) and the P-CSCF (port_pc) shall change.

If the UE has an already active pair of security associations, then it shall use this to protect the REGISTER message. If the S-CSCF is notified by the P-CSCF that the REGISTER message from the UE was integrity-protected it may decide not to authenticate the user by means of the AKA protocol. However, the UE may send unprotected REGISTER messages at any time. In this case, the S-CSCF shall authenticate the user by means of the AKA protocol. In particular, if the UE considers the SAs no longer active at the P-CSCF, e.g., after receiving no response to several protected messages, then the UE should send an unprotected REGISTER message."

Prior releases failed to change the protected client ports after a successful re-registration.

Protected Ports

Within IMS networks, the P-CSCF provides the network access point and serves as the outbound proxy server for user equipment -- smart phones, tablets, and similar devices. The UE must connect to the P-CSCF prior to registration and initiation of SIP sessions. Connection to the P-CSCF, which can be in the user's home network, or in a visited network if the UE is roaming, is accomplished using Dynamic Host Control Protocol (DHCP) P-CSCF discovery procedures.

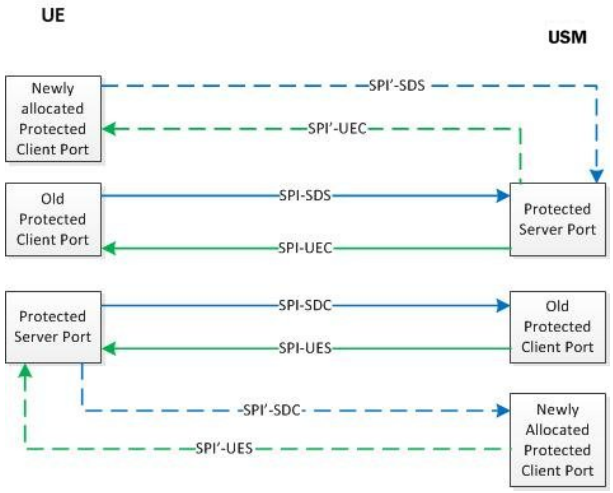
After successful discovery, the P-CSCF and UE negotiate IPSec security associations (SAs) which are used to establish four protected (authenticated and encrypted using Encapsulating Security Payload protocol) ports between the UE and the P-CSCF.

The four protected ports are shown in the following illustration:

As required by Section 7.4 of 3GPP TS 33.203, the protected client ports, one on the UE and the other on the Oracle Communications Unified Session Manager, must be changed after each successful re-registration.

To fulfill this requirement, this release adds a new attribute to the existing ims-aka-profile configuration object. This attribute (**end-protected-client-port**) works in conjunction with **start-protected-client-port** (**protected-client-port** in previous releases) to enable the identification of a pool of protected client ports, which will be used for re-registration scenarios where the Oracle Communications Unified Session Manager is required to change the client port.

The Oracle Communications Unified Session Manager creates new protected client ports, one on the UE and the other on the Oracle Communications Unified Session Manager, after every re-registration. Old protected client ports, along with their associated SAs, are maintained for 30 seconds after re-registration to ensure correct handling or any pending responses to previously transmitted messages.



After successful re-registration, the Oracle Communications Unified Session Manager updates the registration cache with updated port information and checkpoint with the HA peer, if present.

IMS-AKA Change Client Port Configuration

An IMS-AKA profile establishes the client and server ports to be protected, and it defines lists of encryption and authentication algorithms the profile supports. You can configure multiple IMS-AKA profiles, which are uniquely identified by their names.

You apply an IMS-AKA profile to a SIP port configuration using the name.

To configure an IMS-AKA profile:

1. From Superuser mode, use the following command sequence to navigate to ims-aka-profile configuration mode.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ORACLE(security)# ims-aka-profile
   ORACLE(ims-aka-profile)#
   ```

2. **name**—Enter the name you want to give this IMS-AKA profile. This is the value you will use to apply the profile to a SIP port configuration. This parameter is required, and it has no default value.

3. **protected-server-port**—Enter the port number of the protected server port, which is the port on which the Oracle Communications Unified Session Manager receives protected messages. The protected server port should not overlap with the port range defined in the steering ports configuration using the same IP address and the SIP interface. If there is overlap, the NAT table entry for the steering port used in a call will prevent SIP messages from reaching the system's host processor.

   This parameter defaults to 0, which disables the function associated with the parameter. The valid range for values is 1025 to 65535.

4. **start-protected-client-port (protected-client-port in Release S-CX6.3.3M2 and earlier releases)**—Enter the start value for the pool of port numbers available following a successful re-authentication. Like the protected server port, the protected client port pool should not overlap with the port range defined in the steering ports configuration using the same IP address and the SIP interface. If there is overlap, the NAT table entry for the

steering port used in a call will prevent SIP messages from reaching the system's host processor.

Any existing configuration for **protected-client-port** will be mapped to both **start-protected-client-port** and **end-protected-client-port** parameter values.

This parameter defaults to 0, which disables the function associated with the parameter. The valid range for values is 1025 to 65535.

5. **end-protected-client-port**—Enter the end value for the pool of port numbers available following a successful re-authentication. Ensure that this value is greater than the value assigned to **start-protected-client-port**. Note that the maximum supported pool contains 5 entries. Like the protected server port, the protected client port pool should not overlap with the port range defined in the steering ports configuration using the same IP address and the SIP interface. If there is overlap, the NAT table entry for the steering port used in a call will prevent SIP messages from reaching the system's host processor.

   This parameter defaults to 0, which disables the function associated with the parameter. The valid range for values is 1025 to 65535.

6. **encr-alg-list**—Enter the list of encryption algorithms. You enter more than one value by separating the algorithms by <Spaces> and enclosing all values in quotations marks:

   This parameter defaults to the following three values: **aes-cbc**, **des-ede3-cbc**, and **null**.

7. **auth-alg-list**—Enter the list of authentication algorithms. You enter more than one value by separating the algorithms by <Spaces> and enclosing all values in quotations marks:

   This parameter defaults to hmac-sha-1-96.

Sample IMS-AKA Configuration

The following formatted extract from **show running-config** ACLI output shows a sample IMS-AKA profile configuration.

```
ims-aka-profile
name    TS33.203
start-protected-client-port    4060
end-protected-client-port    4064
protected-server-port    4070
encr-alg-list    aes-cbc des-ede3-cbc null
auth-alg-list    hmac-sha-1-96 hmac-md5-96
last-modified-by    admin@172.30.11.18
last-modified-date    2013-06-15 14:58:08
```

SIP IMS P-CSCF P-Asserted Identity in Responses

In releases earlier than Release S-C6.1.0, the Oracle Communications Unified Session Manager —operating as a P-CSCF—removes the P-Preferred-Identity header (if present) on receipt of a 1xx or 2xx response. It also inserts a P-Asserted-Identity header with the value received in the P-Preferred-Identity header.

Release S-C6.1.0 changes this behavior. Now the Oracle Communications Unified Session Manager:

- Caches a copy of the P-Called-Party-ID header when it receives one of the following destined for a UE prior to forwarding the request:
  - An initial request for dialog
  - A request for a standalone transaction

  - A request for an unknown method that does not related to an existing dialog
    The SIP interface receiving the request should have the SIP IMS feature enabled.
- Removes the P-Preferred-Identity header (if present) and inserts a P-Asserted-Identity header with the value saved from the P-Called-Party-ID header on receipt of a 1xx or 2xx response.

Important Notes

Note the following:

- The endpoint to which the response is being sent must be a trusted endpoint. The option **disable-ppi-to-pai** should not be configured in the global SIP configuration's **options** list.
- If the P-Preferred-Identity header is present in the response, the Oracle Communications Unified Session Manager will delete the header.
- If the P-Asserted-Identity header is present in the response, the Oracle Communications Unified Session Manager will overwrite that -Asserted-Identity.

SIP IMS P-CSCF P-Asserted Identity in Responses Configuration

This behavior is enabled automatically. You do not need to perform any configuration steps.

E-CSCF Support

An Emergency Call Session Control Function (E-CSCF) is an IMS core element that aids in routing emergency calls to an appropriate destination, such as a PSAP. E-CSCF functionality can be performed by the Oracle Communications Unified Session Manager with appropriate local policy and network management control configuration.

The E-CSCF feature let the Oracle Communications Unified Session Manager internally prioritize and route emergency calls to the corresponding Emergency Service Center, based either on the calling party's request URI, or based on location information retrieved from a CLF (Connectivity Location Function) for wireline/TISPAN networks.

By integrating E-CSCF functionality into the P-CSCF (Oracle Communications Unified Session Manager), networks can satisfy the common local requirement that certain telephony elements be deployed locally, rather than use single, centralized elements. Funcitons like the E-CSCF likely fall into this category.

Service URN Support

To enable E-CSCF functionality, the Oracle Communications Unified Session Manager can parse service URNs for local policy lookup keys, and as desitnation identifiers in network management controls (NMC). Ensure that the match-URN is entered correctly as: urn:service:sos" or "urn:service:sos.type or the Oracle Communications Unified Session Manager will interpret the URN as a hostname. Please see RFC 5031 for more information on compliant URN construction.

E-CSCF Configuration Architecture

There are four elements which comprise and enable E-CSCF support on the Oracle Communications Unified Session Manager :

- CLF Connectivity
- NMC Emergency Call Control
- Local Policy
- Emergency Local Route Table

CLF Connectivity

The Oracle Communications Unified Session Manager must be configured with Diameter-based CLF support. This is accomplished by creating an appropriate external policy server configuration.

When the Oracle Communications Unified Session Manager requests authorization from the CLF server, a Line-Identifier AVP which includes a location string is expected to be returned for the call. The returned location string will be used later for an LRT query.

NMC Emergency Call Control

By configuring a Network Management Control (NMC), the Oracle Communications Unified Session Manager can flag a call for special priority early after it is received and validated by the system. The **destination identifier** must be configured in the NMC with the service URN of an incoming emergency call. Also, the NMC configuration must have its **next hop** parameter left blank. This lets the Oracle Communications Unified Session Manager route the emergency call with local policies.

For example, if **urn:service:sos** is the configured value in the NMC's **destination identifier**, and an INVITE arrives on the Oracle Communications Unified Session Manager with urn:service:sos in the request URI, the call will be flagged for emergency handling. The next step in call processing is for the INVITE to be evaluated by local policy.

Local Policy

Local policies must be configured to match and then route an incoming emergency call. Once a local policy match is made, the Oracle Communications Unified Session Manager looks to the configured policy attributes for where to forward the INVITE. A matching policy attribute's next hop should be configured to point to an emergency LRT that contains specific destinations for emergency calls. In addition, the **elec str lkup** parameter must be set to enabled so the Oracle Communications Unified Session Manager will perform an LRT lookup based on the location string returned in the CLF response.

The **eloc str match** parameter identifies the attribute, whose value in the location string will be used as the lookup key in the emergency LRT. For example, if the returned location string is:

```
loc=xxx;noc=yyyy;line-code=zzzz
```

and the **eloc str match** parameter is set to **noc**, then when the Oracle Communications Unified Session Manager performs a local policy route search, it will search the LRT for yyyy. If the **eloc str match** parameter left empty or if there is no match when **eloc str lkup** is enabled, the entire location string is used as the lookup key.

Emergency LRT

The Oracle Communications Unified Session Manager needs to be configured with an emergency LRT to route emergency calls to their destination.

As stated in the previous section, when searching an emergency LRT, any user defined parameter within a Location String may be used as the key to look up next-hop routing information.

LRT files support <user type = string> which enables the Oracle Communications Unified Session Manager to perform searches on free form attributes that may appear in the returned location-string. The <user type = string> value for an entry in the emergency LRT should be set to a part or whole value returned in the CLF's location string. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<localRoutes>
    <route>
        <user type="string">1234</user>
        <next type="regex">!^.*$!sip:911@192.168.200.140:5060!</next>
    </route>
    <route>
        <user type="string">loc=xxx;noc=yyyy;line-code=zzzz</user>
        <next type="regex">!^.*$!sip:911@192.168.1.139:5060!</next>
    </route>
</localRoutes>
```

> **Note:**
>
> Given that the Location String is not a well-defined string, care should be taken when defining and configuring the LRT tables.

LRTs must be individually uploaded to both the active and standby systems in an HA node; LRTs are not automatically replicated across nodes.

CLF Response Failure

If there is no location string in a CLF's repsone or the CLF rejects the call, the Oracle Communications Unified Session Manager uses the **default location string** parameter from the ingress SIP interface to populate the PANI header. The emergency call proceeds normally using this location string's information for emergency LRT lookups.

E-CSCF Configuration

This procedure assumes that the Oracle Communications Unified Session Manager is configured to communicate with a CLF. In addition, this procedure assumes an the Oracle Communications Unified Session Manager is configured and loaded with an appropriate LRT for E-CSCF Use.

To configure an NMC for E-CSCF use (baseline parameters are not mentioned):

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

   ```
   ORACLE# configure terminal
   ```

2. Type **session-router** and press Enter to access the signaling-related configurations.

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

3. Type **net-management-control** and press Enter.

   ```
   ORACLE(session-router)# net-management-control
   ```

4. **name**—Enter the name of this network management control rule; this value uniquely identifies the control rule. There is no default for this parameter.
5. **state**—Enable or disable this network management control rule. The default value is **enabled**. The valid values are:
   - enabled | disabled
6. **type—**Set this parameter to **priority** so that the Oracle Communications Unified Session Manager will flag incoming calls with a matching destination identifier as a priority calls.
7. **treatment**—Set this parameter to **divert**.
8. **next-hop**—Leave this parameter blank so that the call's processing will go directly to local policy.
9. **destination-identifier**—Enter the service URN that endpoints in your network include in their request URIs to identify themselves as emergency calls.
10. Save your configuration.

    To configure local policy for E-CSCF use (baseline parameters are not mentioned):

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

    ```
    ORACLE# configure terminal
    ORACLE(configure)#
    ```

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

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

13. Type **local-policy** and press Enter. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

    ```
    ORACLE(session-router)# local-policy
    ORACLE(local-policy)#
    ```

14. **to-address**—Set this parameter to the lookup key for matching emergency calls. You can now use a service URN as lookup key criteria.
15. Save your configuration.

    To configure policy attributes for E-CSCF use (baseline parameters are not mentioned):

16. Type **policy-attributes** and press Enter. If you are adding this feature to a pre-existing local policy configuration, you will need to select and edit your local policy.

    ```
    ORACLE(local-policy))# policy-attributes
    ORACLE(policy-attributes)#
    ```

17. **next-hop**—Set this parameter to **lrt: name-of-elrt-file.gz** for this policy attribute to lookup routes in the named lrt file.
18. **eloc-str-lkup**—Set this parameter to **enabled** for the Oracle Communications Unified Session Manager to parse the emergency location string, as received in a CLF Line Identifyier AVP, for emergency LRT lookup.
19. **eloc-str-match**—Set this parameter to the attribute name found in the location string whose value will be used as a lookup key in the LRT named in the next-hop parameter. Common values include "loc" or noc.

20. Save and activate your configuration.

Maintenance and Troubleshooting

The **show lrt route-entry** command displays two entries, if the username 1234 has a "string" type and "E164" type entries.

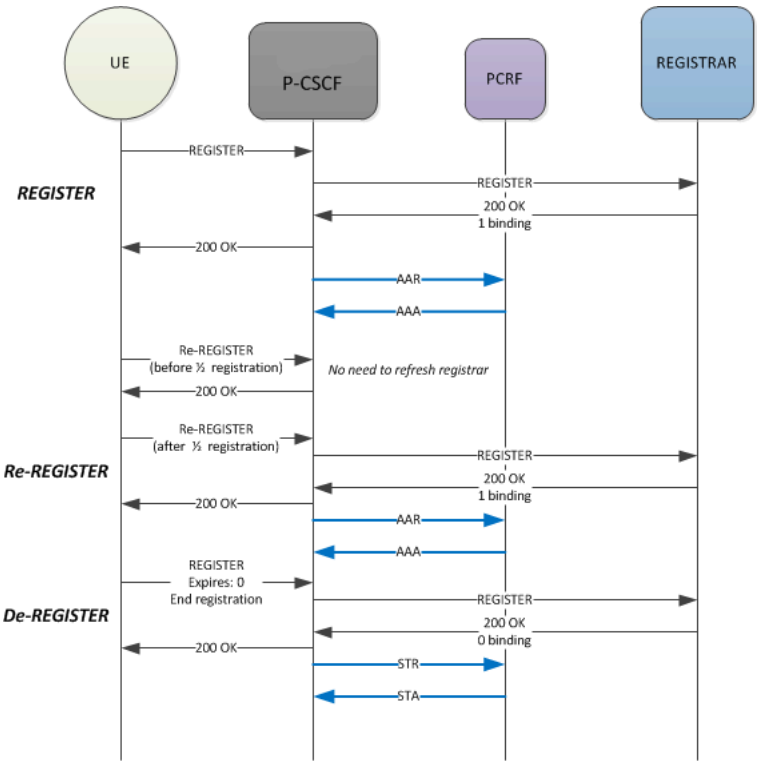
```
ACMEPACKET# show lrt route-entry emergency_lrt 1234
UserName <1234>
 User Type= E164
 NextHop= !^.*$!sip:911@192.168.200.139:5060!
 NextHop Type= regexp
UserName <1234>
 User Type= string
 NextHop= !^.*$!sip:911@192.168.200.140:5060!
 NextHop Type= regexp
```

2774 - Provisioning of SIP Signaling Flow Information

This feature supports of Provisioning of SIP Signaling Flow Information as described in 3GPP TS 29.213 section B1b [1], and the procedures specified in TS 29.214 section 4.4.5a.

The feature applies to the scenario when the Oracle Communications Unified Session Manager (A-SBC / P-CSCF) uses its external policy server functionality and connects to a PCRF via Rx interface in bandwidth-management mode.

This feature deals with the information that the Oracle Communications Unified Session Manager includes in an AAR message it sends to the PCRF when an endpoint registers, re-registers, and de-registers. The following diagram shows the typical scenario.



Initial Registration

When an endpoint registers, the Oracle Communications Unified Session Manager creates and sends an AAR message to a PCRF which includes the following information:

<AA-Request>::= < Diameter Header: 265, REQ, PXY >

AVP	AVP Contents
Session-ID	OriginHost;0;0;<Key to Registration Cache> Key is usually the endpoint's AoR.
Auth-Application-ID	Application-ID of Rx (16777236)
Origin-Host	<ext-policy-server_name>.<ext-policy-realm>.<domain-name-suffix>
Origin-Realm	<ext-policy-realm>.<domain-name-suffix>
Destination-Realm	<IP Address of endpoint>@<destination realm of this AAR>
Framed-IP-Address AVP - v4 address	Layer 3 Endpoint-IP-address v4 - Framed-IP-Addressor
Framed-IPv6-Prefix AVP - v6 address	v6 - Framed-IPv6-Prefix
Media-Component	Grouped AVP description follows

Media-Component-Description AVP ::= < AVP Header: 517 >

AVP	AVP Contents
Media-Component-Number	0
Media-Sub-Component	Grouped AVP description follows

Media-Sub-Component ::= < AVP Header: 519 >

AVP	AVP Contents
Flow-Number	1
Flow-Description	Permit in <ip> from <Endpoint IP:Port> to <OCUSM Sip Interface IP:Port> Permit out <ip> from <OCUSM SIP Interface IP:Port> to <Endpoint IP:Port> Where <ip> is (UDP: 17, TCP: 6) if wildcard-trans-protocol = disabled.
Flow-Status	Set to: ENABLED (2)
Flow-Usage	Set to: AF_SIGNALLING (2)
AF-Signalling-Protocol	Set to: SIP (1)

Register Refresh

When a registration Refresh is received before the half time of the registration expiry, the registration cache is not updated and the Oracle Communications Unified Session Manager responds with 200OK to the UE. This is standard registration cache functionality. No additional AAR is sent to the PCRF.

If a registration Refresh is received after the half time of the registration expiry, or if any registration information changes, the Oracle Communications Unified Session Manager sends an AAR to the PCRF after it receives and forwards a 200 OK response from registrar. The

AAR includes the same session-id as the initial AAR that was to PCRF. This lets PCRF correlate the AAR with the earlier AAR that it received.

De-Registration

When an contact de-registers with expires=0, it means that the endpoint is removing all of its contacts from registration. When this occurs, the Oracle Communications Unified Session Manager sends an STR message to the PCRF to terminate the session. If the de-registration message does not reflect a complete de-registration, the Oracle Communications Unified Session Manager does not send the STR message to the PCRF. The STR message includes the following information:

<ST-Request> ::= < Diameter Header: 275, REQ, PXY >

AVP	AVP Contents
Session-ID	OriginHost;0;0;<Key to Registration Cache> Key is usually the endpoint's AoR.
Auth-Application-ID	Application-ID of Rx (16777236)
Origin-Host	<ext-policy-server_name>.<ext-policy-realm>.<domain-name-suffix>
Origin-Realm	<ext-policy-realm>.<domain-name-suffix>
Destination-Realm	<IP Address of endpoint>@<destination realm of this AAR>
Destination-Host	N/A
Termination-Cause	DIAMETER LOGOUT (1) - User initiated a disconnect

Failure Response to Re-Register

Upon reception of a failure response from the REGISTRAR for a subsequent Registration refresh from endpoint, the Oracle Communications Unified Session Manager performs de-registration actions, i.e. an STR message to the PCRF.

Provisioning SIP Signaling Flows Configuration

To enable correct provisioning of signaling flows upon registration:

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

   ```
   ORACLE# configure terminal
   ORACLE(configure)#
   ```

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

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

3. Type **ext-policy-server** and press Enter.

   ```
   ORACLE(media-manager)# ext-policy-server
   ORACLE(ext-policy-server)#
   ```

   If you are adding support for this feature to a pre-existing realm, then you must select (using the ACLI select command) the external policy server that you want to edit.

4. **provision-signaling-flow**—Set this to enabled for the Oracle Communications Unified Session Manager to send AAR messages to a PCRF on endpoint registrations.

5. Save and activate your work.

Troubleshooting

Upon receipt of AAA response from PCRF for the AAR sent, the Oracle Communications Unified Session Manager logs the status of the AAA received. Successful AAAs are noted by session-ID in EBMD logs. The log will look like:

```
Provision of SIP Signaling Flow (<session-ID>) via Diameter Rx on realm <realm-id> successful.
```

Failed AAA response are noted at NOTICE level EBMD logs citing the failure response to AAR request with the session-ID sent. An a MINOR non-health-affecting alarm is also raised with the following text:

```
Provision of SIP Signaling Flow (<session-ID>) via Diameter Rx on realm <realm-id> failed.
```

show ext-band-mgr

The show ext-band-mgr command has been augmented to include the Register and DeRegister counts. For example:

```
ORACLE# show ext-band-mgr
13:55:49-166
EBM Status                  -- Period -- -------- Lifetime --------
                  Active    High   Total      Total  PerMax    High
Client Trans           0       0       0          7       4       1
Server Trans           0       0       0          0       0       0
Sockets                1       1       0          1       1       1
Connections            0       0       0          1       1       1
                           ---- Lifetime ----
                     Recent      Total  PerMax
Reserve                   0          0       0
Modify                    0          0       0
Commit                    0          4       2
Remove                    0          2       1
Register                  1          1       1
DeRegister                1          1       1
EBM Requests              0          6       3
EBM Installs              0          6       3
EBM Req. Errors           0          0       0
EBM Rejects               0          0       0
EBM Expires               0          0       0
EBMD Errors               0          0       0
```

RTP and RTCP Bandwidth Calculation and Reporting

The Oracle Communications Unified Session Manager supports changing bandwidth requirements in an ad-hoc mult-party conference by tracking reduced bandwidth needs as parties are placed on hold during the initiation of a multi-party call. The combination of the 5 AVPs are considered by network elements for this functionality.

This section is applicable to the Oracle Communications Unified Session Manager's Rx implementation when acting as a P-CSCF and connecting with a PCRF. The 5 AVPs considered in this section are created and sent in AAR messages.

Max-Requested-Bandwidth-UL & Max-Requested-Bandwidth-DL AVPs

These AVPs reflect RTP bandwidth requirements for a call. The default behavior (i.e., no options are configured) that dictates how these AVPs are created follows.

The average rate limit scenario is when no b=AS: parameter in the SDP and the **media-profile**, **average-rate-limit** parameter is configured to a value greater than 0. The Oracle Communications Unified Session Manager inserts the **media-profile**, **average-rate-limit** parameter ×8 into both Max-Requested-Bandwidth-UL & Max-Requested-Bandwidth-DL AVPs.

The SDP bandwidth scenario is when there is a b=AS: parameter in the SDP and the media-profile configuration element has the following configurations:

- average-rate-limit = 0
- sdp-rate-limit-headroom > 0
- sdp-bandwidth = ENABLED

When these conditions are met, the Max-Requested-Bandwidth-UL & Max-Requested-Bandwidth-DL AVPs are populated with the value in the b=AS: parameter + **sdp-rate-limit-headroom** ACLI parameter.

Optional AVP Creation

When the **get-bw-from-sdp** option is configured in the sip-config, the following occurs:

When SDP contains the

```
b=AS:
```

parameter with valid value, the Oracle Communications Unified Session Manager multiplies that value ×1000 and inserts the result in the Max-Requested-Bandwidth-UL and Max-Requested-Bandwidth-DL AVPs in an AAR message.

If there is no b=AS: line in the SDP, the value is taken from the **media-profile**, **average-rate-limit** parameter multiplied ×8 to get a bps value then inserted into the Max-Requested-Bandwidth-UL and Max-Requested-Bandwidth-DL AVPs in an AAR message.

The Oracle Communications Unified Session Manager uses the b=AS: line (or chosen codec via the media-profile) from the final answer SDP for the session as the value for creation of these two AVPs.

RR-Bandwidth & RS-Bandwidth AVPs

These AVPs reflect RTCP bandwidth requirements for a call. When SDP contains:

```
b=RR:
b=RS:
```

parameters and values, the Oracle Communications Unified Session Manager inserts the values (after the :) into the respective RR-Bandwidth (521) and RS-Bandwith (522) AVPs. When these parameters are not present in the SDP, the RR-Bandwidth and RS-Bandwidth AVPs are not created in AAR messages sent to the PCRF. The **sip-config**, **get-bw-from-sdp** option must be configured to enable creation of these AVPs.

Flow-status AVP

The Flow-status AVP (511) is based on SDP direction. Tests for SDP are performed in the following order:

SDP State	Flow-Status AVP (511) Set to
port in the SDP m-line is 0	REMOVED (4)
Transport in m-line is "TCP" or " TCP/MSRP"	ENABLED (2)
a=recvonly and <SDP direction> = UE originated	ENABLED-DOWNLINK (1)
a=recvonly and <SDP direction> = UE terminated	ENABLED-UPLINK (0)
a=sendonly and <SDP direction> = UE originated	ENABLED-UPLINK (0)
a=sendonly and <SDP direction> = UE terminated	ENABLED-DOWNLINK (1)
a=inactive	DISABLED (3)
a=sendrecv or no direction attribute	ENABLED (2)

UE originated - Call originator creates the SDP being considered.

UE terminated - Call terminator creates the SDP being considered.

Flow-status AVP is always created according and requires no configuration.

2629 - IR.92 Compliance via SIP 380 Response

This feature furthers the Oracle Communications Unified Session Manager's compliance with GSMA's Voice over LTE specification (IR.92) to redirect VoLTE originated emergency calls to a circuit switched network.

When the Oracle Communications Unified Session Manager receives an emergency call, which it can not complete, it returns a 380 (Alternative Service) response to the sender. Some examples of when the Oracle Communications Unified Session Manager can not forward such an emergency call are:

- The Next-hop does not exist to route the call via NMC means
- The NSEP calls are rejected due to the Oracle Communications Unified Session Manager hitting a load limit
- The NSEP calls are rejected due to the target session agent exceeding constraints
- The NMC treatment for the call is set to Reject

380 Response Format

The Oracle Communications Unified Session Manager's 380 SIP response to the sender includes:

- Content-Type header field set to application/3gpp-ims+xml
- P-Asserted-Identity header field set to the value of the SIP URI of the last entry on the Path header field value received during registration; It is the value of the SIP URI of the P-CSCF.
- 3GPP IM CN subsystem XML body containing an <ims-3gpp> element with the "version" attribute will be set to "1" and with an <alternative-service> child element set to "alternative service"

- a <type> child element set to emergency
- a <reason> child element set to the value of configuration element send-380-response
- an <action> child element set to emergency-registration

380 Response Example

```
SIP/2.0 380 Alternative Service
Via: SIP/2.0/UDP 192.168.15.2:5060;branch=z9hG4bK-23615-1-0
From: sipp <sip:911@192.168.15.2:5060>;tag=1
To: sut <sip:911@192.168.101.11:5060>
Call-ID: 1-23615@192.168.15.2
CSeq: 1 INVITE
Content-Type: application/3gpp-ims+xml
Content-Length: 209
P-Asserted-Identity: sip:911-44e2etbufgibf@172.16.101.11:5060
Reason: Q.850; cause=63
<?xml version='1.0' encoding="UTF-8"?>
<ims-3gpp version="1.0">
<alternative-service>
<type>emergency</type>
<action>emergency-registration</action>
<reason>sample reason</reason>
</alternative-service>
</ims-3gpp>
```

IR.92 Compliance Configuration

To configure the 380 SIP response for IR.92 compliance:

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

   ```
   ORACLE# configure terminal
   ```

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

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

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

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

   From this point, you can configure SIP interface parameters. To view all sip-interface parameters, enter a **?** at the system prompt.

4. If configuring an existing interface, enter the select command to select the interface.
5. **send-380-response**—Enter a reason phrase enclosed in quotes to place in the reason tag of the <ims-3gpp> element in a 380 response for a failed-to-route emergency call.
6. Type **done** and continue.

IR.94 Support

The Oracle Communications Unified Session Manager supports IR.94 (IMS Profile for Conversational Video Service) for use in an IMS and LTE environment.

The Oracle Communications Unified Session Manager supports both video and audio and can keep a call up by switching to audio mode when the video session is dropped. The Oracle Communications Unified Session Manager receives a Specific-Action AVP in the RAR from the PCRF indicating the loss of video bearer. The Oracle Communications Unified Session Manager then replies with a RAA, and finally sends a STR failing the call.

This behavior contradicts TS 29.214 that indicates the P-CSCF should wait for an ASR from the PCRF (indicating the bearer has been removed), and should not fail the call until all bearers have been removed.

The following behaviors enable IR.94 Support:

- The Oracle Communications Unified Session Manager supports the IR.94 re-INVITE procedures for adding or removing video mid-call.
- The Oracle Communications Unified Session Manager recognizes that a UE is capable of handling video calls when in the REGISTER message, the sip.video media feature is present in the Contact: header.
- The Oracle Communications Unified Session Manager supports Audio-Video Profile with Feedback (AVPF) and SDP Capability Negotiation (RFC 5939) as documented in TS 26.114 section 6.2.1a.
- When the Oracle Communications Unified Session Manager supports sessions (200 OK received after an INVITE) with a single media stream throughout the session duration, it may receive any of the following notifications in the Specific-Action AVP of the RAR from the PCRF:
  - INDICATION_OF_LOSS_OF_BEARER (2)
  - INDICATION_OF_RELEASE_OF_BEARER (4)
  - INDICATION_OF_OUT_OF_CREDIT (7)
  - INDICATION_OF_FAILED_RESOURCES_ALLOCATION (9)

  The Oracle Communications Unified Session Manager then:

  1. Replies with an RAA acknowledging the RAR
  2. Sends a BYE to both the UE and the Core, then
  3. Sends an STR to the PCRF indicating the session has been terminated.
- When the Oracle Communications Unified Session Manager supports sessions (200 OK received after an INVITE) with multiple simultaneous media streams sometime during the session, it may receive any of the following notifications in the Specific-Action AVP of the RAR from the PCRF:
  - INDICATION_OF_LOSS_OF_BEARER (2)
  - INDICATION_OF_RELEASE_OF_BEARER (4)
  - INDICATION_OF_OUT_OF_CREDIT (7)
  - INDICATION_OF_FAILED_RESOURCES_ALLOCATION (9)

  The Oracle Communications Unified Session Manager then:

  1. Sends an RAA acknowledging the RAR, and no further action is required When an eSR-VCC handover occurs, the ATCF (Oracle Communications Unified Session Manager) sends a INVITE/UPDATE without video (voice-only SDP) toward the target UE.

IR.94 Loss Of Voice Bearer

The Oracle Communications Unified Session Manager provides compliance with an IMS Profile for Conversational Video Service (IR.94) requirement that specifies the termination of a multi-media session (voice and video) if the voice bearer is lost.

Version S-CZ7.2.0 adds a new parameter (options **terminate-on-voice-bearer-release**) that addresses an unlikely, but possible, corner case in which the voice bearer is lost but the video bearer remains in service. Prior to Version S-CZ7.2.0, the Oracle Communications Unified Session Manager would retain the call as a video-only session. This behavior is not compliant with IR.94, which specifies that the video-only session should not be allowed to continue.

The option **terminate-on-voice-bearer-release** enables compliance with the IR.94 standard. Compliance is ensured by basing the decision to terminate or continue a multi-media session on the state of the voice bearer exclusively; the state of any other media bearer (video, for example) plays no role in the decision process. Consequently, if the voice bearer fails, the call terminates; failure of the video bearer, or other media stream, is not pertinent to call termination or continuance.

IR.94 Loss Of Voice Bearer Configuration

Use the following procedure to enable IR.94 compliance.

1. Access the **ext-policy-server** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# media-manager
ORACLE(media-manager)# ext-policy-server
ORACLE(ext-policy-server)#
```

2. Select the **ext-policy-server** object to edit.

```
ORACLE(ext-policy-server)# select
<name>:1:  name=extpol1

selection: 1
ORACLE(ext-policy-server)#
```

3. Use options terminate-on-voice-bearer-release to enable I.94 compliance.

```
ACMEPACKET(ext-policy-server)# options +terminate-on-voice-bearer-release+
    ACMEPACKET(ext-policy-server)#
```

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

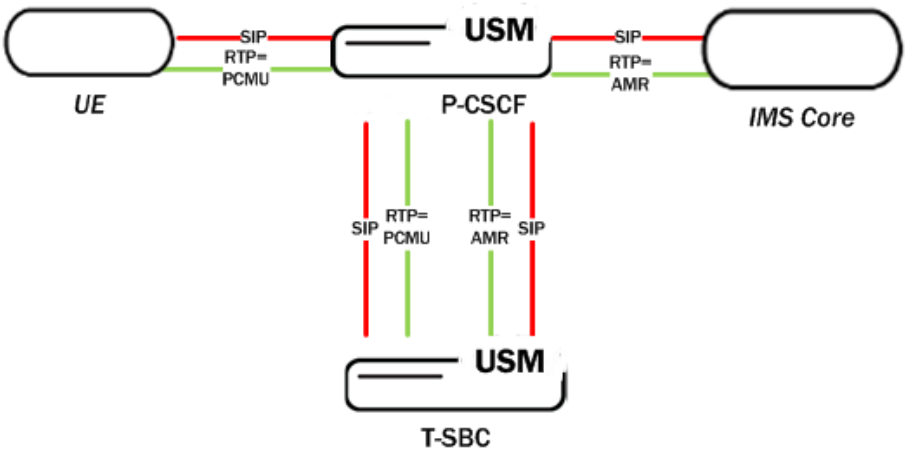
Pooled Transcoding

The term pooled transcoding refers to a deployment model for IMS environments involving two or more Oracle Communications Unified Session Managers. The first is an Oracle Communications Unified Session Manager acting as the P-CSCF, and the others are one or more Oracle Communications Unified Session Managers deployed with transcoding capabilities (referred to as T-SBCs). The T-SBC provides transcoding resources—a pool—that the Oracle Communications Unified Session Manager can invoke on-demand.

In the pooled transcoding model, the Oracle Communications Unified Session Manager sits between realms or between user endpoints that require transcoding between their preferred codecs to communicate. This deployment model conserves resources on both the Oracle Communications Unified Session Manager and the T-SBC. While the Oracle Communications

Unified Session Manager serves as the access function with encryption support, the T-SBC supports transcoding in a tunneling gateway (TG) configuration to meet high-density transcoding requirements.

The following diagram shows an Oracle Communications Unified Session Manager positioned between an access UE and the IMS core. The Oracle Communications Unified Session Manager compares SDP offers and answers from the elements it sits between, and uses the results to determine whether or not a given session requires transcoding. If transcoding is required, the Oracle Communications Unified Session Manager invokes T-SBC's services. Acting as a B2BUA, T-SBC uses information from the P-CSCF's SIP messaging to transcode between the applicable codecs and then to route SIP signaling back to the P-CSCF on the egress.



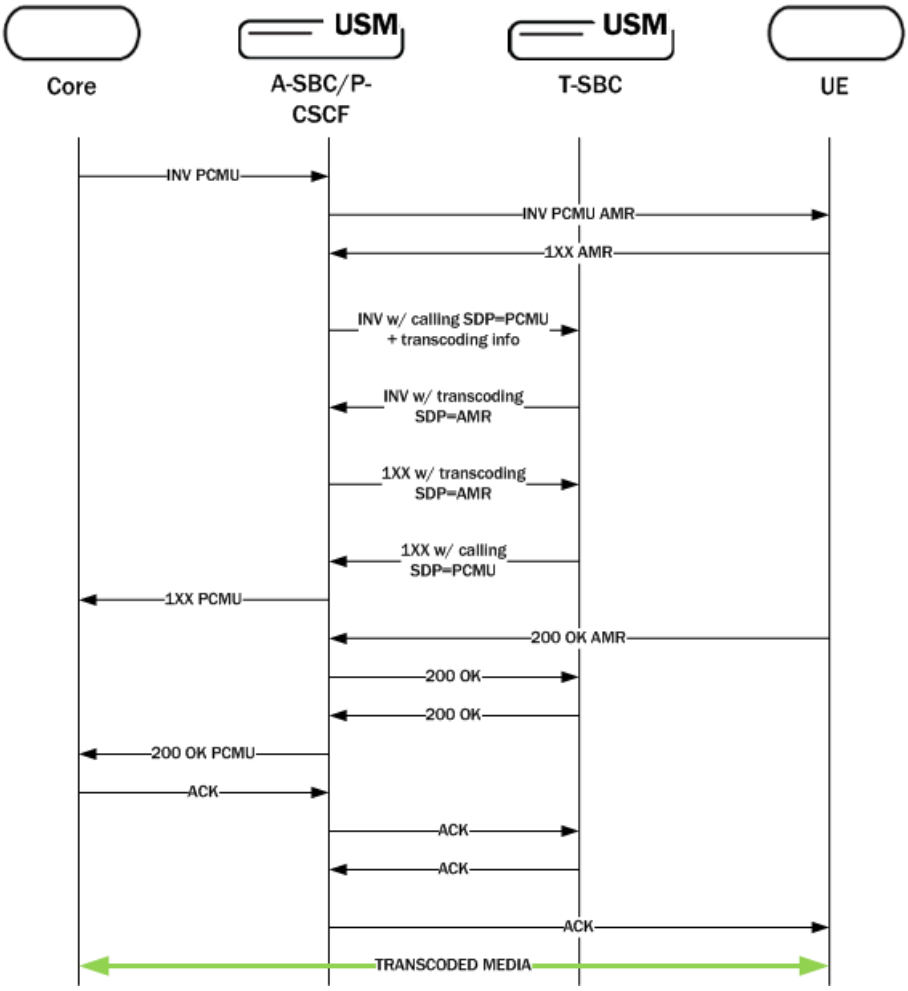
When a call comes in, the system creates two ports for the caller and callee realms respectively. When the call needs pooled transcoding, Oracle Communications Unified Session Manager creates four ports on the pooled transcoded realm and two additional ports on the caller realm. All the ports are removed at the end of the call.

For example, for a call between the realms net172 (Offer) and net145 (Answer) :

Before pooled transcoding	During pooled transcoding	After pooled transcoding
Net172 -> 2 Ports	Net172 -> 4 Ports	Net172 -> 2 Ports
Net145 -> 2 Ports	Net145 -> 2 Ports	Net145 -> 2 Ports
Net192 (Transcoding Realm) -> 0 Ports	Net192 (Transcoding Realm) -> 4 Ports	Net192 (Transcoding Realm) -> 0 Ports

In the previous example, the OCUSM uses four ports on Net192 are for communicating with the transcoding OCUSM. The OCUSM uses the two newly created ports on Net172 for RTP communication, while the original two ports are not used.

The following diagram shows what a call flow between entities in such a deployment model looks like:



Supported Codecs

The is a list of transcodable codecs Oracle Communications Unified Session Manager supports. The pooled transcoding deployment model also supports transcoding for these codecs.

- PCMU
- CMA
- G729
- G729A
- iLBC
- telephone-event
- G726
- G726-16
- G726-24
- G726-32
- G726-40

- G722
- G723
- GSM
- AMR
- AMR-WB

Implementation Details

From the Oracle Communications Unified Session Manager (P-CSCF) perspective, the T-SBC is represented as a transcoding agent whose services the P-CSCF can invoke when offer-answer exchanges reveal transcoding is required. Once that determination is made, the P-CSCF initiates communication with the T-SBC. The P-CSCF uses its public SIP interface and a corresponding realm reserved for communication with a T-SBC, which is configured as a transcoding agent on the P-CSCF. Multiple transcoding agents can be configured, and can be IP addresses, session agents, or session agent groups (SAGs).

If there is more than one transcoding agent for the P-CSCF to choose from, the P-CSCF bases its selection on the order in which the transcoding agents were entered in the list. When conditions call for it, the P-CSCF will try each transcoding agent listed one by one until it:

- Receives a 2xx response from a transcoding agent,
- The list is exhausted, or
- The original transaction times out.

If the transcoding agents are session agents with hostnames, the P-CSCF uses DNS to resolve the hostnames and then tries the hosts in order. In the case when transcoding agents are SAGs, the P-CSCF selects the session agent according to the selection strategy configured for the SAG. The P-CSCF will recurse through all members of the SAG when SAG recursing is enabled. Whenever session agents and SAGs do not have ports or transport protocols specified, the defaults are 5060 and UDP respectively.

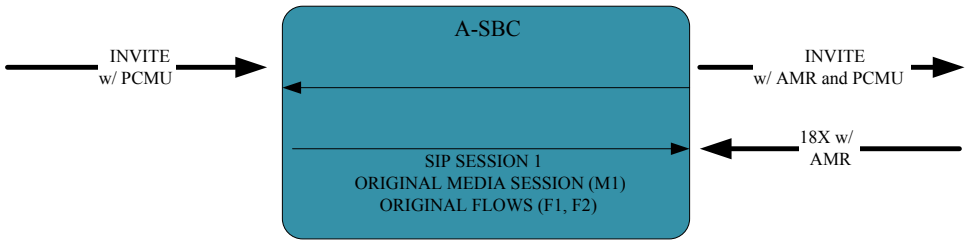
Once it identifies a transcoding agent, the P-CSCF sends an INVITE to which the transcoding agent—the T-SBC—responds with a 2xx message. Configuring the P-CSCFs as a session agent and disabling dialog transparency (in global SIP configuration) on the T-SBC allows it to accept SIP messages from the P-CSCF. Then the T-SBC acts as a B2BUA, using the information received in the P-CSCF's INVITEs to invoke transcoding and to route SIP signaling back to the P-CSCF on the egress.

Application Scenarios

This section discusses two application scenarios for pooled transcoding, one for how the P-CSCF handles an INVITE with SDP and one for how it handles an INVITE without SDP.

Scenario 1 INVITE with SDP

When the Access Session Border Controller (A-SBC) receives an INVITE with SDP, the A-SBC creates a SIP session and an associated media session with two flows for audio. The A-SBC applies the appropriate codec policy (with **add-on-egress** configured), so that the egress INVITE contains the necessary codec.

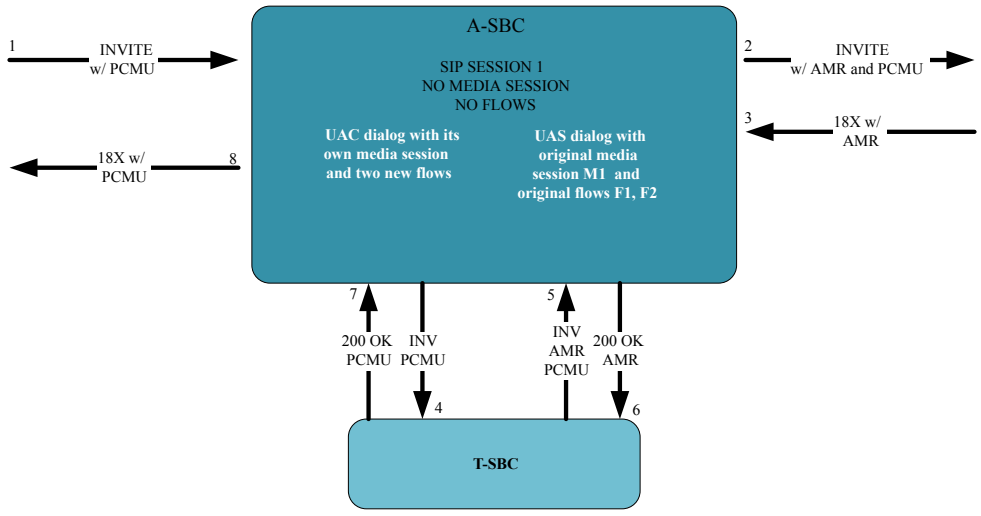


The A-SBC receives an answer to the INVITE, and when the answer contains the added codec the A-SBC invokes the Transcoding Session Border Controller (T-SBC) using an INVITE with the same SDP as the INVITE received on ingress. The communication between the A-SBC and the T-SBC is a separate dialog associated with a new media session.

The following code block shows and example of the INVITE the A-SBC sends to the T-SBC.

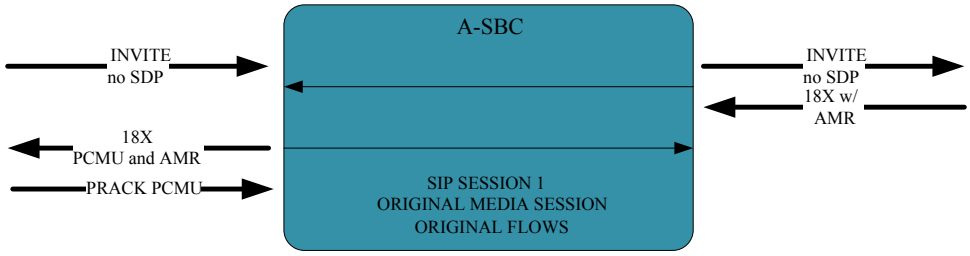
```
INVITE sip:192.168.101.78:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.101.18:5060;branch=z9hG4bK10dspa3058b1io4rk721
Max-Forwards: 70
Call-ID: 9fc71d79b43b4b95de41acefd71a8bc4@192.168.101.78
To: sip:192.168.101.78:5060
Contact: sip:172.16.101.18
From: sip:172.16.101.18;tag=bcf80756721880b7996ccb488b6a1b2f
CSeq: 1 INVITE
Target-Dialog: 1-16881@172.16.18.5;local-tag=1;remote-
tag=16880SIPpTag011;realm=net192
Acme-Codec-Policy: ;ingress;name="in-2833";allow-codecs="* ";add-codecs-on-
egress="";force-ptime="disabled";packetization-time="20";dtmf-in-
audio="disabled";order-codecs=""
Acme-Codec-Policy: ;egress;name="out-2833";allow-codecs="* ";add-codecs-on-
egress="AMR ";force-ptime="disabled";packetization-time="20";dtmf-in-
audio="disabled";order-codecs=""
Content-Type: application/sdp
Content-Length: 196^M
P-Visited-Network-ID: open-ims.test
Route: <sip:192.168.101.18:5060;lr;transport=UDP>
v=0
o=user1 53655765 2353687637 IN IP4 192.168.101.18
s=-
c=IN IP4 192.168.101.18
t=0 0
m=audio 20002 RTP/AVP 96 0
a=rtpmap: 96 AMR/8000
a=rtpmap:0 PCMU/8000
```

Note that the Target Dialog and Acme-Codec-Policy headers communicate operational parameters for pooled transcoding. Using such information, the T-SBC applies two codec policies and returns the INVITE to the A-SBC. The A-SBC moves the media session to itself, but the SIP session does not have a media session at this time.



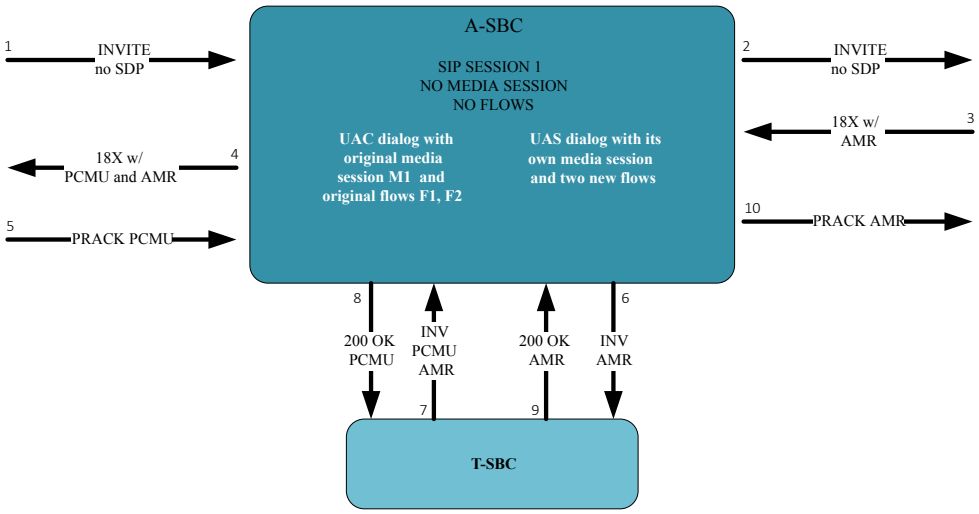
Scenario 2 INVITE without SDP

For an offerless call flow, the system creates a media session when the offer comes in a reliable provisional or final response. The Access Session Border Controller (A-SBC) applies the codec policy and sends the egress offer to the calling UE.



When the A-SBC receives an answer in either a PRACK or ACK message, the A-SBC compares the offer and answer. An answer containing the codec added in the egress offer causes the A-SBC to invoke the Transcoding Session Border Controller (T-SBC) and create a standalone UAC dialog, just as in Scenario 1.

Because the A-SBC advertised its media address in the egress offer to the calling UE, the UAC dialog uses the original media session.



Re-INVITES and Updates with SDP

When a specific session on the Access Session Border Controller (A-SBC) invokes the resources of the Transcoding Session Border Controller (T-SBC), the A-SBC continues to use the same T-SBC for the duration of the session. Anytime when the A-SBC receives a SIP message containing modified SDP, the A-SBC communicates the modification to the T-SBC.

RFC 2833 Considerations

For legacy RFC 2833 inter-working to function properly, the Access Session Border Controller (A-SBC) communicates the RFC 2833 configuration to the Transcoding Session Border Controller (T-SBC) in the UAC dialog.

The following example shows an INVITE with RFC 2833 information sent from the A-SBC to the T-SBC.

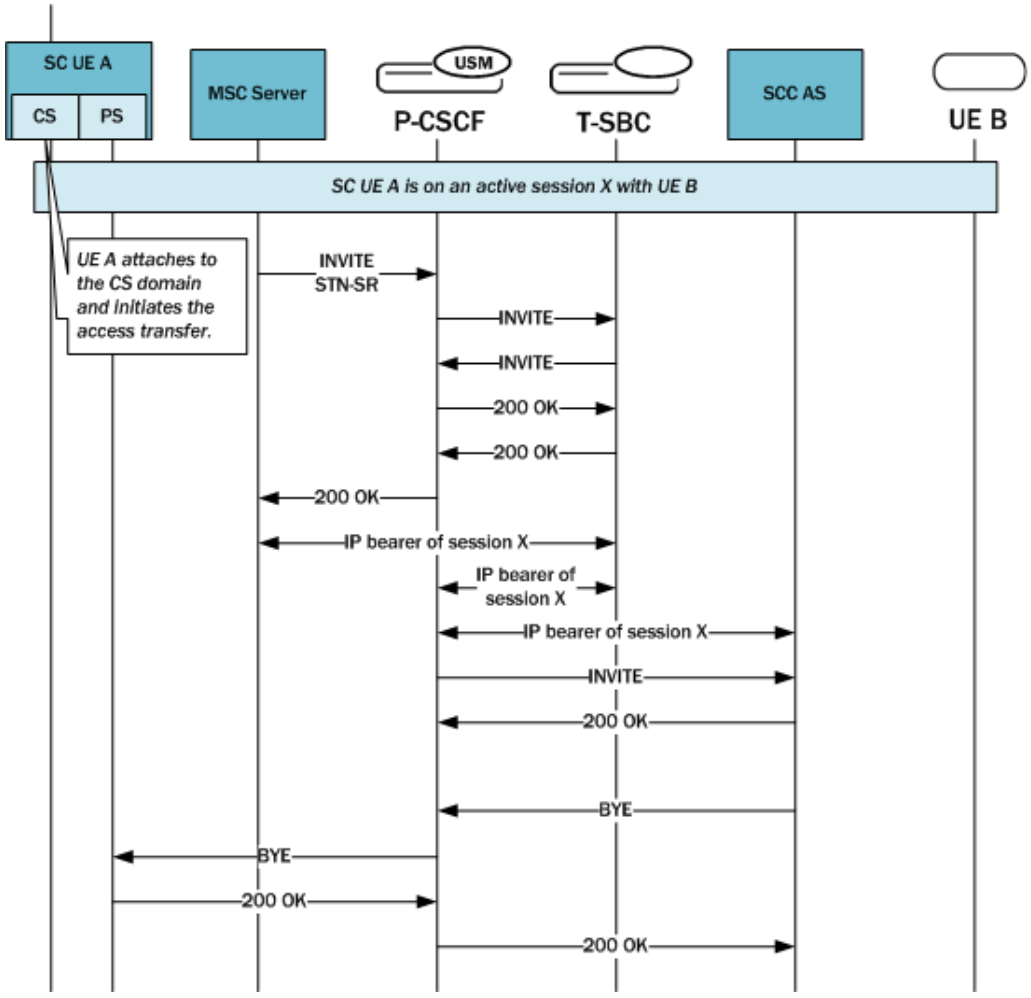
```
INVITE sip:192.168.101.78:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.101.18:5060;branch=z9hG4bK10dspa3058b1io4rk721
Max-Forwards: 70
Call-ID: 9fc71d79b43b4b95de41acefd71a8bc4@192.168.101.78
To: sip:192.168.101.78:5060
Contact: sip:172.16.101.18
From: sip:172.16.101.18;tag=bcf80756721880b7996ccb488b6a1b2f
CSeq: 1 INVITE
Target-Dialog: 1-16881@172.16.18.5;local-tag=1;remote-
tag=16880SIPpTag011;realm=net192
Acme-Codec-Policy: ;ingress;name="in-2833";allow-codecs="* ";add-codecs-on-
egress="";force-ptime="disabled";packetization-time="20";dtmf-in-
audio="disabled";order-codecs=""
Acme-Codec-Policy: ;egress;name="out-2833";allow-codecs="* ";add-codecs-on-
egress="PCMA ";force-ptime="disabled";packetization-time="20";dtmf-in-
audio="disabled";order-codecs=""
Acme-Rfc2833: ;ingress;Digit-
Mode=0;PtTo=101;PtFrom=0;TransNon2833=disabled;TransNonInband=disabled
Acme-Rfc2833: ;egress;Digit-Mode=1;PtTo=101;PtFrom=0
Content-Type: application/sdp
Content-Length: 196
P-Visited-Network-ID: open-ims.test
Route: <sip:192.168.101.18:5060;lr;transport=UDP>
```

```
v=0
o=user1 53655765 2353687637 IN IP4 192.168.101.18
s=-
c=IN IP4 192.168.101.18
t=0 0
m=audio 20002 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
```

eSRVCC Support

For enhanced Signal Radio Voice Call Continuity (eSRVCC), the pooled transcoding deployment model supports instances when:

- Original call is not transcoded, but the handover call is
- Original call is transcoded, but the handover call is not
- Original call is transcoded, and so is handover call



Configuration Requirements and Verification

Pooled transcoding requires specific configuration of the Access Session Border Controller (A-SBC) and the Transcoding Session Border Controller (T-SBC). Note that the Oracle

Communications Unified Session Manager, the A-SBC, does not enforce the requirements. Oracle recommends that you configure your pooled transcoding deployment with care, and verify the configuration to help to identify any potential issues.

A-SBC Configuration Requirements

The Access Session Border Controller (A-SBC) configuration must include the following:

- A public SIP interface the A-SBC can use for communication with the Transcoding Session Border Controller (T-SBC).
- A transcoding realm, which is a separate realm for the public SIP interface used only for communication with the T-SBC.
- Appropriate codec policies, including ones set up to add SDP on the egress leg of the session.
- A global SIP configuration, with **transcoding-realm** set to the name where valid transcoding agents reside, and a **transcoding-agents** list configured with one or any combination of
  - A DNS hostname for a single session agent that can resolve to one or more IP addresses
  - An IPv4 or IPv6 address with or without the port specified (when not specified, the system defaults to port 5060).
  - The name of a session agent group.

T-SBC Requirements

A valid Transcoding Session Border Controller (T-SBC) configuration must include the following:

- A global SIP configuration, with **dialog-transparency** set to **disabled**. You must disable Dialog transparency on the T-SBC, so that the INVITE from the T-SBC contains a different call ID and From tag.
- A public realm with **codec-manip-in-realm** set to **enabled** because the system uses the same realm for transcoding. You must also enable the **mm-in-realm** parameter.

Configuration Verification

You can verify the configuration by using the ACLI **verify-config** command. When verifying the configuration on the Access Session Border Controller (A-SBC), the system displays errors messages when:

- The **transcoding-realm** value configured is not a valid realm.
- Either one of the **transcoding-realm** or **transcoding-agents** parameters is not configured.
- One or more session agent names defined in the **transcoding-agents** list is not a valid session agent.
- The IP address version for a agent in the **transcoding-agents** list is not supported in the transcoding realm you identify. For example, if you list an IPv6 agent in the list and the transcoding realm does not support IPv6.
- The transcoding agent is a hostname and not a valid session agent.

Configure Pooled Transcoding

You must configure a transcoding realm and transcoding agents on your Access Session Border Controller (A-SBC), when used in a pooled transcoding deployment model. Set the parameters as part of the global SIP configuration.

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

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# sip-config
   ORACLE(sip-config)#
   ```

2. If you are adding transcoding support to a pre-existing SIP configuration, you must select the configuration before you can make changes.

3. Enter the name of a configured realm designated as the separate realm for the public SIP interface for exclusive communication with the Transcoding Session Border Controller (T-SBC) in as pooled transcoding deployment.

   ```
   ORACLE(sip-config)# transcoding-realm net157
   ORACLE(sip-config)#
   ```

4. **transcoding-agents**—Enter any IP address, IP address and port combination, session agent hostname, or SAG name to use as a transcoding agent. You can make multiple entries in any combination of these values. For example, you might list an IPv6 address and port, a session agent, and a SAG.

   - To make multiple entries in the list using in one command line, enclose the entire list of value in parentheses (( )), separating each with a Space as in the example below.
   - To add a transcoding agent to an existing list, put a plus sign before the value you want to add, e.g. +154.124.2.8.
   - To remove a transcoding agent from an existing list, put a minus sign before the value you want to remove, e.g. -154.124.2.8.

     ```
     ORACLE(sip-config)# transcoding-agent (sag:sag1 192.168.4.7.3
     192.168.2.6:5061)
     ORACLE(sip-config)#
     ```

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

Monitor Dialogs Between the A-SBC and the T-SBC

You can monitor pooled transcoding communications between the Access Session Border Controller (A-SBC) and the Transcoding Session Border Controller (T-SBC) by using the **show sipd pooled-transcoding** ACLI command. The display shows you information for the client and server User Agents on the A-SBC.

- UAC Dialogs—shows the number of UAC dialogs the A-SBC initiated with the T-SBC. The count is cumulative, and not for any specific session.
- UAS Dialogs—shows the number of UAS dialogs the A-SBC created upon receipt of an INVITE from the T-SBC. The count is cumulative, and not for any specific session.
- XCodeSessions—counts the number of transcoded sessions on the A-SBC.

```
ACMEPACKET# show sipd pooled-transcoding
18:11:28-125
SIP Pooled Transcoding Status
                    -- Period --                 -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
UAC Dialogs          1       1       1          1       1       1
  Early              0       0       0          0       0       0
  Confirmed          1       1       1          1       1       1
  Terminated         0       0       0          0       0       0
UAS Dialogs          1       1       1          1       1       1
  Early              0       0       0          0       0       0
  Confirmed          1       1       1          1       1       1
  Terminated         0       0       0          0       0       0
XCodeSessions        1       1       1          1       1       1
```

Per-Method Statistics

When you set **extra-method-stats** to **disabled** in the global SIP configuration, the system displays the per-method statistics for the Access Session Border Controller (A-SBC) public transcoding-realm.

```
ACMEPACKET# show sipd realms net157 invite
INVITE (18:14:59-36)
                        --------- Server --------   --------- Client --------
Message/Event           Recent      Total  PerMax   Recent      Total  PerMax
                        ------  ---------  ------   ------  ---------  ------
INVITE Requests              0          1       1        0          1       1
Retransmissions              0          0       0        0          0       0
100 Trying                   0          1       1        0          1       1
200 OK                       0          1       1        0          1       1
Response Retrans             0          0       0        0          0       0
Transaction Timeouts         -          -       -        0          0       0
Locally Throttled            -          -       -        0          0       0

Avg Latency=0.000 for 0
Max Latency=0.000
BurstRate Incoming=1 Outgoing=0

ACMEPACKET# show sipd realms net157 bye
BYE (18:15:13-50)
                        --------- Server --------   --------- Client --------
Message/Event           Recent      Total  PerMax   Recent      Total  PerMax
                        ------  ---------  ------   ------  ---------  ------
BYE Requests                 0          1       1        0          1       1
Retransmissions              0          0       0        0          0       0
200 OK                       0          1       1        0          1       1
Transaction Timeouts         -          -       -        0          0       0
Locally Throttled            -          -       -        0          0       0

Avg Latency=0.000 for 0
Max Latency=0.000
BurstRate Incoming=1 Outgoing=0
```

Notes on the DIAMETER Rx Interface

DIAMETER Rx support for external bandwidth management is specialized for pooled transcoding.

For pooled transcoding, the identifier appears in this altered format: <policy server name>;<policy server realm>;<dns suffix>. Because there are two dialogs (one for the server UA and one for the client UA), there are two separate media sessions. This situation requires the Oracle Communications Unified Session Manager to generate two different DIAMETER session identifiers. At the same time, the Oracle Communications Unified Session Manager needs to maintain the AAR and STR associations with the original SIP session. To keep this

relationship clear, the Oracle Communications Unified Session Manager keeps the DIAMETER session identifier between the two media sessions the same for the two UAs.

The MBCD session identifier associated with the media session for the original SIP session is retained for the DIAMETER session identifier for the duration of the SIP session.

Accounting and Transcoding

An Access Control Record (ACR) record for a typical SIP session looks the same as one for a transcoded session, except that the forward codec (Acme-FlowType_FS1__F) differs from the reverse codec (Acme-FlowType_FS1__R), due to transcoding.

For the RADIUS and Diameter protocols, the ACR record shows only the IP address and port number of the two original endpoints. The record shows no details about the Transcoding Session Border Controller (T-SBC) because the system does not support transcoding for RADIUS and Diameter.

Dynamic Sessions Agents for Home-Remote S-CSCF Liveliness

For IMS applications, you can configure your Oracle Communications Unified Session Manager to create session agents dynamically for remote S-CSCFs on in-coming service routes. Dynamic session agents inherit properties of the static session agents with which they are associated, and the Oracle Communications Unified Session Manager takes them out of service when they are deemed no longer responsive according to the liveliness mechanism you set.

Discovery

The Oracle Communications Unified Session Manager, acting as a P-CSCF, can discover remote S-CSCFs using Service-Route header that returns with a 200 OK response from the registrar for a REGISTER request from the endpoint. The system takes the top Service Route from the response and uses it as the first hop in the egress route for the endpoint. Because the creation of dynamic session agents is based on the Service Route returned in the 200 OK, there is no impact to handling AoRs with multiple contacts.

In addition, the system stores the Service Route header data with the endpoint's registration cache. If the Service Route is an FQDN, a DNS look-up is used to provide the route to the S-CSCF.

Creation

For your Oracle Communications Unified Session Manager to create session agents dynamically, you must enable the **create-dynamic-sa** in the global SIP configuration. This parameter defaults to **disabled**, meaning that no dynamic session agents will be created.

When you set the parameter to **enabled**, the Oracle Communications Unified Session Manager decides whether or not to create dynamic session agents in the following ways. The system will create up to five (5) dynamic session agents.

- If the Service Route is an IP address, the Oracle Communications Unified Session Manager attempts to find an exact match for that IP address against pre-existing session agents. If no exact match appears, the system will not create dynamic session agents.

- If the Service Route is an FQDN and matches an existing session agent, the Oracle Communications Unified Session Manager assigns the remote S-CSCF to that session agent.
  Service Routes with FQDNs might not match any session agents. However, the Service Route's FQDN can match the DNS suffix of a wildcard session agent. When this wildcard matching occurs, the Oracle Communications Unified Session Manager creates a new dynamic session agent with the original wildcard session agent as its parent. The new dynamic session agent inherits all configuration properties of the parent session agent.
- If the Service Route is neither an IP address nor an FQDN and the system is unable to match any statically defined or wildcard session agents, then the Service Route is not associated with any session agents.

Property Inheritance

Because dynamic session agents are created on the basis of static or wildcarded session agents, you can think of them as children of the original--or parent--session agent. This relationship means that the child, dynamic session agent inherits the configuration properties of its parent, static or wildcard session agent.

These inherited configuration properties and their effects are:

- The FQDN's DNS resolution of the new dynamic session agent to IP addresses.
- Monitoring for liveliness via your configuration settings for the ping method and interval, transaction timeout, or OOS response. The dynamic session agent also inherits its parent's criteria for being taken out of service, with the exception that a dynamic session agent will not be taken out of service until it expires on its own. This gives the dynamic session agent time to return to service.
- The setting for **ping-all-address**, which when enabled causes new routes (internal session agents) fork and makes the dynamic session agent the parent. The system caps the limit of five routes (or internal session agents) per dynamic session agent. If the FQDN resolves to more than five IP addresses, the system only uses the first five to create routes (internal session agents), and then pings the internal session agents.
- Suppression of the heartbeat (or ping) of the IP address in the presence of traffic. If traffic to a specific IP address stops, then the Oracle Communications Unified Session Manager resumes pinging within the time you set for the **ping-interval**.
- The setting for the **invalidate-registration** option. If the dynamic child session agent goes out of service, the corresponding registration cache entries of users that have the Service Route pointing to this session agent will be invalidated.
  This invalidation means that the next REGISTER request from that user will not receive a local response. Any other services this user requires will not use the Service Route information stored in its registration cache. Instead, the system will route it to the next hop as determined by other means, such as the local policy. At this time, the user would must be re-registered by the registrar, a process that might return a new Service Route to be updated in the registration cache.

Deletion

The Oracle Communications Unified Session Manager needs to delete dynamic session agents no longer in use. But dynamic session agents should not be deleted too soon, in case they return to service. So, deletion occurs according to the process this section describes.

For each dynamically created session agent, the Oracle Communications Unified Session Manager assigns and tracks the last registration expiry time. It determines this time by doubling

the registered endpoint's core side expiry. Where X is the last registration expiry value and Y is the registered endpoint's core side expiry, the Oracle Communications Unified Session Manager performs checks according to this criteria:

X > (2 * Y)

The the system updates the expiry time with the greater value. This way, whenever the dynamic session agent re-registers, it will have an updated expiry timer value.

Timeouts can also cause dynamic session agents to be deleted. Pings, status changes, transaction timeouts, DNS expiries and other system occurrences can trigger timeouts. When the Oracle Communications Unified Session Manager detects that a timeout has occurred, the dynamic session agent is deleted.

How to Wildcard a Session Agent

You can create a wildcard session agent using the session agent's hostname parameter when configured with an FQDN.

To configure a wildcard session agent:

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

   ```
   ORACLE# configure terminal
   ORACLE(configure)#
   ```

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

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

3. Type **session-agent** and press Enter.

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

   If you are wildcarding and existing session agent, you have to select the configuration before you make changes.

4. **hostname**—To wildcard a session agent, you simply replace the value you want to wildcard with an asterisk (*). Also note that your value must lead with the asterisk (*), as in the following example.

   ```
   ORACLE(session-router)# hostname *xyz.com
   ORACLE(session-agent)#
   ```

5. Save and activate your configuration.

Enabling the Global SIP Configuration for Dynamic Session Agents

To use dynamic session agents for remote S-CSCFs on in-coming service routes, you need to set the create-dynamic-sa parameter in the global SIP configuration to enabled.

To configure the ping mode for a session agent:

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

   ```
   ORACLE# configure terminal
   ORACLE(configure)#
   ```

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

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

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

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

   If you are adding this support to a pre-existing SIP configuration, you have to select the configuration before you make changes.

4. **create-dynamic-sa**—To support the creation of dynamic session agents for remote S-CSCFs on in-coming service routes, change this parameter from **disabled** (default) to **enabled**.
5. Save and activate your configuration.

Enhanced eSRVCC Call Continuity

As the LTE Evolved Packet Core (EPC) continues to expand, voice deployments (VoLTE) appear more and more, with the Oracle SBC and P-CSCF playing key roles. 3GPP standards that define how LTE communications take place also continue to evolve, identifying and solving critical issues to increase effectiveness and efficiency. One such issue is session continuity, keeping session transfers between LTE and existing 2G and 3G networks as seamless as possible. Single Radio Voice Call Continuity (eSRVCC) offers one solution for the session continuity issue.

In its role as the P-CSCF and IMS-GW, the Oracle Communications Unified Session Manager can provide eSRVCC by acting as signaling and media anchor points to handover calls to 3G networks smoothly. These anchoring points are called Access Transfer Control Function (ATCF) and the Access Transfer Gateway (ATGW), both of which are logical additions to the Oracle Communications Unified Session Manager's IMS support.

The behavior of these two anchoring points, ATCF and ATGW, is defined by 3GPP in Release 12 of Technical Specification TS 24.237. Oracle Communications developed these functional entities based on the initial version of TS 24.237 Release 10. To align with Release 12, Requirement 4944 introduces the following two new elements to the functional design:

Handsets and Session Continuity

Session continuity can be effected by handsets, which can be dual-mode (3G+WiFi and 3G +WiMAX, for example). Such a handset has two receivers or radios to initiate calls simultaneously. An LTE handset has only one receiver and is able to attach to a single LTE or 3G network at a given time.

The question of session continuity appears in 3GPP standards Release 8, 9, and 10—each building on the previous. Release 8 defines Single Radio Voice Call Continuity (SRVCC) as the mechanism for moving active voice sessions between LTE and existing 2G or 3G circuit networks. In moving over sessions such as these, it is key to keep latency as low as possible to increase the possibility of successful handovers.

Release 9, because of variable signaling latencies within the core network, could not guarantee smooth handovers to circuit networks. And though IMS provides great flexibility in locating application servers remotely from the UE, that flexibility actually increases the latency in signaling media changes to the access network. Due to the high total signaling latency and the

difficulty of successfully coordinating handover timing between 3G and 4G call legs, the possibility of call drops increased.

Anchors for Signaling and Media

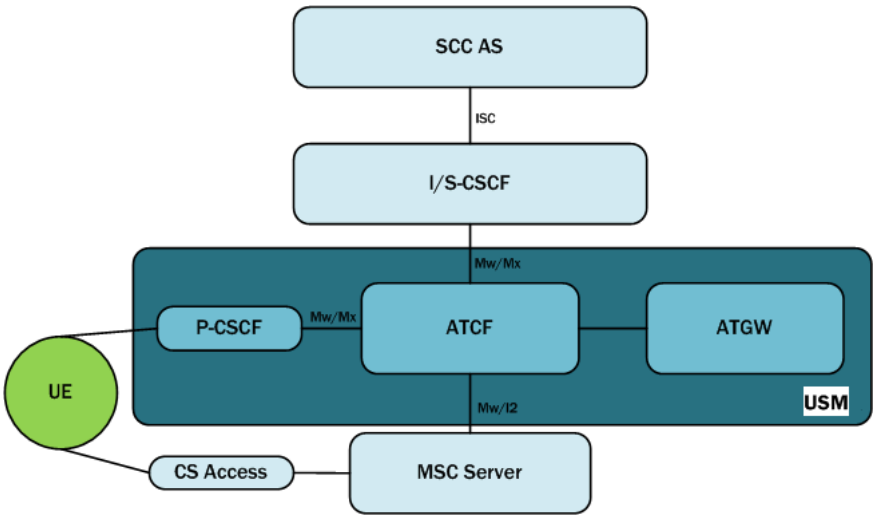
Release 10 addresses the latency concern by proposing these two logical entities, the anchoring points called the ATCF and ATGW. The Oracle Communications Unified Session Manager (OCUSM) can fulfill the tasks set to both entries:

- ATCF—A signaling anchor point co-located with the UE access network. The ATCF is responsible for:
  - Allocating the Session Transfer Number for Single Radio (STN-SR).
  - Instructing the ATGW to anchor the media path for originating and terminating sessions.
  - Tracking sessions (in alerting, active, or held states) so it can perform the access transfer of the selected session. Tracking this information allows the ATCF to support transferring the first session.
  - Performing Access Transfer and updating the ATGW with the new media path for the access call leg, without requiring updating from the remote leg.
  - After the access transfer, updating the Service Centralization and Continuity Application Server (SCC AS) that the transfer has taken place, ensuring that the Terminating Access Domain Selection (T-ADS) has updated information about the access currently used.
  - Handling failures during the access transfer.
  - Handling mid-call support for the access transfer using MSC server-assisted mid-call support.
- Access Transfer Gateway (ATGW)—A media anchor point co-located with the UE access network. Controlled by the ATCF, the ATGW anchors media both for the duration of a call and after the access transfer, based on the local configuration in the serving network.

Originating and terminating sessions are anchored in the ATCF and ATGW already during session set-up. For the first transferred session and the second established session, the SCC-AS provides session state information for the alerting, held, and conference states.

When a UE makes or receives a call, signaling and media are anchored at the ATCF and ATGW. At the point call handover point, the Visited-Mobile Switching Center (V-MSC) receives the handover message from Mobility Management Entity (MME, an EPC network element). The V-MSC then sends a call request to the local ATCF rather than sending the call request home or to the SCC-AS, as defined in 3GPP Release 8. Sending the call request to the ATCF reduces the number of hops required to initiate a media stream change to a new access network.

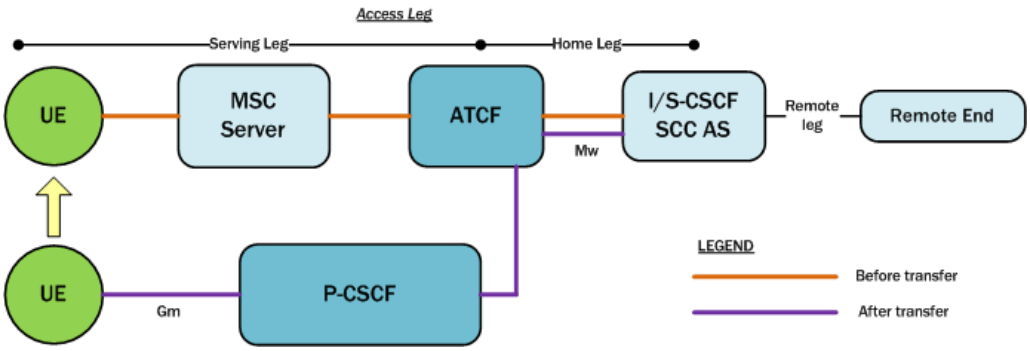
Acting as the ATGW, the OCUSM can immediately affect media switchovers without further core signaling. In this role, the OCUSM keeps RTP media continuity between endpoints using its Hide Media Update (HMU) functionality: The elements on the core side of the eSRVCC update will not see changes to SDP sources and destinations because the Synchronization Source identifiers (SSRCs) are masked. Media can thus flow directly to and from the 3G-attached MSC and onward to the UE with minimal interruption.



Note that if the MSC server-assisted mid-call feature is not supported or MSC server is not enhanced for ICS support, the interface between the MSC server and the ATCF will be Mw. If the MSC server-assisted mid-call feature is supported or the MSC server is enhanced for ICS, the interface between the MSC server and the ATCF will be I2.

Architectural View

This diagram is an architectural view of the control and user planes, depicting communication both before and after transfer. This depiction assumes the PGW and the P-CSCF are in the serving network and support IMS voice roaming (if not the home network). So, the ATCF resides in the serving networking (home network if not roaming). The access leg of the session is divided into the serving leg and home leg. In an actual network, there might be other server IMS nodes.



IMS Registration Details

This section discussed the IMS registration process when a UE attempts to register with the home network, but must do so using an ATCF/P-CSCF. Based on the operator policy and the home network's support for eSRVCC, the ATCF allocates an STN-SR to the session and includes itself in the signaling path for subsequent registration-related messaging. To understand whether or not the home network supports eSRVCC, the ATCF uses service-level agreements and can also determine if eSRVCC is activated in the SCC AS by the reception of a C-MSISDN/ATU-SI during session start-up.

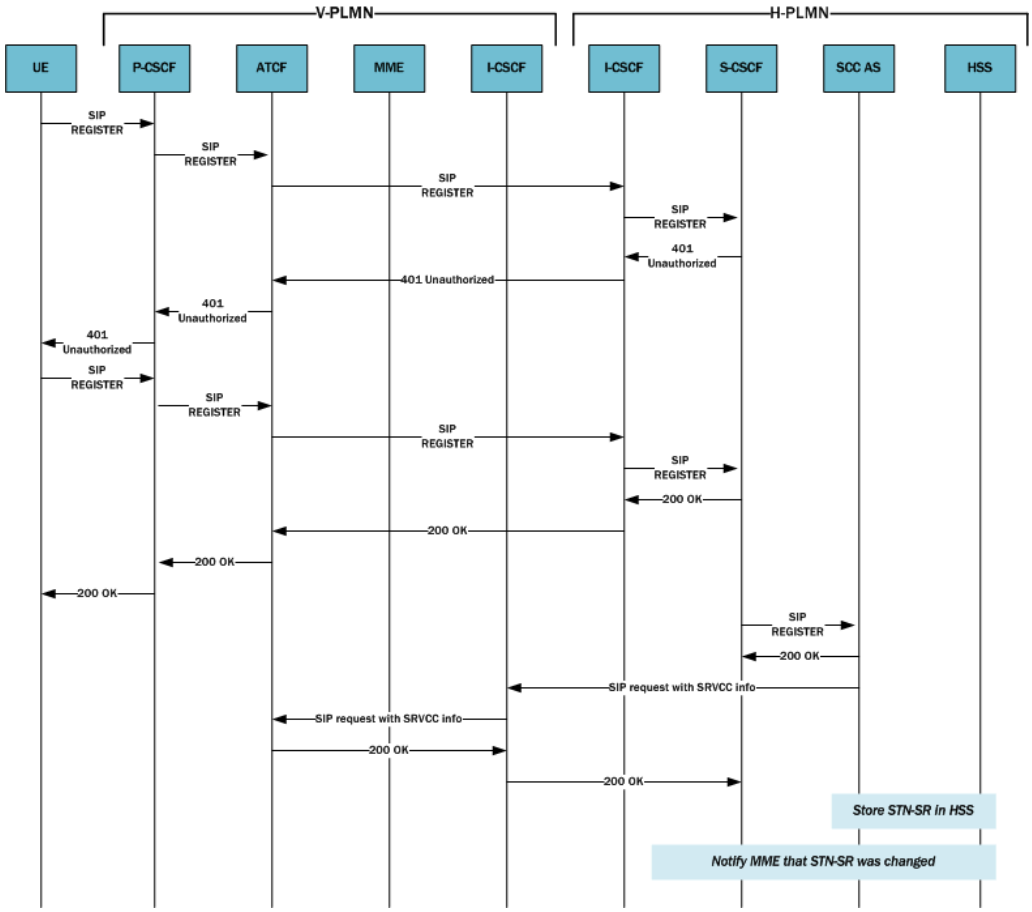
If the ATCF generates an STN-SR, it includes the STN-SR in requests it forwards to the I/S-CSCF. The path and route information for the SIP REGISTER request sent to the S-CSCF is:

- Path—ATCF URI for terminating requests (uniquely identifies registration or registration flow), followed by P-CSCF URI for terminating requests.
  The header field containing the ATCF URI for terminating requests contains the g. 3gpp.atcf media feature tag (indicating this URI supports the ATCF role). This tag contains the STN-SR and the ATCF PSI, showing this URI can receive SIP message requests with SRVCC-related information.
- Route—URI of the entry point of the UE's home network.

The ATCF tracks existing registrations for the UEs it has served. Each registration is identified by the P-CSCF path URI. The following information remains in the ATCF's registration cache: the S-CSCF service route URI, the ATU-STI, and the C-MSISDN. When a UE's registration expires or it is de-registered, the ATCF can remove any SRVCC information bound to the registration.

This ladder diagram shows the IMS registration process between these entities:

- The Visited Public Land Mobile Network (V-PLMN)—The network a mobile subscriber uses when that subscriber is roaming.
- The Home Public Land Mobile Network (H-PLMN)—The mobile subscriber's home network.



SIP Register Request UE to ATCF

The following is an example of the SIP REGISTER request the UE sends to the ATCF/ P-CSCF.

```
REGISTER sip:home1.net SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:131313:ccc:eee];comp.sigcomp;branch=z9hG4bRnasiuen8
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151DOFCE11
From: <sip:user1_public1@home1.net>;tag=2hiue
To: <sip:user1_public1@home1.net>
Contact: <sip:[5555::aaa:bbb:ccc:eee];comp=sigcomp>;
+sip.instance="<urn:gsma:imei:90420156-
    025763-0>;+g.3gpp.icsi-ref="urn%3Aurn-7%3gpp-service.ims.icsi.mmtel"
Call-ID: E05133BD26DD
Authorization: Digest username="user1_private@home1.net",
realm="registrar.home1.net", nonce="",
    uri="sip:home1.net", response=""
Security-Client: ipsec-3gpp; alg=hmac-sha-1-96; spi-c=23456789; spi-s=12845678;
port-c=1234;
    port-s=5678
Require: sec-agree
Proxy-Require: sec-agree
CSeq: 1 REGISTER
Supported: path, gruu
Content-Length: 0
```

SIP Register Request ATCF to S-CSCF

The following is an example of the SIP REGISTER request the ATCF sends to the S-CSCF.

```
REGISTER sip:homel.net SIP/2.0
Path: <sip:termsdgfdfwe@actf.visited2.net>;+g.3gpp.atcf="<tel:+1-237-888-9999>";
+g.3gpp.atef-
    psi="<sip:actf.visited2.net>",<sip:aga2gfgf@pcscfl.visited2.net:5070;ob>
Route: <sip:icscf.homel.net;lr>
P-Visited-Network-ID:
P-Charging-Vector:
Via: SIP/2.0/UDP actf.visited2.net:5060;branch=z9hG4bKnas5889; SIP/2.0/UDP
    pcscfl.visited2.net:5060;branch=z9hG4bKnas56565, SIP/2.0/UDP

[5555::aaa:bbb:ccc:eee];comp=sigcomp;branch=z9hG4bKnasiuen8;rport=5060;received=5
555::aaa:bbb:ccc:eee
Max-Forwards: 68
P-Access-Network-Info:
From:
To:
Contact:
Call-ID: Authorization:
Require:
Proxy-Require:
CSeq:
Supported:
Content-Length:
```

SIP 200 OK from S-CSCF

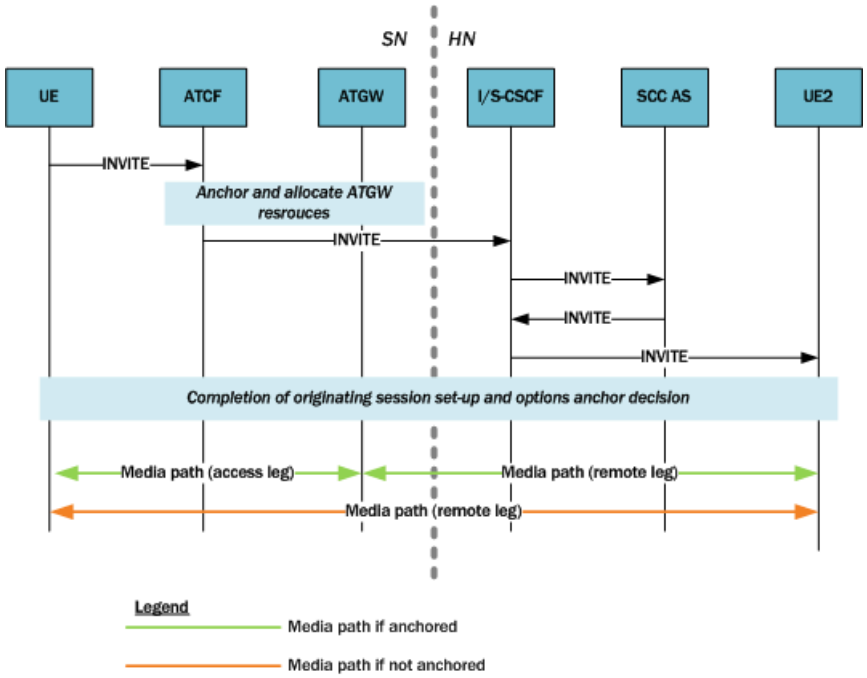
The following is an example of the SIP 200 OK from the S-CSCF.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP icscf1_p.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
     pcscf1.visited1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
     [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Path: <sip:term@pcscf1.visited1.net;lr;ob>
Service-Route: <sip:orig@scscf1.home1.net;lr>
From:
To:
Call-ID:
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>;
   pub-gruu=" sip:user1_public1@home1.net;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
   ;temp-gruu="sip:tgruu.7hs==jd7vnzga5w7fajsc7-ajd6fabz0f8g5@example.com;gr"
   ;+sip.instance="< urn:gsma:imei:90420156-025763-0 >"+g.3gpp.icsi-ref="urn%3Aurn-7%3gpp-
   service.ims.icsi.mmtel";+g.3gpp.ics="principal";+g.3gpp.accesstype="cellular1"
   ;expires=600000;+g.3gpp.iut-controller
CSeq:
Supported: path, outbound
Require: outbound
Date: Wed, 11 July 2001 08:49:37 GMT
P-Associated-URI: <sip:user1_public2@home1.net>, <sip:user1_public3@home1.net>, <sip:+1-212-555-
   1111@home1.net;user=phone>
Content-Length:
```

Originating Sessions for SRVCC with ATCF

For initial SIP requests, the ATCF distinguishes SIP INVITE requests with the ATCF URI for originating requests in the topmost Route header field. And when receiving such an originating SIP INVITE request, the ATCF will do the following prior to forwarding it:

- Insert the Record-Route header field with its own SIP URI.
- If the latest SRVCC information received for a session contains a C-MSISDN and ATU-STI:
  - The ATCF will associate the session being established with the C-MSISDN and ATU-STI bound to the registration.
  - The ATCF will replace the SDP offer in the originating SIP INVITE with updated SDP the ATGW provides. Replacement occurs if the originating SIP INVITE contains SDP and a determination to anchor media has been made (according to the operator policy as specified in the 3GPP standard).



SIP INVITE for SRVCC Using the ATCF

The following is an example of the SIP INVITE the UE sends to the ATCF/P-CSCF.

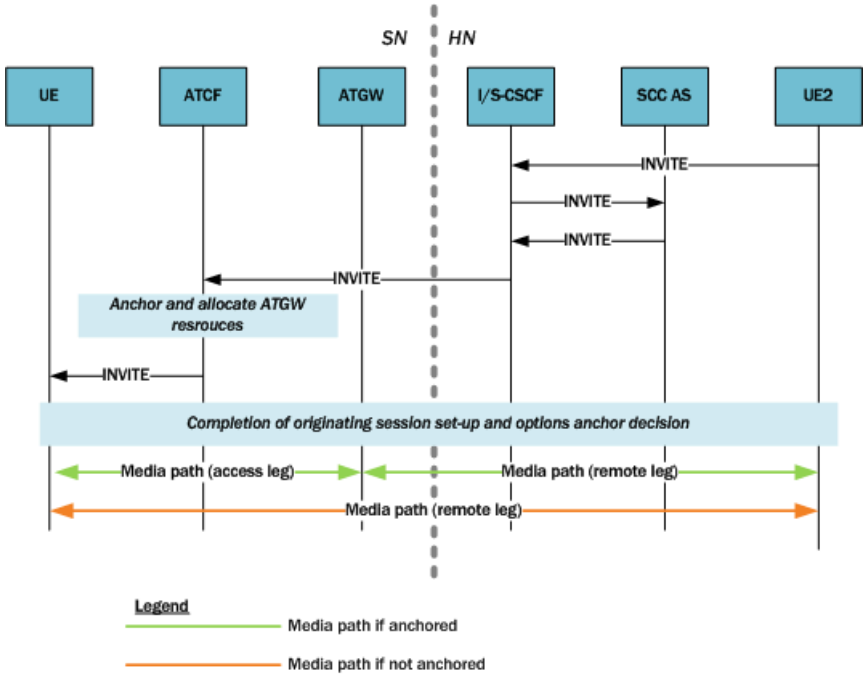
```
INVITE tel:+1-212-555-2222 SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 70
Route: <sip:pcscf1.visited2.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
P-Preferred-Identity: "John Doe" <sip:user1_public1@home1.net>
P-Preferred-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
Privacy: none
From: <sip:user1_public1@home1.net>;tag=171828
To: <tel:+1-212-555-2222>
Call-ID: cb03a0s09a2sdfglkj490333
Cseq: 127 INVITE
Require: sec-agree
Supported: precondition, 100rel, gruu
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432; spi-s=87654321; port-
   c=8642; port-s=7531
Contact: <sip:user1_public1@home1.net;gr=urn:uuid:f81d4fae-7dec-11d0-a765-
   00a0c91e6bf6;comp=sigcomp>;+g.3gpp.icsi-ref="urn%3Aurn-7%3gpp-service.ims.icsi.mmtel"
Accept-Contact: *;+g.3gpp.icsi-ref="urn%3Aurn-7%3gpp-service.ims.icsi.mmtel"
Allow: INVITE, ACK, CANCEL, BYE, PRACK, UPDATE, REFER, MESSAGE
Content-Type: application/sdp
Content-Length: (…)

v=0
o=- 2987933615 2987933615 IN IP6 5555::aaa:bbb:ccc:ddd
s=-
c=IN IP6 5555::aaa:bbb:ccc:ddd
t=0 0
m=audio 3456 RTP/AVP 97 96
b=AS:25.4
a=curr:qos local sendrecv
a=curr:qos remote none
a=des:qos mandatory local sendrecv
a=des:qos none remote sendrecv
a=rtpmap:97 AMR
a=fmtp:97 mode-set=0,2,5,7; maxframes=2
a=rtpmap:96 telephone-event
```

Terminating Sessions for SRVCC with ATCF

For initial SIP requests, the ATCF identifies SIP INVITE requests with the ATCF URI for terminating requests in the topmost Route header field. These are called terminating SIP INVITE requests.

When it receives a terminating SIP INVITE, the ATCF does the following if the INVITE has a Record-Route header field with the g.3gpp.srvcc media feature tag:

- The ATCF inserts a Record-Route header field with its own SIP URI
- If the latest SRVCC information received for a session contains a C-MSISDN and ATU-STI,
  - The ATCF associates the session being established with the C-MSISDN and ATU-STI bound to the registration, and
  - The ATCF replaces the SDP offer in the originating SIP INVITE with updated SDP provided by the ATGW. Replacement occurs if the terminating SIP INVITE contains an SDP offer and a determination to anchor media has been made (according to the operator policy as specified in the 3GPP standard).



SIP INVITE from UE2 ATCF

The following is an example of the SIP INVITE UE2 sends to the ATCF/P-CSCF.

```
INVITE <sip:user1_public1@home1.net;gr=urn:uuid:f81d4fae-7dec-11d0-
a765-00a0c91e6bf6> SIP/2.0
Via: SIP/2.0/UDP sccas1.home1.net;branch=z9hG4bKnas34r5
Max-Forwards: 67
Route: <sip:scscf1.home1.net:lr>
P-Asserted-Identity: <tel: +1-237-555-2222>
P-Charging-Function-Addresses: ccf=[5555.1399:c88:d77:e66];
ccf=[5555::a55:1344:c33:d22];
    ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555.6aa:7bb:8cc:9dd]
P-Charging-Vector: icid-value="AyretyU0dm+602IrT5tAFrbHLso=023551024"; orig-
ioi="type3home1.net"
P-Access-Network-Info:
Privacy: none
From: <tel: +1-237-555-2222; gr=hdg7777ad7aflzig8sf7>;tag=171828
To: <tel:+1-237-555-1111>
Call-ID: cb03a0s09a2sdfg1kj490333
Cseq: 127 INVITE
Supported: 100re1, precondition
Require: sec-agree
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi=87654321; port=7531
Contact: <sip:user2_public1@home2.net;gr=urn:uuid:2ad8950e-48a5-4a74-8d99-
    ad76cc7fc74>;+g.3gpp.icsi-ref="urn%3Aurn-7%3gpp-service.ims.icsi.mmtel"
Allow: INVITE, ACK, CANCEL, BYE, PRACK, UPDATE, REFER, MESSAGE
Accept: application/sdp, application/3gpp-ims+xml
Content-Type: application/sdp
Content-Length: (...)

v=0
o=- 2987933615 2987933615 IN IP6 5555::aaa:bbb:ccc:ddd
s=
```

```
c=IN IP6 5555::aaa:bbb:ccc:ddd
t=0 0
m=audio 3456 RTP/AVP 97 96
b=AS:25.4
a=curr:qos local sendrecv
a=curr:qos remote none
a=des:qos mandatory local sendrecv
a=des:qos none remote sendrecv
a=rtpmap:97 AMR
a=fmtp:97 mode-set=0,2,5,7; mode-change-period=2
a=rtpmap:96 telephone-event
a=maxptime:20
m=message 0 TCP/MSRP 98
a=accept-types:text/plain
```

TS 24.237 Proposed Changes

The Oracle Communications Unified Session Manager implements a proposed change in the processing of failed or cancelled SRVCC sessions. The new processing model has been presented to the 3GPP for inclusion in TS 24.237, IP Multimedia (IM) Core Network (CN) subsystem IP Multimedia Subsystem (IMS) service continuity; Stage 3.

Sections 12.2.4.13, 12.3.3.1, and 12.3.3.1A of 3GPP TS 24.237, IP Multimedia (IM) Core Network (CN) subsystem IP Multimedia Subsystem (IMS) service continuity; Stage 3, describe procedures in response to SRVCC handover cancellation. These procedures allow the UE to generate a e-INVITE request with the Reason header field containing a value of “SIP”, a “cause” parameter set to a value of “487” (Request Terminated), and with reason-text parameter set to a value of either “handover cancelled” or “failure to transition to CS domain”.

SCC AS processing, which does nor release the original access leg, is defined in Section 12.3.3.1 and subclause 12.3.3.1A. Section 13.3.1 describes the handling of subsequent UPDATE requests. There are, however, no defined procedures at the ATCF response to this scenario with the result that the ATCF fails to reconfigure the ATGW to reconnect the bearer in LTE.

Oracle Corporation, in conjunction with other interested parties has proposed the addition of a new Section 12.7.2.3.3 to TS 24.237. This section requires the following ACTF behavior.

Requirement 1: When the ATCF receives either a

- SIP BYE request on the Source Access Leg containing a Reason header field containing a SIP 503 (Service Unavailable) response code, that is terminating an established dialog or an early dialog on the Source Access Leg
- SIP CANCEL request on the Source Access Leg with the Reason header field containing a SIP 503 (Service Unavailable) response code then, that is terminating an early dialog on the Source Access Leg originated by the SC UE, or
- SIP 503 (Service Unavailable) response on the Source Access Leg, that is terminating an early dialog on the Source Access Leg terminating at the SC UE

Then -- The ATCF shall retain session state information and ATGW resources associated with the session until either it receives a SIP INVITE request due to STN-SR, or a specified time period elapses (default value is 8 seconds).

The session remains recognizable for SRVCC access transfer as described in Section 12.7.2.1.

The SIP BYE request is forwarded to the SCC AS, which also delays release of the session, as described in Section 12.3.3.2.

Requirement 2: If the transferable session set determined in Section 12.7.2.1 does not contain any sessions and the identity in the P-Asserted-Identity header field is a C-MSISDN that is not bound to a registration path in the ATCF, the ATCF shall respond with a SIP 404 (Not Found) response.

Requirement 3: When the ATCF receives a SIP re-INVITE request containing Reason header field containing protocol “SIP” and reason parameter “cause” with value “487” on the original source access leg,

- after having initiated an access transfer that was triggered by a SIP INVITE request due to STN-SR, and
- the SIP INVITE request due to ATU-STI transaction is not yet completed

Then -- The ATCF shall wait until this transaction has completed and then continue with the steps described in Requirement 4.

Requirement 4: When the ATCF receives a SIP re-INVITE request(s) containing protocol “SIP” and reason parameter “cause” with value “487” after having performed an access transfer that was triggered by a SIP INVITE request due to STN-SR,

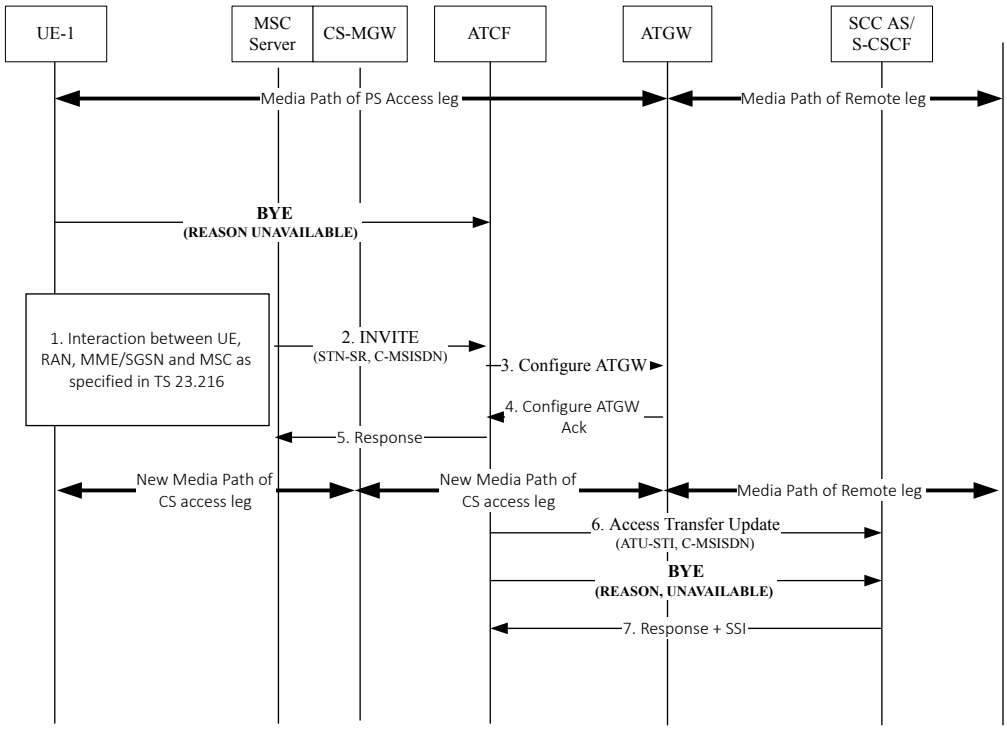
Then -- The ATCF shall act as B2BUA as described in Section 5.6 and shall.

1. Interact with ATGW to provide information needed in the procedures below and to request ATGW to start forwarding the media from the remote UE to the local UE. The details of interaction between ATCF and ATGW are out of scope of this document.
2. Send a SIP 200 (OK) response to the received SIP re-INVITE request. The SIP 200 (OK) response contains the SDP answer that includes the ATGW ports and the IP addresses as provided by the ATGW and the media used on the original source access leg as before the access transfer; and
3. Forward the received reINVITE with the Reason header intact to the SCC AS on the existing source dialog with the SDP offer containing the ATGW IP addresses and ports towards the remote UE as provided by the ATGW.

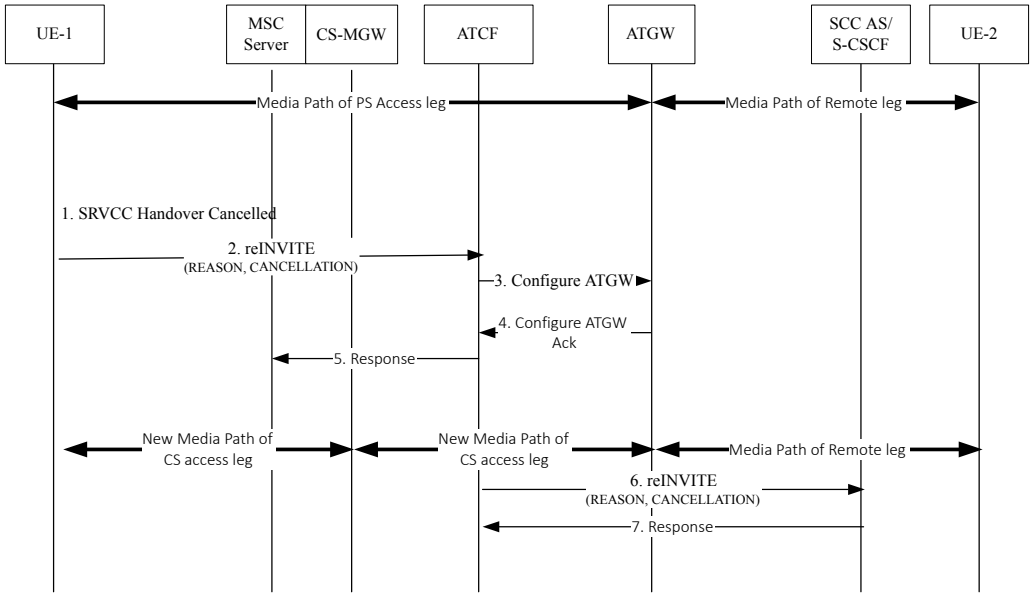
This proposed behavior, which requires no user configuration) is implemented in Version S-CX7.2.0 and later releases.

Related call flows are shown below.

BYE before handover INVITE



Cancellation after handover complete.



Accounting

eSRVCC calls involve two SIP sessions, the accounting records for which need clear association. To correlate the records, the Oracle Communications Unified Session Manager performs these actions for eSRVCC calls:

- When it detects a session handover, the system generates an Interim accounting record for the initial SIP session. This way, the records provide the exact time of the handover (i.e., the timestamp of the Interim record).
- The Oracle specific AVP Generic-ID will appear in Start, Interim, and Stop records for the handover SIP session. The Generic-ID contains the Call ID of the initial SIP session, which helps to correlate the two SIP sessions.
  - For RADIUS accounting records, refer to Oracle specific VSA 40.
  - For DIAMETER accounting records, refer to Oracle specific VSA 30.
- The Stop record for the initial SIP session will not have media flow information because the media session is considered part of the handover SIP session.
- If you are using QoS, the Stop records for the handover SIP session reflects the cumulative QoS statistics for both the initial SIP session and the handover SIP session.

External Bandwidth Management

Is you are using the Oracle Communications Unified Session Manager's external bandwidth management for eSRVCC calls, note these considerations:

- At the time of handover and if the new handover realm has the same external policy server configured as the initial SIP session, the Rx will continue seamlessly with the same DIAMETER session identifier directed toward the policy server. From the perspective of the policy server, the same session is simply continuing.
- At the time of handover and if the new handover realm does not have an external policy server configured, policy server services will stop.

ATCF Configuration

ATCF functionality requires configuration in the **sip-config** and **sip-interface** configuration elements.

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

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# sip-config
   ORACLE(sip-config)#
   ```

2. **atcf-stn-sr**—Enter the value of the Session Transfer Interface, Single Radio (STN-SR). Your entry will resemble this example: tel:+1-237-555-9999. This value will be included in the g.3gpp.atcf media feature tag that the ATCF allocates in the REGISTER message.
3. **atcf-psi-dn**—Enter the value to use for the Public Service Identity Domain Name (PSI-DN). Your entry will resemble this example: sip:atcf.visited2.com. If configured, this value will be included in the g.3gpp.atcf media feature tag that the ATCF allocates in the REGISTER message. If you leave this parameter blank, the Oracle Communications Unified Session Manager will set this value to the SIP interface accress.
4. **atcf-route-to-sccas**—If you leave this parameter set to disabled (default), the handover update, an INVITE, is routed to the IMS Core. If you set this parameter to enabled, the Oracle Communications Unified Session Manager will route the handover update directly to the Service Centralization and Continuity Application Server (SCC-AS).
5. Type **done** to save your configuration.

Continue to and select the existing sip-interface configuration element targeted for this ATCF configuration:

```
ORACLE(sip-config)# exit
ORACLE(session-router)# sip-interface
ORACLE(sip-interface)# select
<RealmID>:
1: public

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

6. **sip-atcf-feature**—Change this parameter from **disabled** (default) to **enabled** to turn on ATCF functionality for the ingress SIP interface.
7. Type **done** to save your configuration.

ATCF INVITE ICSI Matching

The Oracle Communications Unified Session Manager can check, on reception of an INVITE on an ingress sip-interface that has a configured ATCF and before applying any of the already implemented logic, whether the incoming INVITE includes the ICSI (Instantaneous Channel-State Information) of the requested service. The ATCF will be involved in the call flow when the configured ICSI value matches the ICSI value in the original INVITE; otherwise the handoff call will be rejected with code 480 (Temporarily Unavailable) or 404 (Not Found).

The system looks for the ICSI string in the following headers:

- P-Preferred-Service
- P-Asserted-Service
- Feature-Caps (within the "g.3gpp.icsi-ref" feature-capability indicator)
- Accept-Contact (within the tag-value within the g.3gpp.icsi-ref media feature tag)

An example of the ICSI string in the P-Preferred-Service or P-Asserted-Service header is "urn:urn-7:3gpp-service.ims.icsi.mmtel". Examples of the ICSI string in the Feature-Caps or Accept-Contact headers are "+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel" and "g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.mmtel". The ATCF will be involved in the call flow only when the ICSI matches. Configure the **atcf-icsi-match** parameter with the ICSI string you want to match. If **atcf-icsi-match** is blank, the check is not done and the behavior remains the same as before.

ATCF INVITE ICSI Matching Configuration

You can configure the Oracle Communications Unified Session Manager to check, on reception of an INVITE on an ingress sip-interface that has a configured ATCF and before applying any of the already implemented logic, whether the incoming INVITE includes the ICSI (Instantaneous Channel-State Information) of the requested service and, if so, to involve the ATCF in the call flow.

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

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

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

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

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

3. **atcf-icsi-match** — enter the ICSI string you want to match.
4. Type **done** to save your configuration.

SRVCC PS-CS Access Transfer

This section describes various scenarios for transferring sessions between Packet-Switched (PS) and Circuit-Switched (CS) networks, providing details about the following cases:

- Mobile Switching Center (MSC) server-assisted mid-call feature supported by the SCC AS
- Call flows for the MSC server-assisted mid-call feature supported by the SCC AS
- Failure cases
- Confirmed dialog
- Early dialog

For PS-CS transfers in SRVCC, the ATCF performs several actions based on the STN-SR when it first receives the SIP INVITE. First, the ATCF determines the set of transferrable sessions. It defines this set as the SRVCC-transferrable sessions that are associated with a C-MSISDN matching the URI in the P-Asserted-Identity header field in the INVITE. The ATCF compares the INVITE to its set of transferrable sessions to see if the INVITE is part of the set; if so, its responds with 2xx response for the initial INVITE.

Active session transfers require the ATCF to act as a back-to-back user agent (B2BUA) if the session undergoing transfer has media anchored at the ATGW. In this case, the ATCF responds with a 200 OK to the INVITE; the OK carries an SDP answer with ATGW IP address and port information. Then the ATCF updates the following information and sends the INVITE to the remote UE:

- Original SDP offer is replaced with an SDP offer containing media information currently in use with the ATGW IP address and port information.
- The Request-URI with the ATU-SI associated with the session being transferred is inserted.
- The Target-Dialog header field with the dialog identifier of the session being transferred is inserted.

When the ATCF receives an SDP answer and if media is anchored at the ATGW for the session being transferred, the ATCF directs the ATGW to start sending and receiving media to/from the remote UE according to the newly received answer.

If it receives a BYE with a 503 Service Unavailable header on the source access leg, the ATCF retains session state information and ATGW resources associated with the session until either:

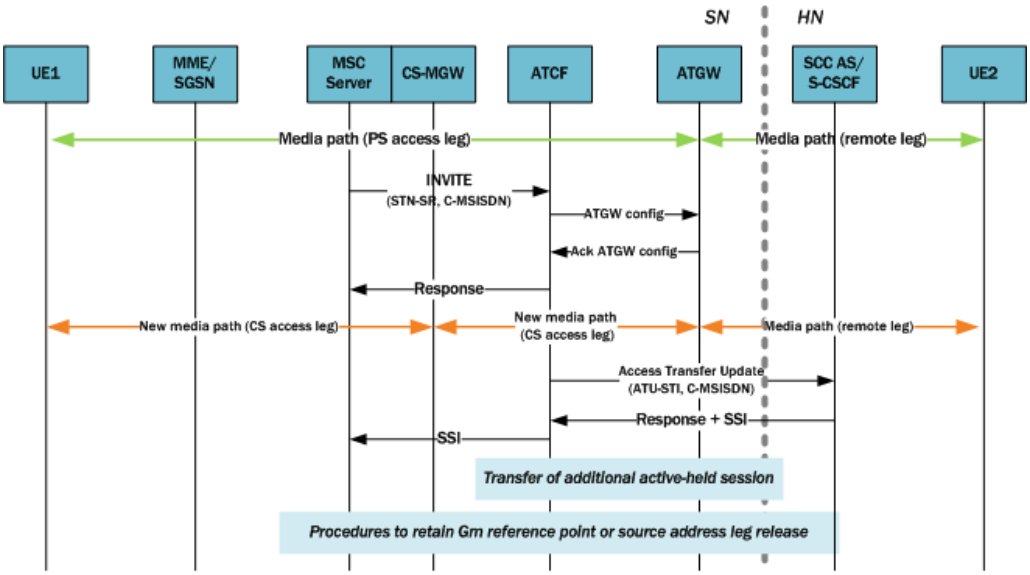
- The ATCF receives an INVITE request because of STN-SR, or
- An operator-defined time period elapses. The default value for the operator-defined time period is eight (8) seconds.

Thus, the session remains recognizable for SRVCC transfer for a time. Release of the session is also delayed by the ATCF's forwarding the BYE to the SCC AS. If the transferrable session cannot progress, the ATCF responds with a 404 Not Found. For transfer cases when only hold

or alerting sessions exist and the session in question is an early dialog or a confirmed dialog with inactive media, the ATCF replaces the Request-URI received in the original INVITE with the ATU-STI associated with a session in the transferrable set before forwarding the request; this proxy-role behavior complies with the 3GPP standard.

MSC Server-Assisted Mid-Call Feature Supported by SCC AS

The following scenario assumes an active session between UA1 and UA2, and that UA1 attaches to the CS domain. It is also assumed that the CS-MGW supports the codecs used for LTE voice calls, minimizing the likelihood that the ATCF will need to instruct the ATGW to insert codecs.



Upon receiving the Access Transfer message, the ATCF identifies the correct anchored session and proceeds with the transfer of the most recently active session. Using a Configure ATGW message, the ATCF replaces the existing PS access leg media path information with new CS access leg media path information for the ATGW. However, the ATCF might instruct the ATGW to continue using the local port of the PS access leg media path. Once the ATGW acknowledges the ATCF's Configure message, the ATCF sends the MSC server an Access Transfer response and the media path is moved to the CS when receiving SDP information.

Voice break interruption starts either when media moves to the CS MGW (controlled by the MSC server enhanced for SRVCC) or when the UE relocates to the target—whichever comes first. When the UE tunes to the target or media switches to the CS MGW (whichever is last), the voice break interruption ends. The assumption is that media is switched to the CS MGW during the time to UE tunes to the target.

After receiving the Access Transfer message, the ATCF re-establishes communication with the SCC AS and updates the SCC AS. When the MSC server supports the mid-call feature, the ATCF also communicates this fact to the SCC AS. The ATU message initiates a new dialog between the ATCF and SCC AS, a dialog the SCC AS associates with the old dialog using the C-MSISDN. This new dialog is necessary because it replaces the old dialog set up over PS access, thereby assuring that if the PS user registration expires the new home leg will not be released or even affected.

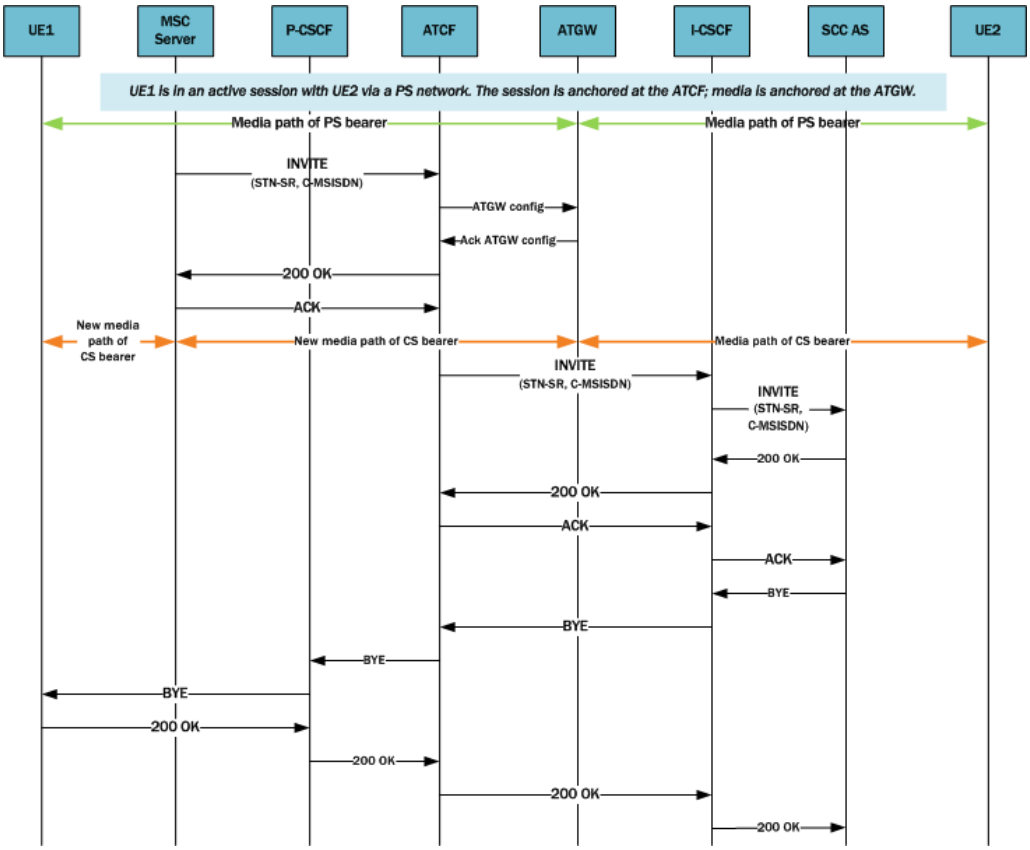
The following shows a sample INVITE request the ATCF sends to the S-CSCF. Note the following:

- The Request-URI header contains the ATU-STI, as routed to the SCC AS.
- The Target dialog header specifies the existing dialog is associated with this request.
- The P-Asserted-Identity header reflects the C-MSISDN of the UE being served.
- The From header reflects the C-MSISDN of the serving UE.
- The SDP reflects the media information at the ATGW.

```
INVITE sip:AUT-STI1@sccas.home1.net SIP/2.0
Via: SIP/2.0/UDP msc1.visit1.net;branch=z9hG4bk731b87
Max-Forwards: 70
P-Asserted-Identity: <tel:+1-237-555-2222>
P-Charging-Vector: icid-value="AyretyU0dm+602IrT5tAFrbHLso=023551024";orig-
ioi=visit1.net
Privacy: none
From: <tel:+1-237-555-9999>;tag=171828
To: <tel: +1-237-555-4444>
Call-ID: cb03a0s09a2sdfg1kj490334
Cseq: 127 INVITE
Supported: 100re1, precondition, gruu
Target-Dialog: me03a0s09a2sdfgjk1491777; to-tag=774321; from-tag=64727891
Accept-Contact: *;+g.3gpp.icsi-ref="urn%3Aurn-7%3gpp-service.ims.icsi.mmtel"
P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.mmtel
Contact: <sip:msc1.home1.net;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6>;
+g.3gpp.icsi-ref="urn%3Aurn-7%3gpp-service.ims.icsi.mmtel"
Allow: INVITE, ACK, CANCEL, BYE, PRACK, UPDATE, REFER
Content-Type: application/sdp
Content-Length: (...)

v=0
o=- 2987933615 2987933615 IN IP6 5555::aaa:bbb:ccc:ggg
s=
c=IN IP6 5555::aaa:bbb:ccc:ggg
t=0 0
m=audio 3456 RTP/AVP 97 96
a=tcap:1 RTP/AVPF
a=pcfg:1 t=1
b=AS:25.4
a=curr:qos local sendrecv
a=curr:qos remote none
a=des:qos mandatory local sendrecv
a=des:qos none remote sendrecv
a=rtpmap:97 AMR
a=fmtp:97 mode-set=0,2,5,7; mode-change-period=2
a=rtpmap:96 telephone-event
a=maxptime:20
```

The SCC AS sends the ATCF a confirmation, including the session state information (SSI) if the SCC AS and the MSC server support the mid-call feature. The MSC server receives the SSI from the ATCF; the access leg for the control has moved to the CS access. When MSC server receives SSI for more multiple active or inactive speech sessions, it initiates transfer directed at the SSC AS for the additional sessions.

The following diagram shows a call session in the active phase for MSC server assisted midcall feature when supported by the SCC AS.



Failure and Cancellation

For cases of failure and/or cancellation, the ATCF behaves in accordance with TS 23.216 [3], Section 8:

- In the case of session failure before the MSC server initiates session transfer, the standard failover procedures (as defined in TS 23.401 [2])apply and no further action is required by the UE.
- In the case of session failure after the UE receives the handover command, the UE attempts to return to the Universal Mobile Telecommunications System (UTRAN) or evolved Universal Mobile Telecommunications System (eUTRAN). The UE initiates signaling to transfer the session back, which the ATCF handles if the ATCF recognizes the session transfer.
- In the case when handover is cancelled, the UE—having received the handover cancellation message—begins the re-establishment procedure as though it needed to transfer the session to the eUTRAN/UTRAN. If he ATCF identifies this as a transfer session, it handles the session transfer back to the eUTRAN/UTRAN.

Confirmed Dialogs

When an SC UE is engaged in one or more ongoing IMS sessions, it will send a re-INVITE containing:

- An SDP offer, including the media characteristics used in the existing dialog, and

- A Reason header field containing the protocol SIP and reason parameters cause with the value 487 (as specified in RFC 3326 [57]), with reason text reading either Handover cancelled or Failure to transition to CS domain.

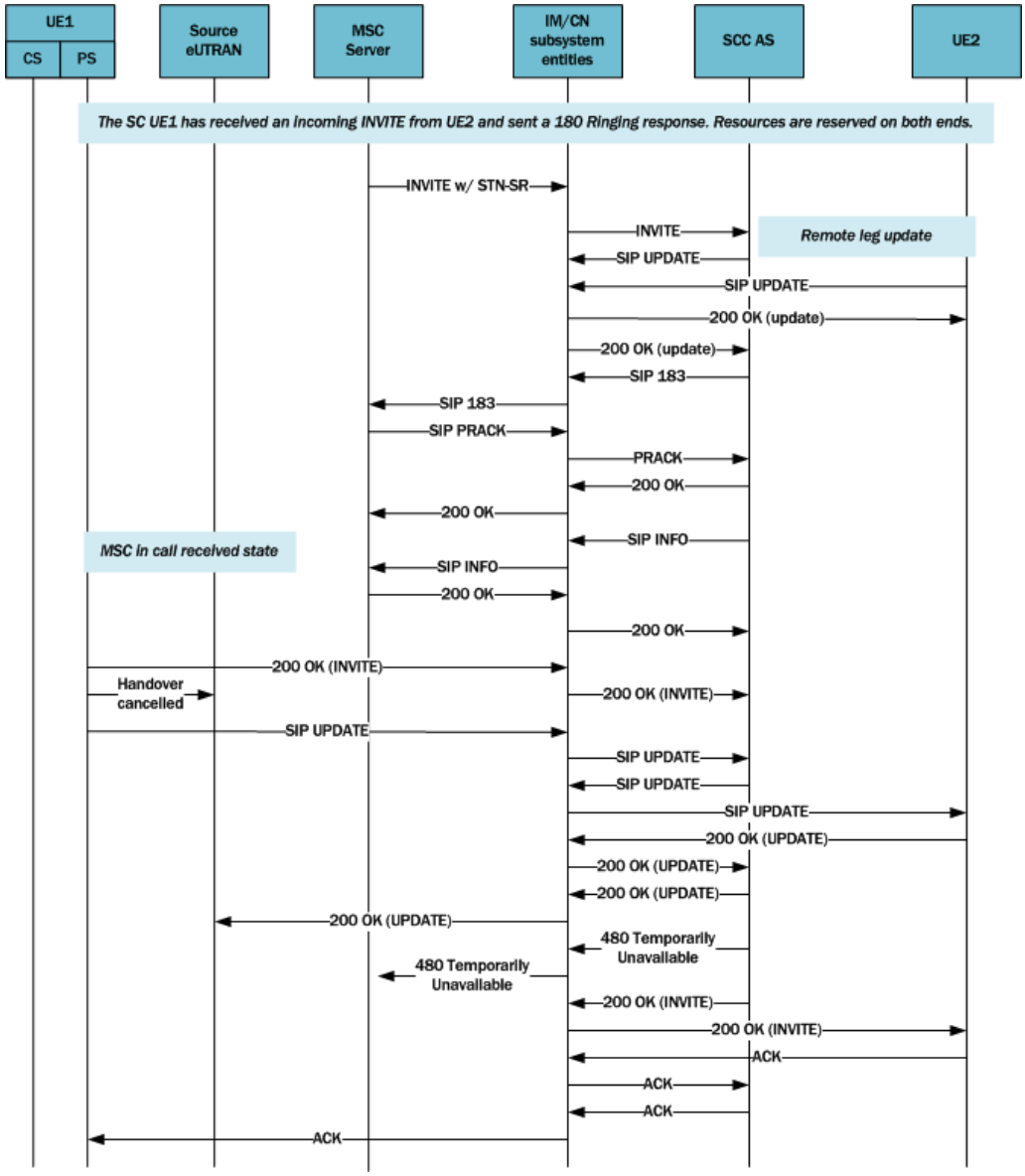
In all other ways, confirmed dialogs behave in accordance with TS 24.229 [2].

Early Dialogs

If the SC UE engages in a session which is in early dialog state, it will send a SIP UPDATE containing:

- An SDP offer, including the media characteristics used in the existing dialog, and
- A Reason header field containing the protocol SIP and reason parameters cause with the value 487 (as specified in RFC 3326 [57]), with reason text reading either Handover cancelled or Failure to transition to CS domain.

In all other ways, early dialogs behave in accordance with TS 24.229 [2].



SRVCC Handover Support in the Pre-Alerting Phase

In addition to other SRVCC support, the Oracle Communications Unified Session Manager (OCUSM) supports procedures to manage the handover from 4G to 3G/2G of sessions in pre-alerting phase. The conditions by which a session is defined as in the pre-alerting phase include the calling party has not yet received a 180 RINGING message.

Refer to TS 24.237 to review the anchoring endpoints' behavior.

To ensure that calls in a pre-alerting phase are transferred between PS and CS networks, set the **sip-feature-caps** value as follows:

- Set **state** to **enabled**
- Set **atcf-management-uri** to **management** or **psi**
- Set **atcf-pre-alerting** to **enabled**

For information on the results of these settings, refer to the *SIP Feature Capabilities* section.

The OCUSM, as Proxy Call Session Control Function (P-CSCF) or Active Transfer Control Function (ATCF), selects any of the early dialogs as the session being transferred when:

- There are no confirmed dialogs supporting a session with an inactive speech media component ("sendonly" or "inactive" directionality) in the transferable session set, and
- There are one or more dialogs in the transferable session set supporting one session where the SC UE has completed a reliable offer / answer procedure and has an active speech media component ("recvonly" or "sendrecv" directionality).

Applicable dialogs include those for which:

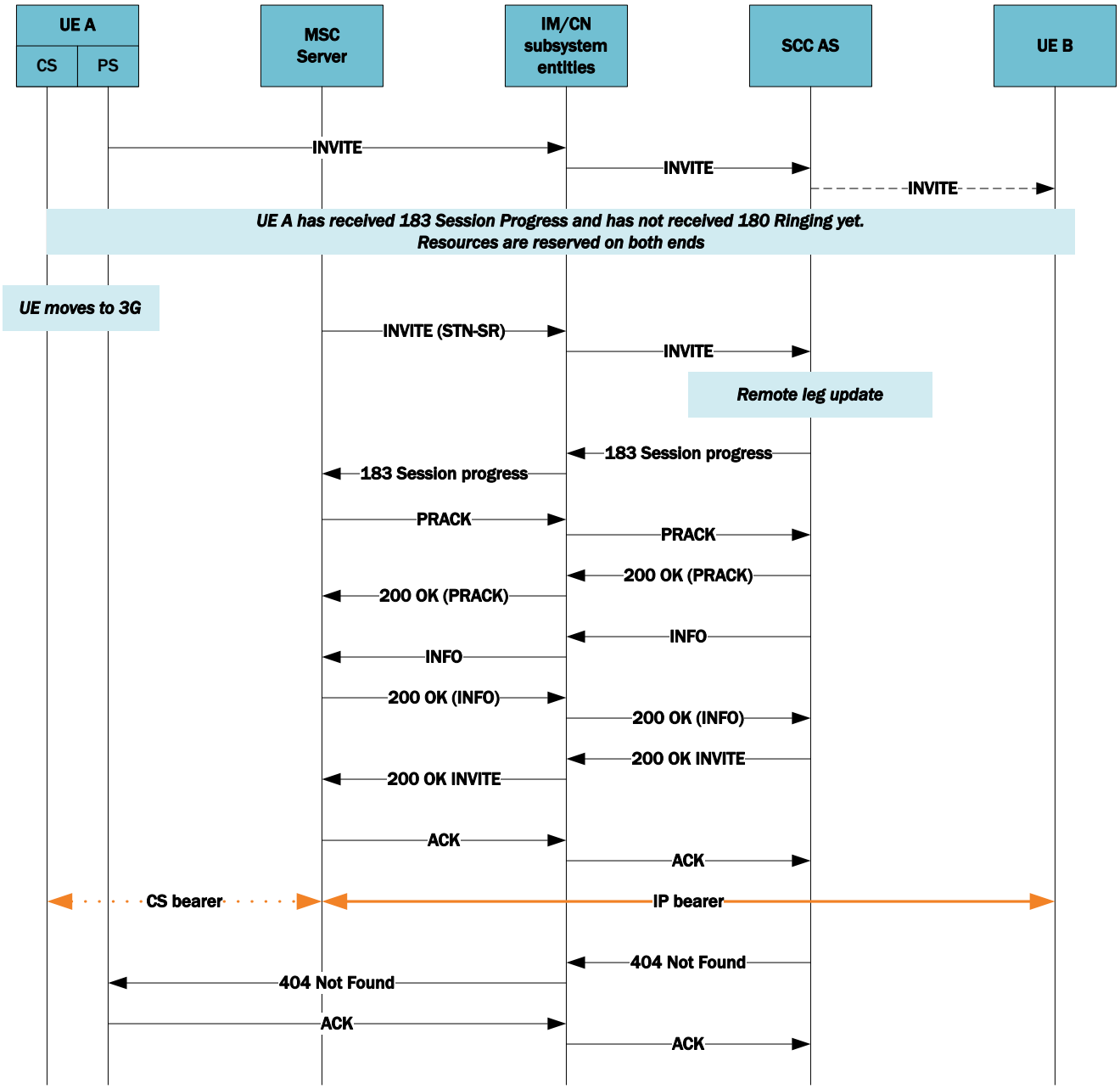
- A SIP 180 (Ringing) response to the SIP INVITE request has not been received yet in any of the existing dialogs.
- All dialogs are early dialogs created by the same SIP INVITE request.
- The Contact header field provided by the SC UE includes the **g.3gpp.ps2cs-srvcc-orig-pre-alerting** media feature tag; and
- The Feature-Caps header field provided by the SCC AS towards the SC UE includes the **g.3gpp.ps2cs-srvcc-orig-pre-alerting feature-capability** indicator.

The range of OCUSM SRVCC pre-alerting phase support includes:

- SRVCC support for both outgoing (originating) and incoming (terminating) calls
- Responds to media feature tag and fcaps indicator
- SRVCC support of EATF emergency calls
- Handover (HO) cancelling scenarios
- Rx interactions
- SRVCC Error scenarios, including:
  - HO cancellation failures
  - Rx failures
  - Standard SIP transaction failures
- ACLI, SNMP and HDR statistics for applicable, successful and failed SRVCC handovers

The diagram below shows a flow for an outgoing call that includes an SRVCC handover during the pre-alerting phase. Prior to the alerting phase, the OCUSM, acting as ATCF (IM/CN subsystem) receives and forwards the new INVITE to update the call to the circuit switched (CS) leg. Ultimately, the signaling between the MSC, the ATCF and the SOC AS succeeds. The components proceed through the alerting phase, eliminating the packet switched based (PS) leg and using the MSC server to manage the CS call.

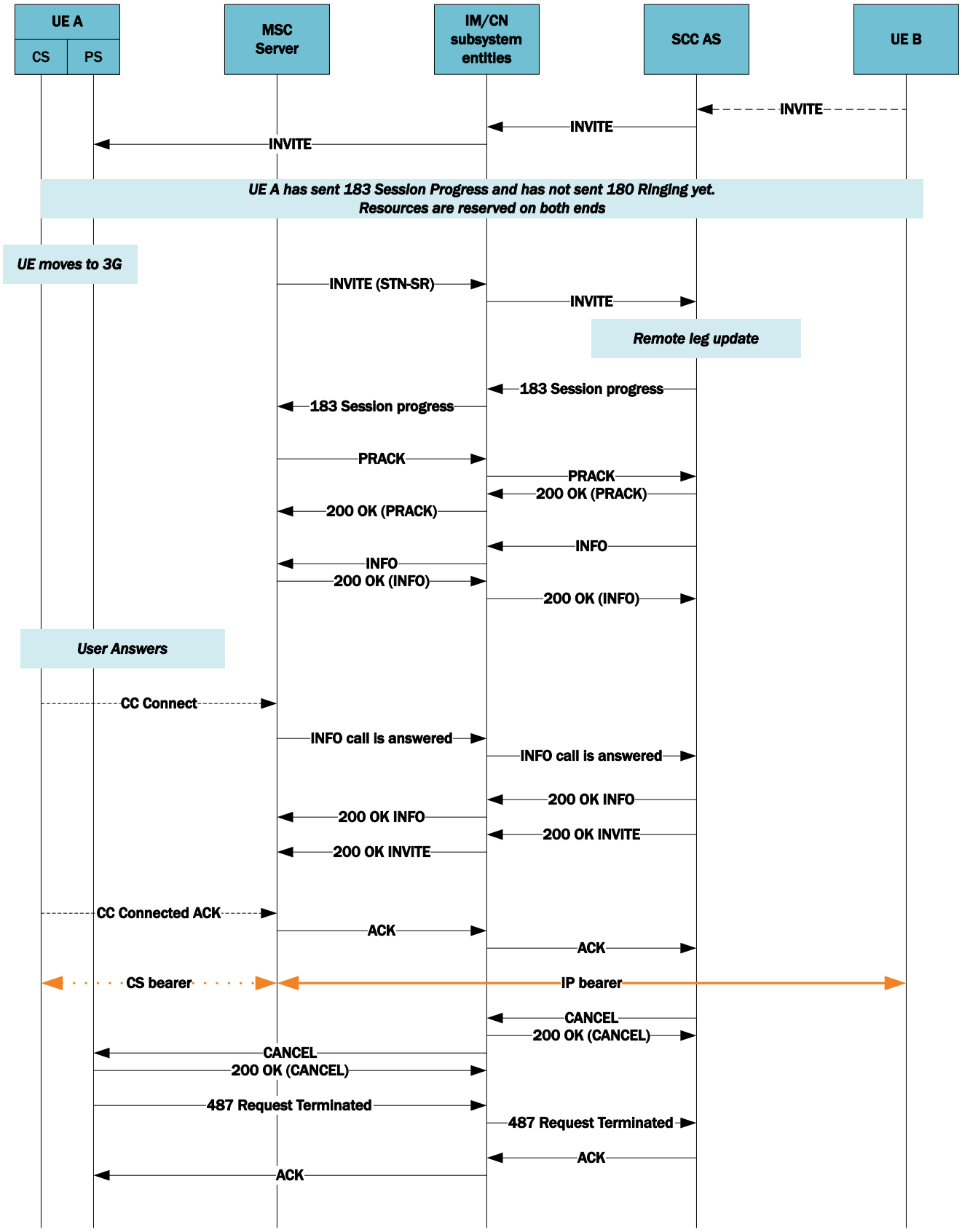
Figure 4-2 Mobile Originating Call in Pre-Alerting Phase



The diagram below shows a flow for an incoming call that includes an SRVCC handover during the pre-alerting phase. Prior to the alerting phase, the OCUSM, acting as ATCF (IM/CN subsystem) receives and forwards the new INVITE to update the call to the circuit switched (CS) leg. Ultimately, the signaling between the MSC, the ATCF and the SOC AS succeeds. The

components proceed through the alerting phase, eliminating the packet switched based (PS) leg and using the MSC server to manage the CS call.

Figure 4-3 Mobile Terminating Call in Pre-Alerting Phase



SRVCC Handover Support in Alerting Phase

The OCUSM supports handovers between Packet-Switched (PS) and Circuit-Switched(CS) networks for calls in an alerting phase; that is, a 180 ringing response for the initial INVITE has been sent or received and the SIP final response has not been sent or received.

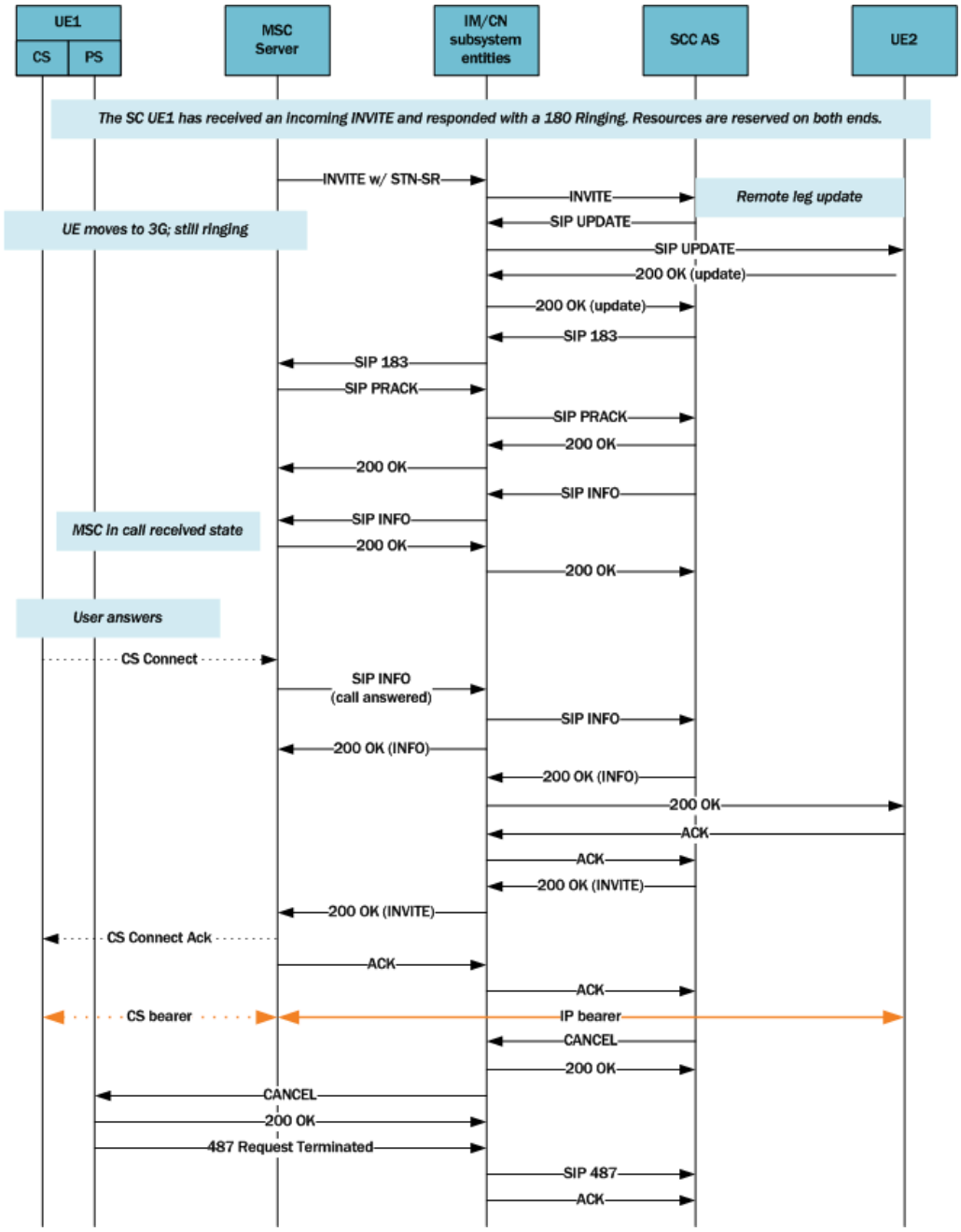
The behavior of the two anchoring points, ATCF and ATGW, is defined by 3GPP in Release 12 of Technical Specification TS 24.237. Oracle Communications developed these functional entities based on the initial version of TS 24.237 Release 10, and has added the **sip-feature-caps** configuration element to align with Release 12.

To ensure that calls in an alerting phase are transferred between PS and CS networks, set the **sip-feature-caps** value as follows:

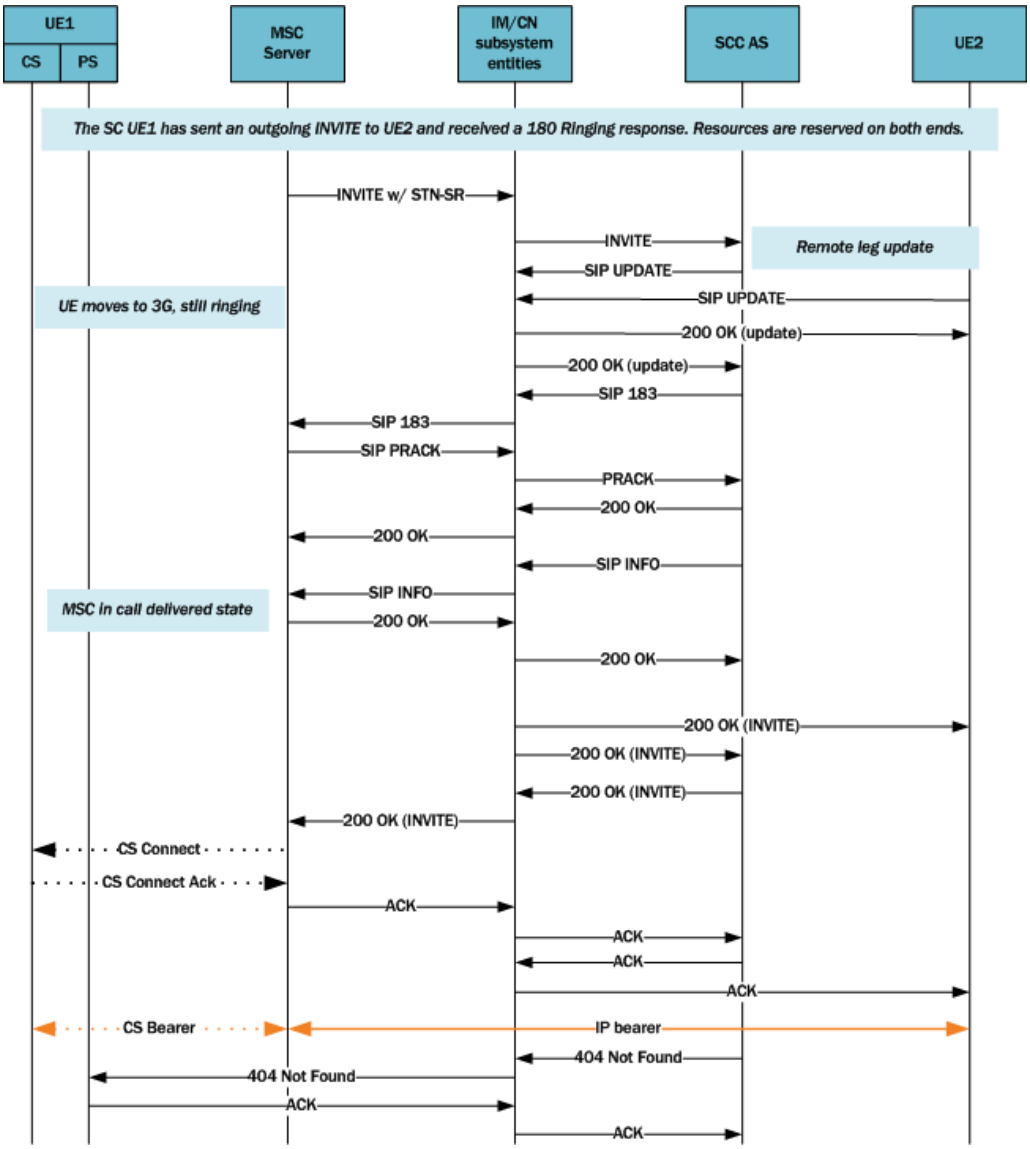
- Set **state** to “enabled”
- Set **atcf-management-uri** to "management" or "psi"
- Set **atcf-alerting** to "enabled"

For information on the results of these settings, refer to the *SIP Feature Capabilities* section.

The diagram below shows an incoming call session during the alerting phase.



This diagram shows an outgoing call session during the alerting phase.



SIP Feature Capabilities

The ATCF can announce a feature capability in a message by transporting the information in the Feature-Caps header, which is supported by the OCUSM.

The behavior of the two anchoring points, ATCF and ATGW, is defined by 3GPP in Release 12 of Technical Specification TS 24.237. Oracle Communications developed these functional entities based on the initial version of TS 24.237 Release 10, and has added the **sip-feature-caps** configuration element to align with Release 12.

When the **state** is enabled, the OCUSM includes the Feature-Caps header in applicable SIP messages. The information in the header is dependent on **sip-feature-caps** configuration.

Specifically, When you enable **sip-feature-caps**, regardless of other settings, the OCUSM inserts the Feature-Caps header with:

- The **g.3gpp.mid-call** capability indicator.

- The **g.3gpp.atcf-path** parameter with a value set to the ATCF URI for terminating requests

Assume the **state** is **enabled** and both **atcf-alerting pre-atcf-alerting** are **disabled**. The OCUSM adds the Feature-Caps header with the above and:

- The **g.3gpp.atcf** parameter with a value of the configured **atcf-stn-sr** in **sip-config**
- The **g.3gpp.atcf-mgmt-uri** parameter with a value of the configured **atcf-psi-dn** in **sip-config**

Finally, when **state** is **enabled**, **atcf-management-uri** is not disabled, and **atcf-alerting** is **enabled**, the OCUSM adds the Feature-Caps header with the above and also adds the **g.3gpp.srvcc-alerting** capability indicator. Similarly, the OCUSM adds the **g.3gpp.srvcc-pre-alerting** capability indicator when **atcf-pre-alerting** is **enabled**.

SIP Feature Capabilities Configuration

You can configure Oracle Communications Unified Session Managers to have the ATCF announce a feature capability in a message by transporting the information in the Feature-Caps header.

1. Access the **session-router** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# sip-feature-caps
ORACLE(sip-feature-caps)#
```

2. **state** — identifies whether to enable the feature and add the Feature-Caps header to messages. Possible values are "enabled" and "disabled". The default value is "disabled".
3. **atcf-alerting** — identifies whether to turn on the alerting feature and add the alerting Feature-Caps header to messages. Possible values are "enabled" and "disabled". The default value is "disabled".
4. **atcf-pre-alerting** — identifies whether to turn on the alerting feature and add the pre-alerting Feature-Caps header to messages. Possible values are "enabled" and "disabled". The default value is "disabled".
5. **atcf-management-uri** — identifies the feature capability indicator that will be used to transport the ATCF management URI. Possible values are "management" and "psi". The default value is "management". When the value is "management" and the value of **state** is "enabled", the Feature-Caps header “g.3gpp.atcf-mgmt-uri” is added and the value is the value of **atcf-psi-dn** in the **sip-config** configuration element. When the value is "psi" and the value of **state** is "enabled", the Feature-Caps header “g.3gpp.atcf-psi” is added and the value is the value of **atcf-psi-dn** in the **sip-config** configuration element.
6. Type **done** to save your configuration.

Reporting SRVCC Statistics

You access SRVCC Hand-Over (HO) call counters and statistics from the OCUSM's via ACLI Show command, SNMP and HDR tools.

Applicable statistic access details include:

- ACLI—Run the command **show sipd srvcc** to display total, successful and failed calls for each handover phase. Find additional detail on **show sipd** from the *Maintenance and Troubleshooting Guide*.

- SNMP—Access the same detail provided by the ACLI command above via SNMP. Get this detail from **ap-sip.mib** mmm. Find additional detail on **ap-sip.mib** from the *MIB Guide*. Example OID and object identifiers include:
  - 1.3.6.1.4.1.9148.3.15.1.1.3.13 — (apSipSRVCCStatsTotalCallsDuringPreAlerting)
  - 1.3.6.1.4.1.9148.3.15.1.1.3.14 — (apSipSRVCCStatsDuringPreAlertingSuccess)
  - 1.3.6.1.4.1.9148.3.15.1.1.3.15 — (apSipSRVCCStatsDuringPreAlertingFailed)
- HDR—Access the same detail provided by the ACLI command above via HDR. Get this detail from the **sip-srvcc** collect group. Find additional detail on the **sip-srvcc** group from the *HDR Reference Guide*. Example HDR objects include:
  - Calls During Pre-Alerting Counter — (0-2^32-1) — Total calls subjected to SRVCC during pre-alerting.
  - During Pre-Alerting Success Counter — (0-2^32-1) — Total successful SRVCC HO during pre-alerting.
  - During Pre-Alerting Failed Counter — (0-2^32-1) — Total failed SRVCC HO during pre-alerting.

Emergency Access Transfer Function

The Emergency Access Transfer Function (EATF) is a logical, functional service defined in 3GPP TS 23.167, *IP Multimedia Subsystem (IMS) Emergency Sessions*, and TS 23.237, *IP Multimedia Subsystem (IMS) Service Continuity*; Stage 2. The EATF, essentially a special-purpose B2BUA, anchors emergency calls to enable access transfer between packet-switched and circuit-switched networks during eSR-VCC procedures when the LTE equipment is moving outside LTE coverage to either a 2G or 3G carrier network. Similar to the Access Transfer Control Function (ATCF) and ATGW (Access Transfer Gateway), the EATF is always located in the visited network when the user equipment is roaming.

Lacking this capability, the LTE equipment would be forced to re-establish the emergency session in the circuit-switched network through the legacy accesses (2G or 3G).

When the LTE equipment initiates a packet-switched emergency session, the INVITE is sent to the EATF thru the P-CSCF/E-CSCF (collocated on the SBC). This original INVITE is identified as an emergency session by the A-SBC/P-CSCF because it contains either an emergency short number (112, 991, and so forth) or an emergency service URN such as urn:service:sos.fire.

In the event that handoff to a circuit-switched network is required, the Mobile Switching Center (MSC) server initiates the transfer with a SIP INVITE containing an E-STN-SR (Emergency Session Transfer Number for Single Radio VCC) in the Request URI of the INVITE. Each network has a single E-STN-SR, essentially the telephone number of the EATF service, that is used exclusively for emergency session transfer access. The MSC directs the INVITE to the I-CSCF, which, in turn, which forwards the request directly to the EATF.

The EATF checks the E-STN-SR to determine that handoff to the circuit-switched network is requested and proceeds with the access transfer of the active session. The EATF associates the received SIP INVITE with an existing SIP session already anchored at the EATF using the instance-id feature tag. The EATF then sends a re-INVITE to the E-CSCF, which terminates the emergency session.

Once the session modification procedures are complete, as indicated by the reception of the SIP ACK request from the target access leg, the source access leg, previously established via IMS, is released.

Enabling EATF Capability

Use the following procedure to enable EATF operations.

1. Use the following procedure to move to sip-config configuration mode.

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

2. Use the **eatf-stn-sr** parameter to enable EATF operation and to provide E.164 telephone number of the EATF service.

   Provide the E.164 telephone number as either a tel: or a sip: URI. When confirming the E-STN-SR value, the match succeeds based solely on the E.164 portion regardless of the URI type.

   ```
   ACMEPACKET(sip-config)# eatf-stn-sr tel:+1-237-555-9999
   ACMEPACKET(sip-config)#
   ```

3. Use **done** and **exit** to complete configuration.

Monitoring SRVCC Sessions

The **show sipd srvcc** command displays SRVCC handover counts including ATCF and EATF sessions.

```
ORACLE# show sipd srvcc
08:22:13-188
SRVCC Handover Stats                          ---- Lifetime ----
                                    Recent      Total  PerMax
Total Calls                              0          0       0
  Success                                0          0       0
  Failed                                 0          0       0
Calls After Answer                       0          0       0
  Success                                0          0       0
  Failed                                 0          0       0
Calls During Alerting                    0          0       0
  Success                                0          0       0
  Failed                                 0          0       0
ATCF Cancellation                        0          0       0
Emergency Calls                          0          0       0
  Success                                0          0       0
  Failed                                 0          0       0
EATF Cancellation                        0          0       0
```

These counters are defined as follows:

- Total Calls - Total calls subjected to SRVCC
- Total Success - Total successful SRVCC hand-off
- Total Failed - Total failed SRVCC hand-off
- Calls After Answer - Total calls subjected to SRVCC in established phase
- After Answer Success - Total successful SRVCC hand-off in established phase
- After Answer Failed - Total failed SRVCC hand-off in established phase
- Calls During Alerting - Total calls subjected to SRVCC in alerting phase

- During Alerting Success - Total successful SRVCC hand-off in alerting phase
- During Alerting Failed - Total failed SRVCC hand-off in alerting phase
- ATCF Cancellation - Total ATCF cancellations
- Total Emergency Calls - Total SRVCC hand-off for Emergency calls
- Emergency Success - Total successful SRVCC hand-off for Emergency calls
- Emergency Failed - Total failed SRVCC hand-off for Emergency calls
- EATF Cancellation - Total EATF Cancellations

5

ENUM Based Oracle Communications Unified Session Manager

The Oracle Communications Unified Session Manager contacts an ENUM server via DNS for two purposes:

- to obtain authentication data for a registering UA (often referred to as a UE)
- to query the user subscriber database (ENUM) regarding the registration state of the AoR and update the database with the latest information

Message Authentication for SIP Requests

The Oracle Communications Unified Session Manager authenticates requests by configuring the sip authentication profile configuration element. The name of this configuration element is either configured as a parameter in the sip registrar configuration element's authentication profile parameter or in the sip interface configuration element's sip-authentication-profile parameter. This means that the Oracle Communications Unified Session Manager can perform SIP digest authentication either globally, per domain of the Request URI or as received on a SIP interface.

After naming a sip authentication profile, the received methods that trigger digest authentication are configured in the methods parameter. You can also define which anonymous endpoints are subject to authentication based on the request method they send to the Oracle Communications Unified Session Manager by configuring in the anonymous-methods parameter. Consider the following three scenarios:

1. By configuring the methods parameter with REGISTER and leaving the anonymous-methods parameter blank, the Oracle Communications Unified Session Manager authenticates only REGISTER request messages, all other requests are unauthenticated.
2. By configuring the methods parameter with REGISTER and INVITE, and leaving the anonymous-methods parameter blank, the Oracle Communications Unified Session Manager authenticates all REGISTER and INVITE request messages from both registered and anonymous endpoints, all other requests are unauthenticated.
3. By configuring the methods parameter with REGISTER and configuring the anonymous-methods parameter with INVITE, the Oracle Communications Unified Session Manager authenticates REGISTER request messages from all endpoints, while INVITES are only authenticated from anonymous endpoints.

Credential Retrieval

The Oracle Communications Unified Session Manager requests authentication information from an ENUM server via DNS when it receives a REGISTER or other message from an endpoint. This server, which provides authentication information, is defined on the Oracle Communications Unified Session Manager in an enum-config configuration element that includes an enum-servers (the IP addresses of the servers) and realm parameters. Together, these two parameters define the DNS/ENUM server(s) which provide authentication data.

The target ENUM server is determined first by setting the credential retrieval config parameter to **enum-config** so the Oracle Communications Unified Session Manager will reference that enum config. Next, set the credential retrieval config parameter to the name of an enum config configuration element which is populated with the ENUM servers' IP addresses.

User Authentication Query

As soon as a request is received on a SIP interface and has been determined to require authentication, the Oracle Communications Unified Session Manager attempts to authenticate the endpoint. It sends a DNS TXT query including the UA's AoR to an ENUM database and expects the H(A1) defined in RFC2617 for the user being authenticated.

SIP Digest User Authentication

SIP Authentication Challenge

When the Oracle Communications Unified Session Manager receives a response from the ENUM server including the hash value for the user, it sends a SIP authentication challenge to the endpoint, if the endpoint did not provide any authentication headers in its initial contact the with Oracle Communications Unified Session Manager. If the endpoint is registering, the Oracle Communications Unified Session Manager replies with a 401 Unauthorized message with the following WWW-Authenticate header:

```
WWW-Authenticate: Digest realm="atlanta.com", domain="sip:boxesbybob.com",
qop="auth", nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="", stale=FALSE,
algorithm=MD5
```

If the endpoint initiates any other request to the Oracle Communications Unified Session Manager besides REGISTER, the Oracle Communications Unified Session Manager replies with a 407 Proxy Authentication Required message with the following Proxy-Authenticate header:

```
Proxy-Authenticate: Digest realm="atlanta.com", qop="auth",
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="", stale=FALSE, algorithm=MD5
```

Authentication Header Elements

- Digest Realm—This value is configured in the digest-realm parameter in the sip-registrar configuration element. This parameter is mandatory when using the "ENUM-TXT" credential retrieval method.
- Domain—A quoted, space-separated list of URIs that defines the protection space. This is an optional parameter for the "WWW-Authenticate" header.
- Nonce—A unique string generated each time a 401/407 response is sent.
- Qop—A mandatory parameter that is populated with a value of "auth" indicating authentication.
- Opaque—A string of data, specified by the Oracle Communications Unified Session Manager which should be returned by the client unchanged in the Authorization header of subsequent requests with URIs in the same protection space.
- Stale—A flag indicating that the previous request from the client was rejected because the nonce value was stale. This is set to true by the SD when it receives an invalid nonce but a valid digest for that nonce.

- Algorithm—The Oracle Communications Unified Session Manager always sends a value of "MD5"

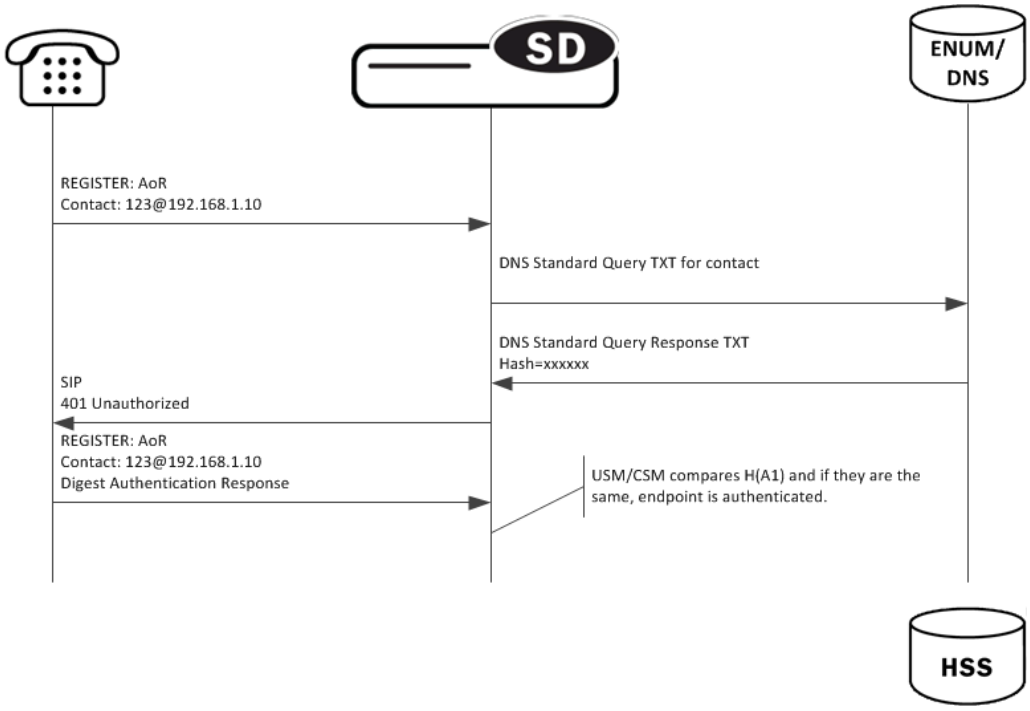
SIP Authentication Response

After receiving the 401/407 message from the Oracle Communications Unified Session Manager, the UA resubmits its original request with an Authorization: header including its own internally generated MD5 hash.

Oracle USM Authentication Check

At this point, the Oracle Communications Unified Session Manager has received an MD5 hash from the ENUM server and an MD5 hash from the UA. The Oracle Communications Unified Session Manager compares the two values and if they are identical, the endpoint is successfully authenticated. Failure to match the two hash values results in a 403 or 503 sent to the authenticating endpoint.

The following image shows the user authentication process.



Oracle USM as Registrar

DDNS Update to User Subscriber Database

As REGISTER messages are received, the Oracle Communications Unified Session Manager updates the ENUM database via DDNS UPDATE messages as defined by RFC2136. New registrations are added to the database while expired or deleted registrations are removed.

The Oracle Communications Unified Session Manager acts as a registrar by configuring the sip registrar configuration element. When registrar functionality is enabled, the Oracle

Communications Unified Session Manager acts as a registrar rather than only caching and forwarding registrations to another device. Oracle Communications Unified Session Manager registry services are enabled globally per domain, not on individual SIP interfaces or other remote logical entities.

On receiving a REGISTER message, the Oracle Communications Unified Session Manager checks if it is responsible for the domain contained in the Request-URI as defined by the domains parameter and finds the corresponding sip registrar configuration. This is a global parameter and all messages are checked against all sip registrar domains. Thus you could create one sip registrar configuration element to handle all *.com domains and one sip registrar configuration element to handle all *.org domains. The Oracle Communications Unified Session Manager begins registrar functions for all requests that match the configured domain per sip-registrar configuration element.

A UA is considered registered after the Oracle Communications Unified Session Manager updates the ENUM server with a DDNS dynamic update. After this action, the Oracle Communications Unified Session Manager sends a 200 OK message back to the registering UA.

TTL

Part of the Oracle Communications Unified Session Manager architecture includes a local ENUM cache which maintains the results from ENUM queries locally. To enable Oracle Communications Unified Session Manager, the TTL value in the enum config must be set to 0. This ensures that whenever an ENUM query is required, the Oracle Communications Unified Session Manager consults the central User Subscriber Database to have the latest network-wide information about the UA it is trying to reach, instead of its ENUM cache.

ENUM Database Correlation

When a UA registers, as the number of associated contacts for an AoR grows or shrinks, the ENUM-based User Subscriber Database is updated in turn with the latest information using a DDNS UPDATE. After a REGISTER including authentication information is received from a UA, the Oracle Communications Unified Session Manager sends a Standard NAPTR query for the AoR to the ENUM server. The ENUM server replies with a Standard Query response, including the NAPTR records.

After receiving the entries from the ENUM database via the ENUM query, the list must be correlated with the Oracle Communications Unified Session Manager's view of the AoR's registration state. The differences will be resolved by sending a DDNS UPDATE to add or remove entries from the ENUM server. The database correlation phase only occurs when endpoints register.

Contacts that need to be added are put on the UPDATE add list. Contacts that are being unregistered (contact with Expires=0) or have expired timestamp are added to the UPDATE remove list.

Entry Expiration

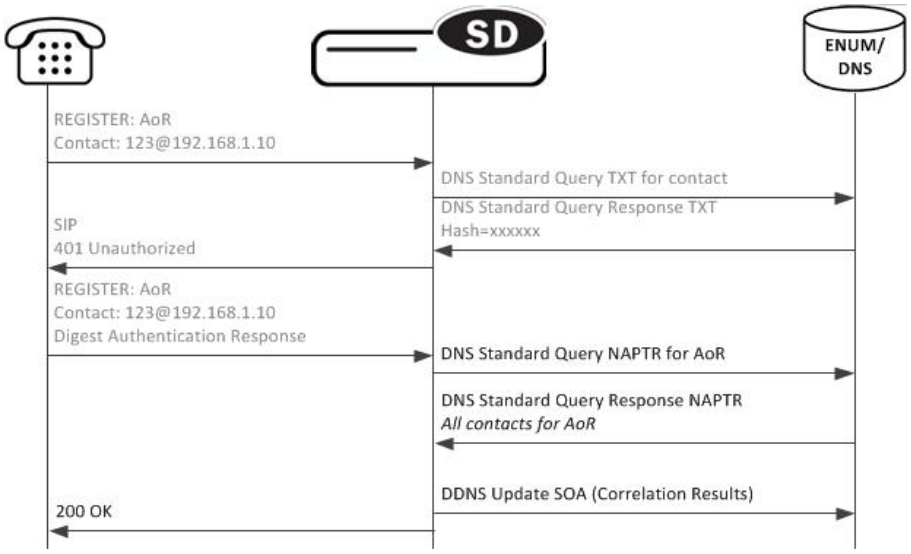
The Oracle Communications Unified Session Manager employs a process to remove expired contacts from the ENUM database whether they were entered by the active Oracle Communications Unified Session Manager or other Oracle Communications Unified Session Manager. After determining that a contact is expired, the Oracle Communications Unified Session Manager removes the contact in a subsequent DDNS update.

To do this, the Oracle Communications Unified Session Manager includes an expiration parameter in the Contacts it insert into the ENUM database. Expiration is indicated with a ts= parameter. This parameter's value is set to the initial registration time measured on the Oracle Communications Unified Session Manager plus the REGISTER message's Expires: header value or Expires parameter in the Contact header value. This value is measured in seconds after the epoch. In a DDNS update, a ts= parameter appears as follows:

```
Regex: "!^.*$!sip:234-hchse6c0d01u2@172.16.101.51:5060;ts=1313493824!"
```

After the Oracle Communications Unified Session Manager retrieves contacts for an AoR in a NAPTR record that are expired, based on ts= parameters, the Oracle Communications Unified Session Manager's next Dynamic update tells the server to remove those contacts from the ENUM database.

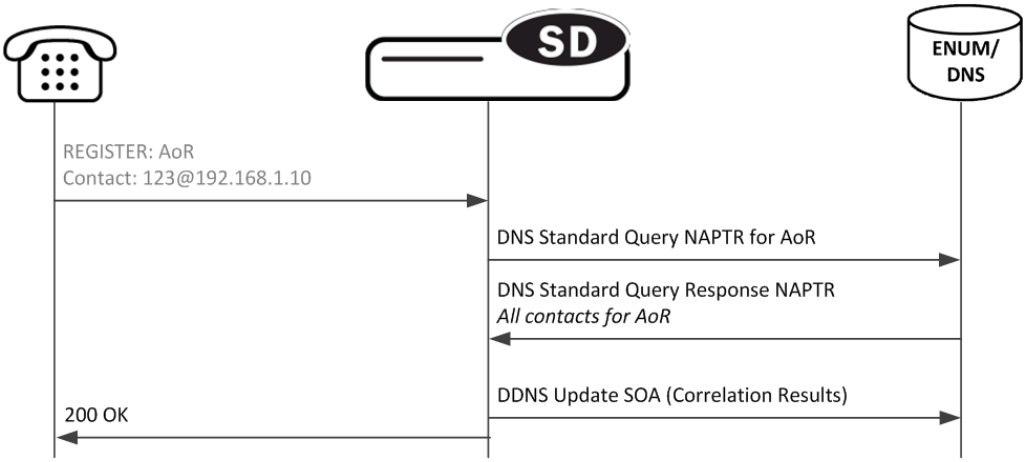
In this way, all Oracle Communications Unified Session Manager members of a domain may remove expired contacts from the ENUM database.



Register Refresh

When a UA sends a register refresh, the Oracle Communications Unified Session Manager first confirms that the authentication exists for that UA's registration cache entry, and then is valid for the REGISTER refresh. Then, the lifetime timer (value in Expires: header) for that registration cache entry is checked.

If the timer has not exceeded half of its lifetime, only a 200 OK is sent back to the UA. If the timer has exceeded half of its lifetime, the Oracle USM sends a NAPTR update to the ENUM database.



In addition to the baseline Oracle Communications Unified Session Manager REGISTER refresh conditions, an ENUM database update is required when one of the following conditions is satisfied:

- The location update interval timer has expired—This value, configured in the sip registrar configuration element ensures that that ENUM database always has the latest user information by periodically sending Standard Queries.
- The message's call-id changes while the f**orward-reg-callid-change** option in the sip config configuration element is set. This covers the case where the UA changes the Oracle Communications Unified Session Manager through which it attaches to the network.
- The REGISTER message's Cseq has skipped a number. This covers the case in which a user registered with Oracle Communications Unified Session Manager1, moves to Oracle Communications Unified Session Manager2, and then returns to Oracle Communications Unified Session Manager1.
- The REGISTER message's contact list has changed.

After receiving the entries from the ENUM database via the NAPTR query, the list is correlated with the internal registration cache. Appropriate DDNS updates are preformed (see: ENUM Database Correlation).

If the Oracle Communications Unified Session Manager updates the ENUM database due to one of the above conditions, the UA's access-side Expires timer is reset to the REGISTER message's Expires: header value, and returned in the 200 OK. This happens even in the case when the reREGISTER was received in the first half of the previous Expires period. In addition, the core-side location update interval timer is refreshed on both active and standby.

When the above four conditions are not met, the registration and reregistration expiration proceeds normally. If the access-side expiration timer has not exceeded half of its lifetime, only a 200 OK is sent back to the UA. If the timer has exceeded half of its lifetime, the Oracle Communications Unified Session Manager refreshes the registration to the ENUM server. Note: Upon a Call-id or contact list change, both the registration cache timer and the ENUM database are updated.

Limiting AOR Contacts

The Oracle Communications Unified Session Manager allows you to limit the number of contacts that apply to AORs. If the Oracle Communications Unified Session Manager receives a registration request that exceeds the maximum that you configured, it responds with a local

response, a 403 Forbidden by default, and does not register the additional contact. The system only rejects registration requests that exceed the maximum. Existing contacts persist normally.

The system checks against the maximum in the following circumstances:

- A new registration is received
- The location-update-interval expires
- A call-id changes (and the forward-reg-callid-change option is enabled)
- A registrar message sequence number has skipped a number
- There is any change to the contact list

If the number of contacts in the initial registration exceeds the maximum, the Oracle Communications Unified Session Manager rejects the entire registration. In addition, if you configure this feature while the system is operational, your setting only applies to new registrations.

You configure these maximums on a per-registrar basis. The value ranges from 0-256. The feature is RTC supported.

User Registration based on Reg-ID and Instance-ID (RFC 5626)

Sometimes a user's device reregisters from a different network than its original registration. This event should be considered a location update rather that a completely new registration for the Contact. The Oracle Communications Unified Session Manager can perform this way by considering the endpoint's reg-id and instance-id parameters defined in RFC 5626.

The Oracle Communications Unified Session Manager identifies new REGISTER requests received on a different access network as a location update of the existing binding between the Contact and AoR. Without this feature, the Oracle Communications Unified Session Manager would create a new binding and leave the old binding untouched in the local registration cache/ ENUM database. This scenario is undesirable and leads to unnecessary load on various network elements including the Oracle Communications Unified Session Manager itself.

The following conditions must be matched to equate a newly registering contact as a location update:

For a received REGISTER:

- The message must not have more than 1 Contact header while 1 of those Contact headers includes a reg-id parameter. (failure to pass this condition prompts the Oracle Communications Unified Session Manager to reply to the requester with a 400 Bad Request).
- The Supported: header contains **outbound** value
- The Contact header contains a **reg-id** parameter
- The Contact header contains a **+sip.instance** parameter

After these steps are affirmed, the Oracle Communications Unified Session Manager determines if it is the First hop. If there is only one Via: header in the REGISTER, the Oracle Communications Unified Session Manager determines it is the first hop and continues to perform Outbound Registration Binding processing.

If there is more than 1 Via: header in the REGISTER message, the Oracle USM performs additional validation by checking that a Path: header corresponding to the last Via: includes an ob URI parameter, Outbound Registration Binding may continue.

If the Oracle Communications Unified Session Manager is neither the first hop nor finds an ob URI in Path headers, it replies to the UA's REGISTER with a 439 First Hop Lack Outbound Support reply.

reREGISTER Example

The user (AoR) bob@example.com registers from a device +sip.instance= <urn:uuid:0001> with a reg-id ="1", contact URI = sip:1.1.1.1:5060. A binding is be created for bob@example.com+<urn:uuid:0001>+reg-id=1 at sip:1.1.1.1.:5060.

Next, Bob@example.com sends a reREGISTER with the same instance-id but with a different reg-id = 2 and contact URI = sip:2.2.2.2:5060.

The previous binding is removed. A binding for the new contact URI and reg-id is created. bob@example.com+<urn:uuid:0001>+reg-id=2 at sip:2.2.2.2:5060

Outbound Registration Binding Processing

An outbound registration binding is created between the AoR, instance-id, reg-id, Contact URI, and other contact parameters. This binding also stores the Path: header.

Matching re-registrations update the local registration cache as expected. REGISTER messages are replied to including a Require: header containing the outbound option-tag.

If the Oracle Communications Unified Session Manager receives requests for the same AOR with some registrations with reg-id + instance-id and some without them, the Oracle Communications Unified Session Manager will store them both as separate Contacts for the AOR; The AoR+sip.instance+reg-id combination becomes the key to this entry.

ENUM Database Update

When a REGISTER message is received:

1. The ENUM user database is queried for the AoR and any existing entries.
2. If there are any entries with the same instance-id as the current REGISTER request in the ENUM query response, then those entries will be marked for subsequent removal in the ENUM database.
3. The ENUM database is updated with a NAPTR request. This request adds the new Contact URI for that AOR+instance-id and removes any existing entries for the same AOR +instance-id.

NAPTR Update Format

The ENUM database update includes the instance-id and reg-id when those parameters are present in a registration. These values are appended to the regex replacement field. For example:

```
!^.*$!sip:3556-1cdstqjt90hve@172.16.101.62:5060;sip.instance=<urn:uuid:
00000000-0000-1000-8000-000A95A0E128>;reg-id=1;ts=1326568408;!
```

Oracle USM Licensing

The Oracle Communications Unified Session Manager connected to an ENUM database requires two licenses: Registration Cache Limit, SIP Authorization/Authentication.

For ENUM-based Oracle Communications Unified Session Manager, the SIP Authorization/Authentication license reveals the SIP Authentication Profile configuration element. Configuring both configuration elements is required to operate a Oracle Communications Unified Session Manager. Refer to the Licensing and Database Registration Limits section for the third license required for Oracle Communications Unified Session Manager operation.

Refer to the Oracle SBC ACLI Configuration guide, Getting Started chapter for how to install licenses in your system.

ACLI Instructions

ENUM Configuration

First the server used for authentication and as the User Subscriber Database is created.

To configure the ENUM Configuration:

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

   ```
   ORACLE# configure terminal
   ```

2. Type **session-router** and press Enter to access the media-related configurations.

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

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

   ```
   ORACLE(session-router)# enum-config
   ORACLE(enum-config)#
   ```

   You may now begin configuring the enum-config configuration element.

4. **name**—Set a name to use to reference this enum configuration from within the Oracle Communications Unified Session Manager.
5. **top-level-domain**—Enter the domain which this ENUM server(s) services and returns results for.
6. **realm-id**—Enter the realm name where this ENUM server exists.
7. **enum-servers**—Enter the IP address of one or more ENUM servers used for registration. Multiple entries are separated by commas.
8. **service-type**—Leave this as its default.
9. **ttl**—Leave this at the default of 0 to set the TTL value (in seconds) for NAPTR entries as populated when sending a DNS update to the ENUM server.
10. **order**—Enter the value to populate the order field with when sending NAPTR entries to the ENUM server.
11. **preference**—Enter the value to populate the preference field with when sending NAPTR entries to the ENUM server.

12. Type **done** when finished.

SIP Authentication Profile

To configure the SIP Authentication Profile:

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

   ```
   ORACLE# configure terminal
   ```

2. Type **session-router** and press Enter to access the media-related configurations.

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

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

   ```
   ORACLE(session-router)# sip-authentication-profile
   ORACLE(sip-authentication-profile)#
   ```

   You may now begin configuring the SIP Authentication Profile configuration element.

4. **name**—Enter the name of this SIP authentication profile that will be referenced from a SIP registrar (or SIP interface).
5. **methods**—Enter all the methods that should be authenticated. Enclose multiple methods in quotes and separated by commas.
6. **anonymous-methods**—Enter the methods from anonymous users that require authentication. Enclose multiple methods in quotes and separated by commas.
7. **digest-realm**—enter the digest realm sent in an authentication challenge (401/407) sent to a UA. This is required in ENUM/DNS deployments,.
8. **credential-retrieval-method**—Enter **ENUM-TXT**.
9. **credential-retrieval-config**—Enter the enum-config name used for retrieving authentication data.
10. Type **done** when finished.

SIP Registrar

To configure the Oracle Communications Unified Session Manager to act as a SIP Registrar:

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 path.

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

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

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

4. **name**—Enter a name for this SIP registrar configuration element.
5. **state**—Set this to **enabled** to use this SIP registrar configuration element.

6. **domains**—Enter one or more domains in the R-URI this configuration element handles. Wildcards are valid for this parameter. Multiple entries can be entered in quotes, separated by commas.
7. **subscriber-database-method**—Set this to **DDNS**.
8. **subscriber-database-config**—Enter the enum-config configuration element name that will handle REGISTER messages for this domain. This should be the same element used for requesting authentication data.
9. **authentication-profile**—Enter a sip-authentication-profile configuration element's name. The sip authentication profile object referenced here will be looked up for a REGISTER message with a matching domain in the request URI. You may also leave this blank for the receiving SIP Interface to handle which messages require authentication if so configured.
10. **location-update-interval**—Keep or change from the default of 1400 minutes (1 day).
11. Type **done** when finished.

Maximum Number of Contacts

To configure a sip-registrar with a maximum of 10 contacts per AOR:

1. From superuser mode, use the following command sequence to access sip-registrar element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# sip-registrar
ORACLE(sip-registrar)# select
```

   Select the registrar you want to configure.

2. Specify the number of contacts.

```
AORACLE(sip-registrar)# max-contacts-per-aor 10
AORACLE(sip-registrar)# done
```

Response to Exceeding Maximum Contacts

To configure local response for the Oracle Communications Unified Session Manager to issue when max-contacts-per-aor is exceeded:

1. From superuser mode, use the following command sequence to access local-response and add an entry.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# local-response-map
```

2. Access the entries configuration.

```
ORACLE(local-response-map)# entries
```

3. Specify the local error you need to configure.

```
ORACLE(local-response-map-entry)# local-error contacts-per-aor-exceed
```

4. Specify the sip-reason for this error.

```
ORACLE(local-response-map-entry)# sip-reason forbidden
```

5. Specify the error code for this error.

```
ORACLE(local-response-map-entry)# sip-status 403
ORACLE(local-response-map-entry)# done
local-response-map-entry
        local-error                     contacts-per-aor-exceed
        sip-status                      403
        q850-cause                      0
        sip-reason                      forbidden
        q850-reason
        method
        register-response-expires
ORACLE(local-response-map-entry)# exit
```

Update to ENUM Database on Endpoint Connection Loss

The Oracle Communications Unified Session Manager can monitor an endpoint's transport-layer status for loss of connectivity in multiple ways. Then, when the endpoint's connection to the Oracle Communications Unified Session Manager has been terminated, the Contact is removed from the registration cache, as associated under a registered AoR. In addition, the user database is immediately updated.

These transactions contribute to the registration cache and ENUM user database being updated in real time to retain only reachable contacts for a registered AoR. This feature helps to alleviate:

- unnecessary transactions and system load spent on attempting to reach an unreachable endpoint
- incorrect and out of date statistics

Connection Reuse

The connection between the Oracle Communications Unified Session Manager and an endpoints must employ the connection reuse mechanism to enable this feature's de-registration and user database updates. Connection reuse is when an endpoint registers to the Oracle Communications Unified Session Manager and all subsequent signaling between the Oracle Communications Unified Session Manager and that endpoint reuses the same socket pair. There are four cases when connection reuse is enabled:

- Connection reuse is enabled on the SIP interface facing the endpoint(s). This is enabled by adding the **reuse-connections=yes** option on a SIP interface.
- The endpoint is behind a NAT. Because of the fundamental method that the Oracle Communications Unified Session Manager uses for maintaining its connection to an endpoint behind a NAT, this case will always force connection reuse.
- The endpoint includes the alias parameter in the Via: header in its REGISTER message to the Oracle Communications Unified Session Manager (RFC 5923).
- If the endpoint is configured as a session agent, the reuse-connections parameter must be set to TCP. When receiving signaling from a remote logical entity such as a session agent defined for an endpoint, if the reuse-connections parameter is set to tcp, the Oracle Communications Unified Session Manager enables connection reuse between itself and the UA.

Unreachability Determination

There are four ways that the Oracle Communications Unified Session Manager determines endpoint is not reachable:

- No CRLF message is returned to the Oracle Communications Unified Session Manager within the expected time frame: this is based on the Oracle USM's RFC5626 support.
- No TCP Keepalive is returned to the Oracle Communications Unified Session Manager within the expected timeframe. This is based on configuring the network parameter configuration element per application interface.
- The endpoint explicitly terminates its transport-layer connection to the Oracle Communications Unified Session Manager.
- The endpoint is otherwise labeled as unreachable from a non-explicit fault condition for a call made to an unreachable endpoint.

RFC 5635 Failure

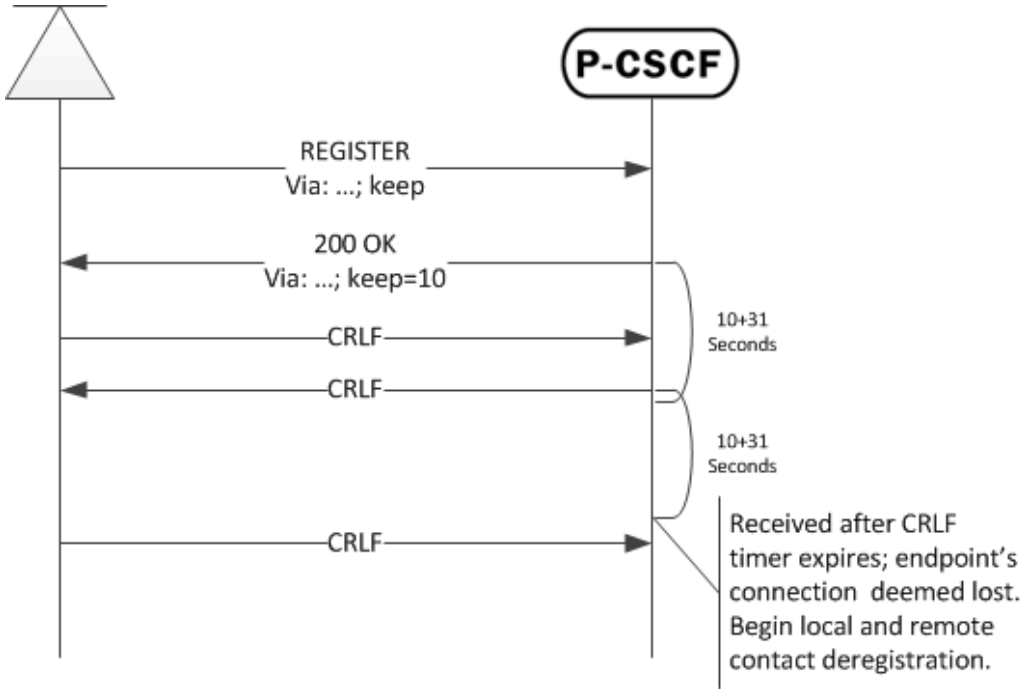
The Oracle Communications Unified Session Manager supports the RFC 5635 method of SIP application keepalives, which are endpoint-initiated, i.e., the endpoint starts the mechanism by including the keep parameter in the initial Via: header. Endpoint reachability is determined the receipt or loss of a CR/LF ping-pong message. In the SIP interface configuration element, you set the **register keep alive** parameter to **always** or **bnat** (behind NAT), to enable RFC 5635 functionality. This applicable to TCP or TLS connections. **Always** forces the Oracle Communications Unified Session Manager to always return a CRLF reply when the keep parameter is in the initial Via: header. **bnat** forces the Oracle Communications Unified Session Manager to replies to RFC 5635 requests when the endpoint is located behind a NAT.

Next, you can accept the Oracle Communications Unified Session Manager's default keep alive window of 30 seconds, or you may set your own by configuring the **tcp nat interval** or **inactive con timeout** value, both found in the sip interface configuration element. The Oracle Communications Unified Session Manager uses the smaller of the two configured values. The chosen value is inserted into the keep parameter in the 200 OK message returned to the registering endpoint.

> **Note:**
>
> The inactive con timeout value otherwise disconnects a TCP/TLS connection after the configured value elapses. The tcp nat interval value is also inserted into the expires parameter in the Contact: header for devices identified as behind a NAT.

In addition, the Oracle Communications Unified Session Manager maintains a keep timer. The Oracle Communications Unified Session Manager adds 31 seconds to the keep value it returns and begins counting down. 31 is the chosen margin to account for any network or application delay.

If the endpoint returns the CRLF before the timer expires, the endpoint is considered up and a new CRLF ping is sent to the endpoint; the timer begins counting down again. If the Oracle Communications Unified Session Manager fails to receive the CRLF ping before the timer expires, then the UA is considered unreachable. Provisions begin to remove that contact from the registration cache and then the ENUM user database.



TCP Keepalive Failure

Endpoint reachability can be determined from TCP keepalives, as enabled per SIP interface. This method of determining connectedness is based on configuring the global network parameters configuration element that is enabled on a SIP interface with the tcp-keepalive parameter. See the System TCP Keepalive Settings section of the Oracle SBC ACLI Configuration guide for how to configure the TCP keepalive feature. When an endpoint fails the TCP keepalive test, provisions begin to remove that contact from the registration cache and then the ENUM user database.

Explicit and undetermined connection termination

Ideally, a UA will gracefully close its TCP connection to the Oracle Communications Unified Session Manager, and in turn the socket pair will be considered closed with the UA being unreachable. In less ideal cases, the UA goes dark and no response is received when expected. The Oracle Communications Unified Session Manager considers the unresponsive UA as unreachable as call attempts time out. Provisions then begin to remove that contact from the registration cache and then the ENUM user database.

Registration Cache and User Database Removal

When the Oracle Communications Unified Session Manager sets up the initial REGISTER request from a UA, an internal binding is created between the contact and the AoR for that user. If a UA is considered unreachable by either of the four conditions explained in the previous section, the following occur:

- The contact's entry in the registration cache is removed. If this is the last contact registered for an AoR, the entire entry for the AoR is removed from the registration cache.
- The Oracle Communications Unified Session Manager sends an UPDATE to the ENUM user database removing the Contact.

To enable these actions, you must configure the **force unregistration** option sip config configuration element and also set the **unregister on connection loss** parameter in the sip interface configuration element to **enabled**.

ACLI Instructions

To globally enable force-unregistration at the sip-config level:

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 path.

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

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

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

4. Type **select** to continue.

   ```
   ORACLE(sip-config)# select
   ORACLE(sip-config)#
   ```

5. **options**—Set the options parameter by typing **options**, a Space, **force-unregistration** with a plus sign in front of it, and then press Enter.

   ```
   ORACLE(sip-config)# options +force-unregistration
   ```

   If you type the option without the plus sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration's options list, you must prepend the new option with a plus sign as shown in the previous example.

6. Type **done** and **exit** to complete configuration of this **sip-config** configuration element.

SIP Interface Configuration

To configure the SIP Interface configuration element portion of the ENUM Database update feature:

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 path.

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

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

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

4. Type **select** and choose the number of the pre-configured sip interface you want to configure.

   ```
   ORACLE(sip-interface)# select
   <realm-id>:
   1: private 192.168.101.17:5060
   ```

```
2: public  172.16.101.17:5060
selection: 1
```

5. **unregister-on-connection-loss**—Set this parameter to **enabled** for the Oracle Communications Unified Session Manager to update the ENUM server when an endpoint is deemed failed.

   Parameters for determining endpoint reachability:

6. **tcp-keepalive**—You may set this parameter to enabled to enforce the network-parameters configuration element located in the system-config path. See the Oracle SBC ACLI Configuration guide, System TCP Keepalive Settings section for more information.

7. **register-keep-alive**—Set this parameter to **always** for the Oracle Communications Unified Session Manager to return the **keep** parameter, with optional value (as configured in the next two steps) in the Via: header to an endpoint including an empty keep value in its initial REGISTER message.

   You may set none, one, of both the following for a keep value returned to the initiating endpoint. Read the RFC 5635 Failure section for how the actual value is determined:

8. **inactive con timeout**—Set this parameter value to the value in seconds inserted in the returned keep parameter for RFC 5635 support

9. **tcp nat interval**—Set this parameter value to the value in seconds inserted in the returned keep parameter for RFC 5635 support

   Type **done** and **exit** to complete configuration of this **sip-interface** configuration element.

OAuth 2.0 Support

The Oracle Communications Unified Session Manager supports Open Authorization (OAuth) in addition to SIP digest authentication for user authorization within ENUM deployments. Both authorization methods can be operational simultaneously, allowing some users to authorize via OAuth and others via SIP digest. Applicable scenarios include authorizing registrations, subscriptions and invites.

OAuth uses HTTP to provide end users with access to services from OAuth 2.0 protected resources using various clients. OAuth also allows users to authorize third-party access to their services using user-agent redirections rather than sharing username password pairs. Any party presenting the proper bearer token can be authenticated. The methodology avoids the use of cryptographic keys, and requires protection from token disclosure during transit and storage. OAuth assumes a secure exchange of credential validation information between end points, specifically an OAuth client and server, prior to operation.

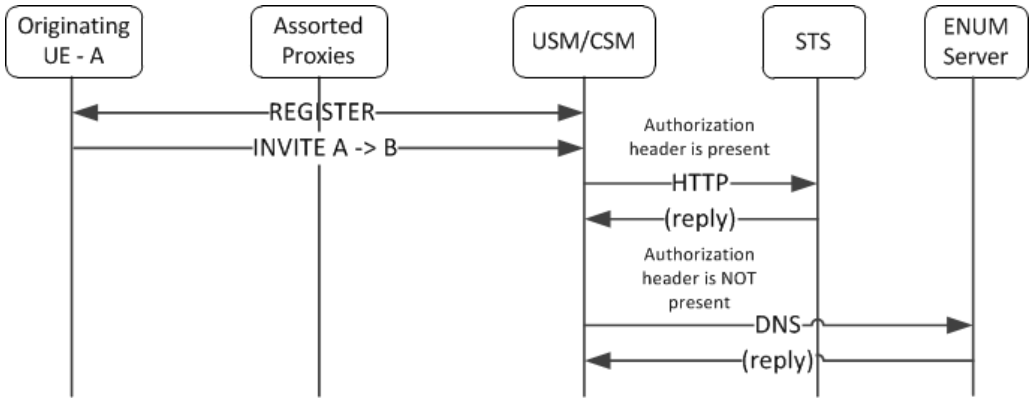
The Oracle Communications Unified Session Manager implements OAuth in compliance with RFCs 6749 and 6750. OAuth typically requires deployment-specific compliance beyond RFC compliance. The Oracle Communications Unified Session Manager allows for typical deployment environments via configuration.

You configure the Oracle Communications Unified Session Manager to use OAuth by creating OAuth profiles and applying them globally and/or to specific interfaces. Interface profiles take precedence. In addition, you specify the server that the Oracle Communications Unified Session Manager must contact for authentication using Oracle ’s Session Processing Language (SPL). Note that this functionality requires that you upload a script file, provided by Oracle , and run an ACLI command to configure the script to use your server. This script file is referred to as a plug-in.

OAuth Operation

The Oracle Communications Unified Session Manager assumes a client requesting OAuth authentication has previously acquired a token from its authorization server. This and token refresh procedures are performed outside of the scope of Oracle Communications Unified Session Manager operation. Within the context of OAuth processes, the Oracle Communications Unified Session Manager performs the functions of a resource server only.

Upon receiving applicable requests from a UA, the Oracle Communications Unified Session Manager attempts to authenticate using OAuth or Digest. If the request includes a bearer string and a token, the Oracle Communications Unified Session Manager initiates OAuth authentication by sending an HTTP query (GET request) to a Secure Token Service Server (STS), as shown below.



If the STS returns a 200 OK, the Oracle Communications Unified Session Manager proceeds with authentication. If the STS does not reply or is not reachable, the Oracle Communications Unified Session Manager replies to the UA with a 500 Internal server error. If the STS replies with any message other than a 200OK, the Oracle Communications Unified Session Manager replies to the UA with a 403 Forbidden.

The Oracle Communications Unified Session Manager uses the User ID within the STS 200OK to authenticate and authorize service, as follows:

- For register requests, the system compares this to the user ID field in the TO header of the SIP request.
- For other requests, the system compares it to the user ID field in the FROM header.

If the user ID matches, the Oracle Communications Unified Session Manager proceeds with authentication. If this procedure concludes with matches, the Oracle Communications Unified Session Manager provides service to the UA. If not, it replies to the UA with a 403 forbidden.

Note the following operational caveats:

- If the bearer token is present, but the Oracle Communications Unified Session Manager is not configured for OAuth, the Oracle Communications Unified Session Manager responds with a 500 internal service error message. Such misconfiguration includes the OAuth SPL plugin, described below, being absent or disabled.
- If the bearer token is not in the request, the Oracle Communications Unified Session Manager proceeds with SIP digest authentication.
- If the request arrives over a secure channel, the Oracle Communications Unified Session Manager follows the procedure defined by the applicable sip-authentication-profile, which may not require authentication.

Configuring OAuth Support

Configuring the Oracle Communications Unified Session Manager for OAuth support consists of:

- Creating one or more OAuth profiles
- Applying each profile globally or to sip-interfaces

To configure an OAuth profile:

1. From superuser mode, use the following command sequence to access http-config element and define your profile.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# http-config
   ORACLE(http-config)# sel
   ```

2. Name your profile for reference within your configuration.
3. You can set the host parameter to an FQDN or an ip address. If you use an FQDN, the Oracle Communications Unified Session Manager resolves it to an ip address when the configuration is loaded.
4. The num-connections parameter specifies the number of connections to be setup with the STS server. The Oracle Communications Unified Session Manager establishes this number of connections when it boots or when the configuration is activated.
5. The max-outstanding-msgs parameter specifies the maximum number of outstanding messages per connection. An outstanding message is a request from the Oracle Communications Unified Session Manager that has not had a response. When reaching this threshold, the Oracle Communications Unified Session Manager stops sending requests on the connection until the number of outstanding messages falls back below this maximum. If the max-outstanding-msgs parameter is set to 1, the Oracle Communications Unified Session Manager waits for a response before sending another request on a connection.
6. The Oracle Communications Unified Session Manager uses the port parameter differently depending on whether the host parameter is configured as an IP address or an FQDN. If the host parameter is an FQDN name, the Oracle Communications Unified Session Manager performs a lookup at a DNS server. In this case, you may or may not configure a port. If you configure the port parameter, the Oracle Communications Unified Session Manager uses the configured port ignoring any port specified in the record returned by the DNS server. If you do not configure the port parameter, the Oracle Communications Unified Session Manager uses the port returned by the DNS server. If the host parameter is set to an ip-address, you must configure the port parameter and the Oracle Communications Unified Session Manager always uses the port parameter's value.

Enabling the SPL Plug-in

Enabling the SPL plug-in is a four step process.

1. Upload the SPL plug-in to a Oracle Communications Unified Session Manager.
2. Add the SPL plug-in to the Oracle Communications Unified Session Manager configuration.
3. Execute the SPL plug-in on the Oracle Communications Unified Session Manager.
4. Synchronize the plug-in across HA pairs.

Uploading the Plug-in

The plug-in must be manually FTPed to the Oracle Communications Unified Session Manager's /code/spl directory using any CLI or GUI-based FTP or SFTP application. The Oracle Communications Unified Session Manager's FTP/SFTP server may be reached from the system's wancom or eth0 management physical interface.

Adding the Plug-in to Your Configuration

The plug-in must be configured in the spl-config configuration element. If multiple plugins are configured on the Oracle Communications Unified Session Manager, the plug-ins are executed in the order of configuration.

To add the SPL Plugin to the configuration:

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
   ORACLE(system)#
   ```

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

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

4. Type **plugins** and press Enter. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMESYSTEM(spl-config)# plugins
   ACMESYSTEM(spl-plugins)#
   ```

5. Type **name**, a <space>, and the name of the SPL plug-in file**.**

   ```
   ACMESYSTEM(spl-plugins)#name SMXOauth2.spl
   ```

6. Type **done** to save your work.

Executing SPL Files

There are two ways to execute SPL files:

1. Perform a **save-config** and **activate-config** after exiting the configuration menu.
2. Execute the **reset spl** command—all configured SPL files are refreshed by with the **reset spl** command. You can also refresh a specific file by typing **reset spl <spl-file>.**

> **Note:**
>
> Oracle suggests that scripts are only refreshed during planned maintenance windows.

If an SPL file exists in the /code/spl directory, but is not configured in the **spl-files** parameter, it will be ignored when the ACLI User Interface is booting.

Synchronizing SPL Files Across HA Pairs

When running in an HA configuration, both the active and the standby systems must have the same version of the SPL plugins installed. To facilitate configuring the standby system, you can executing the **synchronize spl** ACLI command (without any arguments) to copy all files in the /code/spl directory from the active system to the same directory on the standby, overwriting any existing files with the same name.

By adding the specific filename as an argument to the **synchronize spl** command, the individual, specified scripts are copied between systems. For example:

```
ORACLE#synchronize spl SMXOauth2.spl
```

The **synchronize spl** command can only be executed from the active system in a HA pair. There is no means to synchronize SPL files automatically during a save and activate of the Oracle Communications Unified Session Manager.

To sychronize all SPL Plug-ins to the configuration:

1. In Superuser mode, type **synchronize spl** and press Enter.

   ```
   ORACLE# synchonize spl
   ```

Configuring the Plug-in Option

For the SPL to work, you must configure it to recognize your http-server configuration. You do this by setting the **sts-server** option at the sip-registrar. This option is required for plug-in functionality.

To configure the sts-server option:

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

   ```
   ORACLE# configure terminal
   ```

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

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

3. Access the target sip-registrar by typing **sip-registrar,** selecting your registrar and pressing Enter.

   ```
   ORACLE(session-router)# sip-registrar
   ORACLE(sip-registrar)# select
   ```

4. Type **sts-server=<http-config>, where ”<http-config>”** is the name of your http configuration element, and press Enter.

   ```
   ACMESYSTEM(sip-registrar)# spl-options sts-server=http-server1
   ```

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

Message Routing

The Oracle Communications Unified Session Manager performs routing in two ways depending on the routing precedence parameter in the sip registrar. Routing precedence can be set to either **registrar** (ENUM) or **local policy**. Routing precedence is set to registrar by default.

Registrar routing uses the configured SIP registrar/ENUM server for destination address queries. Local policy routing lets you configure routing decisions within the Oracle Communications Unified Session Manager's local policy routing functionality.

If the Oracle Communications Unified Session Manager is performing any services for the call it performs those services prior to routing.

If, for any reason, the Oracle Communications Unified Session Manager is unable to proceed with routing a request, it replies to the station that sent the request with a 4xx response.

Registrar Routing

When the routing precedence parameter is set to **registrar**, the Oracle Communications Unified Session Manager is using the ENUM server as a resource within the context of its routing decisions.

When an INVITE arrives, the Oracle Communications Unified Session Manager issues an query to the ENUM server. If the query's reply includes a match, the Oracle Communications Unified Session Manager proceeds by forwarding the message via the information in the reply.

Note that you can configure the Oracle Communications Unified Session Manager to fallback to a local policy lookup if the lookup via the registrar fails. Configure this by adding the **fallback-to-localpolicy** option to the sip-registrar configuration element.

For situations where the database routing decision needs to be done in lieu of the default, you can set routing precedence to local-policy. Note that you can configure a routing entry that points to an HSS by setting a policy attribute with a next-hop of enum:<server-name> within the local-policy.

Default Egress Realm

The sip registrar configuration element should be configured with a default egress realm id. This is the name of the realm config which defines the IMS control plane through which all Oracle Communications Unified Session Managers, ENUM servers, and other network elements communicate and exchange SIP messaging. It is advisable to configure this parameter in order to ensure well defined reachability among Oracle Communications Unified Session Managers.

SIP Registrar

To configure a SIP registrar configuration element for message routing:

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 path.

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

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

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

4. Type **select** and choose the number of the pre-configured sip registrar you want to configure.

5. **routing-precedence**— Set this to either **registrar** or **local-policy** depending on your deployment.
6. **egress-realm-id**—Enter the default egress realm for Oracle Communications Unified Session Manager messaging.
7. Type **done** when finished.

Segmentation of ENUM Zones

The Oracle Communications Unified Session Manager supports operations with an ENUM root server, from which it can obtain partition delegation information and dynamically reach delegated authoritative ENUM servers for access information by ENUM zone. Obtaining this information is dynamic and supports query re-direction to other relevant authoritative servers the Oracle Communications Unified Session Manager may learn from the root server. This configuration applies only to DDNS-based registration deployments. The user configures the **ddns-ns-caching** option to enable the Oracle Communications Unified Session Manager for dynamically learned ENUM server caching.

For DDNS queries, the Oracle Communications Unified Session Manager uses a manually-configured ENUM server configuration to reach a DNS root server. The root server can respond with the authoritative name server that manages the zone. The Oracle Communications Unified Session Manager establishes dynamic entries in its ENUM server cache, with detail about the zones those servers service. This allows it to issue subsequent lookup queries directly to the correct server based on a match between telephone number and zone prefixes.

This feature assumes ENUM servers that are bind 8.0 compliant and return authority sections compliant with RFC 1304, section 4.3.2.

Additional operational details include:

- Resolution lookups are longest match, providing the Oracle Communications Unified Session Manager with the ability to send a query to authoritative servers servicing zones and sub-zones.
- Dynamic server timeouts are equal to the TTL values received in the name server's resource record, provided in the NS response. The Oracle Communications Unified Session Manager removes dynamic servers, as well as any associated ACL entries upon timeout.
- Redundancy for DDNS name server lookups is round-robin, triggered by standard DNS query timeout procedures, and following the order the system learns servers for the target zone.
- If a dynamic server fails to respond to a DDNS lookup, the Oracle Communications Unified Session Manager removes that server from the lookup list. If there are no further servers in the list, the Oracle Communications Unified Session Manager sends a 404 back to the station that originated the request. The Oracle Communications Unified Session Manager then starts any subsequent lookups for that zone by querying the root server.
- Multiple NS records are supported. The Oracle Communications Unified Session Manager creates NS records for the same zones in the order provided to it to establish name server redundancy.
- Should the Oracle Communications Unified Session Manager need to obtain a resolution for a prefix that is unknown, the Oracle Communications Unified Session Manager queries the original root server again.

- The Oracle Communications Unified Session Manager removes any dynamically-learned name server from a query list when any query fails. That server is only added back to a list when it is dynamically re-learned.
- Should the root server become unavailable, the Oracle Communications Unified Session Manager sends applicable queries to the next server in your manually-configured ENUM list.
- The Oracle Communications Unified Session Manager does not execute its configurable health-query check processes with dynamically learned name servers.
- With respect to high availability deployments, the server list is not maintained on a standby Oracle Communications Unified Session Manager. The ENUM configuration, however, is maintained on the standby via configuration synchronization. In the event of a failover, the standby Oracle Communications Unified Session Manager learns all ENUM servers using the root server as a starting point.
- This **ddns-ns-caching** function interoperates with the Oracle Communications Unified Session Manager's alphanumeric user name function. The zone match is done on the 3-bit checksum of the alpha-numberic user.

> **Note:**
>
> Name server records must present the Oracle Communications Unified Session Manager with IP addresses for the server. Name server records using FQDNs are not supported.

Obtaining Information about Dynamic ENUM Servers

The Oracle Communications Unified Session Manager provides ACLI commands that display real-time information about ENUM server interaction, including dynamically learned servers. The **show enum** command includes a section of statistics on cached entries

```
ORACLE# show enum stats localenum
 Parameter-> localenum lastArg  3 enumStatsType  0

SIP ENUM Statistics:
16:01:51-48
ENUM Agent localenum
                            -- Period -- -------- Lifetime --------
                   Active    High   Total      Total  PerMax    High
Queries                 -       -       3          3       3       -
  Successful            -       -       3          3       3       -
  NotFound              -       -       0          0       0       -
  TimedOut              -       -       0          0       0       -
  Bad Status            -       -       3          3       3       -
  Other Failures        -       -       0          0       0       -
Transactions            0       1       4          4       4       1
Cache Hits
  Successful            -       -       0          0       0       -
  NotFound              -       -       0          0       0       -
Cache Entries
  Successful            0       0       0          0       0       0
  NotFound              0       0       0          0       0       0
DropdCacheEntries       -       -       0          0       0       -
```

When appended with the **cache-console** argument, **show enum** displays detailed information about dynamic server configuration and status.

```
ORACLE# show enum status all cache-console
Showing 2 Enum Agents: state=Active cache
-------------------------------------
Enum Agent:       localenum
  Realm:          net192.4
  EnumServers:    192.168.53.199
  Query Timeout:  11
  Lookup Length:  3
  Health Query:   '' every 0 sec
  Failover To:
  IncludeSrcInfo: disabled
  Options:
  RecursiveQuery: disabled
DNS Agent:        EnumAgent[0x3b75a628(3)] ENUM w/o-stats
  Domain:         bogus.gy
  Local Address:  [256:0]192.168.53.170
  Service Types:  E2U+sip sip+E2U
  Trans ID:       7
  Cache Size:     0(0) inact-tmr=0 cache-addl=no
  Max Resp Size:  512
  Query Method:   hunt
servers:
    1=[256:0]192.168.53.199:53 OK
dynamic servers:
    1=[256:0]192.168.53.205:53 OK
```

Configuring Support for DDNS Server Caching

Use the procedure below to enable the Oracle Communications Unified Session Manager to cache and manage DDNS servers for the purpose of efficiently forwarding ENUM queries to the servers responsible for the DNS zones identified in applicable SIP requests.

1. From superuser mode, use the following command sequence to access **enum-config** configuration mode.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# enum-config
   ```

2. Use the **select** command to access the **enum-config** element or create a new one for the target ENUM root server.

3. Enable the **ddns-ns-caching** option.

   ```
   ORACLE(enum-config)# +option ddns-ns-caching
   ```

4. Use **done**, exit configuration mode, and run **verify-config** to complete DDNS cache support configuration.

   ```
   ORACLE(enum-config)# done
   ORACLE(enum-config)# exit
   ORACLE(session-router)# exit
   ORACLE(configure)# exit
   ORACLE# verify-config
   ----------------------------------------------------------------------
   Verification successful! No errors nor warnings in the configuration
   ORACLE#
   ```

Tel-URI Resolution

The Oracle Communications Unified Session Manager can initiate number resolution procedures for requests that have tel-URI or SIP-URI (with user=phone) numbers in the R-URI. It does this by querying number resolutions services, including the local routing table(s) or ENUM server(s) to resolve the R-URI to a SIP URI. In addition, the original R-URI may not include a full E.164 number. As such, you can also configure the Oracle Communications Unified Session Manager to perform a number normalization procedure and ensure it presents a full E.164 number for resolution. Upon successful resolution, the Oracle Communications Unified Session Manager proceeds with ensuing signaling procedures.

To configure the Oracle Communications Unified Session Manager to perform these lookups, you create applicable **local-routing-config** or **enum-config** elements and set an option within the **sip-registrar** that specifies a primary and, optionally, a secondary l**ocal-routing-config** or **enum-config** that the sip-registrar uses for LRT or ENUM lookups. If there is no ENUM configuration on the sip-registrar, the Oracle Communications Unified Session Manager forwards applicable requests to a border gateway function via local policy.

Refer to the *Oracle Communications Session Border Controller ACLI Configuration Guide*, Session Routing and Load Balancing chapter for complete information on how to configure a **local-routing-config** and an **enum-config** elements.

Number Lookup Triggers

Use cases that are applicable to number lookups and the associated Oracle Communications Unified Session Manager procedures include:

- Request from the access side:
  1. The Oracle Communications Unified Session Manager performs originating services
  2. If the R-URI is a tel-URI or SIP-URI (with user=phone) and is not in the Oracle Communications Unified Session Manager cache, it requests e.164 resolution from the ENUM server(s).
- Request from core side including request for originating services:
  1. The Oracle Communications Unified Session Manager performs originating services
  2. If the R-URI is a tel-URI or SIP-URI (with user=phone) and is not in the Oracle Communications Unified Session Manager cache, it requests e.164 resolution from the ENUM server(s).
- Request from core side, for terminating services only:
  1. If the R-URI is a tel-URI or SIP-URI (with user=phone) and is not in the Oracle Communications Unified Session Manager cache, it performs a NAPTR lookup.
  2. If the reply indicates the tel-URI or SIP-URI (with user=phone) is not provisioned, the Oracle Communications Unified Session Manager requests e.164 resolution from the ENUM server(s).

Actions Based on Lookup Results

The Oracle Communications Unified Session Manager forwards to the resultant SIP-URI under the following conditions:

- The SIP-URI is in the Oracle Communications Unified Session Manager cache, in which case the Oracle Communications Unified Session Manager performs terminating services.
- The SIP-URI is not in the Oracle Communications Unified Session Manager cache, and the Oracle Communications Unified Session Manager is configured to service the returned domain.
  In this case, the Oracle Communications Unified Session Manager performs the following:
  1. The Oracle Communications Unified Session Manager issues an LIR for the SIP-URI.
  2. The Oracle Communications Unified Session Manager forwards the message to the correct S-CSCF.
- The SIP-URI is not in the Oracle Communications Unified Session Manager cache, and the Oracle Communications Unified Session Manager is not configured to service the returned domain.
  In this case, the Oracle Communications Unified Session Manager performs refers to local policy to forward the message via local policy.

PSTN Breakout Routing

The Oracle Communications Unified Session Manager complies with RFC 4694 for operation with request-URIs that include carrier identification code/route number/number portability database dip indicator (cic/rn/npdi) information and routes those requests according to the rn information. The routing process includes utilization of local policy configured to break the request out of the home network via gateways such as a BGCF.

The Oracle Communications Unified Session Manager does not validate any rn or cic information. Instead, it simply routes the request. Note that the Oracle Communications Unified Session Manager uses cic information instead of rn if both are present in the request. RFC 4694 compliant circumstances under which the Oracle Communications Unified Session Manager does not use rn, cic and npdi information include:

- Invalid routing information, including rn present, but npdi missing.
- Invalid routing information, including npdi present, but rn missing.
- Request uses a `sip-URI` presented without `user=phone`.

If the request includes originating services as well as cic/rn/npdi information, the Oracle Communications Unified Session Manager performs those services rather than break out. If, after completing originating services, the request still includes cic/rn/npdi information, the system performs this breakout.

Primary and Secondary ENUM Configs

For the purpose of redundancy, the Oracle Communications Unified Session Manager allows you to configure these number lookups to use a backup resource in case the lookup at the primary fails. Such scenarios include losing contact with the primary ENUM/LRT server config (query time-out) and the entry is not found at the primary (LRT or ENUM).

To apply primary and secondary number lookup resources to a sip-registrar:

1. From superuser mode, use the following command sequence to access the sip-registrar element and select the registrar you want to configure.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# sip-registrar
ORACLE(sip-registrar)# select
```

2. Specify the resources to use with the options command.

   Prepend the option with the + character if you have multiple options configured that you want to retain. Running the command without the + character causes the system to disable any previously configured options.

   To specify primary and secondary ENUM servers:

   ```
   ORACLE(sip-registrar)# options +e164-primary-config=enum:<enum-config name>
   ORACLE(sip-registrar)# options +e164-secondary-config=enum:<enum-config name>
   ORACLE(sip-registrar)# done
   ```

   To specify primary and secondary LRT resources:

   ```
   ORACLE(sip-registrar)# options +e164-primary-config=lrt:<lrt-config name>
   ORACLE(sip-registrar)# options +e164-secondary-config=lrt:<lrt-config name>
   ORACLE(sip-registrar)# done
   ```

   Bear in mind that an enum-config can reference multiple servers. When the Oracle Communications Unified Session Manager references an enum-config, queries follow the normal enum-config sequence, checking each referenced server in order. If the lookup is not successful at the primary, the Oracle Communications Unified Session Manager checks the servers in the registrar's e164-secondary-config.

   In addition, each enum-config may refer to a different top-level-domain. This allows you to configure the Oracle Communications Unified Session Manager to successfully perform lookups within two domains.

Licensing and Database Registration Limits

The Oracle Communications Unified Session Manager (and the Oracle Communications Session Border Controller) limit the number of unexpired registration cache entries globally. The total number of system registrations is configured with the registration cache limit parameter in the **sip config** configuration element.

The Oracle Communications Unified Session Manager also limits the number of registration cache entries that were obtained from a User Subscriber Database; only REGISTERs that prompted the database query are counted here. As User Subscriber Database entries are added and removed, this counter is updated accordingly. Note that it is the actual number of SD-contacts that count against the license limit. Discrete database registration license values range from 20,000 through 500,000 in increments of 20,000.

When a registering contact is rejected because it will exceed one of these limits, the Oracle Communications Unified Session Manager sends a 503 message to the registering endpoint.

Refer to the *Oracle Communications ACLI Configuration Guide*, chapter 2 Getting Started, Software Licensing section for how to install a license.

Database Registration Limit Alarm

By default, a major alarm is enabled when 98% or more of the licensed number of Database Registrations are used. This alarm is cleared when the number of database registrations falls below 90%. You can configure minor and critical alarms when crossing configured thresholds and you can also reassign the major alarm. This is configured in by creating a system-config > alarm-threshold sub element with type of **database-registration**.

Extended ENUM Record Length

In some cases, when a user registers a contact from a UA, the Contact: header's contents are too long to be inserted into the DNS-based user database as presented in the NAPTR record.

To mitigate this, the Oracle Communications Unified Session Manager stores contact-related metadata, i.e. parameter key and value pairs, in a related TXT record.

If the contents of the Contact: header do not exceed 255 bytes, there is generally no need to extract the metadata to store in an additional TXT record, so the Oracle Communications Unified Session Manager will not enact this process.

NAPTR and TXT Record Creation and Association

When a user initially registers to the Oracle Communications Unified Session Manager, the DDNS update sent to the user database will include a NAPTR record and a TXT record. The NAPTR record contains the Contact: header's SIP URI as a regexp replacement. All URI and header parameters from the Contact: header are excluded from the NAPTR record and are inserted into the accompanying TXT record.

The NAPTR record and the TXT record both contain a Oracle Communications Unified Session Manager-generated key that binds the Contact metadata and the Contact URI sent in the separate records. This common key indicates the data in the NAPTR and TXT record belong to the same registering contact, and is used to recreate the Contact: header for later use. The format of the common key is:

```
p-acme-ckey=<SD-Core-IP>:<integer id>
```

The <SD-Core-IP> is the IP address from which the Oracle Communications Unified Session Manager communicates with the DNS server. The <integer id> enumerates each contact. Contacts are enumerated in the integer-id element because they can be non-unique when a user behind a NAT registers more than one contact from behind the NAT. For example:

```
    TXT "p-acme-ckey=172.16.101.61:1" "$key=value" "^key=value"
NAPTR 1 1 "u" “E2U+sip""!^.*$!sip:642-10u72@172.16.101.61:5060\;p-acme-
key=172.16.101.61:1!"
```

In the previous example, the key=value pair represent parameters in the Contact header and Contact SIP URI. A $key= indicates the parameter existed in the Contact: SIP URI. A ^key= indicates the parameter existed in a Contact: header parameter. The Oracle Communications Unified Session Manager uses this convention to reconstruct the parameters' placement in the original Contact: header.

NAPTR Record Format

Oracle Communications Unified Session Managers learn that an associated TXT record exists for the from a NAPTR lookup for the queried SIP URI.

In the DDNS update, the Oracle Communications Unified Session Manager indicates an associated TXT record by setting the NAPTR resource record with:

- m in the flags field (in addition to the u flag)
- E2U+sip:contact in the service type field

When no additional metadata and no parameters need to stored in ENUM server, the flags field contains u.

TXT Record Retrieval

A Oracle Communications Unified Session Manager performs a separate query to the ENUM server for the TXT metadata records when it detects the 'm' flag in a NAPTR result (and the one-query-txt-naptr option is not configured). You can set the one-query-txt-naptr option to enabled in the enum-config configuration element to force the Oracle Communications Unified Session Manager to request the TXT and NAPTR records both in one query from the ENUM server.

Once the Oracle Communications Unified Session Manager receives the metadata stored in the TXT records, it restores the header and URI parameters that were present on the original registration.

Requirements

BIND 9.8.1-P1 or later hosting the ENUM server supports this feature.

SIP User Parts - RFC 3261 Character Set Support

RFC 3261 specifies the range of characters allowed in a user-part, all of which are supported by the Oracle Communications Unified Session Manager. ENUM databases, however, do not support all these characters.

By default, the Oracle Communications Unified Session Manager presents SIP messages to your DNS server unchanged, assuming that your deployment uses telephone numbers only as URI user parts. You configure the Oracle Communications Unified Session Manager to support the entire range of characters allowed in user parts by enabling it on your ENUM server configuration. Upon configuration, the Oracle Communications Unified Session Manager grooms the user part appropriately for the ENUM server.

Encoding Alpha-Numerics

If your deployment uses non-numeric characters in user parts, you can explicitly enable the Oracle Communications Unified Session Manager to change the user part to be compatible with the ENUM database. In these cases, the Oracle Communications Unified Session Manager encodes SIP message user parts using a proprietary encoding. The Oracle Communications Unified Session Manager provides the encoded string to the DNS server, which then creates the applicable record(s). The Oracle Communications Unified Session Manager decodes these strings as necessary during subsequent interactions with the DNS server and the UAs.

In some cases, deployments include SIP messaging in your environment that does not require encoding. These cases include messages with SIP URI user parts composed of:

- tel-uri
- sip-uri with all numeric characters
- sip-uri with all numeric characters except for a leading +

In these cases, the Oracle Communications Unified Session Manager does not encode these user parts. In addition, for all-numeric user parts proceeded by the + character, the Oracle Communications Unified Session Manager strips the + character, reverses the digits, separates

the digits with periods, and sends the message to the ENUM server with the user part unencoded.

Multiple DNS Zone Support

For smaller deployments, one can assume a single DNS zone within which all applicable UEs reside. Large deployments, however, may use multiple DNS zones, allowing the network administrator to segregate and more easily manage large numbers of UEs.

When the Oracle Communications Unified Session Manager encodes the user part, it performs a modulo 1000 operation on a 24-bit checksum of the user part and appends the resulting 3 digits to the encoded user part. You can configure your DNS zones for these digits to organize users into zones.

For example, encoding the user part acme_user gives the string acmeX5Fuser.2.7.7. Assuming a domain of acme-ims.com, the ENUM entry for this user would be acmeX5Fuser.2.7.7.acme-ims.com. Applicable zone configuration can use the .2.7.7 portion of this string to determine this user's zone.

Alpha-Numeric Name Support

To enable to use of alpha characters in SIP message user parts for a given sip-registrar:

1. From superuser mode, use the following command sequence to access sip-registrar element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# enum-config
   ORACLE(enum-config)# select
   ```

   Select the enum-config you want to configure.

2. Specify alpha-numeric name support for this enum-config and specify the number of DNS zones in your deployment.

   ```
   ORACLE(enum-config)# alpha-numeric-user-support enabled
   ORACLE(enum-config)# done
   ```

   Note that the parameter alpha-numeric-user-support is RTC supported.

Configuring SIP Ping OPTIONS Support

You can configure the Oracle Communications Unified Session Manager to respond to SIP ping OPTIONS. This support is typically configured on an S-CSCF so it can respond to pings OPTIONS sent by a P-CSCF:

To configure an SIP Options Ping response support:

1. From superuser mode, use the following command sequence to access ping-response command on a sip-interface element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# session-router
   ORACLE(session-router)# sip-interface
   ORACLE(sip-interface)# sel
   ```

2. Enable the support with the ping-response command.

```
ORACLE(http-config)# ping-response enabled
ORACLE(http-config)# done
```

ping-response—Enable ping-response to allow your device to respond to ping OPTIONS. For example, this feature is useful within hybrid deployment environments on a P-CSCF as a means of verifying the S-CSCF's availability. This configuration allows the S-CSCF to respond to SIP ping OPTIONS.

6

Local Subscriber Tables

Local Subscriber Table

A local subscriber table (LST) is an XML formatted file that contains one or more usernames associated with a hash as encrypted or plaintext. The LST is saved locally on the Oracle Communications Unified Session Manager's file system.

LSTs enable a standalone Oracle Communications Unified Session Manager node or high-availability (HA) pair to forego relying on an external user database. Thus the Oracle Communications Unified Session Manager does not need to communicate with a server to authenticate users. This can eliminate the operational complexity of deploying a highly available credential storage system.

LST Runtime Execution

The LST is loaded on boot up when the configuration is appropriately set. Incoming messages thereafter can then be authenticated based on the credentials in the LST. If the Oracle Communications Unified Session Manager can not load an LST file, three things occur:

1. The following log message is recorded at the NOTICE level:

   ```
   LST [table-name] was not loaded - [filename] has error loading XML file
   ```

2. The message stated above is printed on the ACLI.
3. A 503 Response is returned to the UA that sent the initial REGISTER message to the Oracle Communications Unified Session Manager.

LST Configuration

To configure the Oracle Communications Unified Session Manager to use LSTs for authentication, you need to create a local subscriber table configuration element that identifies that LST. You then need to set the sip authentication profile configuration to reference that LST configuration so that when messages requiring authentication are received and processed by a sip registrar configuration element, the Oracle Communications Unified Session Manager will use the identified LST for authentication.

In a local subscriber table configuration, you must define an object **name**, identify the specific LST **filename** (and path). If the filename is entered without a path, the Oracle Communications Unified Session Manager looks in the default LST directory, which is /code/lst. If the LST file is located elsewhere on the Oracle Communications Unified Session Manager, you must specify the filename and absolute path. For example /code/path/01302012lst.xml.

The corresponding sip authentication profile must be set to use the **local subscriber table** configuration element you just created. First set **credential retrieval method** to **local**, set the **digest realm** appropriately (this is required for authentication), and finally set the **credential retrieval config** parameter to the **name** of the local subscriber table configuration element that you just created. At this point you may save and activate your configuration.

Unencrypted passwords for each user in the table is computed with the MD5 hash function as follows:

```
MD5(username:digest-realm:password)
```

ACLI Instructions

LST Table

To configure the Oracle Communications Unified Session Manager to use an LST:

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 path.

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

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

   ```
   ORACLE(session-router)# local-subscriber-table
   ORACLE(local-subscriber-table)#
   ```

   You may now begin configuring the local subscriber table configuration element.

4. **name**—Enter the name of this local subscriber table configuration element that will be referenced from a SIP registrar configuration element.
5. **filename**—Enter the filename that describes this LST XML file. If no path is given, the Oracle Communications Unified Session Manager looks in the /code/lst directory. You may provide a complete path if the file is located elsewhere.
6. **secret**—Enter the PSK used in encryption and decryption of the passwords in the XML file. Once saved, this value is not echoed back to the screen in plaintext format. See LST Subscriber Hash and Encryption.
7. Type **done** when finished.

SIP authentication profile

To configure the Oracle Communications Unified Session Manager to utilize an LST, continuing from the previous step:

1. Type **exit** to return to the session router path.
2. Type **sip-authentication-profile** and press Enter.
3. Type **select** to choose the existing sip-authentication-profile configuration element you wish to use LST for authentication.

   ```
   ACMESYSTEM(sip-authentication-profile)# select
   <name>:
   1: name=sipAuthSMX1 digest-realm=acme.com credential-retrieval-method=loacl
   selection: 1
   ACMESYSTEM(sip-authentication-profile)#
   ```

4. **digest-realm**—Enter the digest realm used for authenticating here.
5. **credential-retrieval-method**—Set this parameter to **local** to use an LST.

6. **credential-retrieval-config**—Enter the name of the LST configuration you just configured.
7. Type **done** when finished.

LST Redundancy for HA Systems

LSTs must be synchronized between redundant nodes to ensure that the standby node contains identical LST files. You can either SFTP the same LST file to both the active and standby node, or you can use the synchronize command. The **synchronize** command is always executed from the active system. It copies the specified file from the active to the standby node placing the copy in the same file location on the standby node. Use the **synchronize lst** command as follows:

```
ACMESYSTEM# synchronize lst file.xml
```

> **Note:**
>
> The synchronize command does not reload the LST files.

Reloading the LST

After copying a new LST file to the Oracle Communications Unified Session Manager (and its standby peer), you can reload this newer file from the ACLI using the **refresh lst** command. For example:

```
ORACLE# refresh lst <local-subscriber-table name>
```

Using the **refresh lst** command selects the LST by name to refresh. Alternatively, saving and activating the configuration will reload the configuration as well and should be used when configuration parameters have also changed.

> **Note:**
>
> In an HA pair of Oracle Communications Unified Session Managers, you must independently execute the refresh lst command on both the active and standby systems.

LST File Compression

To save local disk flash space, you can compress the LST XML file using .gz compression. The resultant file must then have an .xml.gz extension.

LST File Format

The LST file format is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<localSubscriberTable encrypt-algo="aes-128-cbc">
<subscriber username="alice@apkt.com" hash="02:5E:78:D8:7E:75:A3:39"
encrypted="true"/>
<subscriber username="bob@apkt.com" hash="bc4b2a76b9719d911017c59"
encrypted="false"/>
```

```
<subscriber username="acme@apkt.com" hash="5d41402abc4b2a76b9719d9"
encrypted="false"/>
</localSubscriberTable>
```

The LST file's elements are as follows:

localSubscriberTable

This is the head element in the XML file. Each file can have only one head element. The following attribute is found in this element:

- encrypt-algo—This indicates the algorithm type used to encrypt the hash in the XML file. The key for this encryption will be a preshared key and is configurable in the local subscriber table configuration element with the secret parameter.
- The value in this element is for display purposes only.
- Currently AES-128-CBC is the only supported encryption algorithm.

subscriber

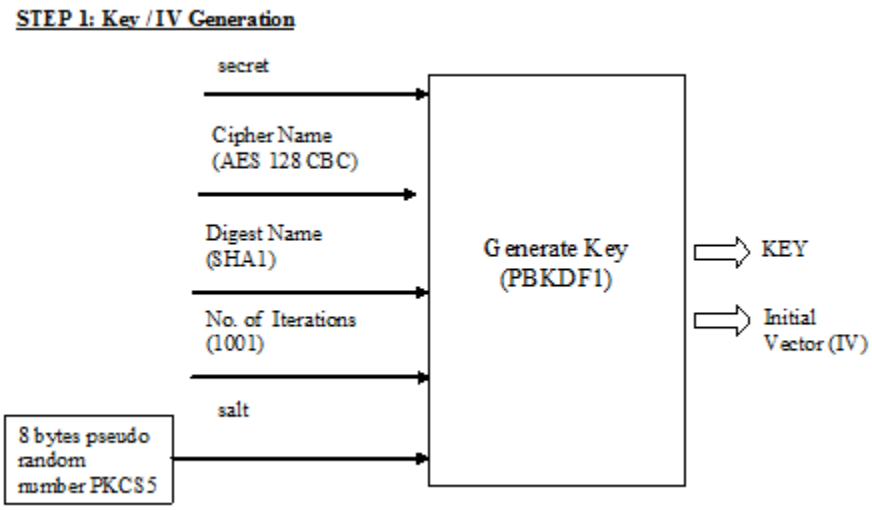
This element has the subscriber information. And has the following 3 attributes:

- username—The value given in the username attribute must be same as the username that will be sent in the Authorization Header in the Request message from the users. Refer RFC 2617 Http Authentication for details.
- hash—The hash provided in the XML must be an MD5 hash of the Username, digest-realm and the password of the user. This is same as the H(A1) described in RFC 2617. hash = md5(username:digest-realm:password)
- encrypted—The encrypted flag indicates if the "hash" given in the XML file is encrypted or not

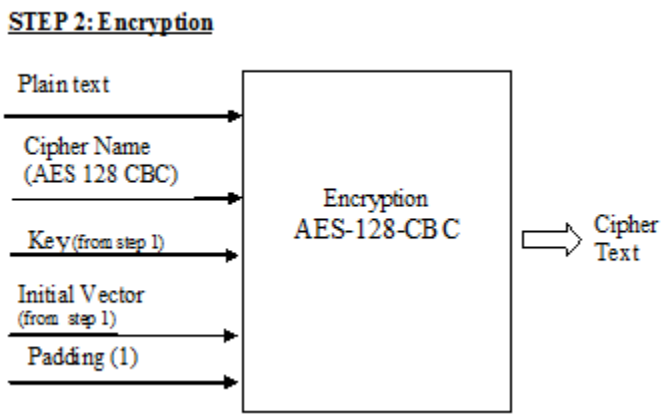
LST Subscriber Hash and Encryption

You may additionally use AES-128 CBC to encrypt the hash in the subscriber element in the LST XML file. The PSK used for encryption is configured in the **secret** parameter and an 8-byte pseudo random number is used as the salt. The LST file must set the encrypted attribute per subscriber element to true. To derive the final encrypted data you place in the XML file, three steps are performed according to the following blocks. The output of the last step, Formatting final Encrypted Data, is inserted into the LST files, subscriber element's hash value, when the encrypted attribute is set to true.

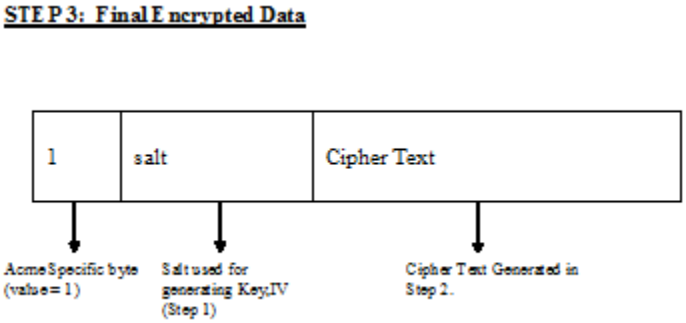
Key Initialization Vector



Encryption



Formatting final Encrypted Data

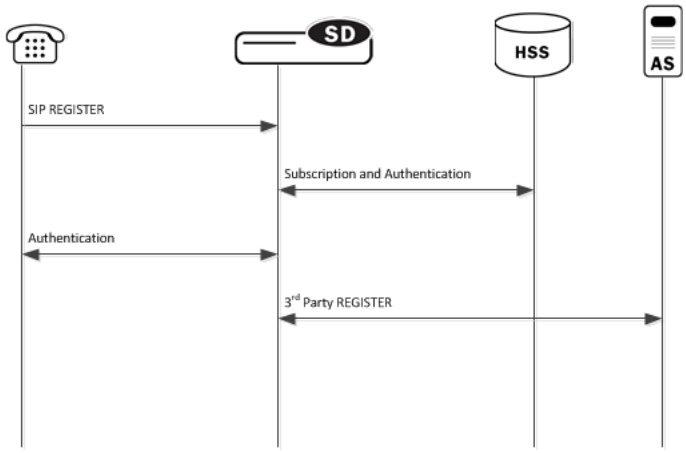


7

Third Party Registration

Third party registration support on the Oracle Communications Unified Session Manager provides a mechanism for sending registration information to a third party server. An IM (Instant Messaging) server might be the recipient of a third party REGISTER message.

The Oracle Communications Unified Session Manager accepts incoming REGISTER requests from UAs. After the UA has been registered with the Oracle Communications Unified Session Manager, the Oracle Communications Unified Session Manager sends a third party REGISTER message to a third party server.



The Oracle Communications Unified Session Manager supports third party registration via two methods:

- For scenarios in which UAs receive iFCs from the HSS and the Oracle Communications Unified Session Manager's default iFC configuration, the Oracle Communications Unified Session Manager generates third party registration requests and responses for matching triggers in its iFC evaluation.
  Some third party servers may want the UA's entire original request to the Oracle Communications Unified Session Manager and response from the Oracle Communications Unified Session Manager to the UA provided to them. The Oracle Communications Unified Session Manager supports these scenarios, in some cases requiring additional configuration.
- For scenarios in which the UA needs a third party registration that is not explicitly prescribed within iFCs, you can configure a third party server on the Oracle Communications Unified Session Manager and achieve third party registration support. For these configurations, the Oracle Communications Unified Session Manager attempts third party registration to those servers for all UAs that register via the applicable Oracle Communications Unified Session Manager registrar.

For both methodologies, you must configure all third party servers as session agents.

Third Party Registrations via iFCs

The Oracle Communications Unified Session Manager performs third party registrations based on the iFC downloaded for the user. If the filter criteria successfully evaluates to a third party server, a third party registration entry is dynamically added in the Oracle Communications Unified Session Manager. The dynamic entry is automatically deleted if there are no more registrations being handled for that third party registration host.

When third party registration is performed by iFCs, the Oracle Communications Unified Session Manager generates the registration messages as follows:

- The Contact: header is populated with the URI from the home server route configuration of the sip-registrar associated with the registration. If the home server route is left blank, the Oracle Communications Unified Session Manager uses the IP address of the egress interface.
- The From: header of the new REGISTER message is the same as the FROM in the original message.
- The To: header of the new REGISTER message is the the same as the TO in original message (AOR).

Embedded REGISTER

As an option within standard iFC third party registration support, the Oracle Communications Unified Session Manager supports 3GPP's methodology of embedding the original UE registration (and/or its response from the S-CSCF/Registrar) as a MIME body in the third party REGISTER sent from the S-CSCF to the third party server. This methodology, presented in 3GPP TS 23.218 and 29.228, uses an optional iFC extension ("IncludeRegisterRequest" and "IncludeRegisterResponse") that tells the third party server to expect the entire original REGISTER request and/or REGISTER 200OK in the mime of the third party REGISTER.

Implementation details for this methodology include the following:

- There may be further configuration required on the Oracle Communications Unified Session Manager.
- The Oracle Communications Unified Session Manager does not embed original registration requests or responses to any third party server outside its trust domain.
- The HSS or configured iFCs must be preconfigured for embedded third party registrations.

An HSS confiuration may not support the optional "IncludeRegisterRequest" and "IncludeRegisterResponse". For these cases, there is a Oracle Communications Unified Session Manager configuration option that allows you to control this inclusion, as follows:

- If the iFCs specify inclusion in an environment where you do not want it, you can set a registrar option to never include the original REGISTER
- If the iFCs do not specify inclusion in an environment where you want it, you can set a registrar option to always include the original REGISTER.

You can set these options for either the third party register, the 200 OK, or both.

ACLI Instructions - Third Party Registration via iFCs

Session Agent

To create a session agent to represent the third party server:

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 path.

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

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

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

4. **hostname**—Enter the name for this session agent.
5. **ip-address**—Enter the IP address for this session agent. This value must be the same as the registrar-host parameter in the third party regs configuration element to which this session agent definition corresponds.

   Continue configuring this session agent's parameters. Not all session agent functionality is applicable to the Oracle Communications Unified Session Manager.
6. Type **done** when finished.

SIP Registrar

Option to set the SIP Registrar to perform embedded REGISTRATION support for third party registration:

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 path.

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

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

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

4. Type **select** and choose the number of the pre-configured SIP registrar configuration element you want to configure.

   ```
   ORACLE(sip-registrar)# select
   name:
   1: registrar1
   selection:1
   ACMEPACKET(sip-registrar)#
   ```

5. **option +include-register-request**—Set this option to control SIP REGISTER embedding in the third party registration.

```
ORACLE(sip-registrar)#options +include-register-request=true
```

Set this option to true to always embed the original REGISTER in the third party registration.

In some cases, the include may already be specified by the iFCs, even though you do not want it used. In these cases, configure the option to false

```
ORACLE(sip-registrar)#options +include-register-request=false
```

6. **option +include-register-response**—Set this option to control SIP REGISTER 200 OK embedding in the third party registration the S-CSCF sends to the AS.

   ```
   ORACLE(sip-registrar)#options +include-register-response=true
   ```

   Set this option to true to always embed the original REGISTER in the third party registration 200 OK.

   In some cases, the include may already be specified by the iFCs, even though you do not want it used. In these cases, configure the option to false.

   ```
   ACMEPACKET(sip-registrar)#options +include-register-response=false
   ```

7. Type **done** when finished.

Third Party Registration via ACLI Configuration

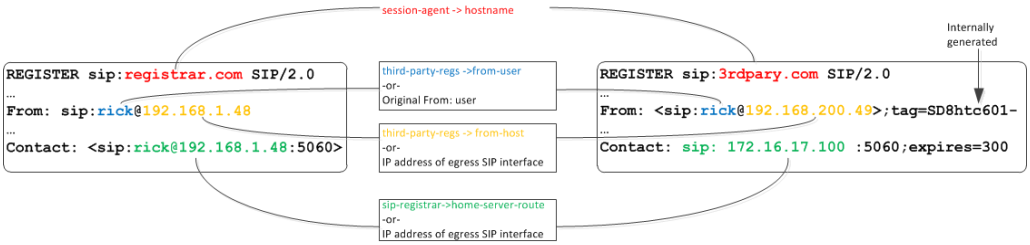
This section specifies the differences between Oracle Communications Unified Session Manager third party registration support via iFC as oppsed to via ACLI configuration.

As is true of the method described above, third party registration is generated by the Oracle Communications Unified Session Manager on behalf of the user in the To: header of REGISTER request.

When third party registration is generated by ACLI configuration on the Oracle Communications Unified Session Manager, the registration messages are generated as follows:

- The request URI of the new REGISTER message uses the value of the hostname parameter in the session agent configuration element.
- The From: header of the new REGISTER message uses the value of the from-user parameter in the third party regs configuration element as the user portion of the URI. If the from-user parameter is left blank, the Oracle Communications Unified Session Manager uses the user in the original From: header.
- The From: header of the new REGISTER message uses the value of the from-host parameter in the third party regs configuration element as the host portion of the URI. If the from-host parameter is left blank, the Oracle Communications Unified Session Manager uses the IP address of the egress SIP interface as the host portion of the from header.
- The Contact: header of the new REGISTER message uses the home server route parameter in the sip registrar configuration element. If the home server route parameter is left blank, the Oracle Communications Unified Session Manager uses the IP address of the egress interface.

See the following diagram:



Third Party Registration Server States

If the third party server does not respond to a REGISTER request, the Oracle Communications Unified Session Manager adheres to standard SIP session agent retransmission/ timeout procedures. If the third party server is set to out of service, the Oracle Communications Unified Session Manager attempts connectivity retry procedures. The retry procedures dictate that the Oracle Communications Unified Session Manager periodically send a REGISTER message to the third party server to check if connectivity has come back. The time interval for checking connectivity to a third party server is set with the retry interval parameter. Retries continue forever or until the third party server responds. The retry mechanism may be disabled by setting the retry interval parameter to 0.

> **Note:**
>
> When using the ACLI generated third party registration method, the time interval for checking connectivity to a third party server is set with the retry interval parameter in the third party regs configuration element.
> When a third party server is out of service, the Oracle Communications Unified Session Manager maintains a queue of outstanding third party registration requests. When the third party server returns to service, the Oracle Communications Unified Session Manager gracefully flushes the queue of outstanding requests. This prevents a registration flood from being directed at the third party server .

Third Party Registration Expiration

The REGISTER message sent from the Oracle Communications Unified Session Manager to the third party server uses the Expires: value returned from the User Subscriber Database or HSS. The third party server sends a 200 OK message containing Contact bindings and an expires value chosen by the third party server itself. The Oracle Communications Unified Session Manager checks each contact address to determine if it created it. For those addresses it created (as SD-Contacts), the Expires value from the 200 OK is used as the final value.

Once the expires timer has reached half the expires period as returned from the third party server, the Oracle Communications Unified Session Manager refreshes the registration.

If the third party server responds to a REGISTER Request with a 423 (Interval Too Brief) response, the Oracle Communications Unified Session Manager updates the contact's expiration interval to the Min-Expires value of the 423 response. It then submits a new REGISTER Request with the updated expires value.

Defining Third Party Servers

To send third party registrations that are generated via ACLI configuration to a third party server, three configuration elements are required. The primary configuration element is the third party regs. One or more may be configured in order to send the REGISTER message to multiple registration servers. You need to configure a name and set the state to enabled. The registrar host must be configured to indicate the value to insert into the Oracle Communications Unified Session Manager-generated request URI in the REGISTER message.

> **Note:**
>
> It is recommended that the list of third party registration servers be restricted to a maximum of 3.

A session agent needs to represent the third party server. Create a session agent as the third party server and note its name. Next, configure the registrar-host parameter with a session agent hostname in the third-party-reg configuration element. This specifies the session agent to be used as the registrar.

Finally, the address of the third party server must be added to the third-party-registrars parameter in the sip-registrar configuration element. This does not supercede any core Oracle Communications Unified Session Manager Registrar functionality. It informs the Oracle Communications Unified Session Manager of the third party server to send messages to after initial registration. Thus the value configured here must exist in the third-party-regs configuration element's registrar-host parameter list.

ACLI Instructions - Third Party Server Configuration

Recall that the configuration below is only required for scenarios in which the iFC does not explicity specify registration for the servers you configure below.

Third Party Registrar

To configure a third party server:

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 path.

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

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

   ```
   ORACLE(session-router)# third-party-regs
   ACMEPACKET(third-party-regs)#
   ```

4. **state**—Set this to enabled to use this configuration.
5. **registrar-host**—Set this value to the complementary session agents' hostname parameter to include those session agents as third party servers. This parameter may be modified like an options parameter. This value also appears in the request URI of the outgoing REGISTER message being sent to the third party server.

6. **from-user**—Configure this parameter to be the user portion of the From: header of the outgoing REGISTER message being sent to the third party server. Leaving this blank sets the user portion that in the original From: header
7. **from-host**—Configure this parameter to be the host portion of the From: header of the outgoing REGISTER message being sent to the third party server. Leaving this blank sets the host portion to the Oracle Communications Unified Session Manager's egress SIP interface.
8. **retry-interval**—Enter the number of seconds the Oracle Communications Unified Session Manager waits before retrying a third party server after a failed registration. Enter **0** to disable this feature.
9. Type **done** when finished.

SIP Registrar

To indicate to a local SIP Registrar when and what third party server to send third party registrations to:

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 path.

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

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

   ```
   ORACLE(session-router)# sip-registrar
   ACMEPACKET(sip-registrar)#
   ```

4. Type **select** and choose the number of the pre-configured SIP registrar configuration element you want to configure.

   ```
   ORACLE(sip-registrar)# select
   name:
   1: registrar1
   selection:1
   ACMEPACKET(sip-registrar)#
   ```

5. **home-server-route**—Enter the value to insert into the REGISTER message's request URI as sent to the third party server. Leaving this blank uses the AoR (or To: header) in the original REGISTER message.
6. **third-party-registrars**—Enter the name of a third party regs configuration element registrar-host parameter to send third part registrations associated with that SIP registrar.
7. Type **done** when finished.

8

References and Debugging

ACLI Configuration Parameters

The following sections describe the Oracle Communications Unified Session Manager's configuration parameters that are unique to S-CZ8.2.5.

sip-registrar

Parameters

name—Configured name of this sip registrar.

- Default: empty

state—Running status of this policy-director-group.

- Default: enabled
- Values: enabled | disabled

domains—List of registration domains that this Oracle Communications Unified Session Manager is responsible for. * means all domains. These domains are compared for an exact match with the domain in the request-uri of the REGISTER message. the wildcard '*' can also be entered as part of this parameter. This is entered as the domains separated by a space in quotes. No quotes required if only one domain is being configured. "+" and "-"are used to add to subtract from the list.

- Default: empty

subscriber-database-method—Protocol used to connect to User Subscriber Database server.

- Default: CX
- Values: CX | DDNS | local

subscriber-database-config—The configuration element that defines the server used for retrieving user subscriber data. For Cx deployments it is a home-subscriber-server name. For ENUM deployments it is an enum-config name.

- Default: empty

authentication-profile—Name of the sip-authentication-profile configuration used to retrieve authentication data when an endpoint is not authenticated.

- Default: empty

home-server-route—The value inserted into the Server Name AVP in an MAR message. This should be entered as a SIP URI as per 3gpp TS 24229 & RFC 3261. The host can be FQDN or IPv4 address, and the port portion should be in the 1025 - 65535 range. Examples: SIP: 12.12.12.12:5060

- Default: empty

third-party-registrars—The third-party-regs configuration element names where third party REGISTER messages will be forwarded to.

- Default: empty

routing-precedence—Indicates whether INVITE routing lookup should use the user database (via the registrar configuration element) or perform local policy lookup immediately.

- Default: registrar
- Values: registrar | local-policy

egress-realm-id—Indicates the default egress/core realm for SIP messaging.

- Default: empty

location-update-interval—Sets the maximum period in minutes in which the core-side user subscriber database is refreshed, per user.

- Default: 1440
- Values: 0-999999999

ifc-profile—References the ifc-profile configuration element's name that is applied to this sip-registrar.

max-contacts-per-aor—Limit to the number of contacts allowed for a given AOR.

- Default: 0 (disabled)
- Values: 1 - 256

ims-restoration—Enables the device to perform standards-based IMS restoration procedures with a compliant HSS deployment.

- Default: disabled
- Values: enabled | disabled

Path

This sip-registrar configuration element is a element in the session-router path. The full path from the topmost ACLI prompt is: **configure terminal**, and then **session-router**, and then **sip-registrar**.

sip-authentication-profile

Parameters

name—Configured name of this sip-authentication profile.

methods—List of SIP methods that prompt authentication. This is entered as the methods separated by a space in quotes. No quotes required if only one method is being configured. "+" and "-"are used to add to subtract from the list.

- Default: empty

anonymous-methods—List of SIP methods that prompt authentication when received from anonymous sources. This is entered as the methods separated by a space in quotes. No quotes required if only one method is being configured. "+" and "-" are used to add or subtract from the list.

- Default: empty

digest-realm—The value inserted into the digest-realm parameter in an authentication challenge header as sent to UA. (not used for Cx deployments)

- Default: empty

credential-retrieval-method—Protocol used to connect to the server providing authentication data.

- Default: ENUM-TXT
- Values: ENUM-TXT | CX

credential-retrieval-config—The home-subscriber-server name used for retrieving authentication data.

- Default: empty

Path

This sip-authentication-profile configuration element is a element in the session-router path. The full path from the topmost ACLI prompt is: **configure terminal**, and then **session-router**, and then **sip-authentication-profile**.

home-subscriber-server

Parameters

name—Configured name of this home subscriber server.

- Default: empty

state—Running status of this home subscriber server.

- Default: enabled
- Values: enabled | disabled

transport— The layer 4 protocol used to communicate with this home subscriber server.

- Default: tcp
- Values: tcp | sctp

address—This home subscriber server's IP address.

- Default: none
- Values: IP address in IPv4 or IPv6 format

port—This home subscriber server's port.

- Default: 80
- Values: 1-65535

realm—Oracle Communications Unified Session Manager realm-config name where this home subscriber server exists.

- Default: none

multi-homed-addrs— Specifies one or more local secondary addresses of the SCTP endpoint. This setting is only applicable to SCTP transport. To enter multiple addresses, bracket an address list with parentheses. At least one address is required if transport is set to SCTP.

Multi-homed addresses must be of the same type (IPv4 or IPv6) as that specified by the address parameter. Like the address parameter, these addresses identify SD physical interfaces.

origin-host-identifier—Used to create segment before the dot in the Origin Host AVP.

- Default: none

origin-realm—Populates the value of the Origin Realm AVP. Populates the segment after the dot in the Origin Host AVP.

- Default: none

destination-host-identifier—Used to create segment before the dot in the Destination Host AVP.

- Default: none

watchdog-ka-timer— The interval in seconds of the watchdog/keep-alive messages.

- Default: 0
- Values: 0-65535

num-auth-vector— The number of authentication vectors downloaded from the HSS per MAR.

- Default: 3
- Values: 1-10

Path

This home-subscriber-server configuration element is a element in the session-router path. The full path from the topmost ACLI prompt is: **configure terminal**, and then **session-router**, and then **home-subscriber-server**.

third-party-regs

Parameters

state—Running status of this third party registration configuration element.

- Default: enabled
- Values: enabled | disabled

name—Configured name of this third party registration configuration element.

- Default: none

registrar-host—hostname of the configured session agent that will be third party server. This value is also used in the request-uri that is sent to the third party server.

- Default: none

from-user—The user part of the From URI in the REGISTER Request that is sent to the third party server in the REGISTER message. When this parameter is blank the user part of the From header from the incoming REGISTER Request will be used.

- Default: none

from-host—The host part of the From URI in the REGISTER Request that is sent the third party server in the REGISTER message. When this parameter is blank the Oracle Communications Unified Session Manager uses the egress hostname/ IP address as the host.

- Default: none
- Values: Format this the same as the "registrar-host" in sip-config.

retry-interval—number of seconds the Oracle Communications Unified Session Manager waits before retrying a 3rd Party Registration server after a failed registration.

- Default: 32
- Values: 0 - 3600

Path

This third-party-regs configuration element is a element in the session-router path. The full path from the topmost ACLI prompt is: **configure terminal**, and then **session-router**, and then **third-party-regs**.

local-subscriber-table

Parameters

name—A given name for this local subscriber table element. This name is referenced from the sip-registrar configuration element when the **credential-retrieval-method** is set to **local**.

filename—The filename of local subscriber table that this element references. If no path is provided, the default location is /code/lst.

secret—PSK used for encrypted passwords. This value is not echoed back to the screen upon viewing the configuration element.

Path

The location of this configuration element is: configure terminatl > session-router > local-subscriber-table.

enum-config

Parameters

name—Name for this enum-config to be referenced from within the system.

top-level-domain—The domain extension used to query the ENUM servers for this configuration.

realm-id—The realm-id is used to determine on which network interface to issue an ENUM query.

enum-servers—List of IP address that service the top level domain.

service-type—The ENUM service types you want supported in this ENUM configuration. Possible entries are E2U+sip and sip+E2U (the default), and the types outlines in RFCs 2916 and 3721.

- Default: E2U+sip,sip+E2U

query-method—the ENUM query distribution strategy

- Default: hunt
- Values: hunt | round-robin

timeout—The total time, in seconds, that should elapse before a query sent to a server (and its retransmissions) will timeout.

- Default: 11

cacheInactivityTimer—Enter the time interval, in seconds, after which you want cache entries created by ENUM requests deleted, if inactive for this interval.

- Default: 3600
- Values: 0-999999999

max-response-size—The maximum size in bytes for UDP datagram responses

- Defaults: 512

health-query-number—The phone number for the ENUM server health query; when this parameter is blank the feature is disabled.

health-query-interval—The interval in seconds at which you want to query ENUM server health.

- Default: 0
- Values: 0-65535

failover-to—Name of the enum-config to which you want to failover.

cache-addl-records—Set this parameter to **enabled** to add additional records received in an ENUM query to the local DNS cache.

- Default: enabled
- Values: enabled | disabled

include-source-info—Set this parameter to enabled to send source URI information to the ENUM server with any ENUM queries.

- Default: disabled
- Values: enabled | disabled

ttl—This value ets the TTL value (in seconds) for NAPTR entries in the local ENUM cache and populates when sending a NAPTR entry to the ENUM server.

- Default: 0
- Values: 1-2592000

order—This parameter value populates the order field with when sending NAPTR entries to the ENUM server.

- Default: 1
- Values: 0-65535

preference—This parameter value populates the preference field with when sending NAPTR entries to the ENUM server.

- Default: 1
- Values: 0-65535

Path

This enum-config configuration element is a element in the session-router path. The full path from the topmost ACLI prompt is: **configure terminal**, and then **session-router**, and then **enum-config**.

ifc-profile

Parameters

name—A given name for this IFC profile element. This name is referenced from the sip-registrar configuration element's **ifc-support** parameter.

state—Running status of this IFC profile.

- Default: enabled
- Values: enabled | disabled

shared-ifc-filename—The name of the file referenced for shared IFC function.

default-ifc-filename—The name of the file referenced for default IFC function. This file may be the same as that used for the shared IFC function.

options—Identifies a set of features that vary depending on the configuration element in which they occur and that are enabled by invocation in the **options** parameter. Set the **options** parameter by typing "options", a Space, and then the option name preceded by a plus sign. If you type the option without the plus sign, you will overwrite any previously configured options. To append the new options to this configuration's options list, you must prefix the new option with a plus sign. Prefixing an option with a minus sign removes it from the list of options.

Path

The location of this configuration element is: **configure terminal**, and then **session-router**, and then **ifc-profile**.

regevent-notification-profile

Parameters

name—A given name for this registration event notification profile element. This name is referenced from the sip-registrar configuration element.

min-subscription-duration—The amount of time, in seconds, before the subscription expires, unless it is refreshed.

- Default: 3761 seconds
- Values: 180-6000005 seconds

Path

The location of this configuration element is: **configure terminal**, and then **session-router**, and then **regevent-notification-profile**.

hss-group

Parameters

name—Enter the name of the hss-group element. This required entry must follow the Name Format, and it must be unique.

state—Enable or disable the hss-group element.

- Default: enabled
- Values: enabled | disabled

origin-host-identifier—Set this to a string for use in constructing a unique Origin Host AVP.

strategy—Select the HSS allocation options for the hss-group. Strategies determine how HSSs will be chosen by this hss-group element.

- Default: hunt
- Values:
  - hunt—Selects HSSs in the order in which they are listed. For example, if the first server is online, all traffic is sent to the first server. If the first server is offline, the second server is selected. If the first and second servers are offline, the third server is selected. When the Oracle Communications Unified Session Manager detects that a higher priority HSS is back in service, it routes all subsequent traffic to that HSS.
  - roundrobin—Selects each HSS in the order in which they are listed in the dest list, selecting each HSS in turn, one per session. After all HSSs have been used, the first HSS is used again and the cycle continues.
  - failover—Selects the first sever in the list until failure is detected. Subsequent signaling goes to the next server in the list.

hss-configs—Identify the home-subscriber-servers available for use by this hss-group. This list can contain as many home subscriber servers as is necessary. An hss-config list value must correspond to a valid hss-group name in another group or to a valid hostname of a configured home-subscriber-server.

A value you enter here must correspond to a valid group name for a configured home-subscriber-server or a valid hostname or IP address for a configured home-subscriber-server.

hss-group is an element under the session-router path. The full path from the topmost ACLI prompt is: **configure terminal**, and then **session-router**, and then **session-group**.

Making Personal Data in Messaging Sent to OCOM Anonymous

When you allow people to examine SIP INVITE or SIP MESSAGE messages in the Oracle Communications Operations Monitor (OCOM), 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 OCOM.

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 OCOM remains available. When the troubleshooter views the SIP INVITE message, OCOM 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 OCOM.

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

Enabling Anonymization of Information Sent to OCOM

When you want to hide certain sensitive information in a SIP **INVITE** message that the Oracle Communications Operations Monitor (OCOM) can display, you can configure the Oracle Communications Unified Session Manager (OCUSM) 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.
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.

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.

SNMP MIBs and Traps

The following MIBs and traps are supported for the Oracle Communications Unified Session Manager. Please consult the Oracle Communications S-CX6.3.0 MIB Reference Guide for more SNMP information.

Acme Packet License MIB (ap-license.mib)

The following table describes the SNMP GET query names for the Oracle License MIB (ap-license.mib).

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apLicenseEntry (1.3.6.1.4.1.9148.3.5.1.1.1)		
apLicenseAuthFeature	apLicenseEntry: 1.3.6.1.4.1.9148.3.5.1.1.1.20	If authorization and authentication is allowed for the Oracle Communications Unified Session Manager, the value is true. If disabled, the value is false.
apLicenseDatabaseRegFeature	apLicenseEntry: 1.3.6.1.4.1.9148.3.5.1.1.1.21	If the Oracle Communications Unified Session Manager is configured as a registrar, the value is true. If registrar functionality is not enabled, this value is false.
apLicenseDatabaseRegCap	apLicenseEntry: 1.3.6.1.4.1.9148.3.5.1.1.1.22	The database registration contact capacity.

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

The following table describes the SNMP GET query names for the Oracle System Management MIB (ap-smgmt.mib).

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apSysMgmtMIBObjects (1.3.6.1.4.1.9148.3.2.1)		
Object Identifier Name: apSysMgmtGeneralObjects (1.3.6.1.4.1.9148.3.2.1.1)		
apSysSipStatsActiveDatabaseContacts	apSysMgmtGeneralObjects: 1.3.6.1.4.1.9148.3.2.1.1.24.0	Number of database-type contacts in the registration cache.

Enterprise Traps

The following table identifies the proprietary traps that Oracle Communications Unified Session Manager system supports.

Trap Name: OID	Description
apSysMgmtDatabaseRegCacheCapTrap: 1.3.6.1.4.1.9148.3.2.6.0.76	Generated when the number of database-type contacts stored in the registration cache exceeds the license threshold.
apSysMgmtDatabaseRegCacheCapClearTrap: 1.3.6.1.4.1.9148.3.2.6.0.77	Trap is generated when the number of database-type contacts stored in the registration cache falls below the license threshold.

Oracle USM Show Commands

show sipd endpoint-ip

The show sipd endpoint-ip <user | IP address> command displays information about each endpoint. For a supplied AoR, the Oracle Communications Unified Session Manager displays all associated contacts (both access and core side), the expiration of each contact entry and associated 3rd Party Registration information. For example:

```
ORACLE# show sipd endpoint-ip 11111
User <sip:111111@172.16.17.100>
  Contact exp=1198
    UA-Contact: <sip:111111@172.16.17.100:5060> UDP keep-acl
         realm=net172 local=172.16.101.13:5060 UA=172.16.17.100:5060
    SD-Contact: <sip:111111-s37q249kvluaa@192.168.101.13:5060> realm=net192
    Call-ID: 1-15822@172.16.17.100'
Third Party Registration:
   Third Party Reg User=<sip:111111@172.16.17.100> state: REGISTERED
   Expire Secs=298 seqNum= 1 refreshInterval=300
   Call-ID: d355a67277d9158e7901e46a12719663@192.168.101.13
   Third Party Reg User=<sip:111111@172.16.17.100> state: REGISTERED
   Expire Secs=178 seqNum= 1 refreshInterval=180
   Call-ID: 07ebbdebfdf64a48985bb82fa8b4c595@192.168.101.13
```

show sipd third-party

The show sipd third-party command displays the current status of third party servers and statistics for messages. The format is:

```
show sipd third-party <all | name>
```

The name argument allows status to be displayed for just the server specified by the name. Not specifying a name results in status being displayed for all third party servers. For example:

```
ORACLE# show sipd third-party-reg all
3rd Party Registrar   SA State  Requests  200OK  Timeouts    Errors
192.168.17.101            INSV         9      9         0         0
192.168.17.102            INSV        14     14         0         0
```

Column definitions are as follows:

- IP Address —IP Address of third party server
- Status —Session Agent State
- Requests —Register requests sent
- 200 OK —200 OK Responses received
- Timeouts —Requests timed out
- Error —Error Responses

show sipd local-subscription

The ACLI **show sipd** command includes an argument that provides information about local subscriptions, as shown below.

```
ORACLE# show sipd local-subscription
19:22:18-152
SIP Local Subscription Status  -- Period -- ----- Lifetime -------
                      Active  High   Total Total  PerMax    High
Server Subscription        0     1       1     1       1       1
Message Statistics
SUBSCRIBE
                    --------- Server --------   --------- Client --------
Message/Event       Recent      Total  PerMax   Recent      Total  PerMax
                    ------  ---------  ------   ------  ---------  ------
SUBSCRIBE Requests       2          2       2        0          0       0
Retransmissions          0          0       0        0          0       0
200 OK                   1          1       1        0          0       0
403 Forbidden            1          1       1        0          0       0
Response Retrans         0          0       0        0          0       0
Transaction Timeouts     -          -       -        0          0       0
Locally Throttled        -          -       -        0          0       0
Avg Latency=0.000 for 0
Max Latency=0.000
NOTIFY
                    --------- Server --------   --------- Client --------
Message/Event       Recent      Total  PerMax   Recent      Total  PerMax
                    ------  ---------  ------   ------  ---------  ------
NOTIFY Requests          0          0       0        2          2       2
Retransmissions          0          0       0       10         10      10
200 OK                   0          0       0        1          1       1
Transaction Timeouts     -          -       -        0          0       0
Locally Throttled        -          -       -        0          0       0
Avg Latency=0.000 for 0
Max Latency=0.000
```

You can extend upon this ACLI **show sipd** command to include an argument that provides information about registration event package traffic, as shown below.

```
ORACLE# show sipd local-subscription regevent
19:23:08-103
SIP Local Subscription Status -- Period -- ------ Lifetime -----
                      Active High   Total Total  PerMax    High
Server Subscription        0    1       1     1       1       1
Message Statistics
SUBSCRIBE
                    --------- Server --------   --------- Client --------
Message/Event       Recent      Total  PerMax   Recent      Total  PerMax
                    ------  ---------  ------   ------  ---------  ------
SUBSCRIBE Requests       2          2       2        0          0       0
Retransmissions          0          0       0        0          0       0
200 OK                   1          1       1        0          0       0
403 Forbidden            1          1       1        0          0       0
Response Retrans         0          0       0        0          0       0
Transaction Timeouts     -          -       -        0          0       0
Locally Throttled        -          -       -        0          0       0
Avg Latency=0.000 for 0
Max Latency=0.000
NOTIFY
                    --------- Server --------   --------- Client --------
Message/Event       Recent      Total  PerMax   Recent      Total  PerMax
                    ------  ---------  ------   ------  ---------  ------
NOTIFY Requests          0          0       0        2          2       2
Retransmissions          0          0       0       10         10      10
200 OK                   0          0       0        1          1       1
Transaction Timeouts     -          -       -        0          0       0
```

```
Locally Throttled        -          -       -        0          0       0
Avg Latency=0.000 for 0
Max Latency=0.000
```

The ACLI **show registration sipd** command includes an argument that provides information about a specific user's registration(s), as shown below.

```
ORACLE# show registration sipd by-user ra1 detailed
User: sip:ra1@apkt.com
  Registered at:  2013-06-05-19:23:40    Surrogate User: false
  Contact Information:
    Contact:
      Name: sip:ra1@apkt.com
      Valid: true
      Challenged: false
      Registered at: 2013-06-05-19:23:40
      Last Registered at: 2013-06-05-19:23:40
      Expire: 3581
      Local expire: 41
      Half: 1781
      Registrar IP: 0.0.0.0
      Transport: UDP
      Secure: false
      Local IP: 192.168.101.62:5060
      User Agent Info:
        Contact: sip:ra1@192.168.13.1:5060
        Realm: net192
        IP: 192.168.13.1:5060
      SD Info:
        Contact: sip:ra1-1cdstqjt90hve@172.16.101.62:5060
        Realm: net172
      Call-ID: 1-28361@192.168.13.1
    Associated URI(s):
      URI: sip:ra1@apkt.com
         Filter Criteria:
           Priority: 0
             Filter: None specified
             Application Server: sip:appserv@apkt.com
    Reg Event Subscriptions Terminated locally:
      Number of Subscriptions: 1
```

Subscriber: appserv<sip:appserv@apkt.com>;tag=1 state=active exp=600114

show registration

The show registration command displays cumulative statistics on all current registrations.

```
ORACLE# show registration
15:35:43-177
SIP Registrations        -- Period -- ----- Lifetime ------
                  Active  High   Total Total  PerMax    High
User Entries           0     0       0     0       0       0
Local Contacts         0     0       0     0       0       0
Via Entries            0     0       0     0       0       0
AURI Entries           0     0       0     0       0       0
Free Map Ports         0     0       0     0       0       0
Used Map Ports         0     0       0     0       0       0
Forwards               -     -       0     0       0       0
Refreshes              -     -       0     0       0       0
Rejects                -     -       0     0       0       0
```

```
Timeouts              -     -       0     0       0       0
Fwd Postponed         -     -       0     0       0       0
Fwd Rejected          -     -       0     0       0       0
Refr Extension        0     0       0     0       0       0
Refresh Extended      -     -       0     0       0       0
ContactsPerAor Reject -     -       0     0       0       0
Surrogate Regs        0     0       0     0       0       0
Surrogate Sent        -     -       0     0       0       0
Surrogate Reject      -     -       0     0       0       0
Surrogate Timeout     -     -       0     0       0       0
HNT Entries           0     0       0     0       0       0
Non-HNT Entries       0     0       0     0       0       0
Database Regs         0     0       0     0       0       0
DDNS Entries          0     0       0     0       0       0
CX Entries            0     0       0     0       0       0
LocalDB Entries       0     0       0     0       0       0
Unreg Users           0     0       0     0       0       0
```

You can extend upon the show registration command by adding the sipd by-user <username> detail arguments. The resulting output reflects user registration information including downloaded IFCs. For example:

```
ORACLE# show registration sipd by-user +19999092907 d
Registration Cache (Detailed View)    MON JUN 25 2012  13:47:46
User: sip:+19999092907@mobile.com
  Registered at: 2012-06-25-13:43:50     Surrogate User: false
  Contact Information:
    Contact:
      Name: sip:+19999092907@mobile.com
      Valid: true
      Challenged: false
      Registered at: 2012-06-25-13:43:50
      Last Registered at: 2012-06-25-13:47:30
      Expire: 48
      Local expire: 13
      Registrar IP: 0.0.0.0
      Transport: UDP
      Secure: false
      Local IP: 155.212.214.175:5060
      User Agent Info:
        Contact: sip:+19999092907@50.76.51.62:5762;transport=udp;acme_nat=
+19999092907+50.76.51.62@10.1.10.20:5762
        Realm: access
        IP: 50.76.51.62:5762
      SD Info:
        Contact: sip:+19999092907-rb8tulsbv3u72@108.108.108.108:5060
        Realm: core
      Call-ID: H_yvkgTAAA@10.1.10.20
    Associated URI(s):
      URI: sip:+19999092907@mobile.com
  Filter Criteria:
    Priority: 0
      Filter: ((case == 'Originating Registered') and (method == INVITE) and
('Accept-Contact'=='+g.app2app')) or
              ((case == 'Originating Registered') and (method == INVITE) and
('Contact'=='+g.app2app')) or
              ((case == 'Originating Registered') and (method == INVITE) and ('P-
Message-Auth'=='.*')) or
              ((case == 'Originating Registered') and (method == INVITE) and ('P-
Application-ID'=='.*'))
      Application Server: sip:pza.mobile.com:5280
```

```
Reg Event Subscriptions Received by Registrar:
Number of Subscriptions : 2
Subscriber: sip:appserv@192.168.13.1:5060; state=active; exp=59978
Subscriber: sip:pcscf@192.168.13.1:5060; state=active; exp=978
```

show home-subscriber-server

The show home-subscriber-server command displays cumulative statistics on all currently configured HSS servers.

show home-subscriber-server [stats <hss-name>| group group-name]

This command allows you to gather a set of information commonly requested by the Oracle TAC when troubleshooting customers.

The show home-subscriber-server command with no arguments displays the status of each HSS as well as the number of transactions and connections per HSS. For example:

```
ORACLE# show home-subscriber-server
Name                 Local-Address        Server-Address       Status
hss1                 192.168.207.21:45463 192.168.200.232:3872 Up
----------------------------------------
18:53:25-105
HSS Status                 -- Period -- -------- Lifetime --------
                  Active    High   Total      Total  PerMax    High
Client Trans           0       1       4      12152       8       1
Server Trans           0       0       0          7       2       1
Connections            1       1       0         53       2       1
```

Note that the Connections statistic indicates the number of connections after successful CER/CEA handshake.

The table below documents the states the

Field	Description
Active	This status is related to HSS failover and load balancing configurations. The diameter connection is up and being used.
Standby	This status is related to HSS failover and load balancing configurations. The diameter connection is up, but is not being used.
Pending	The Oracle Communications Unified Session Manager has sent a CER and is waiting for a CEA response.
Inactive	The Oracle Communications Unified Session Manager has sent a CER but has not received a CEA response.
Down	The Oracle Communications Unified Session Manager is not attempting to establish a connection with the HSS.

Oracle Communications Unified Session Manager reports on each HSS.

The show home-subscriber-server command with the stats argument displays the number of transactions and connections per HSS as well as the number of messages exchanged with all HSS servers per message type. For example:

```
ORACLE# show home-subscriber-server stats
veloster2# show home-subscriber-server stats
Name                 Local-Address        Server-Address       Status
hss1                 192.168.207.21:45463 192.168.200.232:3872 Up
----------------------------------------
18:55:03-103
```

```
HSS Status                -- Period -- -------- Lifetime --------
                 Active    High   Total      Total  PerMax    High
Client Trans          1       1       5      12157       8       1
Server Trans          0       0       0          7       2       1
Connections           1       1       0         53       2       1
                                        ---- Lifetime ----
                                  Recent      Total  PerMax
UAR                                    0          3       1
   SUBSEQ_REG (2002)                   0          3       1
SAR                                    0          6       3
   SUCCESS (2001)                      0          6       3
MAR                                    0          4       2
   SUCCESS (2001)                      0          4       2
LIR                                    0          1       1
   SUCCESS (2001)                      0          1       1
RTR                                    0          1       1
   SUCCESS (2001)                      0          1       1
PPR                                    0          1       1
   SUCCESS (2001)                      0          1       1
CER                                    0         55       3
   SUCCESS (2001)                      0         53       2
DWR                                    5      12088       5
   SUCCESS (2001)                      4      12041       5
   ERR_TIMEOUT                         0         46       1
DWR Recv                               0          5       2
   SUCCESS (2001)                      0          5       2
TCP Failures                           0        267       6
```

By entering the name of a specific HSS as an argument, the ACLI displays all HSS data for that server only. For example:

```
ACMESYSTEM# show home-subscriber-server stats hss1
```

The show home-subscriber-server command with the group argument displays the number of transactions and connections per the HSS group you specify in the command. For example:

```
ORACLE# show home-subscriber-server group hss-group1
display grp hss-group1
HSS Status                -- Period -- -------- Lifetime --------
                 Active    High   Total      Total  PerMax    High
Client Trans          0       0       0          0       0       0
Server Trans          0       0       0          0       0       0
Sockets               0       0       0          0       0       0
Connections           0       0       0          0       0       0
                          ---- Lifetime ----
                    Recent      Total  PerMax
UAR                      0          0       0
SAR                      0          0       0
MAR                      0          0       0
LIR                      0          0       0
RTR                      0          0       0
PPR                      0          0       0
Sent Requests            0          0       0
Sent Req Accepted        0          0       0
Sent Req Rejected        0          0       0
Sent Req Expired         0          0       0
Sent Req Error           0          0       0
Recv Requests            0          0       0
Recv Req Accepted        0          0       0
Recv Req Rejected        0          0       0
HSS  Errors              0          0       0
```

show http-server

The ACLI **show http-server** command provides basic OAuth information as shown below. The command without arguments displays basis statistics on all servers.

```
ORACLE# show http-server
Name               Server-Address              Status
sk                 host.httpsrv.com            Up
sk1                192.168.19.1:8886           Up
sk2                192.168.19.1:8887           Up
sk3                192.168.19.1:8889           Up
12:56:41-184
HTTP Status                 -- Period -- -------- Lifetime --------
                  Active    High   Total      Total  PerMax    High
Client Trans           0       0       0          0       0       0
Server Trans           0       0       0          0       0       0
Sockets                0       0       0          0       0       0
Connections            0       0       0          0       0       0
```

You can extend upon this command to get detailed global statistics by adding the stats argument to the end of this command.

```
ORACLE# show http-server stats
Name               Server-Address              Status
sk                 host.httpsrv.com            Up
sk1                192.168.19.1:8886           Up
sk2                192.168.19.1:8887           Up
sk3                192.168.19.1:8889           Up
HTTP Status                 -- Period -- -------- Lifetime --------
                  Active    High   Total      Total  PerMax    High
Client Trans           0       0       0          0       0       0
Server Trans           0       0       0          0       0       0
Sockets                1       1       1          1       1       1
Connections            1       1       1          1       1       1
                          ---- Lifetime ----
                     Recent      Total  PerMax
Sent Requests             0          0       0
Sent Req Accepted         0          0       0
Sent Req Rejected         0          0       0
Sent Req Expired          0          0       0
HTTP Errors               0          0       0
```

You can limit this output to a single server by appending the command with the name of that server.

```
ORACLE# show http-server stats http-server1
Name = http-server1
----------------------------------------
Server-Address            Status
192.168.19.1:8886            Up
----------------------------------------
12:56:41-184
HTTP Status                 -- Period -- -------- Lifetime --------
                  Active    High   Total      Total  PerMax    High
Client Trans           0       0       0          0       0       0
Server Trans           0       0       0          0       0       0
Sockets                0       0       0          0       0       0
Connections            0       0       0          0       0       0
---- Lifetime ----
```

```
                    Recent      Total  PerMax
Sent Requests            0          0       0
Sent Req Accepted        0          0       0
Sent Req Rejected        0          0       0
Sent Req Expired         0          0       0
HTTP Errors              0          0       0
```

Session Load Balancer Support

In order to rapidly increase the number of supported endpoints, the Oracle Communications Unified Session Manager can interoperate with the Oracle Communications SLB. When paired with an Oracle Communications SLB, the Oracle Communications Unified Session Manager maintains its ability to function with an HSS or ENUM database.

In addition, the Oracle Communications Unified Session Manager and Oracle Communications SLB pair supports Cx or ENUM based registrations.

To communicate with an Oracle Communications SLB, in addition to all baseline Oracle Communications SBC SIP functionality, the Oracle Communications Unified Session Manager advertises its registration capacity to the Oracle Communications SLB. This value is defined in the SIP interface as the reg cache limit parameter. Since the Oracle Communications Unified Session Manager has a database registrar license, the lower of the two will be advertised to the SLB.

Verify Config

The Oracle Communications Unified Session Manager performs application specific verification checks when you save a config with the save-config ACLI command. These checks are in addition to baseline Oracle Communications Unified Session Manager verification checks.

sip authentication profile (CX)

If session-router > sip-authentication-profile > credential-retrieval-method = CX then confirm

session-router > sip-authentication-profile > credential-retrieval-config value =

any existing session-router > home-subscriber-server configuration > name value

Error

If the above check fails:

1. A WARNING is displayed on the ACLI.
2. An INFO log message is generated.

sip authentication profile (ENUM)

If session-router > sip-authentication-profile > credential-retrieval-method = ENUM-TXT then confirm

session-router > sip-authentication-profile > credential-retrieval-config value =

any existing session-router > enum-config > name value

Error

If the above check fails:

1. A WARNING is displayed on the ACLI.
2. An INFO log message is generated.

sip authentication profile (Local)

If session-router > sip-authentication-profile > credential-retrieval-method = local then confirm

session-router > sip-authentication-profile > credential-retrieval-config =

session-router > local-subscriber-table > ame Error

If the above check fails:

1. A WARNING is displayed on the ACLI.
2. An INFO log message is generated.

sip-registrar

If session-router > sip-registrar > subscriber-database-method = DDNS then confirm

session-router > sip-registrar > subscriber-database-config value =

any existing session-router > enum-config > name value

Error

If the above check fails:

1. A WARNING is displayed on the ACLI.
2. An INFO log message is generated.

sip-registrar

If session-router > sip-registrar > authentication-profile is configured, then confirm its value is any existing:

session-router > sip-authentication-profile > name value

Error

If the above check fails:

1. A WARNING is displayed on the ACLI.
2. An INFO log message is generated.

Resource Utilization

The Oracle Communications Unified Session Manager limits resource utilization to maintain operational stability. Resources managed this way include:

- CPU
- Memory (heap)

CPU Overload Protection

CPU overload protection on the Oracle Communications Unified Session Manager is system-oriented in terms of defining the percent utilization that triggers an action. Actions are application-specific.

For the Oracle Communications Unified Session Manager application, if the CPU usage exceeds the configured setting, the system sends a 5xx error in response to any initial dialog request or standalone transactions. The Oracle Communications Unified Session Manager continues to accept registration refreshes and new transactions within a dialog.

> **Note:**
>
> An Oracle CSM configured to operation as an SLRM rejects all messages when CPU utilization exceeds this threshold.

By default the CPU utilization rate is 80%. This value can be changed by the following ACLI command sequence.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# sip-config
ORACLE(sip-config)# options +load-limit=”70”
ORACLE(sip-config)# done
```

Heap Utilization

The Oracle Communications Unified Session Manager limits memory utilization to maintain operational stability, as follows:

- When heap utilization exceeds the default (75%) or configured memory utilization threshold, the Oracle Communications Unified Session Manager no longer accepts new registrations. The Oracle Communications Unified Session Manager replies to these messages with 5xx messages. The Oracle Communications Unified Session Manager continues to accept registration refreshes, in-dialog calls and subscriptions.
- When heap utilization exceeds its default (90%) or configured threshold, the Oracle Communications Unified Session Manager drops all messages.

The user can change these thresholds to higher or lower values to best accommodate their operational environment. The user can also determine current memory utilization using the following command and referring to the heap utilization value, towards the bottom of the command’s output.

```
ORACLE# show platform heap-statistics
```

The user can change the first threshold, for example from its default of 75% to 80%, using the option shown below.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# sip-config
```

```
ORACLE(sip-config)# +options memory-overload-protect 80
ORACLE(sip-config)# done
```

The user can change the default drop-all threshold, from 90% to 85% for example, using the option shown below.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# sip-config
ORACLE(sip-config)# +options heap-threshold 85
ORACLE(sip-config)# done
```

A

USM Base Configuration Elements

This appendix presents base configuration settings required for the following USM deployment types:

- Cx
- ENUM
- LST

USM Base Configuration Elements for Cx

This appendix provides configuration samples of the elements that are required for minimal Oracle USM operation.

The SIP Config must be enabled

```
sip-config
        state                                   enabled
```

You must have a default gateway in your system-config

```
system-config
        default-gateway                          10.0.0.1
```

You must have an access physical interface

```
phy-interface
        name                                    s0p0
        operation-type                          Media
        port                                    0
        slot                                    0
```

You must have a core physical interface

```
phy-interface
        name                                    s0p1
        operation-type                          Media
        port                                    1
        slot                                    0
```

You must have an access network interfaces

```
network-interface
        name                                    s0p0
        sub-port-id                             0
        ip-address                              192.170.1.100
        netmask                                 255.255.255.0
        gateway                                 192.170.1.1
```

You must have a core network interfaces

```
network-interface
        name                                    s0p1
        sub-port-id                             0
        ip-address                              192.170.2.100
        netmask                                 255.255.255.0
        gateway                                 192.170.2.1
```

You must have an access realm

```
realm-config
        identifier                              access1
        addr-prefix                             0.0.0.0
        network-interfaces                      s0p0:0
```

You must have a core realm

```
realm-config
        identifier                              core1
        addr-prefix                             0.0.0.0
        network-interfaces                      s0p1:0
```

You must have an access SIP interface

```
sip-interface
        state                                   enabled
        realm-id                                access1
        sip-port
                address                                 192.170.1.100
                port                                    5060
                transport-protocol                      UDP
                allow-anonymous                         registered
        network-id                              My_Network_Name
        trust-mode                              none
        registration-caching                    enabled
        ims-access                              enabled
```

You must have a core SIP interface

```
sip-interface
        state                                   enabled
        realm-id                                core1
        sip-port
        address                                 192.170.2.100
```

You must have an ENUM Configuration

```
enum-config
        name                                    My_e164_cfg
        realm-id                                core1
        enum-servers                            192.170.2.201
```

You must have a Subscriber Database

```
home-subscriber-server
        name                                    My_HSS
        address                                 192.170.2.202
        realm                                   core1
```

You must have a Registration Event Profile

```
regevent-notification-profile
        name                                    My_reg_event_Profile
```

You must have an Authentication Profile

```
sip-authentication-profile
        name                                    My_Auth_Profile
        methods                                 REGISTER
        anonymous-methods                       *
        digest-realm                            My_Digest_Realm.com
        credential-retrieval-method             Cx
        credential-retrieval-config             My_HSS
```

You must have a Sip-Registrar

```
sip-registrar
        name                                    My_Registrar_Name
        domains                                 my_customer1.com
        subscriber-database-method              Cx
        subscriber-database-config              My_HSS
        authentication-profile                  My_Auth_Profile
        home-server-route                       sip:192.170.2.201:5060
        routing-precedence                      REGISTRAR
        egress-realm-id                         core1
        options                      e164-primary-config=enum:My_e164_Cfg
        regevent-notification-profile           My_reg_event_Profile
```

USM Base Configuration Elements for ENUM

This appendix provides configuration samples of the elements that are required for minimal Oracle USM operation.

The SIP Config must be enabled

```
sip-config
        state                                   enabled
```

You must have a default gateway in your system-config

```
system-config
        default-gateway                          10.0.0.1
```

You must have an access physical interface

```
phy-interface
        name                                    s0p0
        operation-type                          Media
        port                                    0
        slot                                    0
```

You must have a core physical interface

```
phy-interface
        name                                    s0p1
        operation-type                          Media
        port                                    1
        slot                                    0
```

You must have an access network interfaces

```
network-interface
        name                                    s0p0
        sub-port-id                             0
        ip-address                              192.170.1.100
        netmask                                 255.255.255.0
        gateway                                 192.170.1.1
```

You must have a core network interfaces

```
network-interface
        name                                    s0p1
        sub-port-id                             0
        ip-address                              192.170.2.100
        netmask                                 255.255.255.0
        gateway                                 192.170.2.1
```

You must have an access realm

```
realm-config
        identifier                              access1
        addr-prefix                             0.0.0.0
        network-interfaces                      s0p0:0
```

You must have a core realm

```
realm-config
        identifier                              core1
        addr-prefix                             0.0.0.0
        network-interfaces                      s0p1:0
```

You must have an access SIP interface

```
sip-interface
        state                                   enabled
        realm-id                                access1
        sip-port
                address                                 192.170.1.100
                port                                    5060
                transport-protocol                      UDP
                allow-anonymous                         registered
        trust-mode                              none
        registration-caching                    enabled
        ims-access                              enabled
```

You must have a core SIP interface

```
sip-interface
        state                                   enabled
        realm-id                                core1
        sip-port
        address                                 192.170.2.100
```

You must have an ENUM Configuration for e.164 translation

```
enum-config
        name                                    My_e164_cfg
        realm-id                                core1
        enum-servers                            192.170.2.201
```

You must have an ENUM Configuration for accessing ENUM server(s) as subscriber server(s)

```
enum-config
        name                                    My_ENUM_servers
        realm-id                                core1
        enum-servers                            192.170.2.203
        cacheInactivityTimer                    0
        max-response-size                       65535
```

You must have an Authentication Profile

```
sip-authentication-profile
        name                                    My_Auth_Profile
        methods                                 REGISTER
        anonymous-methods                       *
        digest-realm                            My_Digest_Realm.com
        credential-retrieval-method             enum-text
        credential-retrieval-config             My_ENUM_servers
```

You must have a Sip-Registrar

```
sip-registrar
        name                                    My_Registrar_Name
        domains                                 my_customer1.com
        subscriber-database-method              DDNS
        subscriber-database-config              My_ENUM_servers
        authentication-profile                  My_Auth_Profile
        home-server-route                       sip:192.170.2.201:5060
        routing-precedence                      REGISTRAR
        egress-realm-id                         core1
        options                      e164-primary-config=enum:My_e164_Cfg
```

USM Base Configuration Elements for LST

This appendix provides configuration samples of the elements that are required for minimal Oracle USM operation.

The SIP Config must be enabled

```
sip-config
        state                                   enabled
```

You must have a default gateway in your system-config

```
system-config
        default-gateway                          10.0.0.1
```

You must have an access physical interface

```
phy-interface
        name                                     s0p0
        operation-type                           Media
        port                                     0
        slot                                     0
```

You must have an access network interfaces

```
network-interface
        name                                     s0p0
        sub-port-id                              0
        ip-address                               192.170.1.100
        netmask                                  255.255.255.0
        gateway                                  192.170.1.1
```

You must have an access realm

```
realm-config
        identifier                               access1
        addr-prefix                              0.0.0.0
        network-interfaces                       s0p0:0
```

You must have an access SIP interface

```
sip-interface
        state                                    enabled
        realm-id                                 access1
        sip-port
                address                                  192.170.1.100
                port                                     5060
                transport-protocol                       UDP
                allow-anonymous                          registered
        trust-mode                               none
        registration-caching                     enabled
        ims-access                               enabled
```

You must have a Local Subscriber Table

```
local-subscriber-table
        name                                     My_LST
        filename                                 /code/lst/My_LST.xml
        secret                                   *******
```

You must have an Authentication Profile

```
sip-authentication-profile
        name                                     My_Auth_Profile
        methods                                  REGISTER
        anonymous-methods                        *
        digest-realm                             My_Digest_Realm.com
        credential-retrieval-method              local
        credential-retrieval-config              My_LST
```

You must have a Sip-Registrar

```
sip-registrar
        name                                    My_Registrar_Name
        domains                                 my_customer1.com
        subscriber-database-method              LOCAL
        subscriber-database-config              My_LST.xml
        authentication-profile                  My_Auth_Profile
        routing-precedence                      REGISTRAR
```

B

Caveats and Known Issues

This chapter lists the caveats and known issues for this release. Oracle updates this Release Notes document to distribute issue status changes. Check the latest revisions of this document to stay informed about these issues.

Known Issues

This table lists the Oracle Communications Unified Session Manager (OCUSM) known issues in version CZ8.2.0. You can reference known issues by Service Request number and you can identify the issue, any workaround, when the issue was found, and when it was fixed using this table. Issues not carried forward in this table from previous Release Notes are not relevant to this release. You can review delivery information, including defect fixes in this release's Build Notes.

ID	Description	Severity	Found In
29815940	An OCUSM HA pair crashes when you peform configuration changes that include updating and/or removing certificates within a **certificate-record** element. The process that crashes is atcpd.	2	S-CZ7.3.5
26895359	While processing multiple REGISTERs for a single AOR with different contacts that arrive within a few seconds of each other, the OCUSM may lose one or more of those contacts. The issue occurs because of a race condition in the correlation of the NAPTR answers received from the DNS server for the user contact record updates.	3	S-CZ7.3.5
28873421	It can take the OCUSM more than 7 minutes to clear a single session when it is supporting 5000-6000 active concurrent sessions. This issue is caused by unusually large configurations. Instead of routing a message via local policy, the OCUSM incorrectly issues an LIR when the following two conditions exist simultaneously: • The OCUSM is not configured with the e164-primary-config and e164-secondary-config options, and • The OCUSM receives a request with a tel-URI or a sip-URI with the user=phone parameter. Note that the OCUSM sends the request via local-policy if the LIA for a tel-URI or sip-URI with user=phone returns 5001 DIAMETER_ERROR_USER_UNKNOWN. For all other errors in the LIA, the OCUSM returns an error.	3	S-CZ7.3.5

Known Issues Inherited from the S-CZ8.2.0 SBC

Refer to the Known Issues in the S-CZ8.2.0 OCSBC Release Notes to complete your review of issues in this release. Issues withinin the OCSBC, especially including applicable Acme Packet platform, lower-level system issues, and applicable application issues apply across the S-CZ8.2.x product versions, including the OCUSM.

Caveats and Limitations

The following information lists and describes the caveats and limitations for this release. Oracle updates this Release Notes document to distribute issue status changes. Check the latest revisions of this document to stay informed about these issues.

OCUSM Caveats

- Do not load configurations from sibling products, the Oracle SBC for example, on the Oracle Communications Unified Session Manager (OCUSM). Those configurations are incompatible with the OCUSM, causing incorrect operation. Users should configure the OCUSM from scratch or use another valid OCUSM configuration.
- Multi-stage routing does not work for S-CSCF routing functions.

The OCUSM accepts only the first message received from an application server in response to messages from the OCUSM that included an ODI. If it receives subsequent messages from that application server, the OCUSM drops the reused ODI and processes the message as if they were received without an ODI.

- Resolution - Do not configure an AS to fork responses to the OCUSM that include an ODI originally provided by the OCUSM.
- Geo-redundancy is currently not supported by the OCUSM.

Caveats Inherited from the S-CZ8.2.0 SBC

Refer to the Caveats in the S-CZ8.2.0 OCSBC Release Notes to complete your review of issues in this release. Issues withinin the OCSBC, especially including applicable Acme Packet platform, lower-level system issues, and applicable application issues apply across the S-CZ8.2.x product versions, including the OCUSM.