Oracle® Enterprise Communications Broker

User's Guide

Release P-Cz5.0.0
G30548-01
June 2025

Oracle Enterprise Communications Broker User's Guide, Release P-Cz5.0.0

G30548-01

Contents

About This Guide

My Oracle Support xiv

1 Oracle Enterprise Communications Broker Overview

Packet Processing by the Oracle Enterprise Communications Broker 1-2
- Ingress INVITE Processing 1-4
  - Identifying Source Context 1-4
  - Dial Plan Processing 1-5
  - Route Engine Processing 1-5
- Egress Processing 1-5
- Call Handling Example 1-5

2 SIP Signaling Management

The Oracle Enterprise Communications Broker Dial Plan 2-1
Oracle Enterprise Communications Broker Contexts 2-1
- Context Hierarchy 2-2
  - Geographic Contexts 2-2
  - Corporate Contexts 2-3

Oracle Enterprise Communications Broker Agents 2-4
- Why You Need Agents 2-4
- How to Use Agents 2-4
- Agent Groups 2-5

Oracle Enterprise Communications Broker Routing 2-6
- Recursive Routing 2-6
- Identifying Contacts and Specifying Routes 2-8
- Route Selection 2-9
- Forking 2-9
- Fork Groups 2-10
  - Fork Group Assignment 2-11
  - Additional Targets 2-11
  - Fork Groups Configuration Considerations 2-12
- Routing and ENUM 2-14

Route Types and Precedence 2-14
Active Directory and Communications Broker Routing 2-15
LDAP and Oracle Enterprise Communications Broker Routing 2-16
LDAP Messages 2-20
LDAP Failure Events 2-20
Limitations using LDAP 2-21
Configuring LDAP for Routing 2-21
Source Routing 2-22
REFER Source Agent Routing 2-23

3 Registrar and Authentication

Register Refresh 3-2
Proxy Registration 3-2
Message Authentication for SIP Requests 3-2
Authentication 3-3
SIP Authentication Challenge 3-4
Authentication Header Elements 3-4
SIP Authentication Response 3-4
Authentication Check 3-4
Retrieving Information from Active Directory 3-4
LDAP and Authentication 3-5
Configuring LDAP for Authentication 3-5

4 Getting Started

Accessibility Features 4-1
Configuring the Communications Broker for SDM 4-1
User and Administrator Access 4-3
Simultaneous Logons 4-3
RADIUS Server Roles and Access Privileges 4-3
Log on to the Web GUI 4-4
Login Banner 4-5
Log Off from the Web GUI 4-6
Service Provisioning 4-6
Service Provisioning Configuration Objects 4-6
Web GUI Tools 4-7
The User Menu 4-7
Help 4-7
Help Topics 4-7
About This Software 4-8
Configuration Search Tool 4-9

Show Configuration for a Configuration Element 4-11
Customize the Page Display 4-11
Tool Tips 4-12
Configuration Tools and Behavior 4-13
Set System Operations 4-17
Set Boot Parameters 4-18
Set Entitlements Procedure 4-19
Set Initial Configuration Procedure 4-19
Set License Procedure 4-21
Set Login Banner Procedure 4-22
Set Time Zone Procedure 4-22
Upgrade Software Procedure 4-22
Edit, Copy, and Delete Configurations 4-23
Configuration States and Behavior 4-23
Configuration Error Messages 4-24
Uploading and Downloading Configuration File Elements 4-25
Using Tag Fields 4-25
Dashboard Tab Operations 4-26
Add a Widget to the Dashboard 4-28
Configure the Data Sampling Settings for a Dashboard Widget 4-29
View A Widget That is Not on the Dashboard 4-29
Widgets Page 4-30
Widgets and Descriptions 4-31
Command Line Interface Widgets 4-45
Licenses Widget 4-46
Display and Clear Alarms 4-47
Display Users 4-47

5 Agent Configuration

Alphanumeric User Database and Call Routing Entries 5-1
About Traffic Constraints 5-4
In-Service Response Codes for a Session Agent 5-5
Add a Session Agent 5-6
Configure a Session Agent Group 5-14
Configure ENUM Servers 5-14
Enable ENUM Session Agent Group Matching 5-16
Multi-Hop Agent Ping 5-17

6 Dial Plan Configuration

Dial Pattern Configuration 6-1

Dial Pattern Encoding Characters 6-2
Configure a Dial Plan 6-3

7 User Configuration

Configure a User Entry 7-1
Resolving to the Longest Match in the User Database 7-2

8 Using Policy to Refine Routing

The Redirect Action 8-2
Configuring CNAM Replacement 8-3
Using Policy to Normalize SIP Headers 8-3
ANI Masking 8-4
ANI Masking Configurations 8-4
Enabling Policy-based Routing 8-5
Policy Priority 8-5
Define a Policy 8-6
Applying a Policy to a Route 8-10
Runtime Routing with Policies in the User Table 8-10
Applying a Policy to the Registrar 8-11
Configurations Using Policy 8-11
Priority Call Handling 8-11
Priority Call Configurations 8-12
Transcoding and the Oracle Enterprise Communications Broker 8-12
Transcoding Configurations 8-12
Multiple Outbound Translations 8-13
Outbound Translation Configurations 8-13
Routing Action Configurations 8-14
Deny Route Policy Configurations 8-14
Stop Recurse Route Policy Configurations 8-15
Stop Recursion by SIP Response Code 8-16
Skip Route Policy Configurations 8-16

9 Routing Configuration

Routing Fields 9-1
Route Policy 9-2
Deny Patterns in Route Parameter Syntax 9-3
Loop Sensing for PSTN Calls 9-4
Configure Loop Sensing for PSTN Calls 9-4

10 Registrar Configuration

Registrar Configuration Fields 10-1
Local Subscriber Table 10-1
LST Configuration 10-2
- Configure the Registrar with an LST 10-2
- Add an LST File 10-3

LST Runtime Execution 10-4
LST Redundancy for HA Systems 10-4
LST File Compression 10-4
LST File Format 10-4
- localSubscriberTable 10-5
- subscriber 10-5

LST Subscriber Hash and Encryption 10-5

11 LDAP Client Configuration

LDAP Configuration Options 11-1
Making Settings 11-2
Configure LDAP Server Access 11-2
LDAP Groups 11-5
- Matching Criteria in LDAP Groups 11-6
- Configure LDAP Groups 11-6

Configure a Routing Query 11-7
Address of Record Configuration Fields 11-9
SIP Authentication Query Configuration Fields 11-11
Replacing the Calling Number in the FROM Header 11-11

12 Denial of Service Protection and ACLs

Levels of DoS Protection 12-1
About the Process 12-2
- Trusted Path 12-2
  - Address Resolution Protocol Flow 12-3
- Untrusted Path 12-3
- IP Fragment Packet Flow 12-4

Host Path Protection Process 12-4
- Access Control for Hosts 12-4
- Access Control Endpoint Classification Capacity and DoS 12-4
- Static and Dynamic ACL Entry Limits 12-5
- Host Path Traffic Management 12-5
  - Traffic Promotion 12-5

Malicious Source Blocking 12-5
Blocking Actions 12-5

13 ACL Configuration

Configuration Overview 13-1
Configure an ACL 13-1
Access Control for a Realm 13-4
Changing the Default Oracle Enterprise Communications Broker Behavior 13-5
Example 1 Limiting Access to a Specific Address Prefix Range 13-5
Example 2 Classifying the Packets as Trusted 13-6
Example 3 Installing Only Static ACLs 13-6

14 ECB Sync

Synchronizing the Registration Cache 14-3
ECB Sync Operations 14-3
Add Sync Agent 14-3
Enable Sync Config Settings 14-4
Delete a Sync Agent 14-4
ECB Sync Monitoring 14-5

15 HMR Configuration

SIP Header Manipulation 15-1
Multi-Hop SIP Header Manipulation Rules 15-1
SIP Header Manipulation Configuration Dialogs 15-2
SIP Header Manipulation Rules Attributes and Values Reference 15-4
SIP Header Manipulation Configuration 15-8
Configure SIP Manipulation 15-8
Configure a SIP Manipulation Header Rule 15-9
Configure a MIME Rule 15-11
Configure a MIME ISUP Rule 15-13
Configure a MIME SDP Rule 15-16

16 Monitor/Trace and Widgets Tab Operations

Monitor and Trace 16-1
SIP Notable Events Summary 16-3
Display a Notable Events Report 16-4
SIP Registrations Summary 16-4
Display a Registrations Report 16-5

SIP Sessions Summary 16-5
Display a Sessions Report 16-6
SIP Subscriptions Summary 16-7
Display a Subscriptions Report 16-9
Ladder Diagrams and Display Controls 16-9
Display a Ladder Diagram 16-12
Session Summary 16-12
QoS Statistics 16-14
SIP Monitor and Trace Filter Configuration 16-15
Search for a Record 16-17
Search for a Report Record 16-17
Specify Additional Identifiers 16-20
Specify Additional Search Options 16-21
Export Monitor and Trace Information to a Text File 16-21
Export Report Information to a Text File 16-22
Widget Access and Behavior 16-22

17 Troubleshooting and Maintenance

Audit Logs 17-1
Secure FTP Push Configuration 17-4
Configure Secure FTP Push with Public Key Authentication 17-4
Generate an RSA Public Key 17-5
Generate a DSA Public Key 17-5
Import a DSA Public Key 17-5
Copy the RSA Public Key to the SFTP Server 17-6
Configure Audit Logging 17-6
System File Management 17-8
Upload a File 17-10
Download a File 17-11
Delete a File 17-12
Back Up a Configuration File 17-12
Restore a Configuration File 17-13
CSV Configuration File Creation 17-13
Create a CSV Configuration File 17-15
Automatically Upload Updated CSV Configuration Files 17-16
Enable Automatic CSV Configuration File Uploads 17-18
Upgrade Software 17-18
System Restart 17-19
Display Log Files 17-20
Display System Health 17-20

Obtain Support Information - Firefox Browser 17-20

18 Active Directory Modifications

19 Configuration Examples

Configuration Sequence 19-1
Initial Agent Configuration 19-2
Dial Plan Strategies 19-3
Route Strategies 19-3
Small Enterprise Model 19-5
Large Enterprise Model - v2 19-6
Emergency Dial Configurations 19-8
Alternate Translation Modes 19-9
ENUM Example Configuration 19-10

20 Format of Exported Text Files

Exporting Files 20-1
Session Summary Exported Text File 20-2
Session Details Exported Text File 20-3
Ladder Diagram Exported HTML File 20-9

21 SIP HMR (Header Manipulation Rules)

Guidelines for Header and Element Rules 21-2
Splitting and Joining Headers 21-2
Precedence 21-2
Duplicate Header Names 21-3
Performing HMR on a Specific Header 21-3
Multiple SIP HMR Sets 21-3
MIME Support 21-4
Manipulating MIME Attachments 21-4
Escaped Characters 21-5
New Reserved Word 21-5
About the MIME Value Type 21-6
Back Reference Syntax 21-7
Notes on the Regular Expression Library 21-7
SIP Message-Body Separator Normalization 21-7
SIP Header Pre-Processing HMR 21-8
Best Practices 21-8

About Regular Expressions 21-9
Expression Building Using Parentheses 21-10
Configuration Examples 21-10
Example 1 Removing Headers 21-11
Example 2 Manipulating the Request URI 21-12
Example 3 Manipulating a Header 21-13
Example 4 Storing and Using URI Parameters 21-14
Example 5 Manipulating Display Names 21-15
Example 6 Manipulating Element Parameters 21-17
Example 7 Accessing Data from Multiple Headers of the Same Type 21-20
Example 8 Using Header Rule Special Characters 21-21
Example 9 Status-Line Manipulation 21-23
Example 10 Use of SIP HMR Sets 21-25
Example 11 Use of Remote and Local Port Information 21-26
Example 12 Response Status Processing 21-27
Example 13 Remove a Line from SDP 21-29
Example 14 Back Reference Syntax 21-30
Example 15 Change and Remove Lines from SDP 21-31
Example 16 Change and Add New Lines to the SDP 21-32
Dialog-Matching Header Manipulation 21-33
About Dialog-Matching Header Manipulations 21-33
Inbound HMR Challenge 21-34
Outbound HMR Challenge 21-34
Built-In SIP Manipulations 21-35
Unique HMR Regex Patterns and Other Changes 21-35
Manipulation Pattern Per Remote Entity 21-35
Reject Action 21-36
SNMP Support 21-37
Log Action 21-38
Name Restrictions for Manipulation Rules 21-38
New Value Restrictions 21-39
Header Manipulation Rules for SDP 21-39
SDP Manipulation 21-39
sdp-session-rule 21-40
sdp-media-rule 21-40
sdp-line-rule 21-43
Regular Expression Interpolation 21-44
Regular Expressions as Boolean Expressions 21-45
Moving Manipulation Rules 21-47
Rule Nesting and Management 21-48
ACLI Configuration Examples 21-48
Remove SDP 21-48

Remove Video from SDP 21-49
Add SDP 21-49
Manipulate Contacts 21-50
Remove a Codec 21-50
Change Codec 21-51
Remove Last Codec and Change Port 21-52
Remove Codec with Dynamic Payload 21-53
HMR Import-Export 21-54
Exporting 21-54
Importing 21-55
Using SFTP to Move Files 21-55

About This Guide

The *Oracle® Enterprise Communications Broker User's Guide* provides the following information about the Oracle Enterprise Communications BrokerCommunications Brokerhardware and software.

- Configuration examples
- Configuring SIP signaling management
- Configuring dial plans, agents, users, policies, registrars, LDAP, ECB sync, Header Manipulation Rules, and routing
- Maintenance and troubleshooting

Oracle Accessibility

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

Documentation Set

The following table describes the documentation set for the Communications Broker.

Document Name	Document Description
Administrator's Guide	Describes how to deploy the system.
Embedded Help system	Contains task-oriented topics for configuring, administering, maintaining, and troubleshooting the hardware and software.
Release Notes	Contains information about the current release, including specifications, requirements, new features, enhancements, inherited features, known issues, caveats, and limitations.
SBC Family Security Guide	Provides information about security considerations and best practices from a network and application security perspective for the Enterprise family of products.
User's Guide	Describes how to configure SIP signaling management and how to tailor the system to specific needs.

Related Documentation

The following table describes related documentation for the Communications Broker.

Document Name	Document Description
ACLI Reference Guide	Contains explanations of how to use the ACLI, as well alphabetical listings and descriptions of all ACLI commands and configuration parameters.

Document Name	Document Description
Administrative Security Essentials Guide	Contains conceptual and procedural information for supporting the Admin Security and Admin Security with ACP feature sets.

Revision History

The following table lists changes to this document and the corresponding dates of publication.

Date	Description
June 2025	Initial Release for Communications Broker Release 5.0.0.

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.

Access to Oracle Support

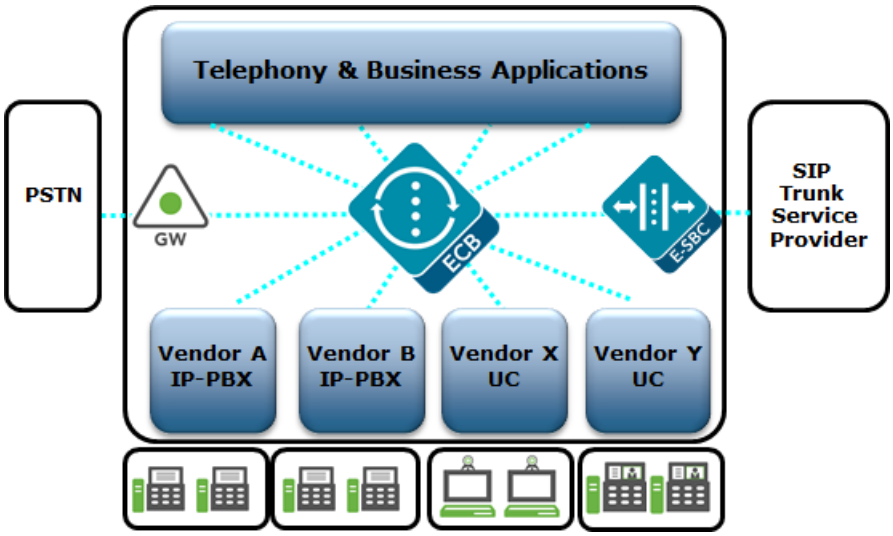
Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1

Oracle Enterprise Communications Broker Overview

The Oracle Enterprise Communications Broker (Communications Broker) is an enterprise-class, core signaling component designed to simplify communications networks. It combines innovative approaches toward dial plan management and SIP topology-aware routing with a purpose-built, Graphical User Interface. While at its best in signaling environments comprised of products and solutions from multiple vendors, it is useful for consolidating policy enforcement decisions, integrating third-party applications, and managing a network-wide routing topology even in homogenous architectures.

The Communications Broker is typically deployed in the core of a multi-vendor communications network where multiple UC, PBX, and service provider trunk interfaces must be interconnected. It normalizes communications between disparate premise-based systems and connects them to service provider networks and hosted applications through Enterprise Oracle Session Border Controllers.



Key benefits include:

- Increases scalability and simplicity
- Protects and extends investments in legacy communications infrastructure
- Reduces operations expenses
- Improves network availability
- Services and Applications

Communications Broker operational functionality focuses around the following:

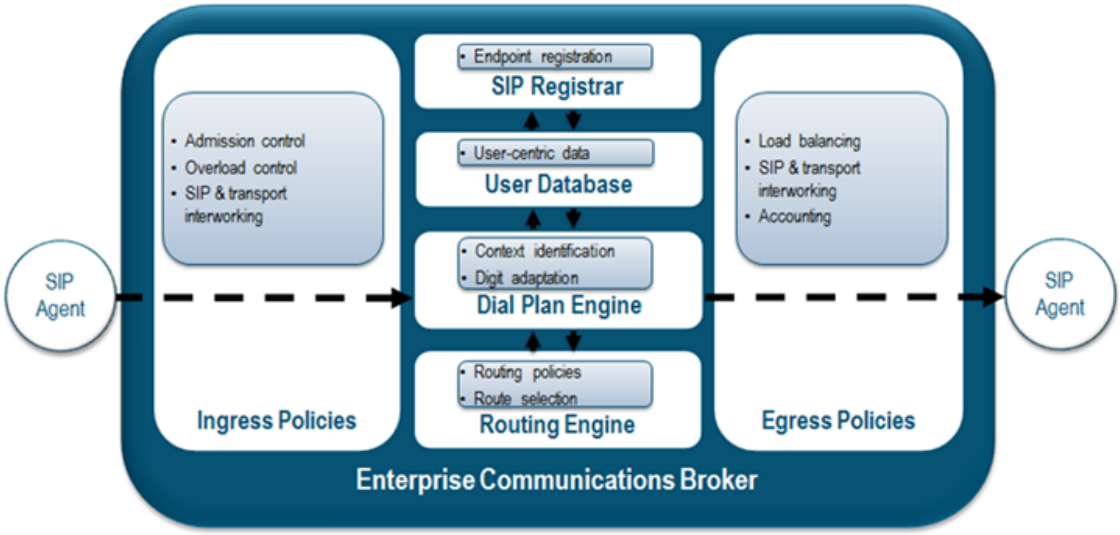
- SIP Signaling Management—The functional components of the Communications Broker's software architecture for SIP signaling management focus around its dial plan, and its routing engine. These two components represent the foundation of the Communications Broker's core SIP processing engine, and were specifically crafted to address, in a generic sense, the problems arising from the organic evolution of SIP-based enterprise communications networks.

- SIP Registrar—Provides a centrally-deployed location service for the enterprise.
- User Authentication—Provides for operation with an internal or external authentication resource, such as Active Directory, for authorization and authentication of users registering at the Communications Broker.
- Header Manipulation—Provides telephony engineering with a means of assembling signaling header information specifically for the enterprise's operations, conformance and interoperability needs.

The *Oracle® Enterprise Communications Broker User Guide* provides operational explanations and configuration instructions for each of these.

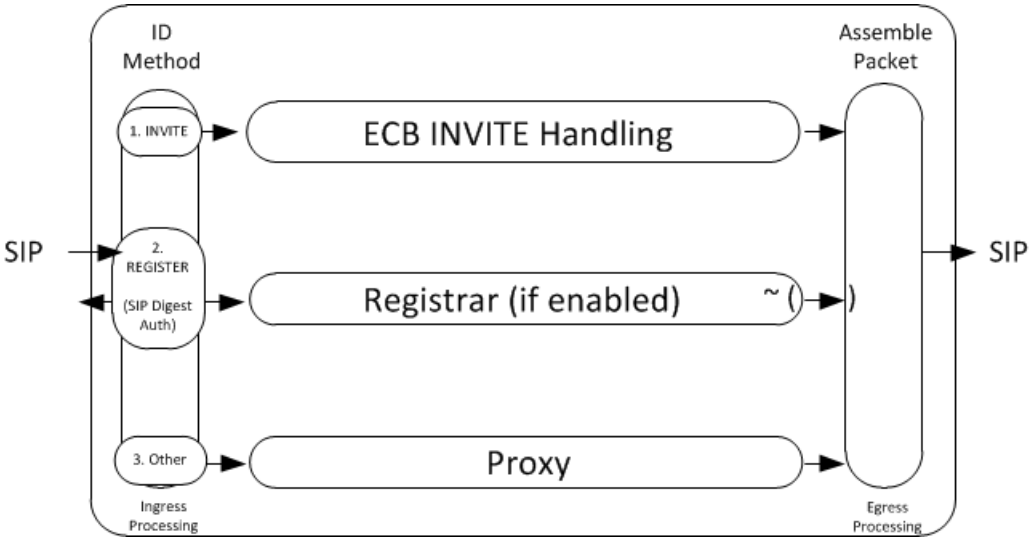
Packet Processing by the Oracle Enterprise Communications Broker

The following diagrams describe, at a high level, the processing performed by the elements of the Communications Broker to all traffic that it handles. Understanding this processing provides insight into configuration and troubleshooting tasks. The following diagram provides an example of the interactions of elements in a call flow between SIP agents.

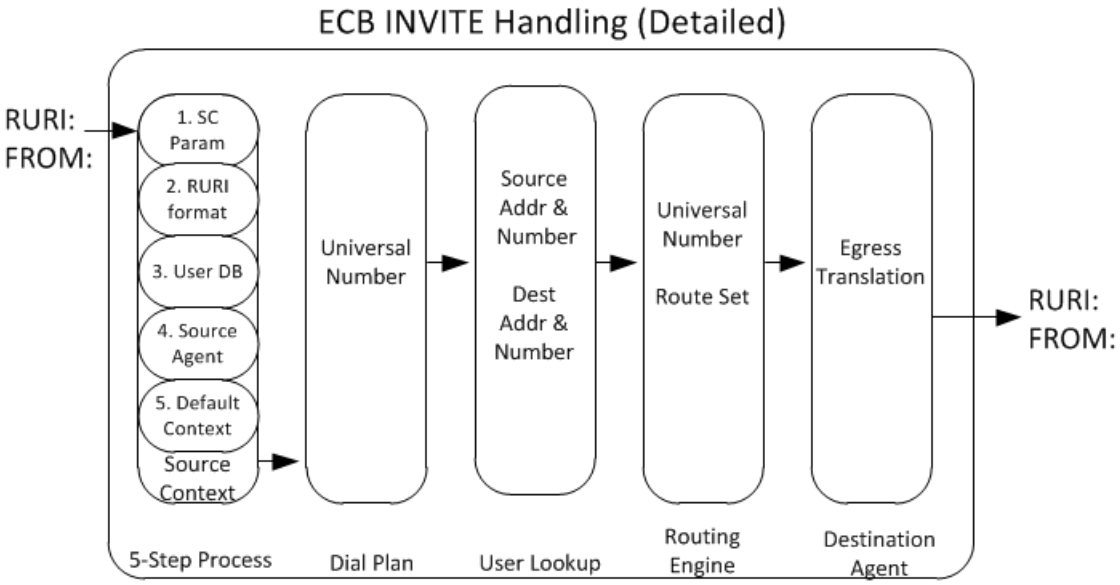


The following image displays Communications Broker processing of different signaling messages, including:

- INVITE—Pass through the INVITE handling processes, which includes number normalization, route optimization, and multi-contact support.
- REGISTER—When configured as a SIP Registrar, registration traffic passes into the registrar for authorization, authentication, and caching. There are multiple means of performing authorization and authentication.
- Other—All other signaling traffic is proxied, based on RFC 3261 standards, including the insertion of VIA and Route Record headers to keep the Communications Broker in the path of each applicable dialog's traffic.

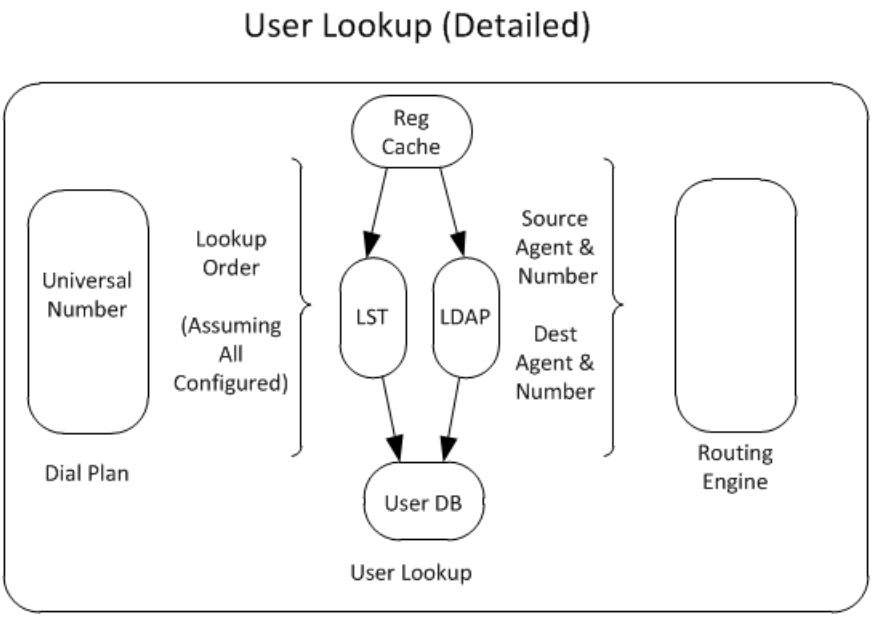


The following diagram displays the key processing elements handling an INVITE, including number normalization, based on context, end station look up, and recursive route set creation.



The following diagram details the elements the system examines to perform user lookup. The Communications Broker queries each of the objects shown in the diagram to identify the destination agent. After identifying the applicable agent, the user lookup hands everything the routing engine needs to recursively specify hop-by-hop routing through agents to reach the target.

Note that utilization of LST versus LDAP resources are independent and exclusive of each other. Either the LST or the LDAP resources perform the functions needed after registration cache procedures. The Communications Broker allows you to configure either LST or LDAP resources.



Ingress INVITE Processing

When an packet arrives at a Oracle Enterprise Communications Broker ingress interface, standard link and network layer processing occurs to prepare the data for processing within the device. Subsequently, the Oracle Enterprise Communications Broker performs admission and overload control procedures to ensure it is both appropriate and possible to proceed with further processing. As discussed, ensuing processing is based on traffic type, of which INVITE processing is key to the overall purpose of the Oracle Enterprise Communications Broker.The sections below describe further INVITE processing.

Identifying Source Context

When receiving an inbound SIP message, the Oracle Enterprise Communications Broker first determines the *source context* of the calling party. This allows the Oracle Enterprise Communications Broker to interpret the dialed digits appropriately.

For example, a user dialing 911 in the United States has different expectations than a user dialing extension 911 in a European office.

The system performs four steps sequentially to identify the source context. If a step identifies a source context, the system skips the next steps and provides the information to the dial plan engine for subsequent processing. These steps include:

1. The system searches the FROM address in the signaling for a source context (SC) parameter. This parameter, if present, identifies the UA's source context.
2. If the number presented in the RURI begins with a "+" sign, assume the RURI is an e.164 number and bypass the source context identification.
3. The system treats the digits received in the userinfo portion of the From header as a universal address and checks to see if the calling party is in its User database.
    a. If there is a match and the user has a source context configured, the system uses that as the call's source context.
    b. If the user has no source context configured, the system check the user's home agent for a source context and, if configured, uses that as the call's source context.

4. The system looks for a Source Context value in the configuration for the Agent from which the message was received.
5. If the above fail, the system uses the default Source Context, as configured in the SIP Interface settings.

Dial Plan Processing

The dial plan receives the dialed digits and the source context of that signaling message, and uses the rules associated with the identified context to prepare the *universal address* from the digits that were dialed. As described in the section on the dial plan engine, this may involve stripping routing digits out of the dial sequence, adding addressing digits into the sequence, or both.

The result of the dial plan processing yields the universal address that the system passes into the routing engine.

Route Engine Processing

The route engine receives the information from the dial plan lookup and builds a search key based on the calling number, called number, source agent, and destination agent for that call. As described in the section on Oracle Enterprise Communications Broker Routing, it recursively processes each route lookup result to construct full route sets.

Egress Processing

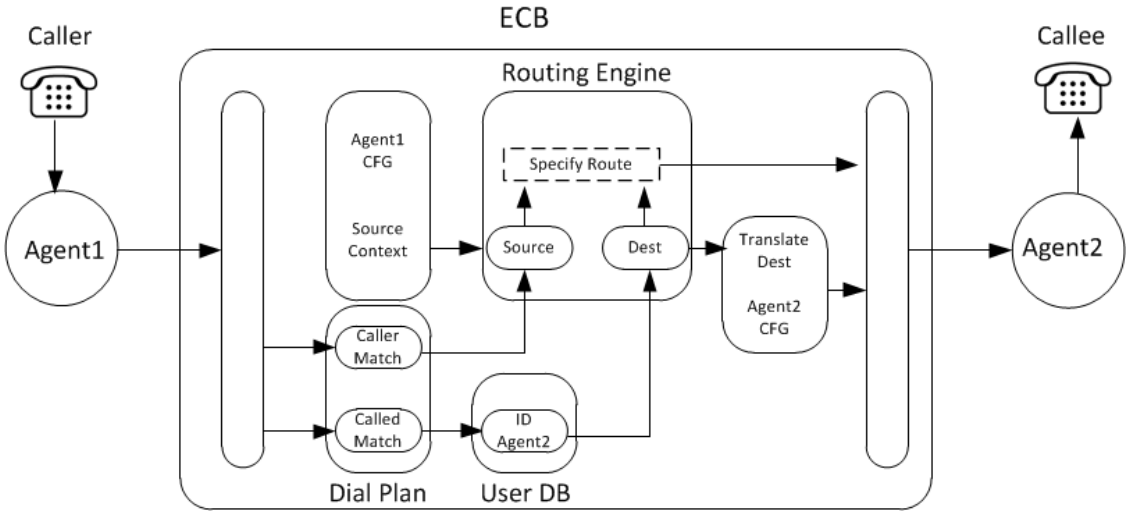
Now that the Oracle Enterprise Communications Broker has a fully qualified universal address, a route or set of routes to use for processing that call, it will prepare the universal address to suit the formatting requirements of the destination. It does this by looking for the Number Translation Mode of the *destination agent* (not any intermediate agents) and applying the transformation identified within that agent's configuration.

Lastly, the message is sent on its way based on the most preferred route. If that route fails, the Oracle Enterprise Communications Broker will try all subsequent route sets that it learned via the routing engine, in order from least cost to highest cost. This may also involve re-writing the universal address to suit the new "last hop".

Call Handling Example

The following illustration and call flow steps describe call handling through the Oracle Enterprise Communications Broker (Communications Broker).This example assumes no registration cache and no LDAP configuration.

The following diagram shows a simple, intra-organization call passing through the Communications Broker.



A user within the organization calls another within the organization residing on a different PBX. The call proceeds through the components of the Communications Broker using the following steps:

1. Call Received by Ingress Processing.
2. Ingress Processing hands FROM and Request-URI to Dial Plan.
3. System runs 5-step process to ID source context.
4. Dial Plan normalizes Source and Destination Numbers.
5. FROM handed to Routing Engine as Source.
6. Dial Plan hands Request-URI to User DB to ID Home Agent.
7. User DB hands Request-URI to Routing Engine.
8. Routing Engine uses FROM and Agent1 CFG to create new Source.
9. Routing Engine builds Route.
10. Routing Engine hands Request-URI to Agent 2 configuration.
11. Agent 2 configuration translates Request-URI into format compatible with Agent 2.
12. Agent 2 configuration hands Request-URI to Egress processing.
13. Egress processing builds INVITE.
14. Egress processing sends new INVITE to Agent2.

2

SIP Signaling Management

Oracle Enterprise Communications Broker (Communications Broker) SIP signaling management requires review of the following topics:

- Dial Plan—Normalizes dialing numbers.
- Contexts—Provides rules for dialing number normalization.
- Agents—Establish hop locations for routes.
- Routing—Builds hop-by-hop path to the end-station's target agent.

The Oracle Enterprise Communications Broker Dial Plan

The Oracle Enterprise Communications Broker's dial plan engine was designed from the ground up to simplify the administration of common, real-world dialing behaviors. Conceptually, the dial plan engine allows administrators to define the rules by which dialed digit strings are built up, or broken down into "universal addresses". A universal address may be thought of as an E.164 number, although this is not strictly required. Universal numbers are required to be globally unique, not E.164-compliant.

These rules are then grouped into a foundation data structure in the Oracle Enterprise Communications Broker, the *context*. The concept of a context is fundamental to the operation of the Oracle Enterprise Communications Broker's dial plan configuration, and is discussed below.

The dial plan engine serves two purposes. First, it constructs universal addresses from input received. Second, it prepares egress translation from universal addresses into contextually-appropriate addresses based upon a message's destination. An example of the latter is the system creating a URI for a remote phone that needs to be addressed with four digits rather than a fully-qualified E.164 number.

Oracle Enterprise Communications Broker Contexts

A context is a collection of rules used to manipulate strings of dialed digits. In most use cases, contexts are associated with a PBX or branch office where the users associated with a given PBX are all subject to its rules for making telephone calls, such as:

- Each user on the PBX dials the same digit for seizing an outside line
- All users may be able to reach other extensions on that PBX by dialing short dial strings
- All users in that environment have access to the same 'tie lines'.

The rules that govern how to interpret the series of digits do not differ from user to user within that PBX.

Note that this does not take user-based entitlements into consideration. For example, users within the same context all dial the same videoconferencing terminal in the corporate boardroom using the same series of digits, even though all of the users are not authorized to use that equipment.

Determination of a SIP message's "source context" is critically important. This is covered in more detail in "Ingress Processing". Phone numbers within a SIP message may have vastly different interpretations when, for example, a user dials "0" from two different branch offices within the same enterprise. The *context* of the dialing user differentiates the dialed pattern for the Oracle Enterprise Communications Broker (Communications Broker).

The following list describes the terminology used to define contexts applicable to the Communications Broker.

- Geographic context—A collection of rules that define the dialing patterns applicable to a geography, usually a country. These rules are outside of an enterprise's control and are pre-configured for you on the Communications Broker by Oracle. You can extend or modify these rules.
- Corporate context—Rules defined by the enterprise that specify routing, policy, access code, and extension range dialing patterns. Rules may vary based on applicable PBX or branch office. Context hierarchy manages such variations.
- Source Context—The context used when an Agent provides context detail for a given call. This is also the context within which a given user resides by way of configuration, to be understood as a user's default location.
- Source Context Param—"sc", meaning source context, is the syntax for a parameter on the FROM header presented by equipment external to the Communications Broker that specifies the context from which the call originated. When presented, the rules of this contexts are always applied.

Refer to "Dial Plan Configuration" for more information about the related fields.

Context Hierarchy

Contexts within the Oracle Enterprise Communications Broker may be defined *hierarchically*, to offer a parent/child inheritance relationship. This is done to avoid data duplication and redundant configuration.

For example, a large enterprise may have a corporate dial plan (common phone numbers for the IT help desk, employee benefits group, travel desk, etc.) that is consistent among all branch offices, and unique extension ranges per branch location. By defining common data in a "parent" context, each child context will inherit these common dial plan values and avoid the need for configuring each of them over and over for each branch office turn-up.

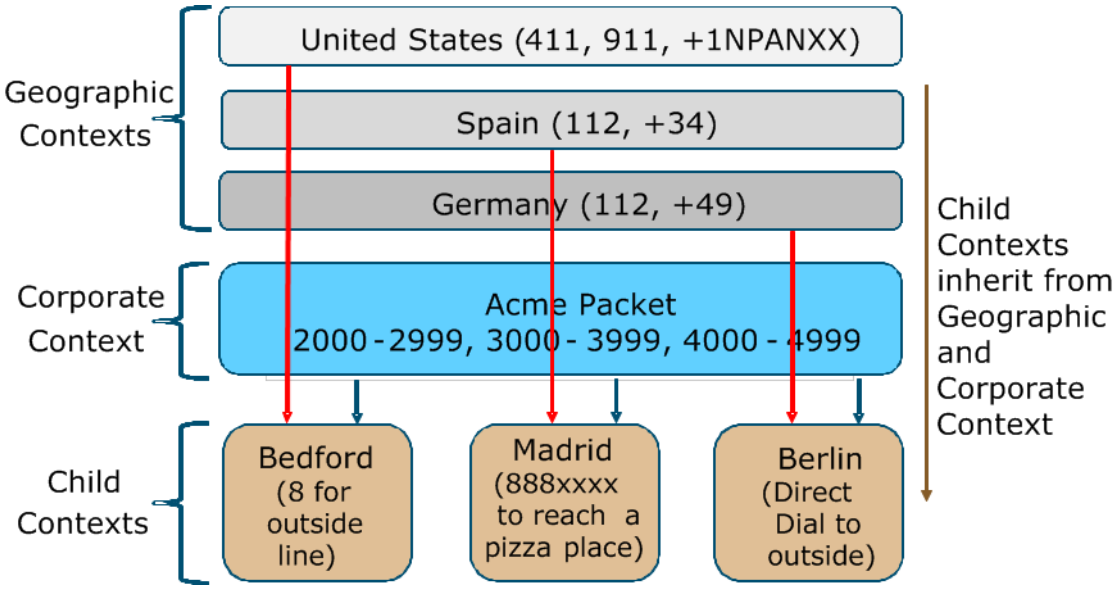
Each dialing context may have one corporate parent (for inheriting dialing rules that are unique for that enterprise) and one geographic parent (for inheriting common dialing rules pertaining to that branch office's physical location). Geographic and corporate contexts are described in the following sections.

Geographic Contexts

A geographic context is the set of rules for dialing within a given geography. It does not matter if you live in New York City or in Los Angeles, you'll still dial 011 for an international long distance call and 911 for emergency services because those are both part of the dial plan for the United States called the North American Numbering Plan (NANP). The Oracle Enterprise Communications Broker (Communications Broker) ships with geographic dial plans for the fifty most populous countries. You can override the default data or refresh it with future data, for example, to account for changes in the ITU dial plans.

Each context that you define on the Communications Broker may have a geographic parent, configured as a geographic location. By configuring a geographic location, the child context inherits the dialing patterns for that geography. You do not need to configure the child contexts

with rules for 011+, 911, 411, and so forth. They inherit these rules because they participate in that geography's parentage relationship.

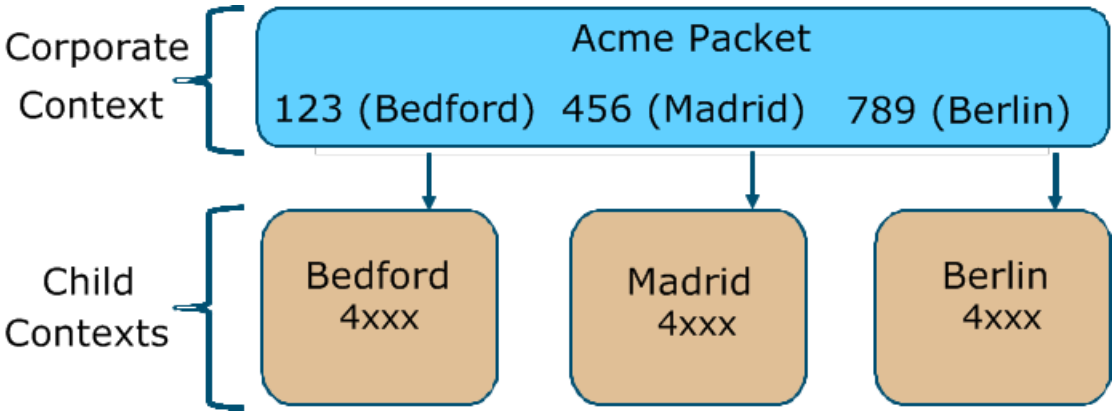


Note:

The digit ranges within the child contexts do not overlap, presenting a simple means of identifying context. This represents a Small Enterprise Communications Broker Configuration Model. The corporate dialing patterns are configured on the corporate context once. In the preceding diagram, when a caller dials 3xxx from any child context, the system always sends the call to Madrid.

Corporate Contexts

In contrast to geographic contexts, which are common for all telephone calls throughout the world and may be supplied with the Oracle Enterprise Communications Broker (Communications Broker) software, corporate contexts are company-specific and define the dialing rules for the enterprise. A corporate context may include all branch offices, remote offices, PBXs, and so forth.



> **Note:**
>
> In contrast to the example shown in "Geographic Contexts," the digit ranges in the child contexts overlap. The preceding diagram represents a Large Enterprise Communications Broker Configuration Model. In this deployment, the system uses the dialed prefix to identify the child context. Each child context inherits the dial patterns of the parent to direct a call. Each child sends calls with the prefix 123 to a Bedford tie line, 456 to Madrid, and 789 to Berlin. You need to configure each pattern only once, on the parent (Acme Packet) context.

Oracle Enterprise Communications Broker Agents

An agent defines a signaling endpoint. It is a next hop signaling entity that applies traffic shaping attributes to flows. Agents provide important properties for Oracle Enterprise Communications Broker operation, including:

- Transit and termination points for Oracle Enterprise Communications Broker routes; and
- Context identification for use by the Oracle Enterprise Communications Broker dial plan.

Agents can include the following types of devices:

- Softswitches
- SIP proxies
- Application servers
- SIP gateways
- Indirect Agents

For each agent, concurrent session capacity and rate attributes can be defined. The Oracle Enterprise Communications Broker can provide load balancing across the defined agents.

Why You Need Agents

You can use agents to define hops the Oracle Enterprise Communications Broker can use in a signaling path. You can also use them to define and identify preferred carriers. This set of carriers is matched against the local policy for requests coming from the agent. You can also set traffic constraints against specific hops via agent configuration.

In addition to functioning as a logical next hop for a signaling message, agents can provide information regarding next hops or previous hops for SIP packets, including providing a list of equivalent next hops.

How to Use Agents

Consider agents as next-hops within routing paths. Before configuring an agent, map out your session network and identify all potential agents. Each agent should be seen as a best hop based on its location, adjacencies and path costs. Redundant paths are also configurable using agents, allowing manual cost configurations for what may otherwise be equal cost paths.

In addition, consider the users for which each agent is a first hop. Agent configuration provides a method of defining routing and policy configuration for groups of users. Agents also provide a mechanism for defining source context for groups of users.

In some cases, specific addressing is not available or needed to access signaling endpoints. It may be that routing to a target domain is preferable to routing to a specific agent. In these cases, you can configure an agent using, for example, only the target domain name rather than a specific endpoint. When doing this, you assume that the domain itself is able to route to any further hops needed to reach the UA and that the same policies must be utilized from all traffic from that domain.

Agent Groups

Agent groups contain multiple agents. Members of an agent group are logically equivalent (although they might vary in their individual constraints) and can be used interchangeably as transit targets for SIP traffic. For one reason or another, a given agent may not be able to service traffic. You configure agent groups to establish multiple transit destinations for purposes such as redundancy.

Examples of agent groups include the following:

- Application Server cluster
- Media Gateway cluster
- Softswitch redundant pair
- SIP Proxy redundant pair
- Gatekeeper redundant pair

Agent group members do not need to reside in the same domain, network, or realm. The Oracle Enterprise Communications Broker (Communications Broker) can allocate traffic among member agents regardless of their location. The Communications Broker uses the allocation strategies configured for an agent group to allocate traffic across the group members.

You configure agent groups from the Agent configuration dialog on the GUI. The configuration consists of naming the group, selecting the allocation strategy, selecting recursion preference, and adding the agent group members.

After you configure the group, you configure agent group names as:

- A `Dest agent` in a routing table entry
- A `Route` in a routing table entry
- A user's `Home agent` in the user database
- A `Default home agent` within an LDAP query

The syntax for these entries appears as the word 'group' followed by a colon (:) and the group name. For example,

```
group:MyGroupName
```

When configuring the group, you select between the following allocation strategies to define the method of selecting the next member of the group for a connection attempt, if the previous connection attempt is unsuccessful:

Hunt	The Oracle Enterprise Communications Broker selects the agents in the order in which they are configured in the agent group. If the first agent is in service, and has not exceeded any defined constraints, all traffic is sent to the first agent.

	If the first agent is out of service, or is in violation of constraints, all traffic is sent to the second agent. And so on for all agents in the agent group. When the first agent returns to service, the new traffic is routed back to it.
Round robin	The Oracle Enterprise Communications Broker selects each agent in the order in which it is configured, routing a session to each agent in turn.

To summarize, agent group operation requires the following configuration:

- Two or more agents
- An agent group containing those agents
- A route, user configuration or LDAP query that directs traffic to that group

Recursion

Agent groups use a recursive process to communicate with agent group members. Recursion behavior is specified by the allocation strategy. You can optionally configure the Oracle Enterprise Communications Broker to attempt communications with only one member of the agent group by leaving the **Try All** control deselected, which means disabled.

The agent group performs its agent selection rotation process independently of the recursion setting. Each allocation strategy rotates agent selection as a means of selecting the first agent to try. This ensures that the system continues to use each agent in the group as a message target.

Routing paths may traverse multiple agent groups. The system performs recursion on the SAG only if it is the next immediate hop in the path. Only the first SA is added from subsequent SAGs in the path.

Oracle Enterprise Communications Broker Routing

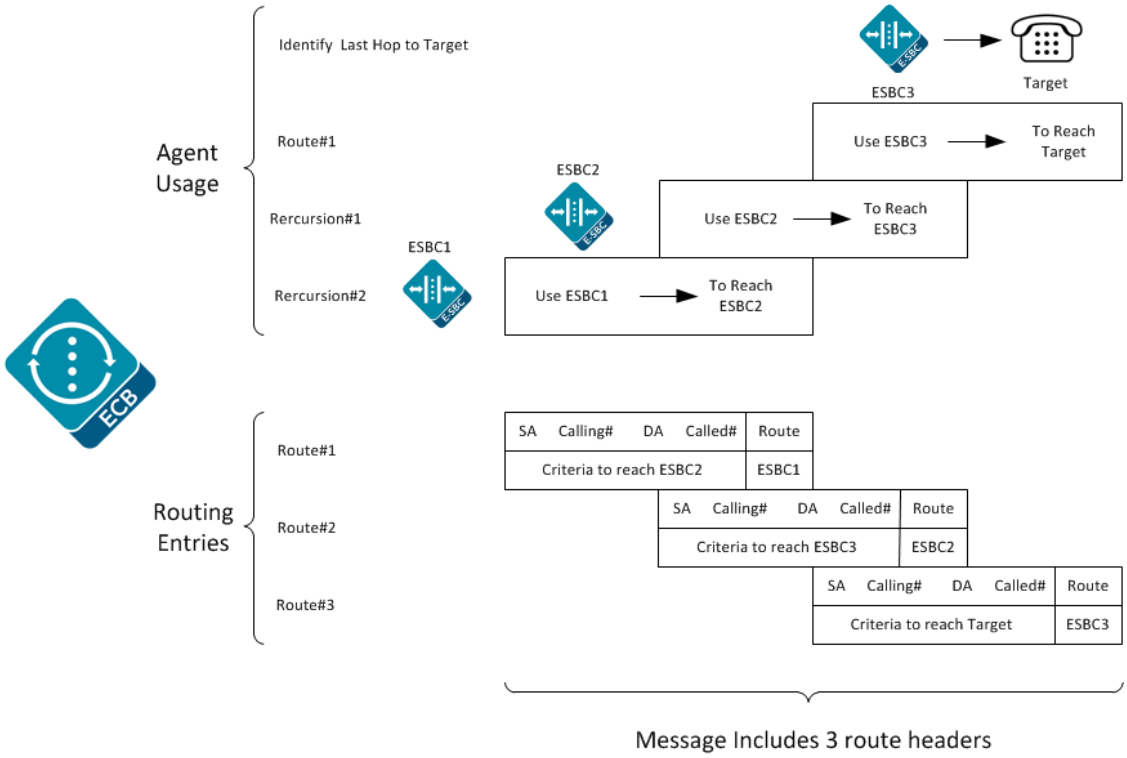
The Oracle Enterprise Communications Broker (Communications Broker) employs a purpose-built SIP routing engine for packet processing. Unlike traditional SIP proxies, application servers, or Session Border Controllers, the Communications Broker may be provisioned with a complete network topology map of all signaling entities, and use this provisioned data to make fully-informed routing decisions on how signaling flows should travel through a SIP network. Beyond choosing a next hop and pushing the signaling message on its way, the Communications Broker will look at the entire path from origin to destination to find the path with the least cost, fewest active sessions, least number of hops, and so forth. The Communications Broker pre-populates the egress signaling message with a specific route set to inform each receiving device on the next element in sequence.

Recursive Routing

Conceptually, the routing engine in the Oracle Enterprise Communications Broker (Communications Broker) is similar to the recursive routing engine in a layer three router. The system provides the route engine with input criteria, including calling number, called number, source agent, and destination agent. The routing engine returns a set of results based on the lookup. The Communications Broker processes each result recursively until complete, with each loop building another hop on the route.

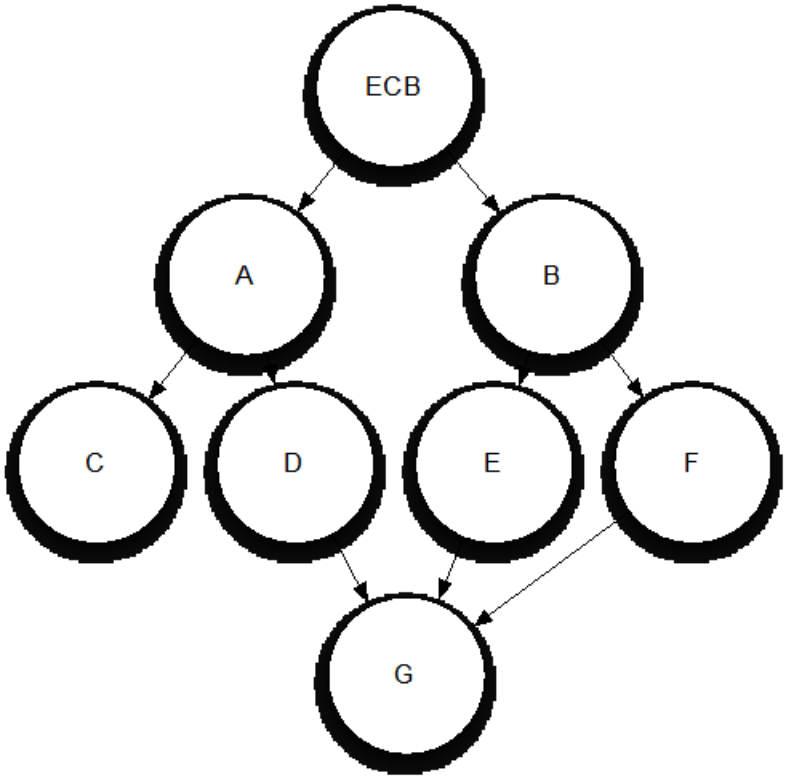
The process of using recursion to create routes consists of identifying individual hops for an end-to-end path beginning with the last hop before the destination. The routing engine selects subsequent elements (agents) to identify the hop that is next after the previously identified hop until it has a full path between itself and the UE. The Communications Broker routing engine

performs this process for all possible paths to each agent, creating multiple choices for the system to use as a route set for every individual call.



The following diagram is an example of a rudimentary network topology to illustrate how the Communications Broker uses recursion to identify a route path.

1. The Communications Broker discovers that Agent G is the last stop on the signaling message path.
2. The Communications Broker resolves each way that it can reach Agent G. It finds that it can reach G by way of agents D, E, and F.
3. The Communications Broker looks up how to reach D, E, and F, and the route table yields Agents A and B. Because Agents A and B are directly connected to the Communications Broker, route path identification is complete.



Identifying Contacts and Specifying Routes

Having determined the target for a call, the Oracle Enterprise Communications Broker creates route sets to the target. In addition, the Oracle Enterprise Communications Broker also finds all of a target's contacts and builds route sets for each of them. Depending on deployment and configuration, these contacts are available from multiple sources, both internal and external to the Oracle Enterprise Communications Broker. These sources provide the Oracle Enterprise Communications Broker with the agent of each contact, from which it can build routes to the contacts.

The Oracle Enterprise Communications Broker uses called, calling number and source agent from the source of the call, and the called number and destination agent to create a route. It goes about collecting this detail for target and all contacts using the following procedure:

1. Find the AoR and all associated contacts in the registration cache. Store agent(s) for route creation.
2. Find caller and callee's source context, as presented earlier in this document.
3. Convert user portion of request-URI and From URI to universal number via the dial plan and source context. These become called and calling number.
4. Lookup called and calling number in the user database. Retrieve each number's agent for route creation.
5. If the user database lookup does not produce source and destination agent, request this information via the LDAP database. If necessary, the system converts the universal number so the lookup format is compatible with the LDAP database.
6. If the LDAP lookup does not return source and/or destination agent, the Oracle Enterprise Communications Broker uses the host portions of the Request-URI and FROM URI or inbound agent (agent on which the call was received) as agents.

7. For any AoR returned from LDAP, the Oracle Enterprise Communications Broker performs another lookup in the registration cache and creates routes for the AoR.
8. If the LDAP lookup identifies other contacts, the system passes those contacts through the registration cache to identify its agent and build the route set.
9. All routes are built and, depending on forking configuration, placed in order.

If any of these sources are not configured or operational, the Oracle Enterprise Communications Broker proceeds to the next source. Note that, if agents are found in the user database, the system does not perform an LDAP lookup.

> **Note:**
>
> The procedures associated with an LDAP resource are equivalent to those with an LST resource. These procedures are also exclusive; LST and LDAP resources cannot be used simultaneously for routing.

The system forwards the request based on the route list and the forking configuration. By default, the system performs serial forking to all contacts using route cost to establish the order. The system can also perform parallel forking, if desired.

Route Selection

After constructing routes to use for a call, the Oracle Enterprise Communications Broker (Communications Broker) often can choose from multiple routes. Cost calculations for each route identify the route that the system uses.

In the network example in "Recursive Routing," the Communications Broker determined the three following potential routes for an inbound message sent to Agent G:

- A to D to G
- B to E to G
- B to F to G

Each route between agents in the Communications Broker routing table may be assigned a cost that represents the desirability of that route. The Communications Broker sums up the total cost for each route and orders them from lowest to highest cost. It then selects the lowest cost route and forwards the message.

Forking

Forking is a routing option that the Oracle Enterprise Communications Broker (Communications Broker) uses to direct an INVITE to multiple targets. Several types of forking control the operations of the function, such as timing and target lists. The Communications Broker performs serial forking to all targets by default. You can enable the Communications Broker to perform parallel forking, which directs the INVITE to all targets for an Address of Record (AOR) simultaneously. When any target responds, the Communications Broker issues a CANCEL to the other targets and ignores any responses from them. Should the Communications Broker receive error messages from all contacts, it provides the lowest number message back to the caller.

The Communications Broker detects a user's multiple contacts from:

- A configured LDAP server

- The local registration cache
- The User Database

The system can fork to multiple contacts that come from the following locations:

- Registration cache (the minimum requirement)
- Registration cache + LDAP
- Registration cache + User Database

When the system finds the contact used in the RURI in the User Database, it uses that contact and then LDAP queries only on the user in the "From:" header.

When the User Database does not contain the number in the RURI, LDAP performs the lookup on the RURI and the system forks the call based on the result.

You cannot configure forking to:

- LDAP directly
- User Database directly
- Registration cache + LDAP + User Database

When the Communications Broker detects multiple routes to a contact, the system uses cost configuration to determine the preferred route. The system uses backup routes only when the primary routes do not respond.

You can enable parallel forking in the **SIP Interface** configuration.

Fork Groups

Fork-groups on the Oracle Enterprise Communications Broker are sets of one or more contacts that the system attempts to reach simultaneously. The system uses fork group order to specify when it tries to reach each fork group's contacts. This results in a hybrid of serial and parallel forking operation. The user can configure fork groups on agents, the registration cache and within the LDAP database. The user can also configure a global fork group timer with a value from 0 to 32 seconds on the sip-interface. If the system does not receive a response from any contact within that time, it tries the next fork group. Parallel forking must be enabled.

By default, the Oracle Enterprise Communications Broker assigns all contacts to fork group 1 and attempts to contact them serially, using the order in which it learns them. If desired, the user can enable parallel forking. By itself, parallel forking causes the system to attempt to reach all contacts simultaneously. Fork groups refine parallel forking, allowing the system to try all contacts in a group, and then move on to the next group.

The user names fork groups using decimal numbers between 1 and 100. This naming defines fork group order, with the system using fork group 1 first. The user configures objects with fork group numbers, based on a forking plan they devise.

The user can also configure a lookup query to LDAP databases to retrieve individual contacts' fork groups. The user must have previously modified the LDAP database to include a custom fork group field in contact records.

A use case for this feature could include the system attempting to reach a user's BYOD and desk phone simultaneously, then forwarding to an enterprise-preferred voicemail server if neither answers. For this to work, the BYOD and desk phone would be in the same fork group. The voicemail server would be a member of a higher numbered fork group. To ensure this order, the system assigns lower numbered fork groups with a higher precedence.

After establishing a session, other contacts may respond to try and start the session themselves. The Oracle Enterprise Communications Broker replies to these messages with a CANCEL.

Fork group operation does not exclude the use of primary and backup routes. The Oracle Enterprise Communications Broker still creates route sets for all contacts. If a contact fails via a primary route, the system attempts to reach the contact using all backup routes, based on cost.

If the Oracle Enterprise Communications Broker receives a redirect from an endpoint, the system adds the redirect target to the current fork-group and tries to contact it before attempting the next fork-group. If the global fork group timer expires before the system receives a redirect, however, the system proceeds to the next fork group.

The flexibility inherent in fork group operation requires the user to carefully plan forking prior to configuration. For each call, the system creates an ordered contact list, based on fork group configuration. Because the fork group assignment may affect multiple contacts, such as agent configuration, the user must be careful not to configure a sequence that would adversely affect calls to different end stations behind that agent.

Fork Group Assignment

You configure fork groups to specify call attempt order for a given call. The Oracle Enterprise Communications Broker (Communications Broker) creates call attempt lists based on each contact's fork group assignment.

Upon configuration, the system assigns fork groups to target endpoints, as follows:

- User database—Each user database entry is assigned to the home-agent's fork-group.
- Registration cache—Each registration cache entry is assigned to the SIP registrar's fork-group.
- LDAP server—Each contact retrieved from an LDAP server is assigned to the a fork-group specified in the server's user record. If no fork-group is configured for the user in the Active Directory, the system assigns the target endpoint to the fork-group of the user's home-agent, as configured on the LDAP server.

The Communications Broker uses the following contact source order:

1. Registration cache contacts
2. User database contact
3. LDAP contacts
4. LDAP AORs generating subsequent contact dips for additional registration cache contacts

By default, the Communications Broker collects contacts from these sources and creates a contact list that follows the order in which the system learns them. This behavior is in accordance with default fork group operation, where all contacts are in fork group 1 and the mode is to fork serially only. As soon as the system finds differentiation between contact fork groups, it arranges contact lists using the fork group order.

Additional Targets

You may want to include forking targets to stations that are not resolved as original call targets. Examples of these scenarios include directing calls to a preferred enterprise voice mail server, if they are not picked up. The Oracle Enterprise Communications Broker (Communications Broker) provides for this using Additional target configurations. You manually configure these devices within **Additional target groups**, which includes one or more end stations. Agent and

registrar configuration allows you to select these groups as additional forking targets for all calls that use that agent or registrar's entries.

An additional target group is a list of agents or end stations that the Communications Broker uses as candidates for either parallel or serial forking. You configure these groups with fork group numbers, which the system then uses to define fork group order. The system adds additional target contacts to the forking sequence the same way it adds contacts for other objects with fork group configurations.

Fork Groups Configuration Considerations

Fork Group configuration requires that you establish a clear plan prior to any configuration. Configurations established by this planning may include:

- Identify or create new agents as fork group targets.
- Identify usage and precedence policy for forking via the Registrar.
- A adjust fork group identification and precedence based on preferred LDAP lookup scenarios.

Coordinating the use of these sources and configuring the applicable objects establishes and refines fork group configuration. Applicable configuration objects include:

- Agent—Create new agents specifically for use in a fork group, or uses existing agents. Configures an agent with a single fork group number, which the system applies to every call using that agent as a route.
- Additional targets—Create sets of targets to manually establish forking targets.
- Registrar—Set the registrar to a single fork group, which the system applies to every contact in the registrar.
- LDAP—Ddefine a lookup query that pulls the pre-configured fork group assignment defined for the queried contact. The query must pull this fork group assignment from a custom attribute established on the LDAP database.

Configure Fork Groups on Agents

1. Access the Agent configuration object.

   Configuration tab, **Service Provisioning** section, **Agents**, **Session Agent**.
2. Click either Add to create a new agent or **Edit** to add the agent to a fork group.
3. On the **Agents** page, do the following:

Additional Target Group	Select a target group from the drop down list.
Fork Group	Enter a digit to specify this agent's fork group priority in a target list. Range: 0-100.

4. Click **OK**.
5. Save the configuration.

Configure Additional Target Groups

Additional targets are agents or end stations that are not contacts already targeted by a given call. You assign additional target groups on a per-agent and a per-registrar basis.

Configure target groups.

To configure additional target groups:

1. Access the Agent configuration object.

   Configuration tab, **Service Provisioning** section, **Agents**, **Additional Target Group**.
2. On the Additional Target page, click the **Add** icon.
3. On the Add Additional Target page, enter a enter a name for this target. Use this name to assign this group to an agent or the registrar.
4. Under Additional Target, click **Add**.

   The system displays the Additional Target Group / Additional Target page.
5. On the Additional Target Group / Additional Target page, do the following:

Additional Session Agent	Select an existing target group from the drop down list, or enter an IP address of a target station.
Fork Group	Enter a digit to specify this agent's fork group priority. The lower the digit, the higher the priority. Default: 1. Range: 1-100.

6. Click **OK**.
7. Click **OK**.
8. Save the configuration.

Assign the Additional Target Groups to the appropriate agent or the registrar.

Configure Fork Groups on a Registrar

1. Access the SIP Registrar configuration object.

   Configuration tab, **System Administration** section, **SIP Registrar**.
2. On the Add SIP Registrar page, do the following:

Additional target Group	Select an existing target group from the drop down list.
Fork Group	Enter a digit to specify the fork group for every contact in the SIP Registrar end points. Default: 1. Range: 0-100.

3. Click **OK**.
4. Save the configuration.

Configure LDAP for Fork Groups

- Configure the LDAP server that you want to use for fork groups.
- Define and populate the custom LDAP attribute for specifying a fork group.

1. Access the LDAP Configuration object.

   Configuration tab, **System Administration** section, **LDAP**, **LDAP Config**.
2. Click on the Action button for an existing LDAP configuration.
3. Click **Edit**.
4. Scroll to **Routing**, **Lookup Queries**, and click **Add** .

   The system displays the Add LDAP Config / Routing / Lookup Query dialog.

5. In the **Fork Group Attribute** field, enter the name of the custom attribute in the LDAP database that includes fork group assignments.
6. Click **OK**.
7. Click **OK**.
8. Save the configuration.

Configure the Global Fork Group Timer

The global fork group timer specifies the amount of time the system waits for responses from a fork group before it tries contacting the next fork group.

After this timeout, the system drops responses received from contacts in the expired fork group.

1. Access the SIP Interface configuration object.

   Configuration tab, **System Administration** section, **SIP Interface**, **SIP Config**.
2. On the SIP Config page, for the **Fork Group Timeout** parameter, enter a time in seconds. Default: 0. When set to the default, the system waits for the standard SIP INVITE transaction timeout to expire before proceeding with the next group. Range: 0-32.
3. Click **OK**.
4. Save the configuration.

Routing and ENUM

The ENUM functionality lets the Oracle Enterprise Communications Broker make an ENUM query for a SIP request. The ENUM lookup capability lets the Oracle Enterprise Communications Broker transform E.164 numbers to URIs during the process of routing (or redirecting) a call. During the routing of a SIP call, the Oracle Enterprise Communications Broker determines if an ENUM query is required and if so which ENUM server(s) need to be queried. A successful ENUM query results in a URI that is used to continue routing or redirecting the call.

Refer to the chapters on Agent and Route configuration for instructions on the related fields.

Route Types and Precedence

There are three types of routes used by the Oracle Enterprise Communications Broker. These include configured, default and implicit. The Oracle Enterprise Communications Broker uses these types, in conjunction with route cost, to determine route order. You create both configured and default routes in your route table. A default route is simply a route configured with wildcards for called number, calling number, source agent and destination agent. The system installs implicit routes dynamically when there are no explicitly configured routes to an agent. The system assumes the agent is to be a directly connected next-hop, and subsequently relies on the network infrastructure to reach that agent when needed.

- The system orders routes by cost first, with the lowest cost being preferred.
- If costs are equal, the system orders by type, with the preference given to configured, then implicit and then default.
- If the route cost and type are all equal, the system orders routes, Exact Match is given the highest priority.

- If multiple routes have the same route cost and type, and are of the same exact match, then the route with least number of hops is given the highest priority.

> **Note:**
>
> If multiple routes having the same cost, type and are of exact match and the same number of hops then the route that gets priority is not defined.

Refer to the chapter on Route configuration for instructions on the related fields.

Active Directory and Communications Broker Routing

A large percentage of Enterprises use call servers with Active Directory (Domain Controller) such as Media Servers, Exchange Servers, Teams Servers, and so on. For Enterprises that integrate these servers in parallel to their existing communications infrastructure, or transition from their legacy Private Branch Exchange (PBX) to these types of servers, Active Directory becomes a more efficient and cost-effective way of routing the incoming calls within the core Enterprise network.

Clients using Microsoft servers such as a Teams Server deploy their own URI. Therefore, a user in a network with both a desk phone and a Teams client may have an IP PBX extension/URI for the desk phone, and a different URI for the Teams client. Currently, all PSTN traffic is sent by default to a legacy PBX in the core network. If the PBX does not recognize the extension/URI, the PBX forwards it to the Teams client. Sending traffic to the PBX first and then to the Teams Server can be costly in terms of computing resources and licensing fees. Routing all incoming sessions from a SIP trunk to the Teams Server first and then to a PBX can be costly.

As a solution, the Oracle Enterprise Communications Broker initiates a query to the Active Directory to determine how to route the call. The data fetched is the agent of the targets pre-configured in the database.

The Communications Broker stores the data used to facilitate the routing decision of the call (performed by Lightweight Directory Access Protocol (LDAP) and routes the call the first time to the applicable destination (PBX or Teams Server).

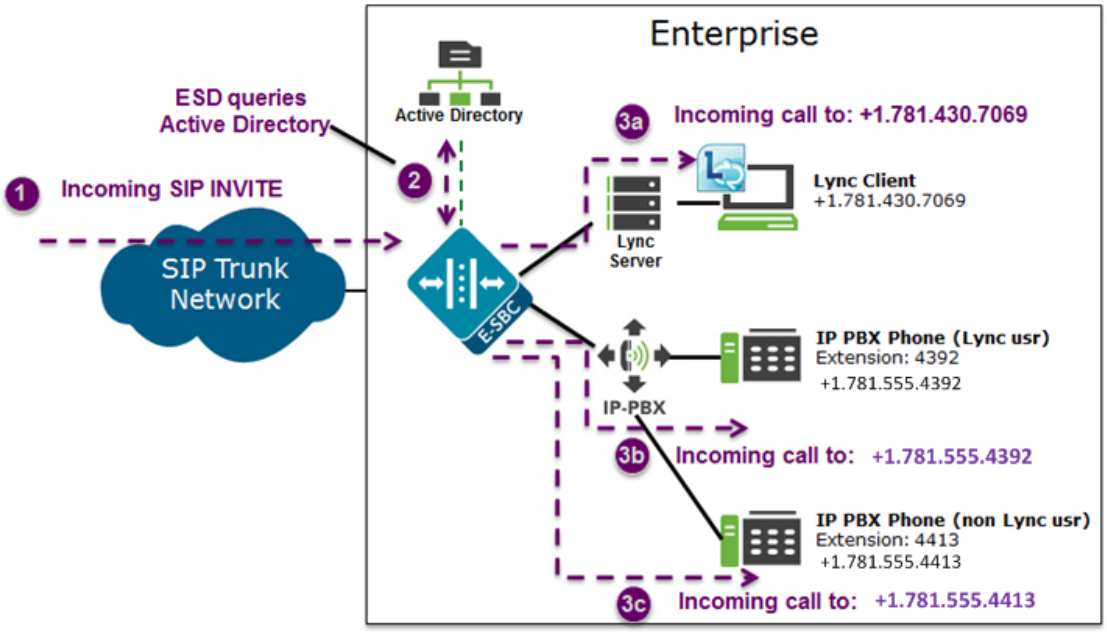
In scenarios where a user has multiple contacts such as both a Teams phone and a legacy PBX phone, calls destined for the Teams phone number can be routed to the PBX phone number, or calls destined for the PBX phone number can be routed to the Teams phone number. The destination is dependent on the current Communications Broker configuration. The Communications Broker uses the information stored in the Enterprise's Active Directory, compares it to the Communications Broker configuration and then determines which phone number to utilize.

> **Note:**
>
> The Active Directory-based call routing feature supports confidential and secure LDAP traffic support by using SSL TLS (LDAPS).

Active Directory-based call routing is a feature of the Communications Broker that facilitates the routing of incoming calls to the appropriate destinations within the Enterprise core network. The Communications Broker LDAP query to the Active Directory yields the agent of the phone number.

The following diagram shows incoming call flow with Active Directory and the Communications Broker in the network. The IP PBX extension (4392) is the primary telephone number (+1.781.555.4392). The secondary transition number (+1.781.430.7069) is assigned to Teams.

1. When the Communications Broker receives an inbound SIP INVITE over a SIP Trunk, it checks the current LDAP configuration in the Communications Broker.
2. Depending on the configuration, the Communications Broker accesses the Enterprise's Active Directory to search for the applicable number being called by way of an LDAP query.
3. When the LDAP query finds the agent of the called number, the Communications Broker builds a route set for the call and routes the call, per the routing engine, directly to the call server client (3a) or to the IP PBX phone (3b) and (3c).



The Enterprise is responsible for migrating phone numbers from the legacy PBX to the call server by making the necessary updates in their Active Directory so the Communications Broker can route the call properly. In the illustration above,

LDAP and Oracle Enterprise Communications Broker Routing

The Oracle Enterprise Communications Broker (Communications Broker) uses Lightweight Directory Access Protocol (LDAP) to perform queries to the Enterprise's Active Directory (AD) to determine where to route incoming calls to the call server or the IP PBX in the Enterprise network. Session requests and responses are sent and received based on the Communications Broker's LDAP routing configuration. LDAP determines the destination, to a call server user or a non-call server user, and forwards the call accordingly.

The Communications Broker, using LDAP, performs the following on an inbound call:

- Creates an LDAP search filter based on the dialed number and the configured LDAP attributes
- Sends an LDAP search query to the configured LDAP Server
- Creates a route list based on the query responses received from the LDAP Server and the applicable attributes it already has (caller number, callee number, caller agent)

- Routes calls based on the route list and routing order. The routing order is dependent on the LDAP attribute configuration and whether there was an exact match for the dialed phone number in the Enterprise's Active Directory.
- If configured, searches for additional AoR matches in Active Directory so that it can create additional routes to target users that have contacts stored in separate records.

To use AD as a source for home agent names, you create look-up queries from the LDAP routing configuration dialogs. The Communications Broker uses LDAP to retrieve that information and create routes. If the system cannot derive a home agent name from the query results, it routes the call to the configured default home agent.

> **Note:**
>
> You must ensure that phone numbers in the LDAP database are unique. If the Communications Broker encounters multiple records with the same number, it cannot process the look-up.

The Communications Broker keeps a permanent LDAP session open to all configured call servers. It sends an LDAP bind request on all established connections to those servers. The first call server is considered the primary LDAP Server, and all others are secondary LDAP servers. If a query request sent to the primary server is unsuccessful, the Communications Broker sends the request to the next configured LDAP Server until the request successfully in gets a response. If the Communications Broker receives no response and cannot find another route successfully, it sends a busy to the caller.

LDAP performs call routing based on LDAP attributes configured on the Communications Broker. The **route-mode** attribute setting determines how LDAP handles the called number when accessing the Enterprise's Active Directory. You can set routing modes to any of the following:

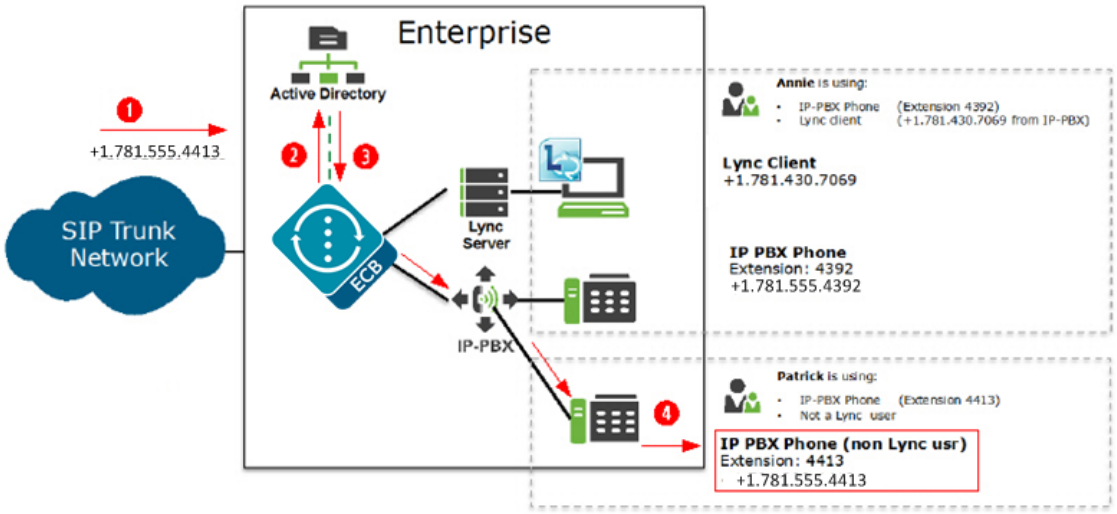
- Match-only (default)
- Attribute-order
- Match-first

Match-only

When you set the LDAP **route-mode** attribute to match-only, the Communications Broker performs as follows.

When the Communications Broker receives an incoming call to the Enterprise network, LDAP queries the Active Directory to find the number that matches the incoming number exactly. If the number is found, the Communications Broker retrieves the entry's agent and builds a route list for the call. The following steps describe the match-only call flow in the following diagram.

1. A call comes into the Enterprise network (+1.781.555.4413).
2. Using the configured route-mode of match-only, LDAP queries the exact matching number in the Enterprise's Active Directory.
3. The Active Directory finds the matching number and includes that number's agent in the response to the LDAP query.
4. The Communications Broker creates a route set for the call and forwards the call towards the destination phone number (same number as the number that initially called into the Enterprise in Step 1 (+1.781.555.4413)).



Attribute-order

When you set the LDAP **route-mode** attribute-order, the Communications Broker performs as follows.

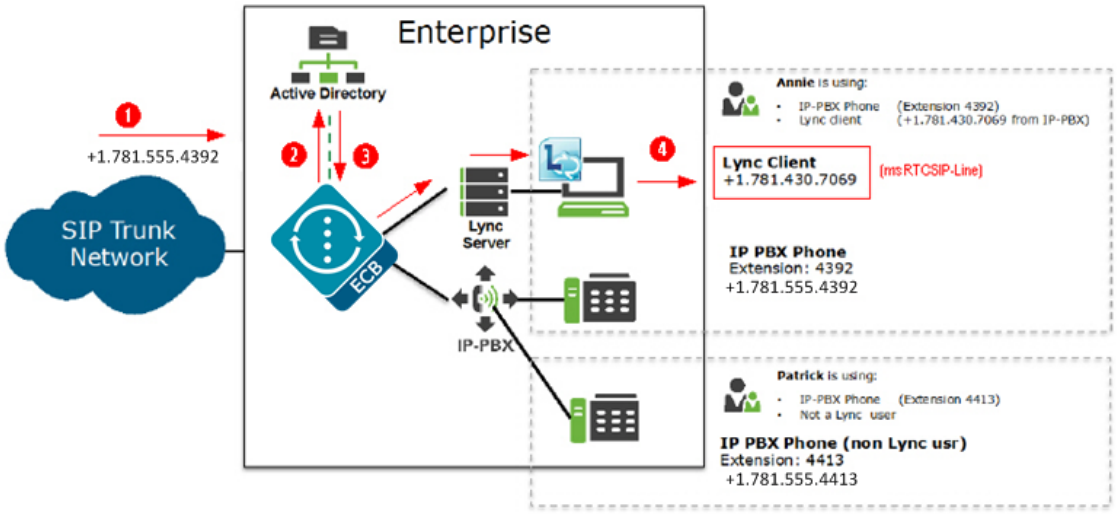
The order in which you configure the LDAP attributes on the Communications Broker determines the priority of each route. If an incoming call is destined for the IP-PBX, but the attribute name for a Teams client is configured first, the Communications Broker uses the corresponding agent (Teams Server) to create the first route in the route list.

An entry in an LDAP search response must have at least one attribute that it matches in the Active Directory.

For example, suppose the incoming phone number is +1.781.555.4392 (which matches the IP-PBX phone number), and the attribute name msRTCSIP-Line (Teams attribute) in the response is+1.781.430.7069 (Teams phone number). The Communications Broker creates a route for the Teams phone number, even though the incoming phone number matches the IP-PBX phone number because the msRTCSIP-Line attribute was configured first. The Communications Broker forwards the call to the Teams destination.

If an Enterprise uses the same phone number for both Teams and IP-PBX phones, and the attribute-name msRTCSIP-Line is configured first (a Teams attribute), the Communications Broker uses the corresponding agent (Teams Server) to create the first route in the route list. The following steps describe the attribute-order call flow in the following diagram.

1. A call comes into the Enterprise network (+1.781.555.4392).
2. Using the configured route-mode of attribute-order, LDAP queries the Active Directory for the agent of the matching number.
3. The Active Directory responds with the agent associated with the first configured LDAP attribute (+1.781.430.7069). In the diagram, the number is associated with a Teams Client (msRTCSIP-Line) that was configured first in the LDAP configuration.
4. The Communications Broker forwards the call to the applicable destination phone number's agent from the Active Directory response. (+1.781.430.7069).



When you configure the attribute name msRTCSIP-Line first, the Communications Broker uses the corresponding next hop (Teams Server) to create the second highest priority route in the route list. For example, the dialed telephone number could be +1.781.555.4392 (IP-PBX phone number), and the attribute-name msRTCSIP-Line in the response could be +1.781.430.7069 (Teams phone number). A route is created for the Teams phone number, even though the dialed telephone number is the PBX phone number.

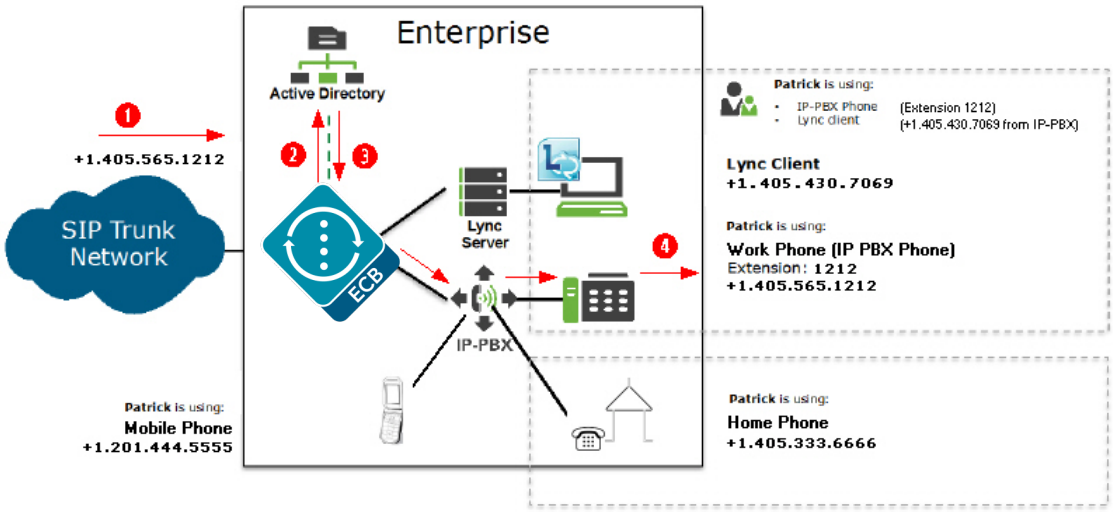
Match-first

When you set the LDAP **route-mode** attribute to match-first, the Communications Broker performs as follows.

When the LDAP query is sent to the Active Directory, the first exact match of the incoming phone number that the LDAP query finds in the Directory is the number whose corresponding route gets the highest priority in the route list. For all other routes configured on the Communications Broker, the ordering of LDAP attributes in the LDAP configuration determines the priority for each route.

For example, if the incoming number is +1.405.565.1212, and the Active Directory includes a configured mobile number first (+1.201.444.5555), a home number second (+1.405.333.6666) , and a work number third (+1.405.565.1212), the LDAP query searches the mobile number first, then the home number, then finds the exact match on the work phone number. The Active Directory responds with the agent information for the work phone number and the Communications Broker creates a route list with this exact phone number and forwards the call accordingly. The following steps describe the match-first call flow in the following diagram.

1. A call comes into the Enterprise network (+1.405.565.1212).
2. Using the configured route-mode of match-first, LDAP queries the Active Directory for the agent of the matching number.
3. The LDAP query searches throughout the Active Directory until it finds the first exact match on the number. Active Directory responds with the exact phone number associated with the incoming number (+1.405.565.1212). In the diagram, the number was associated with the work phone.
4. The Communications Broker forwards the call to the agent of the applicable destination phone number from the Active Directory response. (+1.405.565.1212).



LDAP Messages

If LDAP message logging is enabled in the Active Directory, the Oracle Enterprise Communications Broker sends LDAP messages to a message log called log.sipd. This log records all received and sent LDAP messages. Messages are in ASCII encoded binary format.

Additionally, when LDAP is invoked for routing, the LDAP messages display in the GUI under the Monitor and Trace tab.

> **Note:**
>
> The Oracle Enterprise Communications Broker also supports transmitting LDAP messages using the IPFIX Protocol for the Palladion Mediation Engine.

LDAP Failure Events

If an incoming registration to a primary phone number in Teams fails, the phone number is routed to the IP PBX. If failures occur during LDAP queries for all LDAP Servers, the Oracle Enterprise Communications Broker logs the failure to the log.sipd, and proceeds with normal configured routing policies, if available.

> **Note:**
>
> The Oracle Enterprise Communications Broker always establishes the TCP/TLS connection towards the configured LDAP server(s). If a TCP connection fails, the Oracle Enterprise Communications Broker continues to attempt to re-establish the connection.

An LDAP connection failure can be due to any one of the following events:

- Oracle Enterprise Communications Broker receives a CANCEL message (LDAP connection termination). The Oracle Enterprise Communications Broker detects this if it receives or issues an 'unbind' operation. The session is then closed down at TCP/TLS.

- Oracle Enterprise Communications Broker receives a call failure message from Teams (TCP/TLS socket termination). If either side receives a finish message (FIN) or reset message (RST), the TCP socket closes per standard behavior, which triggers the LDAP layer to detect connection failure. The Oracle Enterprise Communications Broker fails over to a secondary LDAP Server, if configured; otherwise it periodically attempts to reconnect to the Primary LDAP Server.
- Oracle Enterprise Communications Broker is unreachable and SIP session towards Teams times out. User is enabled for Teams but the Teams Server is unreachable by the Oracle Enterprise Communications Broker so a timeout occurs. When consecutive LDAP queries timeout, the Oracle Enterprise Communications Broker concludes that the LDAP session has failed, and then proceeds to terminate the TCP/TLS connection.

The number of consecutive queries that timeout before a connection is considered failed, and the number of successive query timeouts for each LDAP Server can be set via configuration.

Limitations using LDAP

The Communications Broker uses LDAP in the Active Directory when determining the destination of incoming calls. However, the Communications Broker has the following limitations when using LDAP:

- Supports LDAP sessions over the Communications Broker media interfaces only (i.e., not on wancom0).
- Supports LDAPv3 only.
- Establishes a session over the following connections only:
  LDAP over TCP - default

  LDAP over TLS (LDAPS)

Configuring LDAP for Routing

LDAP is the Protocol that the Active Directory uses for general interaction between and LDAP client and an LDAP server. You can configure the LDAP Server(s) in your network, and set the filters and the local policy that the LDAP Server uses when handling inbound Teams and PBX calls in the Enterprise core network.

You can use the following objects in the Web GUI to configure LDAP:

- LDAP Config—Configures the LDAP functionality on the Oracle Enterprise Communications Broker (i.e., name, state, LDAP servers, realm, authentication mode, username, password, LDAP search filters, timeout limits, request timeouts, TCP keepalive, LDAP security type, LDAP TLS profile, and LDAP transactions).
- Routing—Configures Active Directory attribute names, attribute format and regex extractions for routing SIP requests to the target's home agent. You configure this object for LDAP search queries in the Active Directory.
- Address of Record—Configures Active Directory attribute names, attribute format and regex extractions for identifying other addresses of record for the request URI and from. The Oracle Enterprise Communications Broker uses any AoR information provided by these queries to generate additional routes for the session, using the same process it used for the original request URI and from.

Support for FQDN in LDAP Configuration

You can use FQDN format while configuring LDAP servers that provides an easy solution to configure and manage multiple LDAP servers across different LDAP search bases. FQDN is

resolved into IP addresses. Communications Broker supports both A and SRV FQDN configurations.

Communications Broker resolves the IP addresses using the default realm. Configure the network-interface with the DNS parameters. Also, configure the associated Realm under the LDAP Configuration object. Communications Broker makes use of the existing DNS functionality and the associated realm configuration for FQDN resolution.

Communications Broker executes:

- An SRV query when no port is configured
- and an A query when a port is configured.

This provides the flexibility of managing multiple LDAP servers based on LDAP load balancing for A records, weights and priority for SRV records.

When you configure LDAP server as FQDN, here's the workflow executed by Communications Broker

- The order of resolved IP addresses is considered as the actual order of preference of A query responses. For an SRV query, the order is calculated based on priority and weights.
- Only a single FQDN LDAP server configuration is supported by Communications Broker.
- You can either configure static IP Addresses or FQDN in the LDAP Configuration. However, Communications Broker does not support a combination of FQDN and static IP address for a single configuration element.

Source Routing

The Communications Broker supports source routing based on agent hostname. You configure these routes by adding the hostname to the source agent portion of your route. When the Communications Broker sees this hostname in the FROM URI, it uses your source route to direct the traffic. You can also configure the Communications Broker to perform source routing on calling numbers that have a FROM header with a R-URI that contains an IP address or an FQDN. For this, the Communications Broker attempts to determine the hostname by searching for the address or FQDN in the UserDB. If it finds the entry, the Communications Broker inserts the hostname into the FROM and performs a route lookup. You enable source routing using the **Source Based Routing** parameter in the SIP configuration.

If the FROM of the INVITE has an IP address or FQDN instead of a hostname, the Communications Broker does not perform source routing by default. You enable the Communications Broker process that can resolve an IP address or FQDN to an agent's hostname, thereby allowing source routing on this signaling. To accomplish this, the Communications Broker matches the configured session agent hostname with the host portion of the From-URI in incoming INVITE. If there is an IP address in the From-URI, the Communications Broker performs a session-agent IP/FQDN match with the From- URI host.

Assuming you have enabled it, the Communications Broker performs routing using a sequence of configuration checks with source routing being the last. For this sequence, the Communications Broker:

1. Performs a lookup in the Dial-plan for the calling number.
2. Checks for a UserDB entry for the calling number.
3. If LDAP is enabled, performs an LDAP look-up on the host portion of the FROM URI.
4. Checks to match the host portion of the FROM URI with a configured session agent host.
5. Attempts to perform source routing using the host portion of the From-URI as the source agent.

After determining that source routing is required, the Communications Broker:

1. Checks the userDB for an entry that matches the calling number. If the calling number is present in the userDB, the Communications Broker fetches the source agent from userDB entry.
2. If Step (1) doesn't result in any entry, the Communications Broker performs an LDAP lookup for a matching entry. If the Communications Broker finds an entry, it fetches the source agent from that entry.
3. If Step (2) doesn't find an applicable entry, the Communications Broker checks the host portion of the From-URI for an IP address match.
4. If the received IP address doesn't match any of the configured hostnames, the call fails.
5. The Communications Broker performs a routing look up after IP address resolution.

This flexibility with the source agent parameter of a route also allows you to configure source routes using session agent groups (SAGs). In this case, the Communications Broker matches sessions coming from any of the agents included in the group, even if the calling number from the SAG is not in the UserDB.

Oracle recommends that you do not configure overlapping IP addresses to multiple Session Agents when using source routing. The Communications Broker allows you to configure the same IP address for multiple session agents. But if there is overlapping agent addressing, the Communications Broker only uses the first configured SA it finds for setting the source agent, which may not be the intent of you configuration.

Configuring Source Routing

You enable or disable Source based routing using the **Source Based Routing** checkbox in the SIP configuration. Navigate to this setting using **Configuration**, **System Administration**, **SIP Interface**, **Sip-Config**. If this checkbox is disabled, the Communications Broker does not use source IP address/FQDN matches for any incoming calls.

If you configure a routing source routing entry with a destination agent that is not set to `*` , the Communications Broker cannot match the destination agent with the request-uri IP address. Set source route entries with a destination agent of `*`.

You can also support SAG based source routing by configuring an applicable **manipulation-string** to your Session-Agent configuration to replace the configured string with the user-portion of the From URI. The **manipulation-string** configuration only applies if you have configured HMR.

REFER Source Agent Routing

Refer-Source-Agent-Routing simplifies complex call routing involving multiple SIP trunk providers with Communications Broker as the routing agent. It allows you to specify a routing table look-up based on either the source agent of the calling party or the source agent of the referring party. This ensures efficient call transfer between services.

This feature allows source agent-based routing table look-up specifically for terminating REFER scenarios. To enable this, configure the Session Agent with either refer-call-transfer enabled or set to dynamic with the matching dyn-refer-term. This ensures that the REFER is not proxied by the Communications Broker; instead, it is terminated on the Communications Broker, and a new INVITE is created based on the REFER message received.

After enabling the **refer-source-agent-routing** attribute, following the workflow:

1. The headers of the new INVITE after the REFER termination do not change. The overall call flow remains the same, except for the modification of the Source Agent as referring party.
2. When you enable **refer-source-agent-routing**, Dial Plan, User DB and LDAP look-up continue to be based on the originator's number without any change in the previous behavior.
3. Source agent-based look-up for an INVITE in response to the terminating REFER only applies to cases where routing entries are referred. It does not apply to cases where direct hops are identified based on user entries or registration cache
4. When you enable this feature, the Source Agent used to identify the source context for the INVITE will not match the source agent used for Routing. The call originator's context is used.
5. When you disable the **refer-source-agent-routing**, the look-up is based on the Source Agent of the Calling Party.

3
Registrar and Authentication

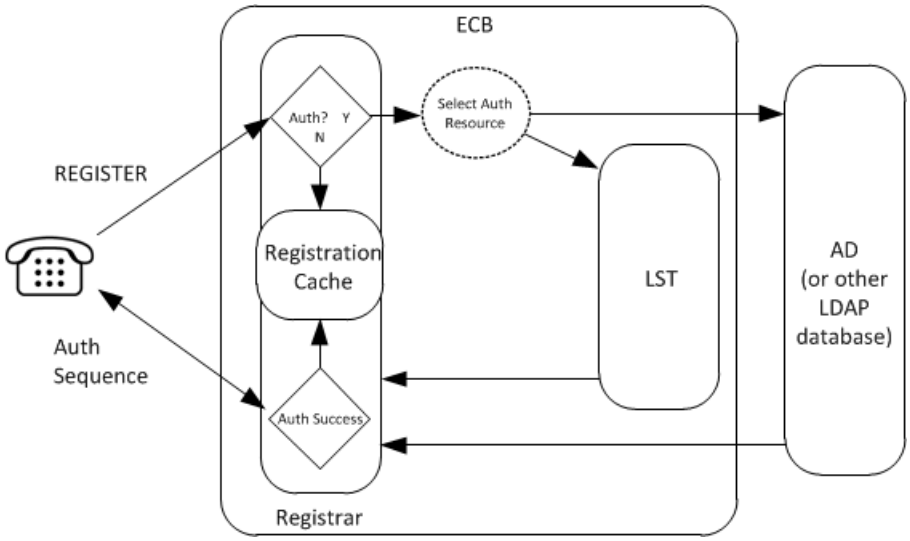
By providing a location service from within it, the Oracle Enterprise Communications Broker (Communications Broker) offloads related infrastructure from providing that information for every session. The Communications Broker can use SIP digest authentication to confirm service authorization and verify user registrations by way of internal or external mechanisms. If using an external mechanism for this purpose, some adaptation of that mechanism is required. You enable the registrar, configure the applicable domains (including serviced and digest domains) and, if required, define the authentication to use for all registrations by way of configuration on the GUI.

You enable the Communications Broker single registry service globally. When registration functionality is enabled, the Communications Broker registers endpoints rather than only caching and forwarding registrations to another device.

On receiving a REGISTER message, the Communications Broker checks if it is responsible for the domain contained in the Request-URI, as configured in the domains list. The Communications Broker begins registrar functions for all requests that match a configured domain.

When there is no authentication configured, the system adds every user that attempts to register to the registration cache. When you configure authentication, the system can authorize and verify the caller by way of the LST or an external LDAP resource. In these use cases, the system uses SIP digest to authenticate the caller, based on authentication information from the LST or LDAP. The following diagram and steps explain a call flow with authentication and interaction with LDAP resources, especially Active Directory.

1. A UA is fully registered after the system installs it in the registration cache.
2. The Communications Broker sends a 200 OK message back to the registering UA.
3. When a user registers with the registrar, the system looks for the To header AoR in the LST. If the LST contains a subscriber with the AoR (or username if no AoR specified) that matches, the system adds the universal number of the subscriber as an alias to the registration cache.



Register Refresh

When a UA sends a register refresh, the Oracle Enterprise Communications Broker 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 Enterprise Communications Broker sends a request to its source database (LST or LDAP) to retrieve authentication data once again).

Next, the Oracle Enterprise Communications Broker determines it can perform a local REGISTER refresh or if the source database needs to be updated. If any of the following 3 conditions exists for the re-registering UA, the Oracle Enterprise Communications Broker updates the database:

- The location update interval timer has expired—This value, configured in the sip registrar configuration element ensures that source database always has the correct Oracle Enterprise Communications Broker address by periodically sending request messages for each registered contact.
- The message's call-id changes while the **forward-reg-callid-change** option in the sip config configuration element is set. This covers the case where the UA changes the Oracle Enterprise Communications Brokers 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 Enterprise Communications Broker1, moves to Oracle Enterprise Communications Broker2 , and then returns to Oracle Enterprise Communications Broker1.

If the Oracle Enterprise Communications Broker updates the source 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 Enterprise Communications Broker just refreshes the access-side expiration timer; the registration cache expiration timer for that AoR begins its count again.

Proxy Registration

By default, the Oracle Enterprise Communications Broker (Communications Broker) rejects a REGISTER request from a domain for which it is not the registrar. You can enable the Communications Broker to proxy such registration requests by way of the **Proxy Registration** control in the **SIP Config** configuration.

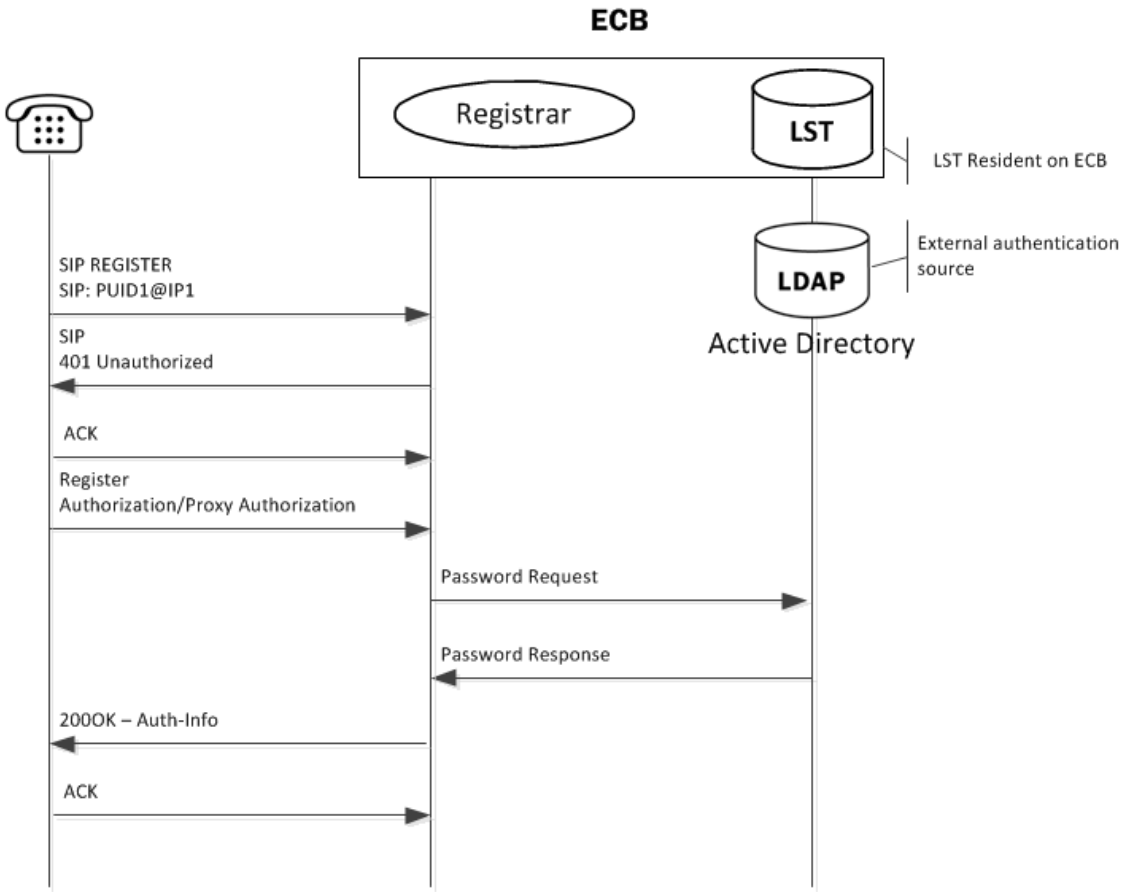
In the **SIP Config** configuration, select **Proxy Registration** to tell the Communications Broker to proxy the registration towards the intended registrar. When you deselect **Proxy Registration**, the Communications Broker responds with a **403: Unauthorized** message.

Message Authentication for SIP Requests

You can configure the Oracle Enterprise Communications Broker (Communications Broker) to authenticate REGISTER requests. The Communications Broker offers a single Registrar for

location services on user-specified listed domains. Registration may or may not include user authentication. If includes authentication, you can select a local, text-based resource called the Local Subscriber Table (LST) as an authentication source. You can also configure the Communications Broker as an LDAP client, allowing it to perform LDAP-compliant processes and retrieve authentication information from an external resource, usually Active Directory. The Communications Broker populates the registration cache with contacts for AORs upon successful authorization/authentication.

The Communications Broker uses SIP digest authentication as a means of challenging an end point for applicable registration attempts. The following diagram shows the overall authentication and authorization sequence, including the Communications Broker confirming the registration by way of an LST or an external LDAP server.



Authentication

To authenticate the registering user, the Oracle Enterprise Communications Broker needs the hash of the end station's password. It requests these from the local LST of an LDAP server by sending it an LDAP query for the configured field.

The hash consists of an MD5 hash made up of the following components:

```
MD5(username:digest-realm:password)
```

The transaction is conducted with the server defined in the Registrar configuration's credential retrieval method parameter. This parameter is populated with the name of the LDAP sever.

SIP Authentication Challenge

When the Oracle Enterprise Communications Broker 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 Enterprise Communications Broker. If the endpoint is registering, the Oracle Enterprise Communications Broker 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
```

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 Enterprise Communications Broker 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 Enterprise Communications Broker always sends a value of "MD5"

SIP Authentication Response

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

Authentication Check

At this point, the Oracle Enterprise Communications Broker has received an MD5 hash from the HSS and an MD5 hash from the UA. The Oracle Enterprise Communications Broker 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.

Retrieving Information from Active Directory

The Oracle Enterprise Communications Broker performs SIP Digest authentication against users attempting to register. It can use pre-configured information from Active Directory to perform such authentication. Access to Active Directory uses standard LDAP processes to retrieve the information needed and to offload the processing from other resources to the Communications Broker.

The Communications Broker can obtain registration authentication information directly from Active Directory when you modify the Active Directory schema to include the Oracle-specific attributes and object classes that the Communications Broker needs to authenticate users.

LDAP and Authentication

Lightweight Directory Access Protocol (LDAP) is the Protocol that the Communications Broker uses to perform queries to the Enterprise's Active Directory to validate registration attempts in the Enterprise network. Requests and responses are sent/received based on the Communications Broker's LDAP configuration. The Communications Broker's LDAP client queries an LDAP server, usually Active Directory for password information for a user attempting to register. This request and response process verifies that the user can get registration servers (authorization) and verifies that the user is who they say they are (authentication). Once both these stages complete successfully, the Communications Broker registers the user.

The Communications Broker, using LDAP, performs the following on a registration attempt:

- Creates an LDAP search filter based on the dialed number and the configured LDAP attributes.
- Sends an LDAP search query to the configured LDAP server.

You configure LDAP servers and filters, on the Communications Broker.

The Communications Broker keeps a permanent LDAP session open to all configured call servers. It sends an LDAP bind request on all established connections, to those servers. The first call server is considered the primary LDAP server, and all others are secondary LDAP servers. If a query request sent to the primary server fails, the Communications Broker sends the request to the next configured LDAP server, until the request is successful in getting a response. If no response is received by the Communications Broker, it replies to the registering endpoint with a (401? authentication failure?).

Configuring LDAP for Authentication

LDAP is the protocol that the Active Directory uses for general interaction between and LDAP client and an LDAP server. You can configure the LDAP server(s) in your network, and set the filters and the local policy that the LDAP server uses when handling inbound Teams and PBX calls in the Enterprise core network.

You can use the following objects in the Web GUI to configure LDAP:

- LDAP Config—Configures the LDAP functionality on the Oracle Enterprise Communications Broker (i.e., name, state, LDAP servers, realm, authentication mode, username, password, LDAP search filters, timeout limits, request timeouts, TCP keepalive, LDAP security type, LDAP TLS profile, and LDAP transactions).
- SIP Authentication—Configures the Active Directory attribute names for the Oracle Enterprise Communications Broker's query-digest-username-attribute and digest-hash-attribute fields. These fields specify where the Oracle Enterprise Communications Broker verifies authentication attempts.

See the section on Active Directory and Oracle ECM Routing for important information about:

- LDAP messages
- LDAP failure events
- Communications Broker limitations using LDAP

That information applies equally to the authentication functionality explained here.

4

Getting Started

Oracle® recommends that you review the topics in "Getting Started" before working with the system to ensure success with the tools and functions provided.

Accessibility Features

The following accessibility features are available in this release.

Keyboard Navigation

You can access GUI functionality using only the keyboard. Use the following keys to navigate:

- Tab key: Move to the next control, such as a dynamic target menu, navigation tree, content pane, or tab in a page. Tab traverses the page left to right, top to bottom. Use Shift +Tab to move to the previous control.
- F6 key: The tab key moves between controls, but does not move between controls after it has, for example, opened a widget toolbar in a Dashboard tile. Use F6 to set focus to the next control on these toolbars. Press again to remove focus.
- Up and Down Arrow keys: Move to the previous or next item in the navigation tree, menu, or table. Down Arrow also opens a menu.
- Left and Right Arrow keys: Collapse and expand an item in the navigation tree or a submenu.
- Spacebar: Activate a control. For example, in a check box, spacebar toggles the state, checking or unchecking the box.
- Enter: Activate a button.

> **Note:**
>
> Some functionality cannot be accessed through keyboard navigation.

Microsoft Screen Narration and Jaws

This product is compatible with common OS's screen reader programs.

Microsoft Magnifier

This product is compatible with common OS's screen magnifying features.

Configuring the Communications Broker for SDM

You can perform configuration and fault management on the Communications Broker and groups using the Communications Broker. Fault management by SDM includes the handling of SNMP traps and logs. Configuration management is based on software version, with each version able to specify which elements you can configure with SDM. The use of SDM for Oracle Enterprise Communications Broker also provides you with the ability to establish

consistent configuration management across multiple Oracle Enterprise Communications Broker deployments.

You must use the Transport Layer Security (TLS) protocol to secure the communications link between the Communications Broker and the Oracle Communications Session Delivery Manager (SDM). Note that the systems use Acme Control Protocol (ACP) for this messaging.

To configure the Oracle Enterprise Communications Broker to use TLS for this ACP messaging:

1. Configure a TLS profile. The **tls-profile** object is located under security, where you add certificates, select cipher lists, and specify the TLS version for each profile.
2. Configure the **system-config** element's **acp-tls-profile** parameter to specify this TLS profile.

```
ORACLE# configure terminal
ORACLE(configure)# system
ORACLE(system)# system-config
ORACLE(system-config)# acp-tls-profile
```

The **acp-tls-profile** parameter is empty by default, which means that ACP is disabled. When the **acp-tls-profile** parameter specifies a valid TLS profile, the Communications Broker negotiates a TLS connection with SDM. You must reboot your Communications Broker after configuring ACP over TLS.

To support this management, the Communications Broker generates an XSD file that specifies its configurable attributes. Using this XSD, SDM determines the configurable and non-configurable attributes on the Communications Broker and to determines what it can display on its configuration GUI.

See your software version's Release Notes for more information about which elements of the Communications Broker you can configure with SDM.

Upgrade Error Messages

The implementation of SDM configuration support required an architectural change that invalidates former configuration objects called templates. As the Communications Broker development progresses, upgrades may generate unexpected results that need troubleshooting. The Communications Broker logs the following applicable error messages, printed in log.web, when an error is observed during upgrade process and may present you with next steps while troubleshooting an upgrade:

- "Could not create new object for <xmlName>" Displayed when error occurs while creating new xmlElement.
- "Could not set attribute <attribute>:<reason>" Displayed when error occurs while setting an attribute in the new xmlElement.
- "Could not delete service for <templateName>, Reason - <reason>" Displayed when service instance deletion on specified template fails.
- "Could not delete for <templateName>, Reason - <reason>" Displayed when profile instance deletion on specified template fails.
- "Could not delete template <templateName>, Reason - <reason>" Displayed when template deletion on specified template fails.

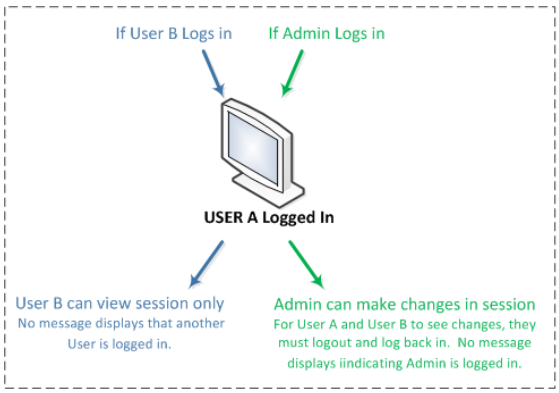
User and Administrator Access

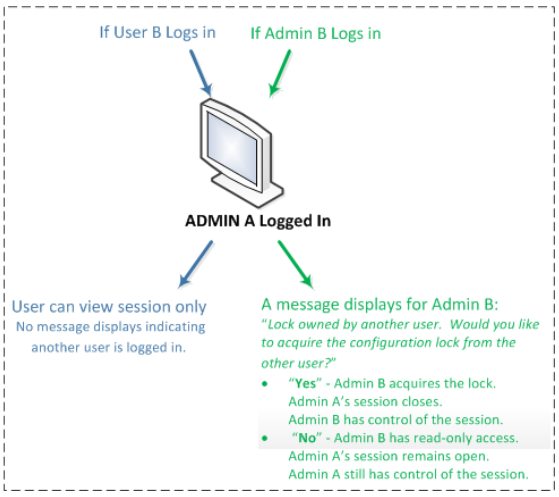
You can log on to the Web GUI using your Web browser. There are two types of user logons:

- **User** - Allows viewing (read-only) access to the Web GUI.
- **Administrator** - Allows Superuser access to the Web GUI.

Simultaneous Logons

The Web GUI allows the User and Administrator to log on simultaneously. Session availability to the User and Admin depends on which type of user is logged on to the session. The following illustrations depict and explain the system behavior when a User and an Administrator log on to a Web GUI session.





> **Note:**
>
> In the illustration where Admin A logged in first, the behavior is as described. If Admin B logs in first, then the system shows the lock message to Admin A. The first Admin to log in acquires the lock.

Up to five users can log onto the same session at the same IP address at the same time. Only one Administrator at a time can have full control of a simultaneous session.

RADIUS Server Roles and Access Privileges

The Web GUI supports RADIUS authentication functionality similar to a user signing on by way of Secure Shell (SSH) and SSH File Transfer Protocol (SFTP).

Available functions depend on the role that you assign to the "userclass" on the RADIUS server.

- When you configure the RADIUS server as userclass=admin, the system allows the Administrator full access to all features and functions after logging onto the GUI.
- When you configure the RADIUS server as userclass=user, the system limits User access to the following features and functions after signing on to the GUI.

- Full access to all System features and functions
- Can download the following files in System File Management:
  - Backup Configuration
  - Configuration CSV
  - Local Subscriber Table
  - Log
  - Audit Log
  - Playback Media
  - Software image
  - SPL Plug-in (SPL)

> **Note:**
>
> The "User" account cannot upload files in System File Management.

For more information about RADIUS support, see the *Administrative Security Guide*.

Log on to the Web GUI

You can log on to the Communications Broker as a User or an as Administrator, depending on your permissions.

You need your User name, Password, and optionally your passcode to log on. The system requires your passcode when two factor authentication is enabled. If your system Administrator configured the optional log on page message, the system displays the message after you enter your log on credentials. After reading the message, click **Close**, and the system displays the GUI. The first time you log on, the GUI displays the Configuration Assistant.

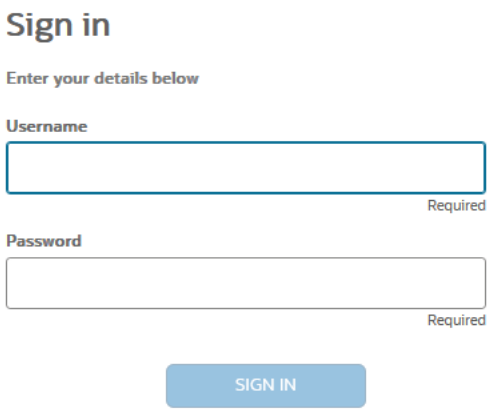
1. On a PC, open a supported Internet browser. See "Browser Support" in your software version's Release Notes for the list of supported browsers.
2. Start the GUI with either the HTTP or HTTPS log on.

```
http://<YourEcbManagementInterfaceIpAddress> previously configured.
https://<YourEcbManagementInterfaceIpAddress> previously configured.
```

> **Note:**
>
> Whether you log on using HTTP or HTTPS depends on the settings for your deployment.

3. In the Sign in dialog, enter your Communications Broker credentials, and click **Sign in**. The Login banner is displayed. For more information on the Sign in Banner, see Login Banner.

Figure 4-1 Sign in



4. If your Communications Broker requires two-factor authentication, do the following:
   a. In the **log on** dialog, enter your passcode.
   b. In the **Confirm** dialog, acknowledge the Last Log on information. Yes—allows you to login. No—ends the session.
5. Click **log on**.

> **Note:**
>
> If you want to open another Web GUI tab on the same Communications Broker in a new tab in the same browser, you can do so without logging on again.

Login Banner

After a successful user authentication and authorization, Communications Broker displays the login banner.

The Login Banner displays:

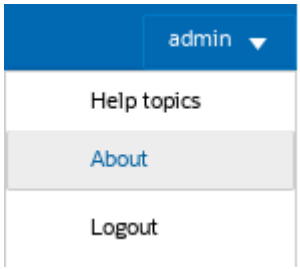
- Last login: Date and time that the current user last successfully logged-in.
- System last accessed: Date and time and user name of the last user who successfully logged-in.
- Unsuccessful login attempts: Date and time of the last five unsuccessful login attempts by the current user.
- Confirm reading: Needs confirmation from the current user. On clicking **Confirm**, the current user can completes login, and the audit-log activity for this user session is started.

If the current user clicks **Cancel**, the current user is logged out of the Communications Broker , and an audit-log entry is created.

The login banner also provides notification of impending password expiration.

Log Off from the Web GUI

To log off from the Web GUI, click **Logout** from the user menu in the upper right corner of the Web GUI.



The system logs you off and re-displays the log on page.

Service Provisioning

After the Oracle Enterprise Communications Broker is operational, network architects and communications service provisioning technicians specify call services using the controls available from the Service Provisioning icons available from the Configuration tab.

System Administration controls, also available from the Configuration tab, specify how to manage the system and are documented in the *Oracle Enterprise Communications Broker Administrator's Guide*.

Service Provisioning Configuration Objects

The following information describes the Oracle Enterprise Communications Broker (Communications Broker) Service Provisioning objects.

Configuration Object	Description
Agents	Add agents. An agent is usually a SIP-aware device that serves as a transit target and/or source for signaling managed by the Communications Broker. Agents are often specified as next-hops for the purposes of routing. Indirect agents, Communications Broker route termination points that require further routing to reach an end station are also configured here. In addition, configuration used to access ENUM servers is performed here.
Dial Plan	Add multiple dialing-contexts and dial-patterns. Dialing-contexts define the system behavior for calls placed to and from either a corporate of geographic focus. Dialing-contexts include multiple dial-patterns, which define the normalization required to most effectively manage diverse signaling structures.
Policy Entries	Add policies and define the conditions of their use.

Configuration Object	Description
Routing Table	Add service routes. Route-entries specify paths for signaling traffic, allowing you to specify policy and cost for traffic based on source and/or destination.
User Entries	Users - Add user and other key phone numbers associated with the enterprise. The user database serves as a directory for phone numbers that need communications services. This database can specify each entry's source context, which can provide a starting point for processing the logic behind a user's call treatment. It also can specify each user's agent, providing a physical location for routing user's calls.

Web GUI Tools

The Web GUI provides various tools to conduct operations. Some tools apply globally, while others apply only to a specific function on the tab.

The User Menu

Oracle Enterprise Communications Broker displays the User menu in the upper right-hand corner of the branding bar, and is labeled with the currently logged in user's name.

The user menu provides the following links.

- Help Topics—Links to the embedded Help system.
- About—Displays information about the software, for example, the release and build numbers.
- Logout—Logs you off of the Web GUI.

Help

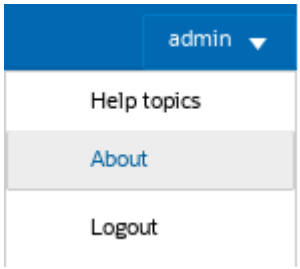
The logged on user button on the Web GUI displays the following information:

- **Help Topics**. Online Help system containing topics about the tasks that you can perform on the Web GUI.
- **About**. Oracle notices and disclaimers, Oracle terms and restrictions, and third-party notices.

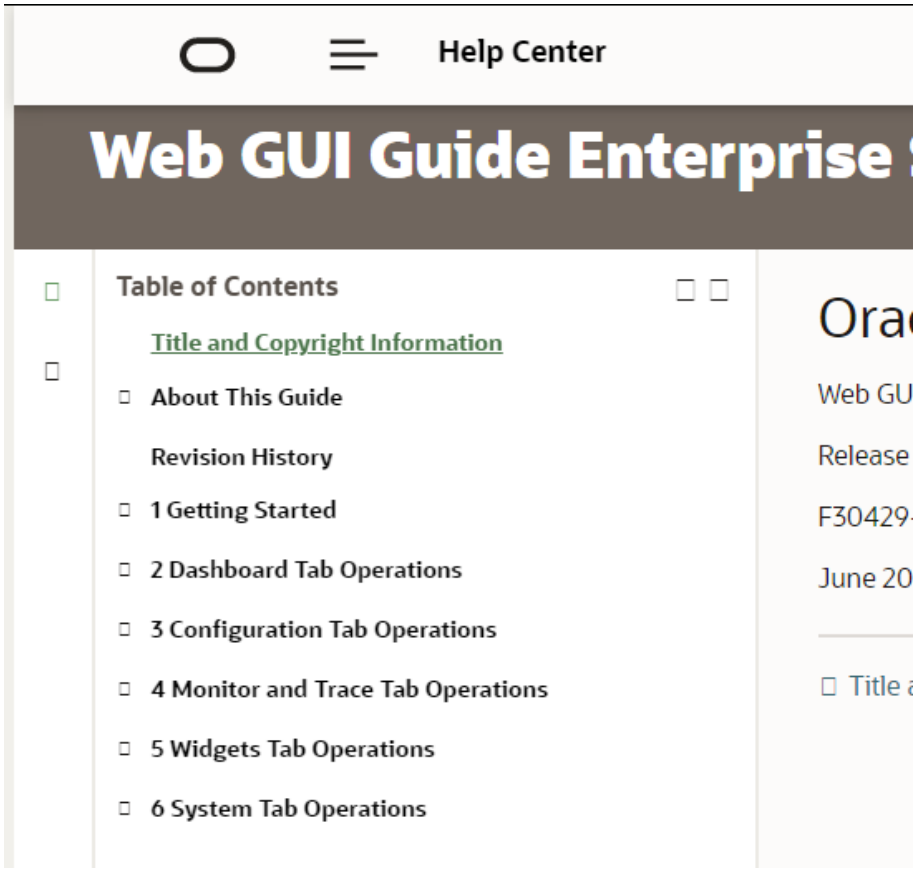
Help Topics

Communications Broker includes an online Help system the provides the concepts and procedures that you need to know for working with the system.

From any page in the GUI, click **admin**, **Help topics** in the upper right corner of the page.



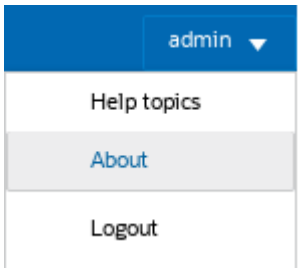
After you click **Help topics**, the system opens a new Tab containing help information.



About This Software

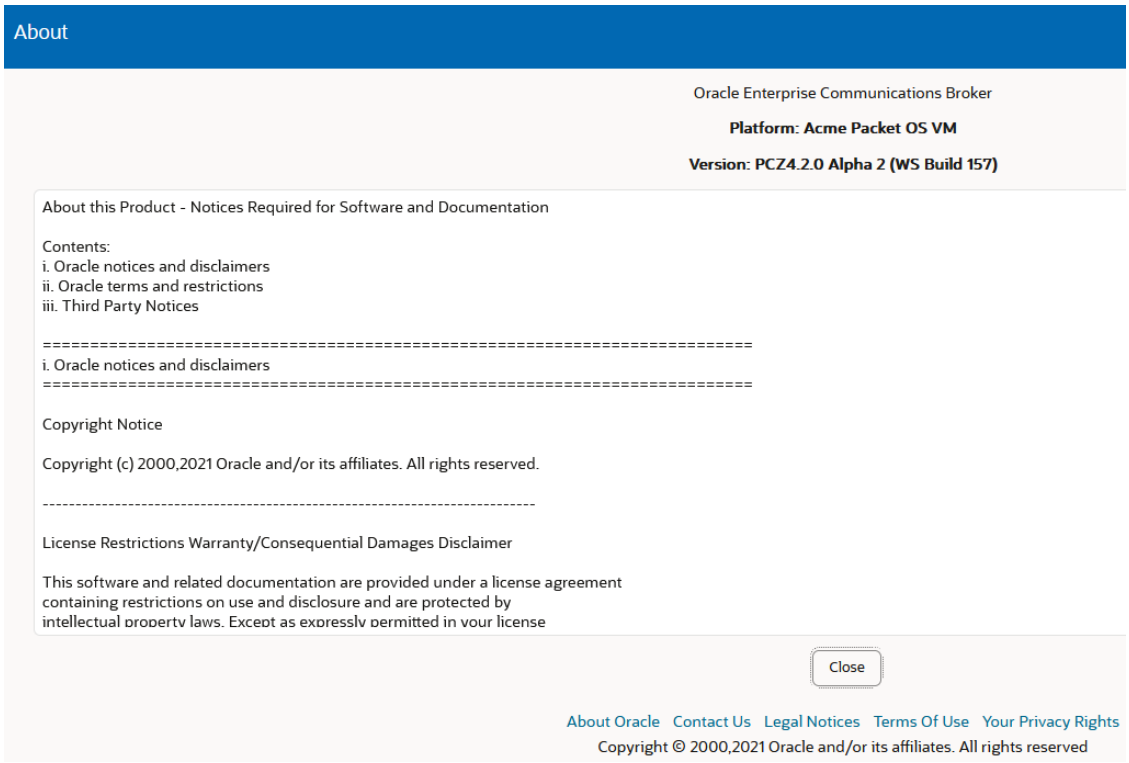
You can display information about the software currently installed on the Communications Brokerthat you are logged on to by clicking **About** on the **User** menu located in the upper right corner of the Web GUI.

The following screen capture shows the location of **About** on the User menu.



The following screen capture shows the information that the About link displays, such as notices and disclaimers, oracle terms and restrictions, third-party notices, and copyright dates, as well as links to About Oracle, Contact Us, Legal Notices, Terms of Use, and Your Privacy Rights.

Figure 4-2 About the Software



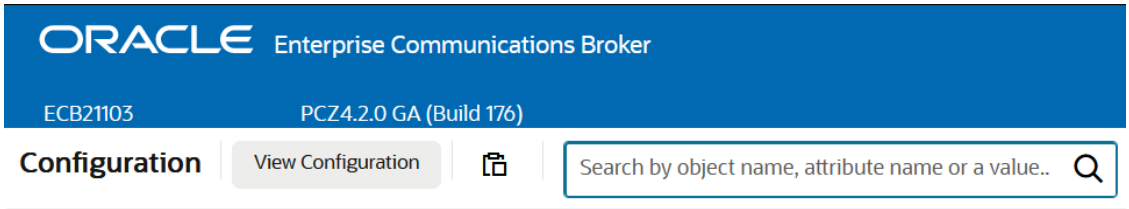
Configuration Search Tool

The Configuration tab provides two levels of search operations. You can use the **Search** tool on the Configuration tab to search for configuration objects. Within a multi-instance configuration object, you can use **Search** to find a specific configuration instance on the list.

The **Search** tool, displayed as a magnifying glass, on the Configuration page header opens the Search text box where you enter the text to search for, such as a configuration object name, an attribute, or value. You can enter the full name of a configuration object to find a specific object or you can enter a phrase to see a list of all objects that contain that phrase. For example, you can enter enable-snmp-auth-traps to find that object or you can enter SNMP to see a list of all configuration objects with SNMP in their name.

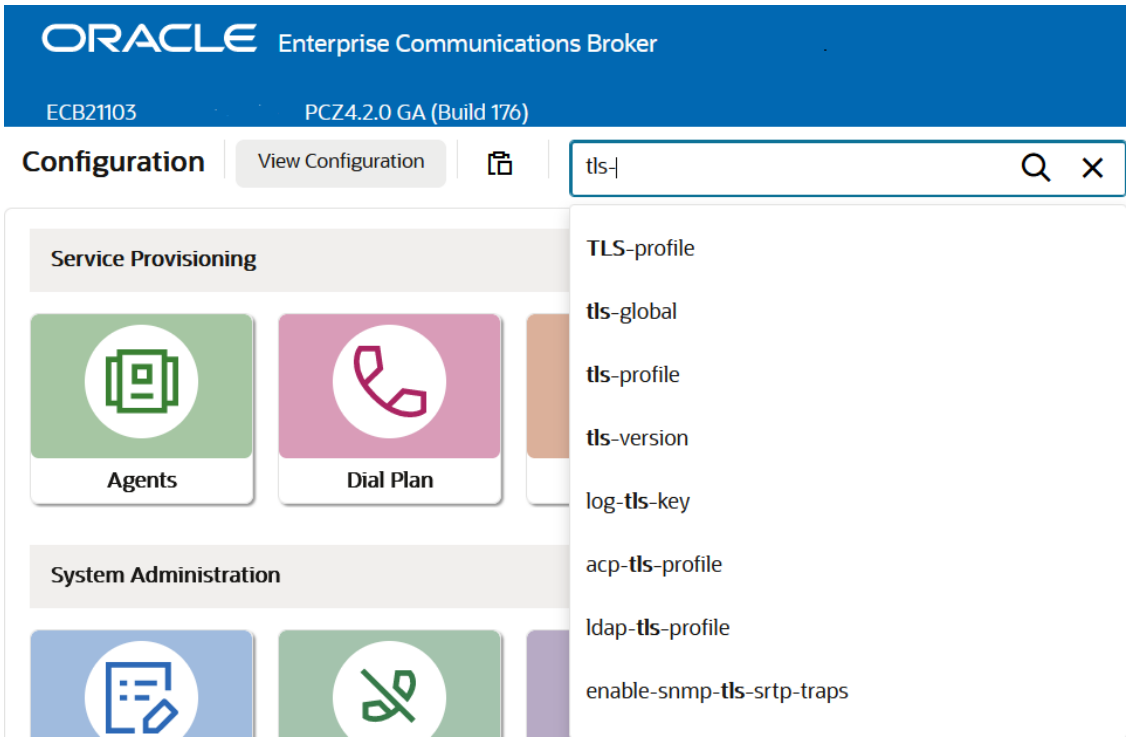
The **Search** magnifying glass is located in the middle of the Configuration tab.

Figure 4-3 Configuration Search



When you click the **Search** tool, the system displays the Configuration Search text box next to the Magnifying Glass, where you enter the configuration object name, attribute, or value that you want to find. When you search for a phrase, rather than a complete object name, the system displays a list of all objects that contain the phrase.

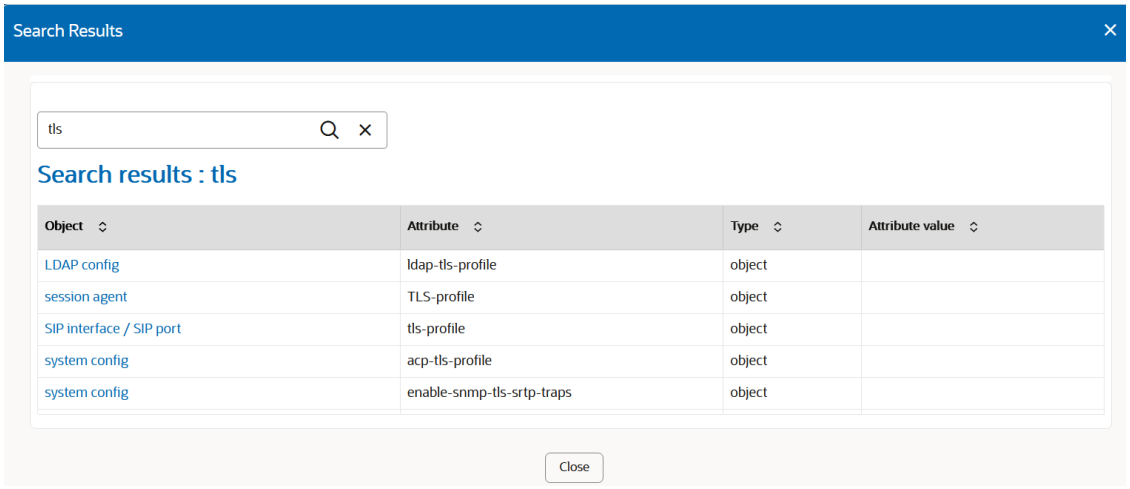
Figure 4-4 Configuration search



Multi-instance configuration objects display a list of configurations and the table includes a text box for local search within the table. You can enter all or part of the configuration name.

When you search for a particular configuration, the results display only that configuration and the system adds a clear entry icon, displayed as an 'x', to the **Search** field.

Figure 4-5 Search Results



When you click the **X** button, or click **Close**, the system resumes displaying all of the configurations on the list.

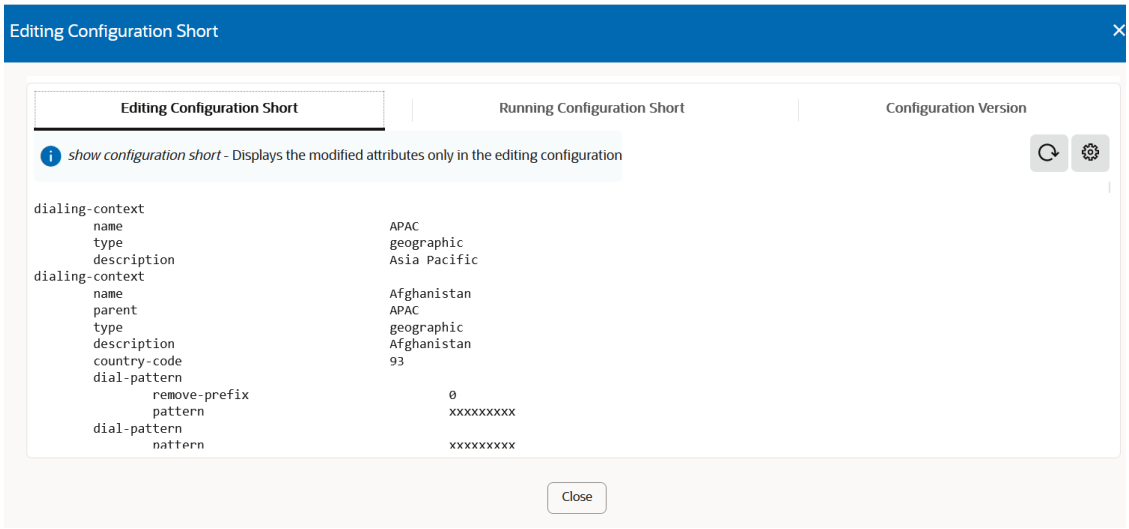
Show Configuration for a Configuration Element

The Show Configuration is located in the top right corner of the heading section in the center content.

For any configuration element, click **View Configuration** to view configuration information in a new dialog box. In this dialog box, you can:

- Refresh
- Edit Configuration Short - Displays the modified attributes only in the editing configuration
- Running Configuration Short - Displays the modified attributes only in the running configuration.
- Configuration Version. - Configuration version number table

Figure 4-6 Show Configuration



Customize the Page Display

You can customize the display of Web GUI pages that display data in tables by selecting which columns display and the sort order of the rows.

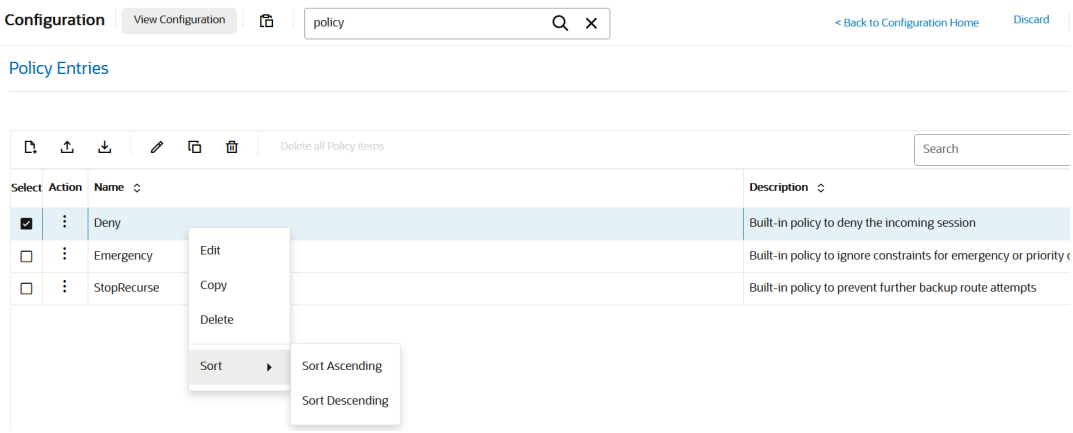
1. Place the cursor on a row, and right-click.

   The system displays the customize menu.

2. Click **Sort**.

   The system displays the sort choices.

Figure 4-7 Customize the Display using the Sort Menu

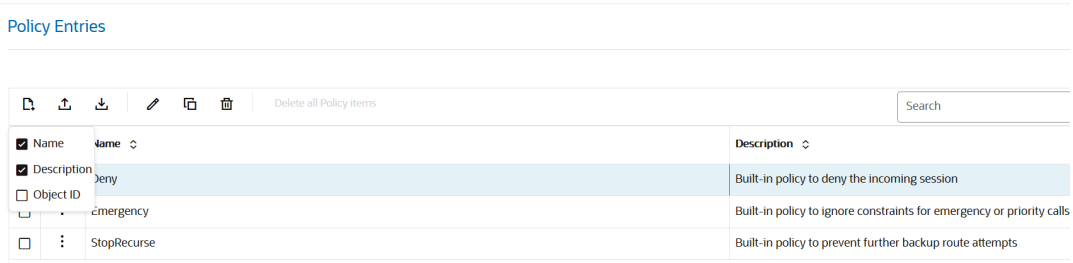


You can also Sort lists alphabetically by hovering over the column header's right corner and clicking the arrow icon when it appears.

3. Do one of the following:
   - Click **Ascending**.
   - Click **Descending**
4. Right-click any column header.

   The system displays the menu, which lists the column types that are available for the table displayed.

Figure 4-8 Customize the Display



5. Select the columns that you want displayed or deselect the columns that you do not want displayed.

Tool Tips

A tool tip is a brief description of a parameter on a configuration screen in the Web GUI, that also includes the default setting and valid values for the corresponding parameter.

To view a tool tip, place the cursor in the parameter field.

To close the tool tip, click away from the parameter.

Configuration Tools and Behavior

The Oracle Enterprise Communications Broker (Communications Broker) Web GUI provides the following tools for working with configurations. Some tools are located in the navigation pane and others are located at the top of the center pane.

The following screen capture shows the locations of all of the **Configuration** tab controls.

From any sub object on the configuration icon's page, there is a Back to Configuration Home link.

Figure 4-9 The Configuration Tab Display

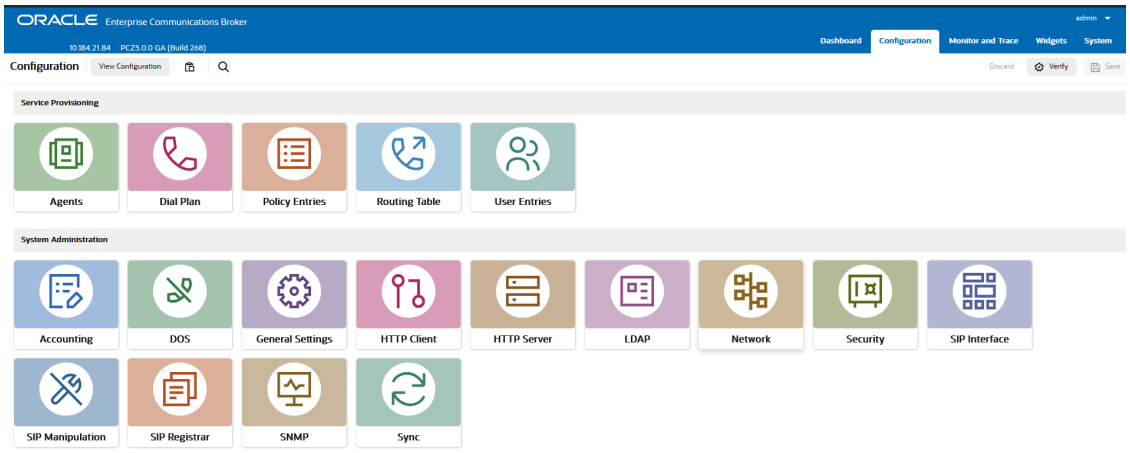
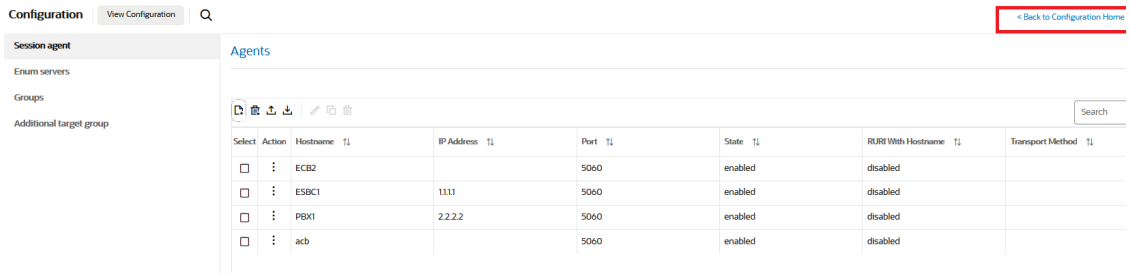
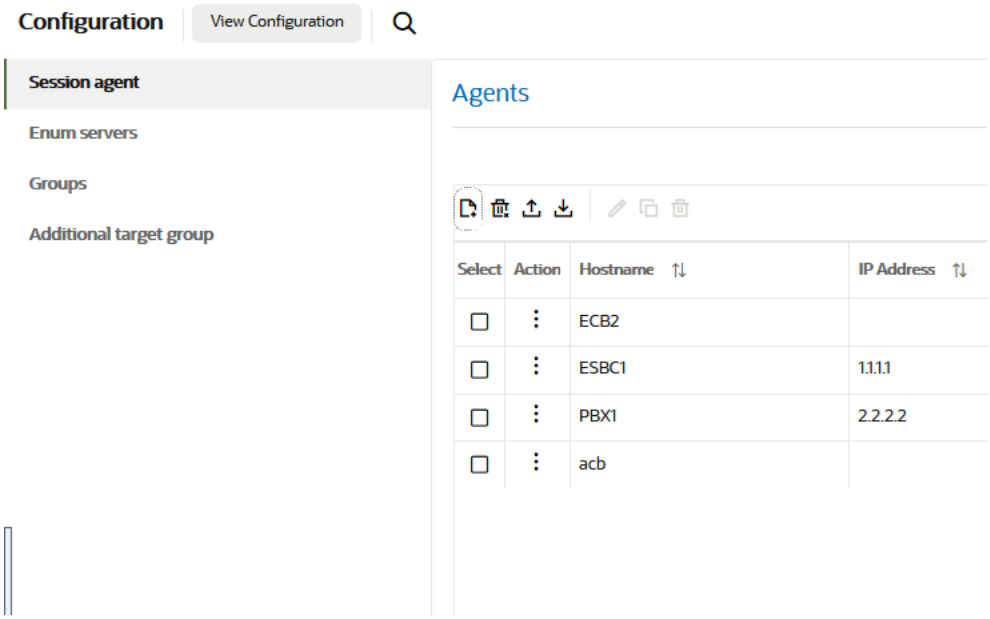


Figure 4-10 Back to Configuration Home



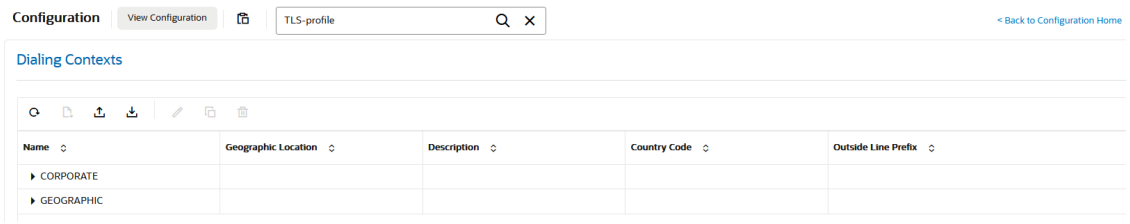
The navigation object on the left side of an icon's configuration dialog is only available for objects that include sublinks.

Figure 4-11 Navigation object on the left side



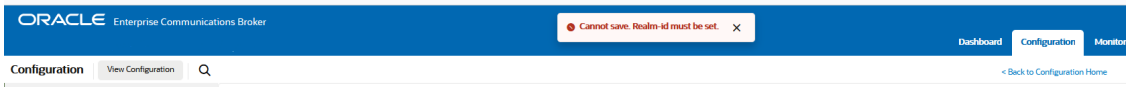
The navigation object on the left side of an icon's configuration dialog does not display for objects that have no sublinks.

Figure 4-12 Navigation object on the left side with no sub links



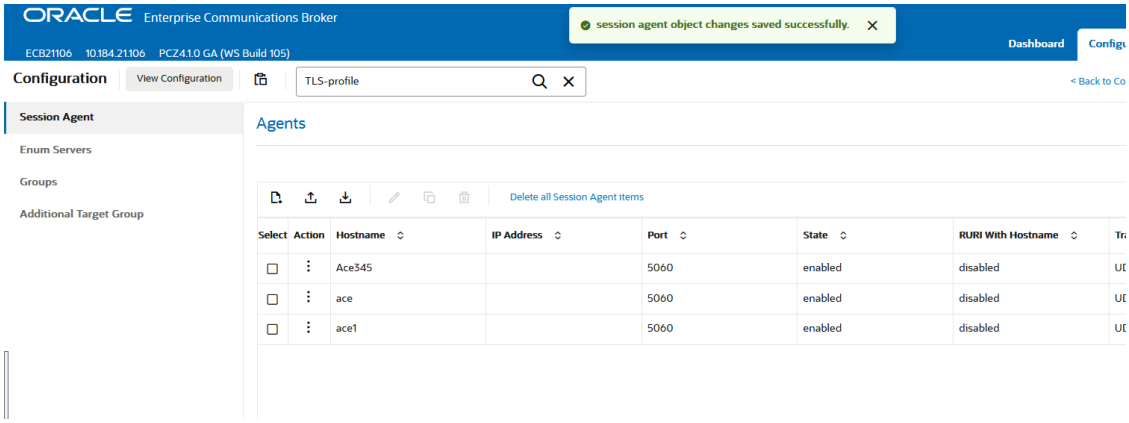
The Communications Broker displays Error and Success messages at the top of each configuration dialog.

Figure 4-13 Error Panel



This image displays a configuration success message.

Figure 4-14 Success Message Panel



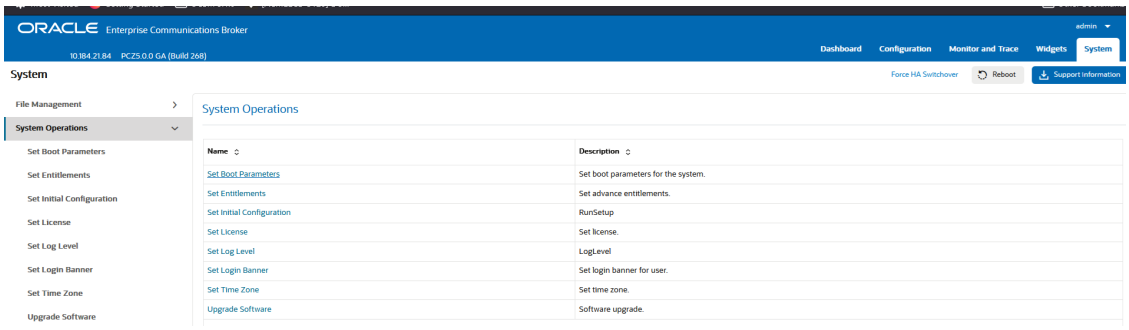
Controls in the Navigation Pane

The navigation pane is on the left side of configuration lists and remains there as screens change within the center pane. You can think of the Navigation Pane as similar to a table of contents. It displays links to configuration objects or drop down controls to drill into sub-categories of configuration objects.

For example, the System Tab includes a navigation pane that shows Systems Operations and File Management category links. When you click the Systems Operations link, the system opens a drop-down showing all System Operations functions and a table with function descriptions in the center pane. When you click a function, from either the Navigation of Center pane, the system displays that function's configuration dialog.

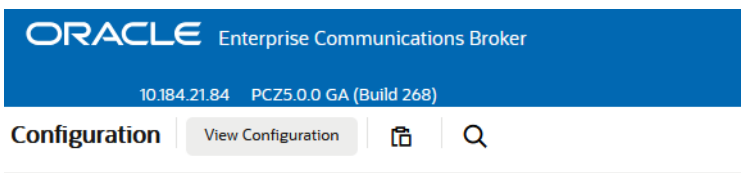
This screen capture shows the list of Systems Operations configurations and provides their descriptions.

Figure 4-15 List of Systems Operations configurations



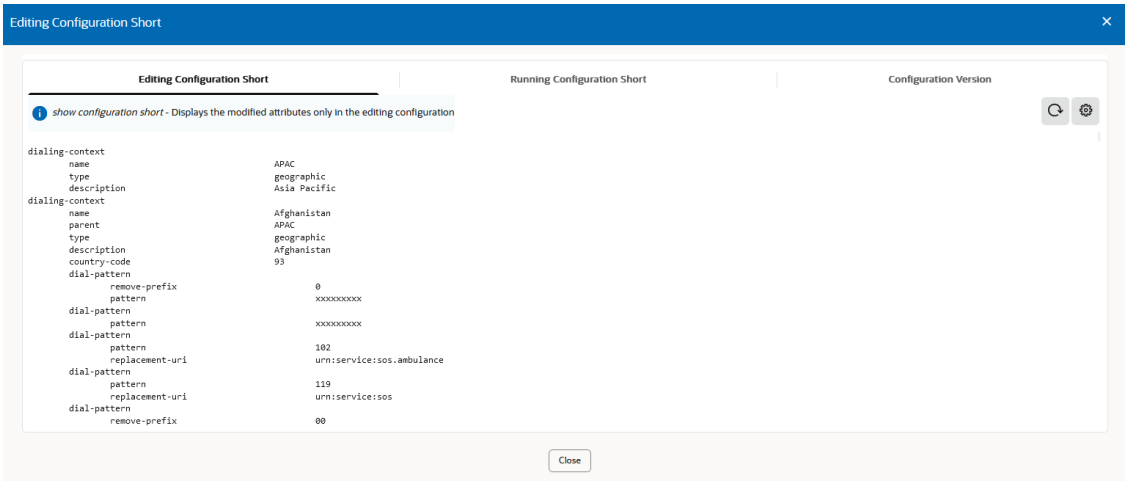
This screen capture shows the View Configuration button.

Figure 4-16 View Configuration button



The Editing, Running and Configuration Version commands are available from the top of this dialog.

Figure 4-17 View Configuration



Controls in the Center Pane

The controls located at the top of the center pane on a configuration page help you manage configuration objects.

This screen capture shows the controls located at the top of the center pane on any configuration pane.

Figure 4-18 Center Panel

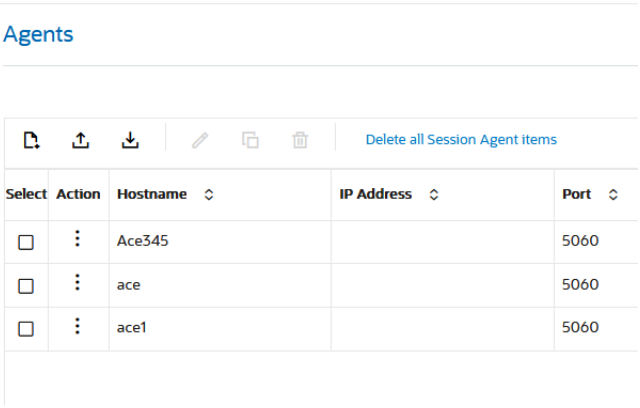
These controls are described in the following table.

Paste	Opens the Paste Configuration window, in which you can paste the configuration details for one or more objects and save it.
Save	Use to save the current configuration session. Upon Save, the system displays a prompt giving you a choice of whether or not to activate the configuration. If you do not activate the configuration, you can continue to make changes and Save again. When finished, you can save and activate all of the configuration changes.
Verify	Use to confirm that the configuration is valid before you save it.
Discard	Discard the current changes to configuration.

Controls for Multi-Instance Objects

The controls located at the top of a list of multi-instance objects help you manage the objects on the list. The following screen capture shows the controls.

Figure 4-19 Controls located at the top of the center pane on any multi-instance list



Add	Use to add another instance of the configuration object with one or more parameters set differently.
Delete All	Use to delete all instances. **Caution:** You cannot select a sub-set of instances and delete only those. The system deletes all instances, regardless of how many you select.
Upload	Use to upload a configuration file in CSV format.
Download	Use to download a configuration file in CSV format.
Search	Use to search the list of multi-instances for a specific instance. When you enter the name of an instance, the Web GUI displays it at the top of the list.
Edit	Use to edit the selected configuration element.
Copy	Use to copy the selected configuration element.
Delete	Use to delete the selected configuration element.

Set System Operations

The **System Operations** link under the **System** tab displays a list of procedures, for performing selected configuration procedures for the Communications Broker.

The Configuration Helpers combine related configuration for you to perform entire, specific system tasks:

Set Boot Parameters	Specify the boot file and the boot parameters.
Set Entitlements	Set the number of sessions that a license entitles you to.

Set Initial Configuration	Configure a new Communications Broker or reconfigure an existing one. Includes configuring High Availability.
Set License	Enter the license number for a feature that requires a license.
Set Login Banner	Customize the text on the Web GUI log on banner.
Set Time Zone	Select the time zone for the deployment.
Upgrade Software	Upload a newer version of the software.

> **Note:**
>
> ACLI output sometimes refers to System Operations using the common term, wizard.

Set Boot Parameters

The Oracle Enterprise Communications Broker (Communications Broker) requires you to enter the necessary parameters to boot the system in your deployment.

You can set the Communications Broker boot parameters from the Set Boot Parameters on the Web GUI.

1. Access the Set Boot Parameters Procedure: **System**, **System Operations**, **Set Boot Parameters**.
2. In the Set Boot Parameters dialog, enter the following information:

Boot File	Name of the image file.
IP Address	Enter the IP address of the Communications Broker.
VLAN	Range: 0-4095
Net Mask	Enter the net mask IP address in dot decimal format. For example, 255.255.0.0.
Gateway	Internet address of the boot host. Leave blank if the host is on the same network.
IPv6 Address	Enter the IPv6 address that you want to use.
IPv6 Gateway	Enter the IPv6 gateway that you want to use.
FTP Host IP	Enter the IP address of the FTP host.
FTP Username	Enter the FTP username for the FTP user on the boot host.
FTP Password	Enter the FTP password for the FTP user on the boot host.
Flags	Hexadecimal. Always starts with 0x. See "Configurable Boot Loader Flags."
Target Name	Name of the Communications Broker, as displayed at the system prompt.
Console Device	Enter the type of console device. For example, VGA.
Console Baud Rate	Select a console baud rate from the drop-down list.

Other	For miscellaneous and deployment-specific boot settings.

3. Click **Complete**.

   The system displays a success message.

4. Click **OK**.

Configurable Boot Loader Flags

You may configure the following boot flags in the boot loader:

- 0x04 - disables autoboot timeout (ap3820 and ap4500 only)
- 0x08 - extend autoboot countdown timer to 15 seconds
- 0x40 - use DHCP for wancom0 (VM Edition only)
- 0x80 - network boot using TFTP instead of FTP

Set Entitlements Procedure

Use the Set Entitlements Procedure to enter the maximum number of sessions that your license allows.

- Note the session limit number from your license.

You can launch the Set Entitlements on the Web GUI.

1. Access the Set Entitlements Procedure: **System**, **System Operations**, **Set Entitlements**.
2. In the Set Entitlements dialog, do the following:

Admin Security	Enabling this feature activates enhanced security functions. Once saved, you cannot revert the security without resetting the system back to factory default state.
Admin Security With ACP/NNC	Inherit Admin Security rules with added restrictions for stronger passwords. Click **Enable** to enforce stronger password rules and restrictions.
Session Capacity	Set the number of licensed sessions. Range is from 0 to 100000 .
Data Integrity (FIPS 140 - 3)	Enable or disable Data Intergrity in compliance with FIPS 140 - 3.

3. Click **Complete**.

   The system displays a success message. It is recommended to do a Save/Activate.

Set Initial Configuration Procedure

Use the Set Initial Configuration procedure to perform the initial configuration on an unconfigured system and to change the configuration on a configured system. During the configuration, you select the scope of configuration that you want to perform, define the boot parameters, opt to set a VLAN, and configure features such as High Availability (HA) and access to the Oracle Communications Session Delivery Manager (OC SDM). A valid license is required to run the Set Initial Configuration procedure.

> **Note:**
>
> ACLI output sometimes refers to System Operations, using the common term, wizard.

You can perform the Set Initial Configuration procedure from the ACLI or the GUI. Use the ACLI for the first system initialization:

- Unconfigured system—Immediately after first boot, setting passwords, product and entitlements, you perform system initialization by running the **run setup** command. When the initial configuration is complete, the system saves the configuration, activates the configuration, and reboots. The system does not backup the initial configuration of an unconfigured system.
- Configured system—From the System tab on the Web GUI, click the System Operations link and then the Set Initial Configuration link. When the re-configuration is complete, the system saves a backup of the existing configuration, saves the new configuration, activates the new configuration, and reboots. The backup is stored in /code/bkups.

Before you can configure the Communications Broker, the procedure requires you to make the following selections that determine which configuration parameters the procedure displays.

Communications Broker Mode: Standalone or High Availability	• If you select Standalone, you can begin configuring the parameters displayed. • If you select High Availability, the GUI also provides the Communications Broker role, Primary or Secondary, to the display.
Communications Broker Role: Primary or Secondary	If you selected High Availability for Communications Broker Mode: 1. Select Primary, and configure the displayed parameters. 2. Select Secondary, and the GUI displays only the parameters needed by a secondary.

> **Note:**
>
> Unlike other Communications Brokers, which provide 2 management interfaces and 2 media interfaces, the Acme Packet 1100 provides 1 management interface and 2 media interfaces. When configuring HA, the configuration dialogs for the Acme Packet 1100 differ from the other Communications Brokers because you must create a second, virtual management interface. For creating the second management interface, the HA dialogs on the Acme Packet 1100 contain more attributes than the dialogs for the other Communications Brokers. Regardless of the Communications Broker model, the path through the Set Initial Configuration procedure to the HA dialogs is the same as described in this topic.

First System Initialization

The system requires an initial configuration of attributes, such as modes and IP addresses, before it can function in the network.

Use the **run setup** to define the attributes for the system immediately after first system boot, and after you have set passwords, product and entitlements.

1. Observe the entire first boot sequence.
2. Login to the Oracle Enterprise Communications Broker.
3. Set passwords.
4. Note that the product is selected automatically as Communications Broker.
5. Set entitlements, using the **set entitlements** command.
6. Initialize the system using the **run setup** command.

   The system saves the configuration, activates the configuration, and re-boots.

Reconfigure the System

You can reconfigure the system from the Web GUI.

Use the Set Initial Configuration operation to change the initial configuration on a configured system, for example, change attributes such as IP addresses and modes.

1. Log on to the system.
2. Access the Set Initial Configuration dialog: **System**, **Systems Operations**, **Set Initial Configuration**.
3. Run the Set Initial Configuration operations and change the attributes, as needed.
4. Click **Complete**.

   The system saves a backup of the existing configuration, saves the new configuration, activates the new configuration, and automatically re-boots.

- (Optional) Reconfigure the system objects.

Set License Procedure

Use the Set License procedure to enter the serial number for your license.

- Obtain the license, which includes the serial number, for the feature that you want to add to the deployment. See your sales representative for information about specific licenses.

You need the license number for the following procedure.

1. Access the Set License procedure: **System**, **System Operations**, **Set License**.
2. In the Set License dialog, enter the license serial number in the Add license field.
3. Click **Complete**.

   The system displays a success message.

Set Login Banner Procedure

Use the Set Login Banner procedure to add customized text to the log on page.

You can customize the log on page by adding text to help the user. For example, Welcome to <company name> <business unit> <location> session border controller <device name>.

1. Access the Set Login Banner procedure: **System**, **System Operations**, **Set Login Banner**.
2. In the Set Login Banner dialog, enter the text that you want to display on the log on page.
3. Click **Complete**.

   The system displays a success message.

Set Time Zone Procedure

The system requires a setting for time zone.

You can set the system time from the Set Time Zone procedure on the Web GUI. You can select a time zone or Coordinated Universal Time (UTC).

1. Access the Set Time Zone procedure: **System**, **System Operations**, **Set Time Zone.**
2. From the drop down list, select one of the following:
   - Time zone by locale
   - UTC
3. Click **Complete**.

   The system displays a success message.

Upgrade Software Procedure

You can upgrade the system software with the Upgrade Software Procedure on the Web GUI. .

Use the Upgrade Software procedure to perform the following tasks:

- Check the system health before the upgrade
- Download new software
- Change boot parameters
- Reboot the system

The system requires a reboot after the upgrade for the changes to take effect.

1. Access the Upgrade Software procedure: **System**, **System Operations**, **Upgrade Software** .
2. (Optional) In the Upgrade Software dialog, click **Verification**, and do the following:
   - Click **View Synchronization Health**, and confirm that the system components are synchronized.
   - Click **View Configuration Version**, and note the Current Version and Running Version.
   - Click **View Disk Usage**, and confirm that the system has enough free space.
3. In the Upgrade Software dialog, do the following:

Upload method	Select an upload method from the drop-down list.
Software file to upload	Browse to the file to upload.
Reboot after upload	Select to reboot the system after the upgrade.

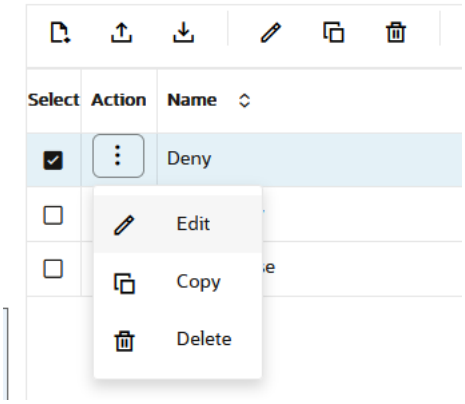
4. Click **Complete**.
   - If you did not select **Reboot After Upload**, the system displays a message stating that a reboot is required for the changes to take effect.
   - If you selected **Reboot After Upload**, the system displays a message stating that it is about to reboot.
   - The system performs the file transfer and any boot parameter changes. If you selected **Reboot After Update**, the system reboots.

Edit, Copy, and Delete Configurations

You can edit, copy, and delete multi-instance configurations by way of the controls that the Web GUI displays on each multi-instance configuration page. The edit and copy functions act only on a single instance of a configuration. The delete function can act on either a single instance or all instances.

To edit, copy, or delete a single multi-instance configuration, select the configuration and right-click or click the instance's Action button. The Web GUI displays the edit, copy and delete menu.

Figure 4-20 Right-click Menu



You can delete multiple objects using the checkbox under the select column. You select multiple rows and click the delete icon from the toolbar.

When you click Delete, the system displays a confirmation dialog before performing the operation. When you click either Copy or Edit, the GUI displays the configuration dialog.

Configuration States and Behavior

The Communications Broker uses three states of memory to manage configuration changes, which allows you to make changes and decide when you want them to take effect.

At any time, the following three versions of the configuration can exist on the Communications Broker.

- Editing—The editing configuration is the version that you are making changes to from the Web GUI. The editing version is stored in the Communications Broker volatile memory. The editing version cannot survive a system restart.
- Saved—The saved configuration is the version of the editing configuration that the system copies into the non-volatile memory when you click **Save** on the Web GUI. The changes do not take effect on the Communications Broker until you activate the saved configuration. Note that the system does not load the saved, but unactivated, configuration as the running configuration upon restart. The Saved configuration requires activation, first.
- Running configuration—The running configuration is the configuration that the system is using. When you activate the saved configuration it becomes the running configuration. Most configuration changes can take effect upon activation. Some configuration changes require a system restart. Upon restart, the system loads the running configuration.

The process for saving and activating a configuration, includes the following steps.

1. **OK**—All configuration dialogs display an **OK** button that saves changes to the editing memory. If you restart before the next step, the Communications Broker does not save the changes.
2. **Verify**—(Optional) Oracle recommends verifying the validity of the configuration before saving because saving shows any errors after saving. Verify displays the configuration changes and any errors found, before saving.
3. **Save**—The **Save** button on the Web GUI toolbar verifies the configuration, saves the current configuration to the last-saved configuration, and displays errors. Save stores the configuration on the Communications Broker. The system displays any errors at the bottom of the Configuration page.
4. **Activate**—After you finish making one or more configuration changes, **OK** and **Save** from the last configuration dialog that you need to edit at this time. The system displays the Confirmation dialog containing the **Activate** button. When you click **Activate**, the Communications Broker activates all of the saved configuration changes and saves the new configuration to the running configuration. If you cancel the activation function, the Communications Broker saves the configuration in a file and does not change the running configuration. You can continue to make changes to the configuration.

Configuration Error Messages

If you save a configuration that contains errors, the system displays the following error message: *There were errors! Are you sure you want to activate the configuration?*

The system displays a list of errors at the bottom the page. Click an error to go to the location in the configuration where the error occurred and edit the configuration as needed.

Severity	Identifies the level of severity that the Oracle Enterprise Session Border Controller assigns to the error. Valid values are: • ERROR—Means that the issue identified in the Message column is not correctly configured or it does not exist. You can still verify, save, and activate the configuration if this severity exists. • WARNING—Means that the configuration contains invalid information for the element field identified in the Message column. You can still verify, save, and activate the configuration if this severity exists. • CRITICAL—Means that a critical error occurred in the configuration and you cannot verify, save, or activate until the error is corrected. The Message column indicates the element field where the error has occurred.

Message	Identifies the element field where the error, warning, or critical error occurred, and the reason for the error.
Object	Identifies the element and the field for that element where the error occurred.
Attribute Name	Identifies the attribute within the element where the error occurred.
Other	Identifies any other pertinent information relating to the error.

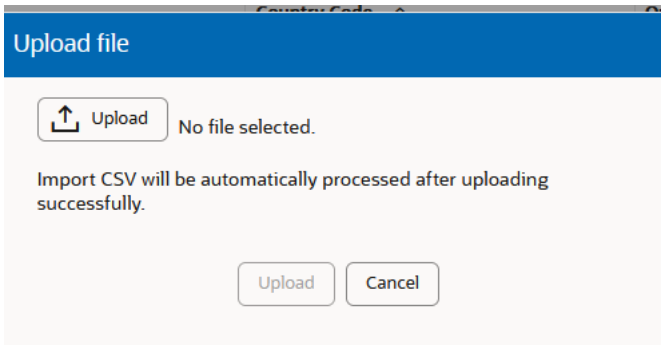
Uploading and Downloading Configuration File Elements

The Oracle Enterprise Communications Broker allows you to upload and download files containing certain key elements of your configuration that are more easily managed separately from the overall system configuration. Such elements include:

- Dial Plans
- Dial Patterns
- User Database Entries
- Route Database Entries
- SIP Manipulation Rules

The corresponding dialogs for the items in the preceding list include controls for uploading and downloading files, depending on which action you choose. When you click the Upload button, for example, the system displays the Upload file dialog, which includes a Browse button. When you click Browse, the system opens a browse dialog, from which you can select the file that you want to upload. Download behavior is similar. The system performs the file transfer to the correct system directories in the correct format without requiring any intervention. The following screen capture shows an example of the Upload File dialog.

Figure 4-21 Upload File



Using Tag Fields

The Communications Broker provides a configuration element data field referred to as a tag. You enter information into the tag field for descriptive and grouping purposes. You can establish your own criteria for labeling configuration elements with these tags. Tag fields have no operational effect on signaling services.

The following configuration objects display the Tags text field:

- Users
- Routes

You can enter any text that you want into the field and you can apply as many tags to a configuration object as needed. You can filter the element list searches using tags as a means of organizing these objects. Applicable element list search fields include a down arrow that exposes a tag drop-down list, from which you select the tag on which to filter the list. Tags have no operational function other than supporting this kind of filtering.

Dashboard Tab Operations

The Oracle Enterprise Communications Broker (Communications Broker) provides a web-based Dashboard that can display SIP data statistics in Widgets to help you monitor and manage the system. The Communications Broker collects only SIP data for the Dashboard Widgets, including the default CPU and Memory Widgets. For this reason, you must set up a valid SIP configuration before the Communications Broker can display any data on a Dashboard Widget.

The Dashboard can display up to eighteen Widgets. Each Widget can display up to 100 data samples in intervals of 1 hour, 1 minute, or 1 second. You can select a chart, graph, table, web form, or text for the display, depending on the Widget. You can customize the Dashboard by adding, deleting, and moving the Widgets. You can refresh the statistics displayed on the Dashboard and you can reset the Dashboard to its default display. The default display includes:

- Highest CPU Usage
- Current Memory Usage
- Historical Memory Usage
- Alarms

The Dashboard page provides the following controls:

Figure 4-22 Dashboard

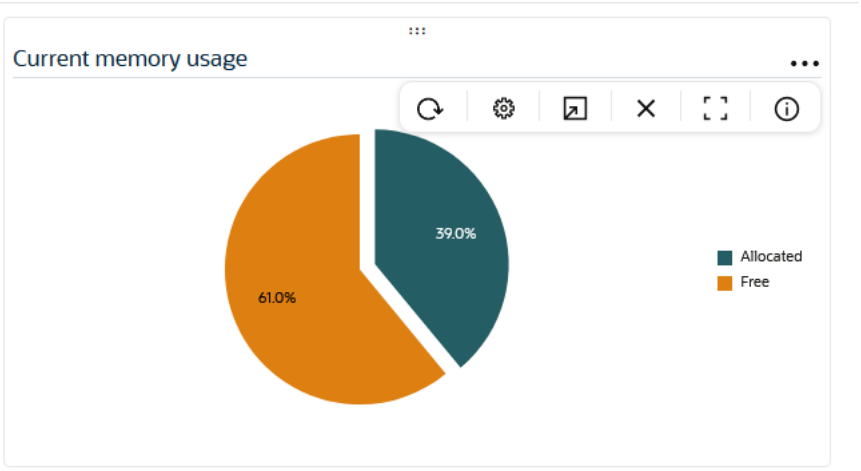
1. Refresh—Updates the data in all of the Widgets on the Dashboard.
2. Reset—Resets the Dashboard to display the default Widgets and removes all other Widgets from the Dashboard.
3. Widgets drop down —
   - Add a Widget - Displays a list of Widgets that you can add to the Dashboard.
   - Upload Widgets - Upon upload, the system backs up your existing widgets file to `/code/dashboard` and updates your Dashboard with the new file.
   - Download All Widgets - Downloads Widgets to `settings-admin.xml`.

A Widget can display a table, text, a pie graph, or a line graph depending on the type of data and the purpose of the display. For example, the SIP Realms All widget displays an actionable table and the Recording widget displays static text. You can access the list of widgets from either the navigation pane on the Monitor and Trace tab or from the Dashboard page by clicking Add Widget.

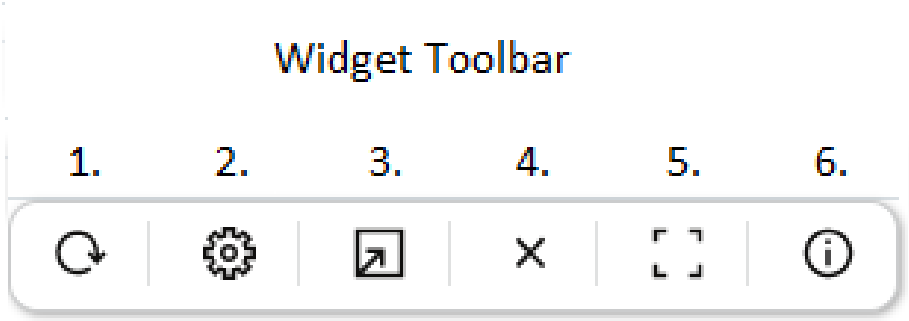
Most of the Widgets automatically display any available data when you click the name of the Widget, but some Widgets require further input. Such Widgets include a Settings button in their display that launches a parameters dialog were you specify the data to display. For example, the Realm Specifics Widget requires you to set the name of the realm and the auto refresh interval.

Each Widget contains the controls that you need to manage the particular Widget, according to the purpose of the Widget. The controls display when you hover over the three ellipses in the upper right corner of the Widget. The following screen capture shows an example of the tool bar that displays in a Widget.

Figure 4-23 Manage Widget



To learn what each control does, hover over the control.



The following list describes the Widget controls.

1. Refresh—Updates the data in the Widget.
2. Settings—Displays the dialog where you set the data sampling parameters for the Widget. When you click the icon, the Widget displays the Settings dialog, where you can set the auto-refresh interval and other parameters that apply to the particular Widget.
3. Export—Downloads the data displayed in the Widget.
4. Remove—Removes the Widget from the Dashboard. When you remove all Widgets from the Dashboard, the GUI displays the following message: " Your Dashboard is empty, please add a Widget or reset to restore the original Dashboard."
5. Maximize—Displays the Widget in full-screen size.

6. Show Information—Describes the data display. For example, in the Current Memory Usage Widget, the information icon says, "Pie graph displays current percentage of free and allocated memory."

> **Note:**
>
> The operation of Widgets, such as those that require the SIP Session module, may affect system performance. The system displays a warning when you add a Widget that may affect performance. Oracle recommends adding such Widgets at a time when the performance impact will not degrade service.

Add a Widget to the Dashboard

You can add up to eighteen Widgets to the Web GUI Dashboard to display SIP and System statistics to help you monitor and manage the system.

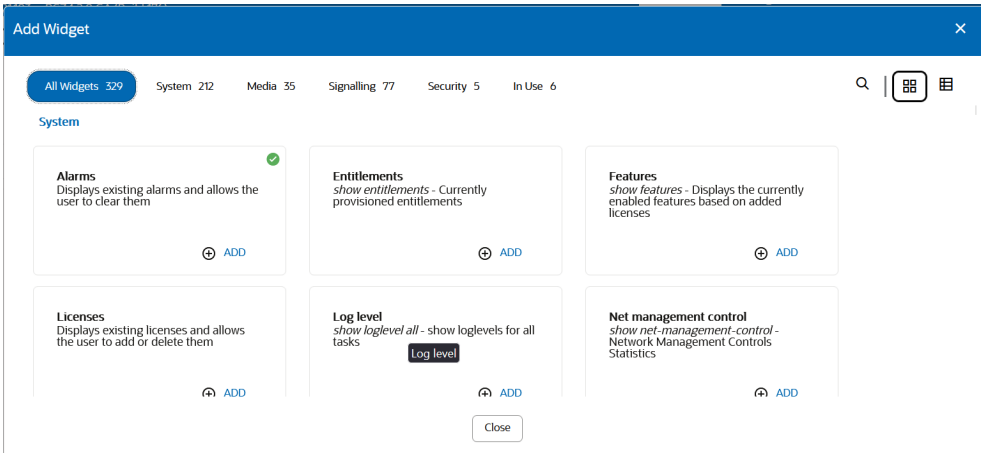
The system does not require a re-boot after performing the following procedure. The system adds the Widget to the Dashboard right away.

1. On the **Dashboard** page, scroll down and click **Add Widget**. icon.

   The Communications Broker displays the Add Widget dialog in card layout. Note the following aspects of the Add Widget dialog:

   - Tabs across the top of the dialog allow you to filter to specific widget categories, including All Widgets, System, Media, Signaling, and In Use.
   - The In Use tab lists widgets already added to the dashboard.
   - The Search (Magnifying Glass) icon provide a dialog for you to type in widget names, which the system finds for you.
   - There are two widget layouts :
     - Card layout—Shows widgets that match the selected filter as text boxes.

Figure 4-24 Card layout for Widgets



     - Table layout—Shows widgets that match the selected filter in lists.
   - Both layouts add a green marker to tell you that the widget is already on the Dashboard.

2. On the **Add Widget** page, use your preferred dialog layout and filters to locate the widget you want to add.

   The Web GUI displays the Widget name in the center pane.

3. In the center pane, under **Command** (See the far, right column header.), click **Add**.

   The system displays a success message.

> **Note:**
>
> If the system displays a warning that adding this Widget requires enabling the SIP.Message module, the system enables the module when you add the Widget.

4. Click **OK**.
5. Click **Close**.

   The system displays the newly added Widget on the Dashboard right away.

See "Configure Data Sampling Settings for a Dashboard Widget."

Configure the Data Sampling Settings for a Dashboard Widget

To display SIP and System statistics on a Dashboard Widget, the system requires a setting for how often to refresh the display. You can use the default interval or select one from the Auto-refresh interval drop-down list on the Widget.

Confirm that the Widget that you want to configure is on the Dashboard. If not, see Add a Dashboard Widget

Some Widgets also display the Table Name drop-down list, where you can set the data sampling frequency. For example, you might configure the Widget to refresh the display every 40 seconds and to display the data samples in one minute increments.

1. Click the **Dashboard** tab.
2. On the Widget, hover over the ellipses to display the toolbar, and click the **Settings** icon.
3. Select a Widget display refresh frequency from the **Auto-Refresh Interval (seconds)** drop down list.
4. If the Widget displays the **Table Name** drop-down list, select a data sampling increment for the Widget display. Valid values: Seconds | Minutes | Hours.
5. Click **OK**.
6. Click **Close**.

View A Widget That is Not on the Dashboard

When you want to see a Widget that is not on the Dashboard, go to the Widgets tab and select the Widget.

1. Access the **Widgets** tab.
2. In the navigation pane, click the name of the Widget that you want to see.

   The system displays the Widget.

3. (Optional)—Add the widget to the Dashboard, if the Dashboard is not fully populated.

Widgets Page

The Widgets Tab provides a place where you can select and view graphical displays of data and statistics about Communications Broker in the form of a widget. Most widgets correspond to an equivalent ACLI `show` command. A widget can display a list, a table, a graph, or text depending on the type of data and the purpose of the display.

The **Widgets** page displays a categorical list of widgets in the left-hand navigation pane and provides their descriptions in the center pane along with the corresponding ACLI show command, when one exists. You can populate the Favorite Widgets list with the widgets that you use the most often. The following screen capture shows the Widgets page display.

Some widgets display static information, while others display actionable information. For example, the IP Connections widget displays a static list of internet connections, and the Licenses widget displays a list of licenses along with controls to Add and Delete licenses. Some widgets immediately display any available information when you click the name of the widget, for example, Entitlements. Other widgets require you specify a few settings before displaying information, for example, Agent Individual. Most widgets allow you to set the Auto Refresh Interval.

Each Widget displays the particular controls used for its purpose. The following table lists all of the possible Widget controls and describes their purpose. The system displays a variable set of tool icons at the right-side top of Widgets. You click these icons to set the controls that each widget supports.

Figure 4-25 Widgets Page

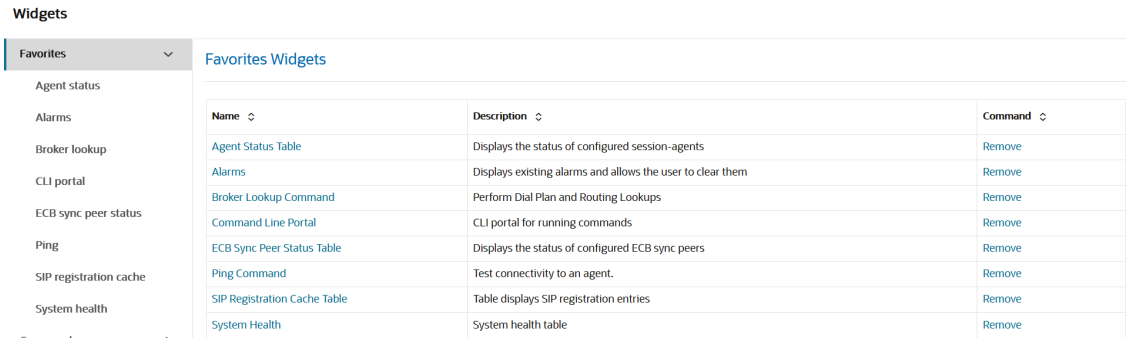


Figure 4-26 Widget control icons

Actions performed using these icons are described in the following table.

Tool	Description
Refresh	Update the data in the Widget.
Add the View to the Dashboard	Add the Widget to the dashboard.

Tool	Description
Add to Favorites	Add the widget to the Favorite Widgets list.

Widgets and Descriptions

For each **show** command that you can use from the ACLI to see Oracle Enterprise Communications Broker Sync, Media, Signaling, and System data about the Communications Broker, the system provides a corresponding Widget on the Web GUI. Widgets display the data in a graphical or textual format.

A Widget can display a table, text, a pie graph, or a line graph depending on the type of data and the purpose of the display. For example, the SIP Realms All widget displays an actionable table and the Recording widget displays static text. You can access the list of widgets from either the navigation pane on the Monitoring tab or from the Home page by clicking Add Widget.

Most of the Widgets automatically display any available data when you click the name of the Widget, but some Widgets require further input. Such Widgets include a Settings button in their display that launches a parameters dialog were you specify the data to display. For example, the Realm Specifics Widget requires you to set the name of the realm and the auto refresh interval.

> **Note:**
>
> You must set up a valid SIP configuration before the Communications Broker can display any SIP data in a Widget, including the default Widgets.

The following tables list all of the widgets in the left column. The middle column displays the corresponding ACLI **show** command, if one exists. The right column describes the data that the Widget displays. Note that a few widgets do not correspond to a **show** command.

The following tables list the Widgets in the same order as displayed on the GUI, which is not alphabetical.

- ECB Sync
- Media
- Signalling
- System
- Security

ECB Sync

Widget	Show Command	Description
ECB Sync Peer Status Table	No show command	Displays the following: • Peer • Status • Receiver Sync State • Receiver Uptime • Receiver Refresh Timer • Receiver Restart Timer • Transmitter Sync State • Transmitter Uptime • Transmitter Refresh Timer • Transmitter Notify Queue

Media

Widget	Show Command	Description
MBCD	show mbcd	MBCD status
MBCD Add	show mbcd add	MBCD Add statistics,Yes
MBCD Media Recorders	show mbcd media-recorders	Dynamic Media Recording Servers
MBCD Modify	show mbcd modify	MBCD Modify statistics
MBCD Monmsrp	show mbcd monmsrp	MBCD MSRP Monitor statistics
MBCD NAT	show mbcd nat	MBCD NAT statistics
MBCD Notify	show mbcd notify	MBCD Notify statistics
MBCD Redundancy	show mbcd redundancy	MBC Redundancy Statistics
MBCD Subtract	show mbcd subtract	MBCD Subtract statistics
Monthly Minutes	show monthly-minutes	Monthly minutes information for a specified realm
NAT By Index	show nat by-index	Displays the specified range of entries in the NAT table, with a maximum of 5024 entries. The default range is 1-200. The range corresponds to the line numbers in the table, not to the number of the entry.
NAT Flow Info Srtp Stats	show nat flow-info srtp statistics	Display global statistics for srtp flows,No
NAT Info	show nat info	Display NAT table information
NAT in Tabular	show nat in-tabular	Displays a specified range of entries in the NAT table.
Realm Specifics	realm-specifics	Display all realm-specific configuration based on the specified realm-id
Realm Summary	show realm	Displays the realm summary statistics
Survivability	show survivability	Display status information related to survivability
Survivability ack	show survivability ack	Survivability ACK statistics
Survivability Agents	show survivability agents	Session Agent Statistics

Widget	Show Command	Description
Survivability All	show survivability all	Display all Survivability Statistics
Survivability Bye	show survivability bye	Survivability BYE statistics,
Survivability Cancel	show survivability cancel	Survivability CANCEL statistics
Survivability Info	show survivability info	Survivability INFO statistics
Survivability Interface	show survivability interface	Interface Statistics
Survivability Invite	show survivability invite	Survivability INVITE statistics
Survivability Message	show survivability message	Survivability MESSAGE statistics
Survivability Notify	show survivability notify	Survivability NOTIFY statistics
Survivability Options	show survivability options	Survivability OPTIONS statistics
Survivability Other	show survivability other	Display other Survivability method statistics
Survivability Prack	show survivability prack	Survivability PRACK statistics
Survivability Publish	show survivability publish	Survivability PUBLISH statistics
Survivability Realms	show survivability realms	Realm Statistics
Survivability Refer	show survivability refer	Survivability REFER statistics
Survivability Register	show survivability register	Survivability REGISTER statistics
Survivability Status	show survivability status	Survivability Server Status
Survivability Subscribe	show survivability subscribe	Survivability SUBSCRIBE statistics
Survivability Update	show survivability update	Survivability UPDATE statistics
Temperature	show temperature	Displays the temperature in Celsius for all given components with temperature sensors
Version Acme	show version acme	Display ACME version information
Version Secinfo	show version secinfo	Display security-related information
Voltage	show voltage	Display current voltage readings
Xcode Api Stats	show xcode api-stats	XCode API statistics
Xcode Dbginfo	show xcode dbginfo	Debug Info
Xcode Dsp Events	show xcode dsp-events	Dsp Events
Xcode Dsp Resource	show xcode dsp-resource	Dsp Resources
Xcode Session All	show xcode session-all	Sessions

Signaling

Widget	Show Command	Description
Agent Details	showsipd agents	Display statistics related to defined SIP session agents
Agent Details	show sipd agents	Displays statistics related to defined DIP session agents.
Agent Groups	showsipdgroups	Display cumulative information for all session agent groups
Agent Groups	show sipd groups	Displays cumulative information for all session agent groups.
Agent Individual	showsipd agents <agent name>	Display statistics related to the entered SIP session agent Displays the status of configured agents.

Widget	Show Command	Description
Agent Individual	show sipd agents <agent name>	Displays statistics related to the specified SIP session agent.
Agent Status Table	NA	Displays the status of the configured Agents.
Agent Status Table	show sipd agents-status	Displays the state of the agent and active inbound and outbound calls.
Built In SIP Manipulation	show built-in-sip-manipulations	Displays all built-in sip-manipulations
Client Trans	showsipd client	Display statistics for SIP client events when the SBC is acting as a SIP client in its B2BUA role
Client Trans	show sipd client	Displays statistics for SIP client events when the SBC is acting as a SIP client in the B2BUA role.
DNS	show dns	Displays statistics for the DNS configuration.
DNSALG	show dnsalg	Displays stats summary of all dnsalg agents
DNSALG	show dnsalg	Displays stats summary of all dnsalg agents
Dynamic Monitoring Filters Global	show monitoring dynamic-monitoring-filters global	Display all global dynamic capture filters,
Dynamic Monitoring Filters Int Ev	show monitoring dynamic-monitoring-filters int-ev	Display all dynamic capture filters on interesting events
Dynamic Monitoring Filters Realm	show monitoring dynamic-monitoring-filters realm	Display all dynamic capture filters on realms
Dynamic Monitoring Filters Session Agent	show monitoring dynamic-monitoring-filters session-agent	Display all dynamic capture filters on session-agent
Dynamic Monitoring Filters Summary	show monitoring dynamic-monitoring-filters summary	Display dynamic capture filters for all mainfilters
ECB Sync Remote Registration	spl show sip ecb-sync registrations	Displays the address of record for the sync agent.
ECB Sync Remote Table		Displays the remote registrations for each Communications Broker.
ENUM	show enum	Displays ENUM statistics.
ENUM All	show enum all	Displays stats summary of all ENUM Agents
ENUM Sipd	show enum sipd	Displays stats summary of all sipd ENUM Agents
ENUM Stats	show enum stats	Enum Statistics,Yes
ENUM Stats All	show enum stats all	Displays stats summary of all ENUM Agents
ENUM Stats H323d	show enum stats h323d	Displays stats summary of all h323d ENUM Agents
Forked Sessions	show sipd forked	Displays forked sessions statistics.
Interface Individual	show sipd interface	Displays SIP interface statistics for the specified realm.
Interface Summary	show sipd interface	Displays all SIP interface statistics.
LDAP	show ldap	Displays LDAP statistics.

Widget	Show Command	Description
LRT	show lrt	Displays the Local Routing Table statistics.
Method Ack	show sipd ack	Displays all SIP ACK method statistics.
Method Bye	show sipd bye	Displays the SIP BYE method statistics.
Method Cancel	show sipd cancel	Displays all SIP CANCEL method statistics.
Method Info	show sipd info	Displays the SIP INFO method statistics.
Method Invite	show sipd invite	Displays the SIP INVITE method statistics.
Method Message	show sipd message	Displays the SIP MESSAGE statistics.
Method Notify	show sipd notify	Displays the SIP NOTIFY statistics.
Method Options	show sipd options	Displays the SIP OPTIONS statistics.
Method Prack	show sipd prack	Displays the SIP PRACK method statistics.
Method Publish	show sipd publish	Displays the SIP PUBLISH method statistics.
Method Refer	show sipd refer	Displays the SIP REFER method statistics.
Method Register	show sipd register	Displays the SIP REGISTER method statistics.
Method Subscribe	show sipd SUBSCRIBE	Displays the SIP SUBSCRIBE method statistics.
Method Update	show sipd update	Displays the SIP UPDATE method statistics.
Monitoring	show monitoring info	Displays info of cached objects
Monitoring Reset	show monitoring reset	Resets selected cache table entry
Monitoring Stats	show monitoring stats	Displays ring buffer information
Monitoring Xml	show monitoring xml	Display all in XML format
Policy Server Connections	show policy-server connections	Display all TCP/SCTP connections
Policy Server Standby	show policy-server standby	Standby external policy server
Recording Redundancy	show rec redundancy	Displays SIPREC Redundancy Statistics
Registration by Realm	show registration sipd by realm	Displays calls that registered through a specified ingress realm for which you want to view cache information.
Registration SIP	show registration SIP	Displays SIP registrations.
Registration Statistics	show registration statistics	Displays a table of counters showing the total and periodic number of registrations by protocol.
Server Trans	show sipd server	Displays statistics for SIP server events when the SBC acts as a SIP server in the B2BUA role.

Widget	Show Command	Description
Sessions	show sessions	Displays the session capacity for license and session use.
SIP Acls	show sipd acls	SIP Access Control statisticsclient trans
SIP All	show sipd all	Display all SIP Statistics
SIP Ccp	show sipd ccp	Display Cluster Control Stats
SIP Directors	show sipd directors	Session Director Statistics
SIP ECB Registration Cache	show registration sipd by-realm <realm id>	Displays user and contact information.
SIP Ep Threadinfo Table	show sipd ep-threadinfo-table	Display SIP Endpoint Thread Info Table
SIP Ep Threadinfo Table Count	show sipd ep-threadinfo-table count	Number of entries in Endpoint Thread Info map
SIP Errors	show sipd errors	Displays statistics for SIP media event errors.
SIP Forked	show sipd forked	show sipd forked Forked Session statistics
SIP Lb Endpoints	show sipd lb-endpoints	Display LB Endpoint Information
SIP Other	show sipd other	Other SIP method statistics
SIP Policy	show sipd policy	SIP Policy/Routing statistics
SIP Pooled Transcoding	show sipd pooled-transcoding	Pooled Transcoding Statistics
SIP Rate	show sipd rate	Display SIP processing level in terms of messages
SIP Rate Agent	show sipd rate agent	Sip Sesion Agent Statistics
SIP Rate Interface	show sipd rate interface	Sip Interface Statistics
SIP Rbt Trfo	show sipd rbt-trfo	Display RBT TrFO stats
SIP Realms All	show sipd realms	Displays realm statistics related to SIP processing.
SIP Realms Individual	show sipd realms <realm name>	Displays realm statistics related to SIP processing for the specified realm.
SIP Redundancy	show sipd redundancy	Displays information about SIP redundancy
SIP Registration Cache Table		SIP Registration Cache Table
SIP Routers	show sipd routers	Session Router Statistics
SIP Sa Nsep Burst	show sipd sa-nsep-burst	Session Agent NSEP Burst Statistics
SIP Sessions	show sipd sessions	Displays the number of sessions and dialogs in various states for the Period and Lifetime spans.
SIP Sessions All	show sipd sessions all	Displays all SIP sessions currently on the system.
SIP siprec	show sipd siprec	SIPREC MESSAGE OR ERRORS
SIP Siprec Ack	show sipd siprec ack	SIPREC ACK statistics
SIP Siprec Bye	show sipd siprec bye	SIPREC BYE statistics
SIP Siprec Cancel	show sipd siprec cancel	SIPREC CANCEL statistics
SIP Siprec Errors	show sipd siprec errors	SIPREC error statistics
SIP Siprec Invite	show sipd siprec invite	SIPREC INVITE statistics
SIP Siprec Options	show sipd siprec options	SIPREC OPTIONS statistics

Widget	Show Command	Description
SIP Siprec Other	show sipd siprec other	SIPREC other method statistics
SIP Sms Context	show sipd sms-context	Display sipd sms-context
SIP Srg	show sipd srg	Session Recording Group Status
SIP Srs	show sipd srs	Session Recording Server Status
SIP Srvcc	show sipd srvcc	SRVCC handover Statistics
SIP Status	show sipd status	Displays information about SIP transactions.
SIP TCP	show sipd tcp	sipd TCP socket statistics
SIP TCP Connections	show sipd tcp connections	Display sipd TCP connections
SIP Thread Affinity	show sipd thread-affinity	Display SIP Thread Affinity stats
SIP Thread Redirect Table	show sipd thread-redirect-table	Display SIP CallID-ThreadIndex Redirect Table
SIP Transcode	show sipd transcode	Transcode Codecs
SIP Tunnels		

Security

Widget	Show Command	Description
DOS Reset	show dos threshold reset	Reset DOS threshold counters
DOS Threshold	show dos threshold counters	Displays DOS threshold counters
SA Information	show sa	Displays security-associations information
SA Information Stats	show sa stats	Displays statistics summary of all Security Associations
SA Information Stats Manual	show sa stats manual	Displays statistics for Manual Security Associations

System

Widget	Show Command	Description
Accounting	show accounting	Displays a summary of statistics for configured external accounting servers.
ACL	show acl all	Displays cumulative and per-interface statistics on ACL traffic and drops, displaying Recent, Total, and PerMax counts. The display also separates traffic from trusted from untrusted sites.
ACL Denied	show acl denied	Displays denied entries
ACL Info	show acl info	Displays acl statistics
ACL Reset	show acl reset	Reset the summary counts of all host acl entries
ACL Summary	show acl summary	Displays a summary of all host acl entries
ACL Trusted	show acl trusted	Displays trusted entries
ACL Untrusted	show acl untrusted	Displays untrusted entries
Alarms	show alarms	Displays existing alarms and allows you to clear them.

Widget	Show Command	Description
ARP Info	show arp info	Displays the current Internet-to-Ethernet address mappings in the ARP table.
ARP Statistics	show arp statistics	Displays ARP statistics.
ARP Summary	show arp	Displays the current Internet-to-Ethernet address mappings in the ARP table.
Authentication RADIUS	show radius all	Displays the status of established RADIUS accounting connections.
Authentication TACACS	show tacacs stats	Displays statistics related to communications between the Communications Broker and configured TACACS servers .
BFD Stats Errors	show bfd-stats errors	BFD global errors
BFD Stats Summary	show bfd-stats	BFD status
Buffers Histogram	show buffers histogram	Buffer pool histogram of buffer sizes
Buffers Statistics	show buffers	Displays buffer pool statistics
Buffers Usage Functions	show buffers usage-threads	Buffer pool usage statistics per function
Buffers Usage Threads	show buffers usage-functions	Buffer pool usage statistics per process
Clock	show clock	Displays the current date and time.
Clock UTC	show clock utc	Displays the current date and time in Coordinated Universal Time (UTC).
Communications Monitor Errors	show comm-monitor errors	Displays Communications Monitor aggregate error statistics information.
Communications Monitor Internal	show comm-monitor internal	Displays Communications Monitor aggregate internal statistics information.
Communications Monitor Stats	show comm-monitor stats ACL Denied	Displays statistics related to connections between the Communications Broker Communications Monitor probe and any configured Communications Monitor servers.
Configuration Inventory	show configuration inventory	Displays the editing and running configuration inventory of all configured elements.
Configuration Version	show version	Displays the configuration version number table.
Current Disk Usage Pie Graph	No show command	Displays the current disk usage in a pie graph.
Current Disk Usage Table	No show command	Displays the current disk usage in a table.
Current Memory Usage Pie Graph	No show command	Displays the current percentage of free and allocated memory in a pie graph.

Widget	Show Command	Description
Current Memory Usage Table	No show command	Displays the current percentage of free and allocated memory in a table.
Directory	show directory	Displays files in a directory
Directory All	show directory	Displays contents of all top-level directories
Editing Configuration	show configuration	Displays the current editing configuration.
Editing Configuration Short	show configuration short	Displays only the parameters that you modified in the editing configuration.
Entitlements	show entitlements	Currently provisioned entitlements
Features	show features	Displays the features that are currently enabled, based on added licenses.
Highest Task CPU Usage Line Graph	No show command	Displays a line graph with 5-10 tasks with the highest CPU usage in percent, during a specific period of time.
Highest Task CPU Usage Table	No show command	Displays a table with 5-10 tasks with the highest CPU usage in percent, during a specific period of time.
Historical Memory Usage Line Graph	No show command	Displays a line graph of the kilobytes of free and allocated memory over a period of time.
Historical Memory Usage Table	No show command	Displays a table of the kilobytes of free and allocated memory over a period of time.
Interface Mapping	show interface mapping	Displays the configured physical interfaces with their MAC addresses and label.
Interfaces	show interfaces	Displays all of the information concerning the rear interfaces.
Interfaces Brief	show interfaces brief	Displays key running statistics about the rear interfaces in one graphic.
Interfaces Ethernet	show interfaces ethernet	Displays network interfaces ethernet
Interfaces Mapping	show interfaces mapping	Ethernet Interfaces Names and MAC Physical ports mapping
IP Connections	show ip connections	Displays all TCP and UDP connections.
IP SCTP	show ip sctp	SCTP statistics
IP Statistics	show ip statistics	IP statistics
IP Summary	show ip	Displays IP statistics.
IP TCP	show ip tcp	Displays all TCP statistics.
IP UDP	show ip udp	Displays all UDP statistics.
Licenses	license	Displays existing licenses and allows you to add or delete them.

Widget	Show Command	Description
Logfiles	show logfile	Display a complete list of all log files
Memory Application	show memory application	Application memory usage statistics
Memory L2	show memory l2	Layer 2 cache status
Memory L3	show memory l3	Layer 3 cache status
Memory Sobjects	show memory sobjects	Displays all sobjects cached in the system
Memory Sobjects Compare	show memory sobjects compare-baseline	Compare the current sobject counts versus baseline
Memory Sobjects Set	show memory sobjects set-baseline	Set a baseline for comparison
Memory Sobjects SkIP Zero	show memory sobjects skip-zero	Display only non-zero entries
Memory Summary	show memory	Displays statistic related to the memory.
Memory Usage	show memory usage	Memory usage statistics
Neighbor Table	show neighbor table	Displays the IPv6 neighbor table and validates that an entry for the link local address exists and that the gateway uses that MAC address.
Net Management Control	show net-management-control	Network Management Controls Statistics
NTP Server	show ntp server	Displays information about the quality of the time used for offset and the delay measurement maximum error bounds.
NTP Status	show ntp status	Displays information about configuration status, NTP daemon synchronization, NTP synchronization in process, and if NTP is not responding.
Packet Trace	show packet-trace	Displays the current packet trace addresses
Platform	show platform	Display platform summary information
Platform All	show platform all	Displays full platform information.
Platform Components	show platform components	Display list of component packages
Platform Cpu	show platform cpu	Display summary CPU information
Platform CPU load	show platform cpu-load	Displays current CPU load.
Platform Current Draw	show platform current-draw	Display power supply current draw
Platform Errors	show platform errors	Displays service pipe write errors.
Platform Health Check	show platform health-check	Display thread health information
Platform Heap Statistics	show platform heap-statistics	Display current heap statistics
Platform Hotswap	show platform hotswap	Hotswap status
Platform Kernel Drivers	show platform kernel-drivers	Display included kernel drivers
Platform Limits	show platform limits	Displays platform related limits.
Platform Memory	show platform memory	Display current memory usage

Widget	Show Command	Description
Platform Paths	show platform paths	Display filesystem paths
Platform Pci	show platform pci	Display PCI information
Power	show power	Current state of each power supply
Power Supply Revision	show power-supply-rev	Power supply revisions
Privilege	show privilege	Display current privilege level
Processes	show processes	Displays statistics for all active processes.
Processes Algd	show processes algd	Display algd process statistics
Processes All	show processes all	Display statistics for all processes
Processes Atcpd	show processes atcpd	Display atcpd process statistics
Processes Authd	show processes authd	Display authd process statistics
Processes Berpd	show processes berpd	Display berpd process statistics
Processes Brokerd	show processes brokerd	Display brokerd process statistics
Processes Ccd	show processes ccd	Display ccd process statistics
Processes Collect	show processes collect	Display Collector process statistics
Processes CommMonitord	show processes commMonitord	Display commMonitord process statistics
Processes Cpu	show processes cpu	Display CPU Usage
Processes Current	show processes current	Display current process statistics
Processes Ebmd	show processes ebmd	Display embd process statistics
Processes Hashtables	show processes hashtables	Display hashmap tables information for all processes
Processes Ipt	show processes ipt	Display ipt process statistics
Processes Lbp	show processes lbp	Display lbp process statistics
Processes Lemd	show processes lemd	Display lemd process statistics
Processes Lemd	show processes lemd	Display lemd process statistics
Processes Lid	show processes lid	Display lid process statistics
Processes Locks	show processes locks	Display lock wait information for all processes
Processes Mbcd	show processes mbcd	Display mbcd process statistics
Processes Memory	show processes memory	Display process memory statistics
Processes Overload	show processes overload	Display process overload statistics
Processes Radd	show processes radd	Display radd process statistics
Processes Secured	show processes secured	Display secured process statistics
Processes Sipd	show processes sipd	Display sipd process statistic
Processes Snmpd	show processes snmpd	Display snmpd process statistics
Processes Spal	show processes spal	Display spal process statistics
Processes Sysmand	show processes sysmand	Display sysmand process statistics
Processes Threads	show processes threads	Display system thread
Processes Top	show processes top	Display CPU top
Processes Top Threads	show processes top-thread	Display CPU top-threads
Processes Total	show processes total	Display total process statistics

Widget	Show Command	Description
PROM Info	show prom-info all	Displays all available PROM information
Queues	show queues	Display command queue status
Queues Atcpd	show queues atcpd	Displays thread level statistics of ATCPD
Queues Ccd	show queues ccd	Displays thread level statistics of CCD
Queues Dns	show queues dns	Displays thread level statistics of DNS
Queues Lbp	show queues lbp	Displays thread level statistics of LBP
Queues Ldap	show queues ldap	Displays thread level statistics of LDAP
Queues Lid	show queues lid	Displays thread level statistics of LID
Queues Mbcd	show queues mbcd	Displays thread level statistics of MBCD
Queues Radd	show queues radd	Displays thread level statistics of RADD
Queues Sipd	show queues sipd	Displays thread level statistics of SIPD
Queues Sipd Commands	show queues sipd commands	Displays thread command queue statistics of SIPD
Queues Spal	show queues spal	Displays thread level statistics of SPAL
RADIUS Accounting	show radius accounting	Display accounting statistics
RADIUS Accounting All	show radius accounting all	Display all Accounting Servers
RADIUS All	show radius all	Displays RADIUS accounting and authentication statistics
RADIUS Authentication	show radius authentication	Display authentication statistics
RADIUS Cdr	show radius cdr	Display cdr statistics
RAMDRV	show ramdrv	Displays disk space usage on /opt
Realm Specifics	realm-specifics	Display all realm-specific configuration based on the specified realm-id
Realm Summary	show realm	Displays the realm summary statistics
Recording Redundancy	show rec redundancy	Displays SIPREC Redundancy Statistics
Redundancy Collect	show redundancy collect	Collect Redundancy Statistics
Redundancy Collect Journals		
Redundancy Collect Journals Perf	show redundancy collect journals perf	Journal Performance
Redundancy Collect Journals Size	show redundancy collect journals size	Journal size
Redundancy Config	show redundancy config	Configuration Redundancy Statistics
Redundancy Config Journals	show redundancy config journals	Configuration Redundancy Journal Statistics

Widget	Show Command	Description
Redundancy Config Journals Perf	show redundancy config journals perf	Journal Performance
Redundancy Config Journals Size	show redundancy config journals size	Journal size
Redundancy Manuald	show redundancy manuald	Manuald Redundancy Statistics
Redundancy Manuald Journals	show redundancy manuald journals	Manuald Redundancy Journal Statistics
Redundancy Manuald Journals Perf	show redundancy manuald journals perf	Journal Performance
Redundancy Manuald Journals Size	show redundancy manuald journals size	Journal size
Redundancy Radius Cdr	show redundancy radius-cdr	Radius CDR Redundancy Statistics
Redundancy Radius Cdr Journals	show redundancy radius-cdr journals	Radius Redundancy Journal Statistics
Redundancy Radius Cdr Journals Perf	show redundancy radius-cdr journals perf	Journal Performance
Redundancy Radius Cdr Journals Size	show redundancy radius-cdr journals size	Journal size
Redundancy Sipd	show redundancy sipd	SIP Redundancy Statistics
Redundancy Sipd Actions	show redundancy sipd actions	SIP Redundancy Actions Statistics
Redundancy Sipd Journals	show redundancy sipd journals	SIP Redundancy Journal Statistics
Redundancy Sipd Journals Perf	show redundancy sipd journals perf	Journal Performance
Redundancy Sipd Journals Size	show redundancy sipd journals size	Journal size
Redundancy Sipd Objects	show redundancy sipd objects	SIP Redundancy Object Statistics
Routes	show routes	Displays the current system routing table.
Route Stats	show route-stats	Display routing statistics
Running Configuration	show running-config	Displays the current running configuration.
Running Configuration Short	show running-config short	Displays only the parameters that you modified in the running configuration.
Snmp	show snmp	Summary of snmp engine information
Snmp Address	show snmp address	Displays snmp address info and statistics
Snmp All	show snmp all	Displays all snmp entry info and brief statistics
SNMP Community Table	show snmp-community-table	Displays all information for configured SNMP communities including requests and responses for each community.
Snmp Group	show snmp group	Displays snmp group info and statistics
Snmp Group Entry	show snmp-group-entry	Display snmp-group-entry
Snmp Info	show snmp-info	Summary of snmp engine information

Widget	Show Command	Description
Snmp Info Address	show snmp-info address	Displays snmp address info and statistics
Snmp Info All	show snmp-info all	Displays all snmp entry info and brief statistics
Snmp Info Community	show snmp-info community	Displays snmp community table
Snmp Info Group	show snmp-info group	Displays snmp group info and statistics
Snmp Info SNMP Address	show snmp-info snmp-address	Displays snmp address info and statistics
Snmp Info SNMP Group	show snmp-info snmp-group	Displays snmp group info and statistics
Snmp Info SNMP User	show snmp-info snmp-user	Displays snmp user info and statistics
Snmp Info SNMP View	show snmp-info snmp-view	Displays snmp view info and statistics
Snmp Info Stats	show snmp-info stats	Displays all snmp statistics
Snmp Info Trap Receiver	show snmp-info trap-receiver	Displays snmp trap receivers
Snmp Info User	show snmp-info user	Displays snmp user info and statistics
Snmp Info View	show snmp-info view	Displays snmp view info and statistics
Snmp SNMP Address	show snmp snmp-address	Displays snmp address info and statistics
Snmp SNMP Group	show snmp snmp-group	Displays snmp group info and statistics
Snmp SNMP User	show snmp snmp-user	Displays snmp user info and statistics
Snmp SNMP View	show snmp snmp-view	Displays snmp view info and statistics
Snmp Stats	show snmp stats	Displays all snmp statistics
Snmp Trap Receiver	show snmp trap-receiver	Displays snmp trap receivers
Snmp User	show snmp user	Displays snmp user info and statistics
Snmp User Entry	show snmp-user-entry	Display snmp-user-entry
Snmp View	show snmp view	Displays snmp view info and statistics
Snmp View Entry	show snmp-view-entry	Display snmp-view-entry
Space	show space	Check the remaining space on the device specified
Space Boot	show space boot	Check the remaining space on / boot
Space Code	show space code	Check the remaining space on / code
Space Crash	show space crash	Check the remaining space on / crash
Space Data Disk	show space data-disk	Check the remaining space on the hard drive user data partitions
Space Hard Disk	show space hard-disk	Check the remaining space on all of the hard drive partitions
Space Opt	show space opt	Check the remaining space on /opt

Widget	Show Command	Description
Space System Disk	show space system-disk	Check the remaining space on the hard drive system partitions
SPL Appstats	show spl appstats	Display SPL statistics for all applications
SPL Memory	show spl memory	Displays SPL memory for each task SPL engine.
SPL Options	show spl-options	Displays information on all SPL options.
SPL Statistics	show spl statistics	Displays statistics for all tasks.
SPL Version	show spl	Displays the version of the SPL engine.
System Health	show health	Displays the system health table for HA pairs.
Time Zone	show timezone	Displays the time zone.
Trap Receiver	show trap-receiver	Displays trap receiver information for each configured SNMP community.
Uptime	show uptime	Displays information about the length of time the system has been running in days, hours, and seconds. Also displays the current date and time information.
User Management	show users	Displays a table that lists all users currently logged on to the system.
Version Boot	show version boot	Displays the boot version.
Version CPU	show version cpu	Displays the CPU version.
Version Hardware	show version hardware	Displays the hardware version.
Version Image	show version image	Displays the image version.
Version Summary	show version	Displays the Operating System version information, including the OS version number, the manufacturing date of the current version, and other details.
Virtual Interfaces	show virtual interfaces	Displays the virtual interfaces for signaling services.
Wancom	show Wancom	Displays negotiated duplex mode and speed for all system control interfaces.

Command Line Interface Widgets

Like many devices, the Communications Broker includes an underlying management interface called the Command Line Interface (CLI).

Support technicians use the CLI to display detailed information about the system in text format. Oracle makes this information available from the graphical user interface (GUI) with CLI Widgets, available from the Communications Broker Widget page. The CLI Widgets can provide useful troubleshooting information, as well as insight into system operations.

To reach the CLI portal, go to the **Widgets** tab, **Command**, **Command Line Portal**. When you click the CLI portal link, the Communications Broker displays the **Settings** page.

CLI portal settings include:

Command	A drop-down list where you choose the CLI command.
Parameter	A text box where you enter additional parameter text to refine the command output with a command argument.
Auto-refresh Interval	A drop-down list where you specify how often the system refreshes the widget's data. Default: Never. Valid values: Never | 30 |40 | 50 | 60.

The portal includes an **OK** button and a **Cancel** button. When you click **OK**, the system displays the output of the command.

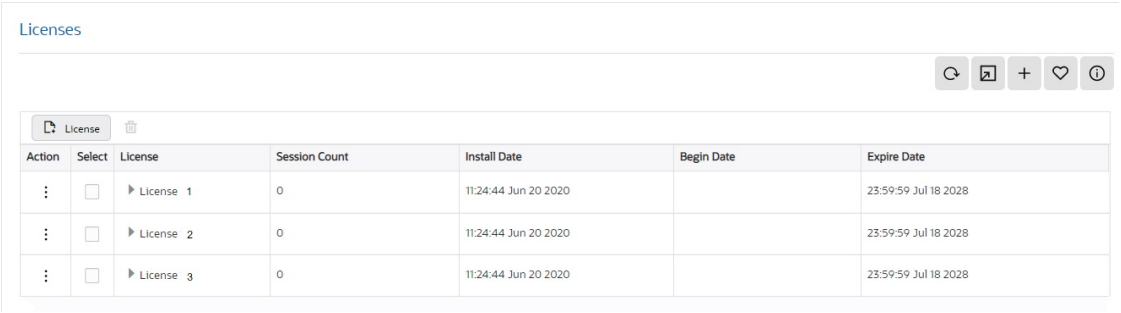
The system produces two types of CLI widgets, depending on the command invoked:

- Text Display—The system displays the output of the command in an all-text format.
- Tabular Display—The system displays the output of these commands in a table.

Licenses Widget

The Licenses widget on the Web GUI provides a workspace where you can view, add, and delete Oracle Enterprise Communications Broker (Communications Broker) licenses.

On the Widgets tab, click System and then click Licenses. The following screen capture shows and example of the Licenses page, which displays the license name, the entitled number of sessions, the installation date, the begin date, and the expire date.

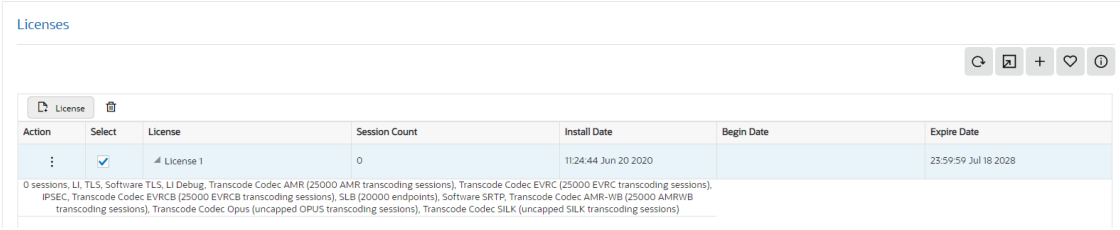


The License widget includes the same toolbar in the top, right-hand corner that Widgets on the Dashboard display. Note that the License widget does not include the Settings icon and the Auto-refresh function because these operations do not apply to licenses.

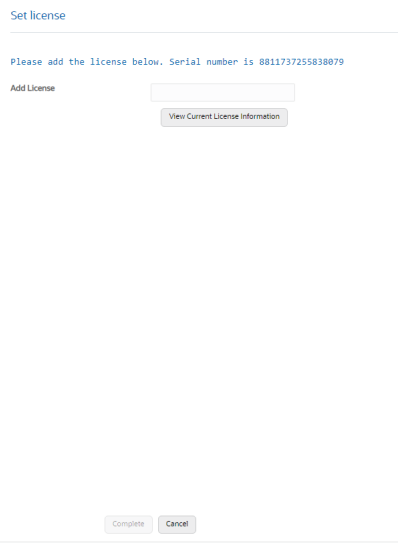
The Licenses page displays a list of your Communications Broker licenses with the following information.

License	The name of the license.
Session Count	The number of session entitlements for the license.
Install Date	The date when the license is added to the system.
Begin Date	The date when the license begins service.
Expire Date	The date when the license ends service.

If you want to see the details of a particular license, click the twister by the license name to expand the view to show all of the details, which display in text below the selected license. The text states the number of allowed sessions and lists all of the features included in the license. The following screen capture shows an example of license details.



When you click **License**, the system displays the License dialog, where you enter the license serial number.



When you select a license from the Licenses list and click the delete icon, the system displays the delete Confirmation dialog.

The Set License function, located under System Operations on the System tab, links to the License widget so you can view your licenses from the Set License dialog. After launching Set License, use the "View current license information" button in the Set License dialog to see a view-only list of your Communications Broker licenses.

The only operations allowed in view mode are Refresh and Download.

Display and Clear Alarms

The Oracle Enterprise Communications Broker provides a Widget that allows you to see all current alarms that the system triggered. You can also see current alarms by clicking the Bell icon the top information bar.

1. Access the **Widgets** Tab. You can also click the **Bell** icon to access the Alarms.
2. Click **System**, **Alarms**.
3. (Optional)—To clear alarms, click **Clear All**. If you have accessed the Alarms using the Bell icon, select the Alarms in the pop-up dialog box.

Display Users

The Oracle Enterprise Communications Broker provides a widget that allows you to see a list of other users currently logged on to the system.

1. Access the Widgets Tab.
2. Click **System**, **User Management**.

The Oracle Enterprise Communications Broker displays the User Management Page.

3. (Optional)—Right-click a user, and click **Disconnect**.

5

Agent Configuration

Agent configurations specify the hops by which the Oracle Enterprise Communications Broker (Communications Broker) defines routes. Network architects select agents to identify and delineate between key logical and/or physical locations in the overall network topology. Mapping out agent topology can also identify logical or physical locations that would benefit from an agent. Agents are usually physical devices, such as proxies, SBCs, PBXs and gateways. But agents can also be logical entities, including domain names.

You can configure agents with:

- Location definition
- Context
- Translation mode
- Inbound and outbound manipulation rules
- Traffic constraints

The last two bullets are particularly impactful when the agent is the first hop in a route, but are applicable to agents multiple hops from the Communications Broker.

Alphanumeric User Database and Call Routing Entries

The Oracle Enterprise Communications Broker (Communications Broker) allows you to enter alphanumeric entries in the User database and in the routing table because it can modify the request URI to use either alphanumeric or numeric entries, depending on the setting you choose for **egress-URI-mode**.

Egress-URI-mode

When the Communications Broker locates the home agent to which the call is routed, it uses one of the following configurable values from the **egress-URI-mode** parameter in the **Agent** configuration to determine whether or not to convert the request URI:

- No-conversion (Default)—The Communications Broker adds only the IP address of the Home Agent in the outbound call.
- Convert-to-aor— When the incoming URI is a number, the Communications Broker replaces it with the configured address of record in the outbound call.
- Convert-to-number—When the incoming URI is an Address of Record, the Communications Broker replaces it with the configured number in the outbound call.

Example

Assume that the User database includes "aor1@oracle.com" for the AOR, "123@oracle.com" for the number, and the destination agent is named Agent123.

When a call comes in, the Communications Broker checks the value in the **egress-URI-mode** parameter for the destination agent Agent123 and does the following:

- If the request URI in the incoming call is "aor1@oracle.com", and the egress-URI-mode for Agent123 is set to "convert-to-number", the Communications Broker modifies the request URI to 123@oracle.com.
- If the request URI in the incoming call is "123@oracle.com", and the egress-URI-mode for Agent123 is set to "convert-to-aor", the Communications Broker modifies the request URI to "aor1@oracle.com."
- If you set the egress-URI-mode for Agent123 to "no-conversion", the user part of the request URI remains unchanged. The Communications Broker changes only the host part of the request URI to use the IP address of Agent123.

Alphanumeric User Database Entries

In the User Entries configuration, both the **Address of Record** parameter and the **Number or Pattern** parameter can accept alphanumeric entries. Note that you can leave either the **Address of Record** or the **Number or Pattern** parameter empty, but not both at the same time. The user database supports the following combinations of entries:

Address of Record	Number or Pattern	Home Agent	Dialing Context
username1@company.com	1234567890@company.com	Agent1	region.country
username2@company.com	1234567890	Agent2	region.country
username3@company.com		Agent3	region.country
username1@1.1.1.1	1234567890	Agent3	
	11234567890	Agent4	
	12324567890	Agent4	
	12345678[01-04]	Agent5	
	xxxxxxxxx	Agent5	

The user database does not support wild card entries in the **Address of Record** parameter, but the **Number or Pattern** parameter supports the following wild card patterns:

- 12345678xx@company.com
- 12345678xx@country1.company.com
- 12345678xx@country2.com
- 123456[01-20]@country.company.com

Example of a valid configuration

The following example shows the **Address of Record** and the **Number or Pattern** entered in the User database in a valid configuration, where the entries are properly formatted.

Address of Record	Number or Pattern	Home Agent	Dialing Context
username@company.com	1234567890@company.com	Agent1	region.country

With "convert-to-number" selected, the Communications Broker modifies the request URI to use the number 1234567890@company.com. With "convert-to-aor" selected, the Communications Broker modifies the request URI to use the **Address of Record** username@company.com.

Example of an invalid configuration

The following example shows an invalid user record which includes two different address of record entries for one user (rows 1 and 2), and duplicate address of records for two different numbers (rows 3 and 4).

Address of Record	Number or Pattern	Home Agent	Dialing Context
username@company.com	1234567890@company.com	Agent1	region.country
user@1.1.1.1	1234567890@company.com	Agent1	region.country
auser@company.com	5551234567@company.com	Agent1	region.country
auser@company.com	0005551234@company.com	Agent1	region.country

Routing Table Entries

The routing table supports the following syntax for alphanumeric entries in the **Calling Number** and **Called Number** parameters:

- 1234567890
- 123[000-999]
- 123[x]xxx
- 1234567890@company.com
- user_name@company.com
- user.name@1.1.1.1

Dial Plan Lookup

When the Communications Broker attempts a dial plan lookup, it first confirms that the user part of the request-uri is a valid phone number. For a valid number, the Communications Broker performs the dial plan lookup. When user part of request-uri is an alphanumeric string, the Communications Broker skips the dial plan lookup.

ENUM

When the Communications Broker uses a configured route to an ENUM server, it does not convert the request-uri as specified in the destination agent configuration. If the destination agent (returned by the ENUM server) is not reachable, and Communications Broker chooses a backup route from the user database, the Communications Broker converts the request-uri of the outgoing INVITE based on the **egress-uri-mode** parameter set for the destination agent in the user database.

LDAP

When a user database entry and an LDAP entry both exist, the Communications Broker gives priority to the entry in user database. Conversion of the request-uri is performed based on the configuration of the destination agent. If a user database entry does not exist, LDAP takes the precedence and the system does not convert the request-uri.

About Traffic Constraints

You use traffic constraint configuration to establish call admission control (CAC) thresholds and triggers that determine whether the Communications Broker can accept and process network traffic, including calls. The Communications Broker provides call CAC based on the following policies:

- Bandwidth
- Session capacity
- Session rate

Bandwidth-Based Admission Control

The Communications Broker is a policy enforcement point for bandwidth-based call admission control. Sessions are admitted or rejected based on bandwidth policies, configured on the Communications Broker for each agent.

As the Communications Broker processes call requests to and from a particular agent, the bandwidth consumed by the traffic is decremented from its bandwidth pool.

Session Capacity- and Rate-based Admission Control

A session agent defines a signaling endpoint. It is a next hop signaling entity that can be configured to apply traffic shaping attributes. You can define concurrent session capacity and rate attributes for each session agent.

You can configure a set of attributes and constraints for each session agent to support session access control. And you can set up session admission control so that the Communications Broker limits the number of concurrent inbound and outbound sessions for any known service element.

The Communications Broker denies a call request to any destination that has exceeded its configured policies for session capacity and session rate. The Communications Broker might reject the call request back to the originator. If multiple destinations are available, the Oracle Enterprise Communications Broker checks current capacity and rate for each destination and attempt to route the call only to destinations whose policy limits have not been reached.

CAC Utilization Statistics via SNMP

The Communications Broker allows you to retrieve information on current session utilization and burst rate as a percentage of their configured maximums on per session-agent basis. The Communications Broker uses the configured max-session and max-burst-rate settings in conjunction with a percentage formula to calculate this value. The system also uses a configuration setting to establish the threshold at which trap and trap clear messages are sent from the SNMP agent to the configured manager(s). You must configure traps and enable the **SNMP Monitor Traps** parameter for this functionality.

You must load the MIB version associated with this software version on all pertinent SNMP managers to query these CAC utilization (occupancy) values and interpret the traps. In addition, the user must configure the threshold at which the system generates the CAC utilization trap. Note that the corresponding clear trap uses the same threshold setting, sending the clear trap when utilization falls below 90% of the threshold.

The Communications Broker can issue a trap when either the value of max-session or CAC burst rate exceeds a configured value. The system only sends one trap when the threshold is

exceeded. When the value falls back under 90% of this threshold, the Communications Broker sends a clear trap.

You configure the value that triggers these traps as a percentage of the max-session and max-burst-rate settings configured for the applicable session agent. The system uses the same setting to specify when to send both the sessions and burst rate traps. The name of this parameter is the **cac-trap-threshold**. You must express the value as a number less than 100. There is no default setting; the system does not generate a trap if you have not configured this setting.

The apSipCACUtilAlertTrap identifies the threshold exceeded on a per-element and per-value (session count or burst rate) for each trap, including:

- apSipSaCacSessionUtilLevel
- apSipSaCacBurstRateUtilLevel

In-Service Response Codes for a Session Agent

You can configure the **ping-in-service-response-codes** parameter in the session agent configuration to refine how the Communications Broker monitors agent status. You define one or more response codes that keep a session agent in service when sent as response to the Communications Broker's ping request in this parameter. The Communications Broker takes the session agent out of service if the session agent sends a response code that is not on this list. The exception to this is a 200 OK response, which, by default, always keeps the agent in-service. This behavior applies to both single and multi-hop pings.

Some session agents ay respond to the ping with, for example, a 4xx message instead of a 200 OK even though it can service the request. To alleviate issues like this, you configure this session agent with the 4xx messages as acceptable responses using the **ping-in-service-response-codes** parameter.

Addition configuration required to support these codes include:

- Ping Method
- Ping Interval
- Ping All Addresses

When the Communications Broker receives a session agent's response to its ping request that is not a 200 OK, it checks to see if there is an in-service list of responses configured for that session agent. If the list is configured and the Communications Broker determines that there is a match, the session agent is deemed in service. If there is no match, the Communications Broker takes the session agent out of service.

Important detail on specific configured response codes includes:

- Create a list of codes by separating each entry with a comma.
- If you configure for the 3xx response code, the Communications Broker makes its in-service decision based on the redirected options response.
- You cannot configure a 6xx response as an in-service response code because it is a global error response. If you do this, the Communications Broker displays an error message.

Add a Session Agent

You can enable and configure a variety of constraints that the Communications Broker applies to regulate session activity with the session agent.

Configure the following before you configure a session agent.

- Out Translation ID
- TLS profile
- SIP header manipulation IDs
- LDAP
- One or more target groups

The following procedure lists all of the possible parameters that you can set on a session agent. The parameters that the Communications Broker allows you to set depends on your deployment configuration, the licenses that you own, whether or not you configured the prerequisite that a particular parameter requires, and whether or not you enabled a requisite function. Within those conditions, you can choose which parameters that you want to set for your deployment.

1. Access the Agent configuration object.

   Configuration tab, **Service Provisioning** section, **Agents**, **Session Agent**.

2. On the **Agents** page, click **Add**, and do the following:

Table 5-1 Fields in the Agents Page

Field	Description
Hostname	Enter the name of the host associated with the agent in host name, FQDN, or IP address format. This field is required and the name cannot include blank spaces. The value entered here must be unique to this agent because no two agents can use the same host name. • If you enter the host name as an IP address, you do not have to enter an IP address in the optional IP address parameter. • If you enter the host name in FQDN format, and you want to specify an IP address, enter it in the optional IP address parameter.
IP address	(Optional) Enter the IP address for the host name that you entered in FQDN format if you want to specify the IP address. Otherwise, you can leave this parameter blank to allow a DNS query to resolve the host name.

Table 5-1 (Cont.) Fields in the Agents Page

Field	Description
Port	Enter the number of the port associated with this agent. • 0. If you enter zero, the Communications Broker cannot initiate communication with this agent (although it will accept calls). • 1025-65535. • The default value is 5060. If the transport method value is TCP, the Communications Broker will initiate communication on the TCP port of the agent.
State	Select State to enable this agent.
RURI with Hostname	• Select to resolve all outgoing requests to the Session Agent to the Session Agent name in the RURI. • Deselect to resolve all outgoing requests to the Session Agent to the Session Agent IP address in the RURI. Default: deselected.
Transport method	Select the transport protocol for communicating with this agent. Valid values: • UDP - Default • UDP+TCP • Dynamic TCP • Static TCP • Dynamic TLS • Static TLS
TLS profile	Select a TLS profile from the drop-down list.
Realm ID	Enter the name of the realm in which this agent resides.
Description	Enter descriptive text to identify this agent.
Source context	Select the dialing context the Communications Broker uses when receiving a call from this agent.
Egress URI mode	Select the method you want for sending the outbound call from this agent. Default: no-conversion. Valid values: • no-conversion—The Communications Broker adds only the IP address of the Home Agent in the outbound call. • convert-to-aor—When the incoming URI is a number, the Communications Broker replaces it with the configured address of record in the outbound call. • convert-to-number—When the incoming URI is an Address of Record, the Communications Broker replaces it with the configured number in the outbound call.

Table 5-1 (Cont.) Fields in the Agents Page

Field	Description
Egress number translation mode	Select a number translation code from the drop-down list. Default: E164. Valid values: • E164—Example: +14445551234 • E164 no plus—Example: 14445551234 • no country code—Example: 4445551234 • pattern only—Use only the portion matching the dial pattern. Example: 1234 • n digit dialing—Use the specified number of digits. Example: 5551234
Number of digits for n digit dialog	Specify the number of digits to use for translation mode (n-digit dialing). Default: 4. Range: 0-25.
Prepend prefix on egress	Specify the characters to prepend to the outbound number after translation. Valid values: Telephony characters 0-9, start, hash sign, and A-D. Maximum: 25 characters.
Outbound translate from number	Select to apply outbound translation to the FROM number in addition to the request-URI. Default: Disabled.
Stop recurse	Enter one or more response codes that you want to cause this session agent to stop route recursion. Default: 401,407. Valid range: 300-599. You can enter individual response codes separated by a comma, such as 301,305 or a range such as 300-380.
Constraints	Check to enable the use of constraints on this agent.
Max Sessions	Set the maximum number of sessions (inbound and outbound) allowed by the session agent. The default value is zero (**0**). The valid range is: • Minimum—0 • Maximum—4294967295
Max Inbound Sessions	Enter the maximum number of inbound sessions allowed from this session agent. The default value is zero (**0**). The valid range is: • Minimum—0 • Maximum—999999999
Max Outbound Sessions	Enter the maximum number of concurrent outbound sessions (outbound from the Communications Broker) that are allowed from this session agent. The default value is zero (**0**). The valid range is: • Minimum—0 • Maximum—4294967295 **Note:** The number you enter here cannot be larger than the number you entered for max-sessions.

Table 5-1 (Cont.) Fields in the Agents Page

Field	Description
Max Burst Rate	Enter a number to set how many SIP session invitations or H.323 SETUPs this session agent can send or receive (per second) within the configured burst rate window value. The default value is zero (**0**). The valid range is: • Minimum—0 • Maximum—4294967295 For the sustained rate, the Communications Broker maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes. For example, if you enter a value of 50 here and a value of 60 (seconds) for the burst rate window constraint, no more than 300 session invitations can arrive at or leave from the session agent in that 60 second time frame (window). Within that 60-second window, any sessions over the limit of 300 are rejected.
Max Inbound Burst Rate	Enter the maximum burst rate (number of session invitations per second) for inbound sessions from this session agent. The default value is zero (**0**). The valid range is: • Minimum—0 • Maximum—999999999
Max Outbound Burst Rate	Enter the maximum burst rate (number of session invitations per second) for outbound sessions to this session agent. The default value is zero (**0**). The valid range is: • Minimum—0 • Maximum—999999999
Max Sustain Rate	Enter a number to set the maximum rate of session invitations (per second) this session agent can send or receive within the current window. The default value is zero (**0**). The valid range is: • Minimum—0 • Maximum—4294967295 The number you enter here must be larger than the number you enter for **max-burst-rate**. For the sustained rate, the Communications Broker maintains a current and previous window size. The period of time over which the rate is calculated is always between one and two window sizes. For example, if you enter a value of 50 here and a value of 36 (seconds) for the sustain rate window constraint, no more than 1800 session invitations can arrive at or leave from the session agent in any given 36 second time frame (window). Within that 36 second window, sessions over the 1800 limit are rejected.

Table 5-1 (Cont.) Fields in the Agents Page

Field	Description
Max Inbound Sustain Rate	Enter the maximum sustain rate (of session invitations allowed within the current window) of inbound sessions from this session agent. This value should be larger than the **max-inbound-burst-rate** value. The default value is zero (**0**). The valid range is: • Minimum—0 • Maximum—999999999
Max Outbound Sustain Rate	Enter the maximum sustain rate (of session invitations allowed within the current window) of outbound sessions to this session agent. This value should be larger than the **max-outbound-burst-rate** value. The default value is zero (**0**). The valid range is: • Minimum—0 • Maximum—999999999
Cac Trap Threshold	The CAC (session or burst-rate) utilization threshold expressed as a percentage that when exceeded generates a trap. • Allowed values: minimum 0, maximum: 99. • Default value: 0
Session Max Life Limit	Enter the maximum interval in seconds before the SBC must terminate long duration calls. The value supersedes the value of session-max-life-limit in the sip-interface and sip-config configuration elements and is itself superseded by the value of session-max-life-limit in the session-agent configuration element. The default value is 0 (off/ignored). The valid range is: • Minimum—0 • Maximum—999999999
Time To Resume	Time in seconds after which the SIP proxy resumes sending session invitations to this session agent. This value only takes effect when the SIP proxy stops sending invitations because a constraint is exceeded. The valid range is: • Minimum—0 • Maximum—4294967295
In Service Period	Amount of time in seconds the session agent must be operational (once communication is re-established) before the session agent is declared as being in-service (ready to accept session invitations). This value gives the session agent adequate time to initialize. The valid range is: • Minimum—0 • Maximum—4294967295
Burst Rate Window	Enter a number to set the burst window period (in seconds) that is used to measure the burst rate. The term window refers to the period of time over which the burst rate is computed. The default value is zero (**0**). The valid range is: • Minimum—0 • Maximum—4294967295

Table 5-1 (Cont.) Fields in the Agents Page

Field	Description
Sustain Rate Window	Enter a number to set the sustained window period (in seconds) that is used to measure the sustained rate. The default value is zero (**0**), which disables the functionality. The valid range is: • Minimum—0 • Maximum—4294967295 The value you set here must be higher than or equal to the value you set for the burst rate window. **Note:** If you are going to use this parameter, you must set it to a minimum value of 10.
Ping Method	SIP only. Indicate the SIP message/method to use to ping a session agent. The ping confirms whether the session agent is in service. If this field is left empty, no session agent will be pinged. Setting this field value to the OPTIONS method might produce a lengthy response from certain session agents and could potentially cause performance degradation on your system.
Ping Interval	(SIP only) Set the time, in seconds, for how often to ping the session agent. Range: 0-4294967295. Default: 0.
Ping All Addresses	Enable to allow DNS load balancing.
Ping in Service Response Codes	Define the response codes that keep a session agent in service when they appear in its response to the ping request. Furthermore, the Communications Broker takes the session agent out of service if it receives a response code that does not appear on this list The exception to this is a 200 OK response. Separate multiple entries with a comma. The Communications Broker displays a configuration validation error if your include a 6xx response as an in-service response code. These global error responses do not apply. **Note:** For configured 3xx response codes, the Communications Broker makes the in-service decision based on the redirected options response.

Table 5-1 (Cont.) Fields in the Agents Page

Field	Description
Load Balance DNS Query	Specifies the type of load balancing the system uses to perform A record resolution. The valid entries include: • Empty—(Default) • Hunt—Use the hunt method to resolve A queries from this agent • Round-Robin—Use the round robin method to resolve A queries from this agent
SPL options	You can add features or parameters here. For example: • feature=<value feature> You can include the original address in the SIP message from the Communications Broker to the proxy in the Via header parameter by entering the following option: via-origin=<parameter-name> The original parameter is included in the Via of the requests sent to the session agent. The via origin feature can take a value that is the parameter name to include in the Via. If the value is not specified for via origin, the parameter name is origin. **Note:** If the feature value is a comma separated list, enclose it in quotation marks.
Apply outbound manipulation on	Specify when to apply outbound header manipulation. Default: next-hop-only. Valid values: last-hop-only, next-and-last-hop.
Ping Response	Enable to allow the Communications Broker to respond to OPTIONS pings.
In manipulation ID	Enter the name of the SIP header manipulation configuration that you want to apply to the traffic entering the Communications Broker through this session agent. No default.
Out manipulation ID	Enter the name of the SIP header manipulation configuration that you want to apply to the traffic exiting the Communications Broker through this session agent. No default.
Manipulation String	Support SAG based source routing by configuring an applicable **manipulation-string** to your Session-Agent configuration. This HMR should replace the configured string with the user-portion of the From URI. This configuration only applies if you have configured HMR for the applicable flows.
Early media inhibit	Select to inhibit early media. Default: disabled. Valid values: Disabled, enabled.

Table 5-1 (Cont.) Fields in the Agents Page

Field	Description
Enable options ping	Select to enable the use of OPTIONS pings to determine the status of this agent.
LDAP	Select the name of the ldap-group or ldap-config-group that you want this agent to use.
Additional target group	Select an additional target group from the drop-down list.
Fork group	Enter the number for the fork group number, which determines this session agent's priority in a target list. The lower the number, the higher the priority. Default: 1.
Refer call transfer	Specifies whether and how to execute a REFER • Disabled—Proxy the REFER (default) • Enabled—Issue a Re-INVITE to the transfer destination • Dynamic—Refer to the target realm to determine whether to proxy or issue a Re-INVITE
Refer notify provisional	https://docs.oracle.com/en/industries/communications/enterprise-communications-broker/3.3.0/index.html
Reuse Connections	Set the protocol that can use the SIP Connection Reuse feature. • None—Do not support the SIP Connection Reuse feature. • TCP—Support the feature for TCP. • UDP—Support the feature for UDP.
TCP Keepalive	Enable or disable standard keepalive probes to determine whether or not connectivity with a remote peer is lost. The default value is none. The valid values are: • None (default) • Disabled—Do not support the interworking. • Enabled—Support the interworking.
TCP Reconn Interval	Enter the amount of time in seconds before retrying a TCP connection. The default for this parameter is 0. The valid range is: • Minimum—0 • Maximum—300
Monitoring filters	lick **Add** to add one or more monitoring filters.
Send TCP Fin	Check to Enable or uncheck to disable the system from sending TCP FIN messages when the session agent is Out of Service. The default is disabled.
Trigger Oos Alarm	Check the Enable checkbox to trigger alram when a Session Agent is OOS. Uncheck the Enable checkbox to disable. Default value is: Disabled.

3. Click **OK**.
4. Save the configuration.

Configure a Session Agent Group

Create a session agent group to define a signaling endpoint configured to apply traffic shaping attributes and information about next hops and previous hops.

Session agent groups contain individual session agents in a logical grouping. You can configure each member of the group with individual constraints. The system does not require all members of a group to apply the same constraints.

Session agent group members do not need to reside in the same domain, network, or realm. The Oracle Enterprise Communications Broker (Communications Broker) can allocate traffic among member session agents regardless of their location by using the allocation strategy that you select allocate traffic across the group members.

1. Access the Groups configuration object.

   Configuration tab, **Service Provisioning** section, **Agents**, **Groups**.

2. On the **Groups** page, click **Add** and do the following:

   Table 5-2 Groups Page

Field	Description
Group Name	Enter a name for this Session Agent Group.
Description	Enter a description of this Session Agent Group.
Strategy	Select a strategy for choosing this agent from the drop-down list. Default: Hunt. Valid values: Hunt | Round Robin.
Stop SAG Recurse	Enter one or more response codes that you want to cause this session agent group to stop route recursion. Default: 401,407. Valid response code values range from 300-599. You can enter individual response codes separated by a comma, such as 301,305 or a range such as 300-380.
Dest	Enter one or more destinations for this group.
SAG Recursion	Select to enable Session Agent Group recursion. Default: Disabled.

3. Click **OK**.
4. Save the configuration.

Configure ENUM Servers

You can create E.164 Number to URI Mapping (ENUM) server configurations on the Oracle Enterprise Communications Broker (Communications Broker) to resolve SIP URIs presented to the Communications Broker in a call. An ENUM server configuration points to one or more ENUM servers from which the Communications Broker can request resolutions. Configuring with multiple servers provides redundancy when a particular server cannot respond or provide a resolution.

- Configure a top-level domain
- Configure one or more ENUM servers to add to the ENUM settings configuration

Similar to an agent configuration, an ENUM server configuration includes number translation settings for the strings returned by the ENUM infrastructure. The ENUM configuration also includes configuration values to support the interaction with the servers.

1. Access the ENUM Servers configuration object.

   Configuration tab, **Service Provisioning** section, **Agents**, **ENUM Servers**.

2. On the ENUM Servers page, click **Add**, and do the following:

Name	Enter a unique name for this ENUM configuration. You may need to use this name in other areas of the Communications Broker configuration to refer to this ENUM configuration. For example, in route configuration.
Top Level Domain	Enter the domain extension to use when querying the ENUM servers for this configuration. For example, e164.arpa. The query name is a concatenation of the number and the domain.
Realm ID	Enter the name (in string) of the Realm in which the ENUM servers are located.
ENUM Servers	Enter one or more ENUM servers (an ENUM server and corresponding redundant servers) to query. Separate each server address with a space and enclose list within parentheses. **Note:** The Communications Broker media interface does not support management traffic for ENUM. When configuring connectivity to a media interface, do not configure these resources within a media interface subnet range.
Number Translation Mode	Select the translation mode required by this agent. The modes define how to format the ENUM request to accommodate the specific ENUM server. Valid values: • E164 - Default. The server can accept numbers in E164 format. • E164-no-plus. The server uses numbers in E164 format, with the exception of the plus sign. • no-country-code. The server cannot use a country code. • pattern-only. The server cannot use any string that varies from the configured pattern. • n-digit-dialing. The server requires the specified number of digits.
Number of Digits for n Digit Dialing	If you selected n-digit-dialing as the Number Translation Mode for this agent, specify the number of digits the Communications Broker must send to this server.
Prepend Prefix	Specify a prefix the Communications Broker must send to this server. For example, the digit 9, which may be required to allow outbound traffic.
Advanced Settings	Expand to display the following settings.

Query Method	Set the strategy the Communications Broker uses to contact ENUM servers. Valid values are: hunt—Directs all ENUM queries toward the first configured ENUM server. When the first server is unreachable, the Communications Broker directs all ENUM queries to the next configured ENUM server, and so on. round-robin—Cycles all ENUM queries sequentially among all configured in-service ENUM servers. The Communications Broker directs query 1 to server 1, query 2 to server 2, query 3 to server 3, and so on.
Timeout	Set the number of seconds to elapse before a query sent to a server (and its retransmissions) times out. Range: 0-4294967295 seconds.
Lookup Length	Set the length of the ENUM query, starting from the most significant digit. Default: 0. Range: 1-255.
Max Response Size	Enter the maximum size in bytes for UDP datagrams in DNS NAPTR responses. Range: 512 (default)-65535. Oracle recommends configuring values that do not exceed 4096 bytes.
Health Query Number	Set a standard ENUM NAPTR query that will consistently return a positive response from the ENUM server. Blank = disabled.
Health Query Interval	Set the number of seconds to perpetually probe ENUM servers for health. Range: 0-65535 seconds.
Options	Enter optional features or parameters. Enter multiple values using the comma as a delimeter.
Translation Mode	Outbound number translation mode for agent. • E164: E164 number. This is the default value. • no country code: number without country code • n-digit dialing: Specify the number of digits to via number-digits field. • pattern-only - Use portion matching dial-pattern only
Number Digits	Enter a value between 0 25
Prepend Prefix	Specify digits to be prepended to an outbound number after the transcation telephone digits. Maximum 25 characters. • 0-9 • star • pound • A-D

3. Click **OK**.
4. Save and activate the configuration.

Enable ENUM Session Agent Group Matching

When you want the Oracle Enterprise Communications Broker (Communications Broker) to consider session agent groups as part of the response from the ENUM server, set the **Enum Sag Match** parameter in **SIP Config** to enable. With this parameter enabled, the

Communications Broker matches session agent group names received from the ENUM server with the hostname portion in the naming authority pointer. In this way, the Communications Broker supports a group of session agents as a SIP designation in the Communications Broker ENUM request.

1. Access the SIP Interface configuration object.

   Configuration tab, **System Administration** section, **SIP Interface**, **SIP Config**.
2. On the **Modify SIP Config** page, enable **Enum Sag Match**.
3. Click **OK**.
4. Save the configuration.

Multi-Hop Agent Ping

The Oracle Enterprise Communications Broker (Communications Broker) ping function can test connectivity to endpoints that are not directly adjacent to the source Communications Broker. This multi-hop ping capability requires that you configure special routes dedicated to sending SIP options pings to these targets.

To enable ping tests to targets that are more than one hop from the Communications Broker, configure routes that have the string "`ping:`" in the **Called number** field. These routes also have a first agent towards the target configured in the **Route** field, and the last agent toward the target configured in the **Destination Agent** field.

The system invokes these `ping:` routes for ping traffic only. In addition, the system prioritizes SIP signaling traffic over `ping:` route traffic.

The following table shows an example of a multi-hop ping route.

Source Agent	Calling Number	Destination Agent	Called Number	Route	Cost	Policy
*	*	Target_Agnt	ping:	Adja_Agnt	0	

Having configured this route, you can initiate a ping to a target, or configure agent pinging to `Target_Agnt` by way of `Adja_Agnt`.

You can also set up multi-hop ping recursion by creating multiple `ping:` routes that specify the complete path to the target. To create these paths, you can configure `ping:` routes using the `Destination Agent` and `Route` fields to define each hop in a "ping path".

Based on the two route entries in the following example, ping attempts to reach the device defined as `Target_Agnt` follow a two-hop path:

1. Communications Broker to `Adja_Agnt`
2. `Adja_Agnt` to `Interim_Agnt`
3. `Interim_Agnt` to `Target_Agnt`

Source Agent	Calling Number	Destination Agent	Called Number	Route	Cost	Policy
*	*	Interim_Agnt	ping:	Adja_Agnt	0	
*	*	Target_Agnt	ping:	Interim_Agnt	0	

The Communications Broker uses an agent status, determined by OPTIONS ping, to validate all routes using that agent. Specifying an agent's status, including in-service and out-of-service, is the same for agents using either single or multi-hop ping. The system does not use routes to out-of-service agents for any signaling traffic.

6

Dial Plan Configuration

Dial plans specify how you want the Oracle Enterprise Communications Broker (Communications Broker) to process calls and route them according to contexts and patterns that you configure. Use the Dial Plan icon on the Configuration tab to access the Dialing Contexts configuration page, where you define how you want the system to handle dial patterns.

The Dialing Contexts configuration page displays the following permanent dialing context hierarchy parents:

- Corporate—The parent of child dialing contexts of groups of users that you add, who share dialing patterns that you specify. For example, the employees in a branch office. You can add thousands of child contexts to the Corporate parent and you can set thousands of dial patterns for each child.
- Geographic—The parent of sets of child dialing contexts pre-defined by the Communications Broker for every geographic location in the world. You can add thousands of child contexts to the Geographic parent and you can set thousands of dial patterns for each child.

Plan and configure your contexts in hierarchical priority. The Communications Broker always uses the most specific match it can find when performing dial pattern matches. When the Communications Broker finds no match in a child context, it searches for a dial pattern match in the parent context.

You can configure a dial plan on the Communications Broker by way of the Web GUI or you can upload a dial plan in a .csv file.

Dial Pattern Configuration

A dial pattern defines the prefix and pattern that the Oracle Enterprise Communications Broker (Oracle Enterprise Communications Broker) receives and defines the transformation that the system performs.

Only one transformation type is valid for each prefix and pattern match. The system displays an error when you try to configure multiple transformation types. Transformation types include replacement prefix, replacement URI, and Go to context. The following table shows examples of patterns, the transformation types, and the results of the transformation.

Prefix and pattern	Transformation type	Result
8 - xxxx	Replacement prefix	Replace with configured digits, which in this case are suitable for an outside line.
911	Replacement URI	Insert the configured service URN for emergency services.
*123	Go to context	Present the prefix/pattern to another context. Matching occurs based on the target context's configured dial patterns.

Overlapping dial pattern matches in the same context that result in finding the same target number produce configuration errors. Such errors produce an ambiguity that the system cannot resolve, so it does not forward the message. You must configure patterns, especially those that use encoding characters, very carefully to avoid ambiguities that the system cannot resolve.

When overlapping patterns result in the same target number, the system forwards to that number without error. When the system finds overlapping patterns in different contexts, the system chooses the most specific context as a match. Child contexts are more specific than parent contexts. When configuring dial patterns, Oracle recommends keeping them unique even across contexts.

Dial Pattern Encoding Characters

The Oracle Enterprise Communications Broker (Communications Broker) allows multiple matches to a specified dial pattern, such as the ones you add to a dialing context. You can use selected characters to help you encode dial patterns to meet your needs.

In the following table, the Character column lists the encoding characters that a dial pattern allows and the Usage column explains what the characters mean in a pattern and how to use them.

Character	Usage
Brackets [ ]	Use brackets to enclose digit ranges you need to express for a pattern. TheCommunications Broker parses the pattern 8[1-20]9 as 8[01-20]9, adding an implied 0 before 1. The Communications Broker considers both values to contain the same number of digits. The Communications Broker strictly enforces the range and the number of characters in the preceding examples, as follows: • 8019 matches • 819 does not match • 8119 matches
The "x" character	Use the x character as a wildcard in dial pattern strings. Use the "x" character can only at the end of a string. When you configure a pattern that includes an "x" character followed by digits, the system displays an error.
Parenthesis ( )	Use parenthesis to enclose wildcard characters and express a pattern. The Communications Broker does not strictly enforce the range and the number of characters in patterns with "x" characters in parenthesis. Consider the pattern 8xx(xx): • 812 matches • 8123 matches • 81234 matches

Note that the use of encoding characters can result in overlapping dial-pattern matches. Overlapping dial-pattern matches that result in multiple targets introduce ambiguity that the Communications Broker cannot resolve. As a result, the system does not forward the signaling.

For example, the following two dial-patterns overlap:

- 4000
- 4xxx

Double check dial patterns made up of encoding characters to avoid overlaps.

Configure a Dial Plan

You can add one or more child dialing contexts to the Corporate and Geographic parent dialing contexts to specify how you want the Oracle Enterprise Communications Broker (Communications Broker) to extrapolate and present universal strings to the routing engine upon ingress, as well as to create target URIs for egress based on these rules.

Configuration includes specifying the applicable:

- Prefixes, including:
  - Access codes
  - Tie line digit sequences
- Services, such as 411 and 911 services
- Dialing Ranges
- Dialing Range exceptions (gaps in dialing ranges)

1. Access the Dial Plan configuration object.

   Click **Configuration** tab, **Service Provisioning** section, **Dial Plan**.

2. On the Dialing Contexts page, click the **Add** icon.

   The attributes and instructions in the next step apply to both Corporate and Geographic.

3. On the Add Dialing Context page, do the following:

Name	Enter a name for this dial plan. Required.
Geographic Location	Select a geographic location from the drop-down list to use for pattern matching when the Communications Broker cannot find a pattern match in this context's dial-patterns.
Description	(Optional) Enter a description of this dialing context.
Country Code	Enter the E.164 country code for which this dialing context exists. For example, 1 for North America.
Outside Line Prefix	Do one of the following: • Enter the characters required for PSTN access. Valid values: 0-9, *, #, and A-D. Maximum: 25 characters. • Leave blank to allow direct dialing.
Dial Pattern	1. Click **Add**, and do the following: • Remove Prefix—Enter the prefix of the dial pattern entry that you want stripped for translation. Valid values: 0-9, *, #, and A-D. Maximum: 25 characters. Allow ranges in brackets []. For example, 555[2000-3999].

- Pattern—Enter telephony digits 0-9. Maximum: 25 characters. Allow ranges in brackets []. For example: 555[2000-3999]. Use x at the end of a pattern to mean 0-9. Use parens ()around optional digits. For example: 555xx(xxxx) means 555 followed by 2 to 6 digits.
- Description—Enter a description of this dial pattern.
- Country Code—Enter the country code for this dial pattern or leave blank to inherit from the context.
- Replacement Prefix—Enter the replacement prefix to add to the translated number. Valid values: 0-9. Maximum: 25 characters.
- Replacement URI—Enter the URI to use without outbound translation.
- Go to Context—Select the target context from the drop-down list.

2. Click **OK**.
3. (Optional) Add another dial pattern.

4. Click **OK**.
5. (Optional) Add another child dial plan.
6. Save the configuration.

7

User Configuration

User configuration is a required task. It is, in fact, the only way to assign an agent to a user. Complex VoIP deployments can derive additional benefits from the user database. Examples include contiguous dial strings being deployed across multiple PBXs, making it impossible to identify the target PBX by dial string alone.

When you need a user database, you configure entries for all source and destination numbers that you know require user database support. The user database effectively performs dial plan tasks on individual numbers. Such tasks include identifying a user's agent and source context. These entries provide the Oracle Enterprise Communications Broker with shortcuts for determining this information.

A user often represents an actual account that exists within an enterprise's dialing network. Examples of users include an extension, a subscriber or a phone number. Each user may have a source context, which translates into "Default" dialing rules for processing that user's calls using the appropriate contextual transformation rules and the agent at which they are located.

You can configure this database manually. Alternatively, you can upload user information in a format pre-configured to translate into a user database.

Configure a User Entry

To build a user database, you configure individual entries for users. Each entry defines the name or dialing pattern, dialing context, agent, and policy for a user. You can optionally add a description and tags to create and filter reports for users. A user can represent an account in your dialing network, such as an extension, a subscriber, or a phone number.

- Configure one or more agents.
- Configure one or more policies.
- Plan any tags that you want to use to create and filter reports for users.

In the following procedure you can add multiple policies and tags to this user record. Repeat the procedure for each user that you want to manually add to the user database.

1. Access the User Entries configuration object.

   Click the **Configuration** tab, **Service Provisioning** section, **User Entries**.
2. On the User Entries page, click **Add**.
3. Do the following on the Add User Entries page:

AoR	Enter the user's address of record. Default: Empty. Default: Empty. Valid values: Alphanumeric characters in the following formats: username@companyname.com | username@ipaddress. See "Alphanumeric User Database Entries" for more information.
Name or pattern	Enter the E.164 number associated with this user using the same pattern rules as in dial plans. Do not include the + character.
Description	Enter a description of this user.

Dialing context	Select or enter the name of the context that defines the preferred dialing rules for this user.
Agent	Select or enter the name of the agent to which this user is connected. This agent is the closest agent to the user.
Policy	Add one or more policies to apply to this user.
Tags	Add one or more tags to this user entry.

4. Click **OK**.
5. Save the configuration.

Resolving to the Longest Match in the User Database

The Oracle Enterprise Communications Broker (Communications Broker) user database supports resolution of overlapping numbers during lookups by selecting entries with the longest matches.

Flexibility in creating user database records allows for overlapping records, which can create ambiguity when the Communications Broker looks for a match. To resolve such ambiguity, the Communications Broker selects the entry that matches the most digits. As you build your user database, consider how you create user database entries to take advantage of flexible matching while preventing ambiguity.

The following table shows a set of overlapping user number entries in the Pattern column. The Match column shows the corresponding number of digits that must exactly match the pattern. Note that the match length does not include number ranges and wildcards.

Pattern	Match
17815551111	11
17815551[000-999]	8
17815551[111-999]	8
17815551xxx	8
1781555(x)xxx	7
xxxxxxxxxx	0

When performing a user database lookup, the system uses the entry that matches the pattern with the longest match length. The following table explains how different dialed numbers match the numbers configured in the user database. The Dialed Number column lists examples of dialed numbers. The Matching Pattern column lists the configured pattern to match. the Reason column explains how the match does or does not work.

Dialed Number	Matching Pattern	Reason
17815551111	17815551111	This dialed number matches all patterns with an exact match and has the highest match length (11).

Dialed Number	Matching Pattern	Reason
17815551112	17815551[000-999] or 17815551[111-999] or 17815551xxx	This dialed number matches all but the first pattern, but the three patterns have the longest match length of 8. When multiple entries have the same match length, the selection between them is undefined and causes an unsuccessful lookup due to the ambiguity.
17815552111	1781555(x)xxx	This number matches only the last two patterns, but this pattern has a higher match length (7).
22222222222	xxxxxxxxxxx	This number matches only the last pattern.

8

Using Policy to Refine Routing

The Oracle Enterprise Communications Broker (Communications Broker) supports policy-based routing, allowing the user to select pre-defined policies or create their own policies and apply them to routes or the Registrar. In turn, these policies impact the behavior of the applicable routes when traffic matches user-defined conditions. The user configures new policies from the Communications Broker's **Policy** icon (or uses pre-built policies) and applies them to routes and/or the Registrar. Routes support multiple policies.

Policy provides the Communications Broker with a generic approach to configuring routing applications. Policy configuration assumes a desired behavior that has been identified by the user. The most common objectives include:

- Establish more specific routing decisions
- Apply additional services

The Communications Broker abstracts policy behavior into three components including:

- Route
- Condition
- Action

The combination of route and condition define when the policy applies. The action refines the way in which the traffic uses the route. Note that possible actions may include not using the route.

This generic approach to route policy provides great flexibility in policy definition, but also imposes a level of complexity on the user, requiring them to:

- Identify the application they want to create.
- Determine how to identify the applicable traffic.
  - Use or create new routes specifically for the application.
  - Define the condition that causes the system to apply the policy to the route.
- Test traffic matching and application action.
- Ensure no overlapping configurations cause the system to use or not use the policy unexpectedly.

There are two components of a policy configuration:

- Conditions:
  - codec-condition—Tell the system to determine the presence of absence of specific codecs within an offer.
  - time-condition—Specify the day(s) of the week and time(s) of the day when the system uses the policy.
- Actions:
  - routing-action—Tells the system to modify how the route is applied.

  - redirect-action—Tells the system to direct traffic to the configured agent when the route and conditions match.
  - outbound-translation-action—Tells the system to perform the configured outbound translation when the route and conditions match.

Policies allow for both multiple conditions and actions. All conditions must be met for the condition to be true. Alternatively, the user can configure no conditions, meaning the policy's condition is always true. When a policy includes multiple actions, it performs all of the actions in the configured order.

The user may also need to create route(s) specific to the policy. Whether an existing or newly configured route, the user configures the route to use one or more policies to complete the application configuration.

The Redirect Action

The redirect action causes the Oracle Enterprise Communications Broker (Communications Broker) to redirect the incoming call through a particular agent by way of policy. You can use redirect to send a call to an external resource or service, such as a transcoding Session Border Controller or a call-recorder.

When applied, the policy engine performs an additional routing lookup for the call to the specified redirect agent. The system pre-pends additional hops from the redirect to the hops that were already calculated for the current route. The redirect action adjusts the routes to send the call to the specified redirect agent first, and then to the call destination.

Redirect action configuration includes the**Hairpin signaling** field. When you enable **Hairpin signaling**, the Communications Broker routes the call to the redirect agent first, then routes it back to the Communications Broker before sending it to the original destination. Hairpin signaling ensures that the Communications Broker can route the call even when the redirect agent cannot reach the final destination. Note that keeping **Hairpin signaling** disabled eliminates the extra hops and extra session required when the destination is reachable by the redirect agent.

The Communications Broker uses the same source agent, calling number, and called number parameters as the original call to reach the redirect agent. Only the dest-agent parameter gets replaced with the redirect agent specified in the redirect policy action. Take special care with default routes or routes that use a wildcard in the dest-agent field because such routes can become part of the path to the redirect agent.

Note the following details when evaluating and configuring redirect action:

- Hops incurred by the redirect action do not affect the route cost. The system determines the route-set and order-set before the redirection takes place.
- The system does not evaluate policies applied to the redirect routes, which prevents redirection loops and other undesirable behavior.
- The system uses only the first (lowest cost) redirection path, which prevents the exponential increase of backup paths.
- You can configure redirection to agent groups, which operate normally.
- The system applies the same routing parameters of the call (source agent/number and dest number) to the redirection route lookup as the original route.
- The system does not use default routes ('*' for all match patterns) for redirection. When the Communications Broker finds no valid routes for the redirect, the Communications Broker rejects the call.

Configuring CNAM Replacement

The Oracle Enterprise Communications Broker provides the user with the ability to specify the value of the caller name (CNAM) in the FROM header. A simple use case consists of an enterprise inserting the name of their company into the FROM value, in place of the original caller name. You configures the system to use other policy or routing configurations to determine when to replace a CNAM. The system applies this policy action on SIP requests immediately prior to egress.

To configure CNAM replacement, add the **cnam-masking-action** action to the affected policy. The **Modify Policy / cnam masking action** dialog includes two fields.

- **Name**—Assigns a name as an identifier to your action.
- **Display name**—Defines the text the Oracle Enterprise Communications Broker inserts as the CNAM value in the FROM header.

Note that, if no user name is in the original FROM, the Oracle Enterprise Communications Broker inserts text configured via the CNAM mask and encloses it in brackets per RFC 2822.

Using Policy to Normalize SIP Headers

The Oracle Enterprise Communications Broker supports policy-based SIP Header Normalization, allowing the user to copy and change information in headers when their user parts are Tel-URIs or SIP URIs composed of numbers. The system writes header changes caused by policy after any inbound and before any outbound manipulation performed by header manipulation rules. These policies work for registered users and targets derived from the user database or LDAP. The user configures header normalization policies from the Oracle Enterprise Communications Broker's **Policy** icon and applies them to routes and/or the Registrar. Routes support multiple policies.

The header normalization policy works with the existing outbound translation policy. Each policy consists of one or more rules by which the system changes headers to messages that match the routes or registrar to which the policy is configured. Multiple rules can be defined for the same SIP header. The system evaluates and executes rules in the configured order and changes the header values with each rule that has a valid **New value** field.

Header normalization configuration fields include:

- **SIP header name**—The name of the SIP header to be normalized. This field cannot be empty.
- **Dialing context**—Name of dialing context that defines the dialing rules to be applied to the phone number in this SIP header. An empty value in this field indicates that no dialing rules need to be applied to the corresponding SIP header in the rule.
- **Result name**—A temporary variable name to store the result of the dialing rules on the SIP header. This name has to be unique within this policy action. The value is saved between policy rule evaluation, but is not saved after the policy is fully evaluated.
- **New value**—The **Result name** whose value will be used as the new value of the SIP header. If this field is left empty, the system does not change the value of the SIP header.

If a rule specifies a SIP header name that is absent in the SIP request and the **New value** field has a valid **Result name**, the system adds the SIP header and set it to the value of **Result name**.

ANI Masking

The Oracle Enterprise Communications Broker provides a means for ensuring that the automatic number identification (ANI) presented to a service provider be recognized as a valid screened telephone number (STN) and, therefore, is not dropped by that service provider. The user configures this function as a Header Normalization policy that rewrites the applicable header with a recognizable STN. Applicable headers are DIVERSION, P-ASSERTED-IDENTIFY or FROM, depending on the service provider's requirements.

ANI Masking Configurations

The following example shows a header normalization policy that ensures that the system presents an Automatic Number Identification (ANI) as a valid Screened Telephone Number (STN). The example assumes that you created a dialing context from which the system can determine an STN.

Policy

Name	Description
ANI_Mask1	North American international dialing

Condition

The default condition, which requires no configuration, is to always apply the policy.

Actions

The following table shows an example configuration for the action named ANI_MaskforSP1, which is of the header-normalization-action element type with two actions. The first row in the table shows the SIP header name as "From," the dialing context, and the result name. The second row shows the SIP header name as "Diversion," the result name, and the new value.

Table 8-1 Example Configuration

SIP Header	Dialing Context	Result Name	New Value
From	find_Verizon_STN	STN	NA
Diversion	NA	original_diversion	NA

The results of the actions in the preceding table follow.

1. Use the value of the "from" header and run it through a dialing context called "find_Verizon_STN" and store the result in a variable called "stn". Because the new value field is empty the value of the "from" header is unaltered.
2. Because the dialing context is empty, copy value of the "Diversion" header to the variable called "original_diversion". If the incoming SIP request did not have a "Diversion" header "original_diversion" will be set to an empty string. Copy the value of the variable "STN" into the "Diversion" header.

Route and Registrars

Configure either routes or registrars as triggers by setting their **Policy** fields to the **ANI_Mask1** policy.

Results

The following table shows the results of this example configuration. The SIP Message Text column shows the SIP To and From addresses. The Mask Function column shows the dialing context used to create the new value. The Result SIP Message Text column shows the SIP To, From, and Diversion addresses.

SIP Message Text	Mask Function	Result SIP Message Text
To: sip:781.630.1111@oracle.com From: sip:781.630.2222@oracle.com	Use Dialing context titled "find_Verizon_STN" to create new value titled "STN"	To: sip:781.630.1111@oracle.com From: sip:781.630.2222@oracle.com Diversion: sip:978.528.1234@oracle.com

Enabling Policy-based Routing

Starting with Release 5.0.0, Oracle Enterprise Communications Broker supports flexible policy-based routing, where you can define policies that control how calls are handled. Each policy has conditions, and actions. All conditions must be true for the policy to apply.

Starting with Oracle Enterprise Communications Broker Release 5.0.0, you can apply policies on a calling number, in addition to the existing facility to apply policies on called number. Applying routing policies to the calling number, and the called number, enables both the origin and destination of a call to be evaluated against their respective policies in the user table.

During the configuration of a policy, a new parameter named `userdb-lookup-mode` controls the application of policies to both TO and FROM calls. For more information, on this seeDefine a Policy.

The `userdb-lookup-mode` parameter does not influence the application of policies from the routing table. Its significance is limited to policies applied from the user table.

In an HA environment, the configuration is replicated to the Standby system. The Standby system uses the configured value of `userdb-lookup-mode` parameter to apply policies for each call.

During an upgrade, the new `userdb-lookup-mode` parameter does not affect the existing behavior. When upgrading from an unsupported version, the `userdb-lookup-mode` parameter defaults to the CALLED_NUMBER. This ensures that the current routing policies work as before. When upgrading from a supported version, the `userdb-lookup-mode` parameter continues to be the same as configured.

In a downgrade scenario, since the `userdb-lookup-mode` parameter is not available, policies are applied apply only to called numbers.

Policy Priority

In case of conflicting policies, Communications Broker retains the existing behavior by adhering to in-built order of priority.

This hierarchy ensures that higher-priority policies override lower-priority ones, maintaining a consistent and predictable system behavior. Each policy serves a specific function—such as blocking actions, controlling recursion, or translating headers - to ensure that the most critical policies are enforced first, while others are processed in a defined sequence.

Define a Policy

The Policy configuration object the Oracle Enterprise Communications Broker (Communications Broker) provides access to the policy list and configuration dialogs, where you can define and manage policies.

- Analyze traffic patterns and message content before configuring a custom policy.
- Identify the message contents and metadata that allows a policy to uniquely target the traffic.
- Configure any agents the you want to use for redirected calls.

Use the following procedure to add a new policy or modify an existing one.

1. Access the Policy Entries configuration object. **Configuration** tab, **Service Provisioning** section, **Policy Entries**.
2. On the Policy Entries page, click **Add**, and do the following:

Parameters	Instructions
Name	Enter a name for the policy.
Description	Enter descriptive text that explains the purpose of the policy.
UserdbLookup Mode	Controls how policies are applied to both the Calling and Called numbers. Administrators will have three options: • CALLING_NUMBER: Apply policies to the matched FROM number. • CALLED_NUMBER: Apply policies to the matched TO number. This is the default setting. • BOTH: Apply policies to both numbers. **Note:** If both calling and called party policies are present, Communications Broker applies the calling party's policy first followed by called party policy for the user.
Conditions	Set the conditions under which the system performs the actions that you specify.

Parameters	Instructions
For Codec Condition	For Codec Condition, do the following: a. Click **Add**. b. Click **Codec Condition**. c. On the Add Policy / Codec Condition page, do the following: i. Name—Enter a name for the Codec Condition. ii. Do one or both of the following: i. Contains Codecs—Click **Add**, select a codec from the drop-down list, and click either **OK** or **Apply/Add another**. ii. Missing Codecs—Click **Add**, select a codec from the drop-down list, and click either **OK** or **Apply/Add another**. d. Click **OK**.
For time-condition	For time-condition, do the following: a. Click **Add**. b. Click **Time Condition**. c. On the Add Policy / Time Condition page, do the following: i. Name—Enter a name for the Time Condition. ii. Days—Click **Add**, select a day from the drop-down list, and click either **OK** or **Apply/Add another**. iii. Start time—Set the time when you want the condition to start on the specified day. iv. End time—Set the time when you want the condition to end on the specified day. d. Click **OK**.
Next hop compare condition	a. Name: Unique name b. Mode for next hop comparison: Allowed values: from-uri, previous-hop. Default value is pre-hop.
Actions	Set the one or more actions that you want the system to perform for each specified condition.

Parameters	Instructions
For Routing Action	For Routing Action, do the following: a. Click **Add**. b. Click Routing Action. c. On the Add policy / Routing Action page, do the following: i. Enter a unique name for the action. ii. Select a routing mode from the drop-down list. d. Click **OK**.
For Redirect Action	For Redirect Action, do the following: a. Click **Add**. b. Click Redirect Action. c. On the Add policy / Redirect Action page, do the following: i. Name—Enter a unique name for the action. ii. Redirect to Agent—Select an agent to redirect the call to before sending the call to the destination. iii. Hairpin Signaling—Select to enable rerouting a call back to the Communications Broker from a redirect agent before sending the call to the destination. d. Click **OK**.
For Outbound Translation Action	For Outbound Translation Action, do the following: a. Click **Add**. b. On the Add policy / Outbound Translation Action page, do the following: i. Name—Enter a unique name for the action. ii. Egress Number Translation Mode—Select a mode from the drop-down list. iii. Number of Digits for n Digit Dialing—If you selected n-digit-dialing as the number translation mode, enter the number of digits the Communications Broker must send to the agent. Range: 0-25. iv. Prepend Prefix on Egress—Specify the number of digits to prepend to the outbound number after translation. Valid entires include telephony digits 0-9, star, pound, and A-D. Max: 25 digits. c. Click **OK**.

Parameters	Instructions
For Constraints Action	For Constraints Action, do the following: a. Click **Add**. b. On the Add policy / Constraints Action page, do the following: i. Name—Enter a unique name for the action. ii. Ignore Constraints—Select to ignore call admission control restraints. Default: Disabled. c. Click **OK**.
For Header Normalization Action	For Header Normalization Action, do the following: a. Click **Add**. b. On the Add Policy / Header Normalization action page, do the following: i. Name—Enter a unique name for the action. ii. Header Normalization rules—Click **Add**, and do the following: i. Header Name—Enter the name of the SIP header to normalize with dialing rules. Required. ii. Dialing Context—Select a dialing context from the drop-down list. iii. Result Store—Specify the temporary variable name to store the result of the dialing rules on the SIP header. iv. New Value—Enter the result name to use as the new value in the SIP header. The system does not change the value of the SIP header unless you specify a value. c. Click **OK**.
For CNAM Masking Action	For CNAM Masking Action, do the following: a. Click **Add**. b. On the Add policy / CNAM Masking Action page, do the following: i. Name—Name—Enter a unique name for the action. ii. Display Name—Enter the caller name replacement value to display in the SIP From header. For example, your company name. c. Click **OK**.

3. Click **OK**.

4. Save the configuration.

You must apply the policy to the Registrar and any route that you want by way of the Communications Broker Registrar or route configuration dialogs.

Applying a Policy to a Route

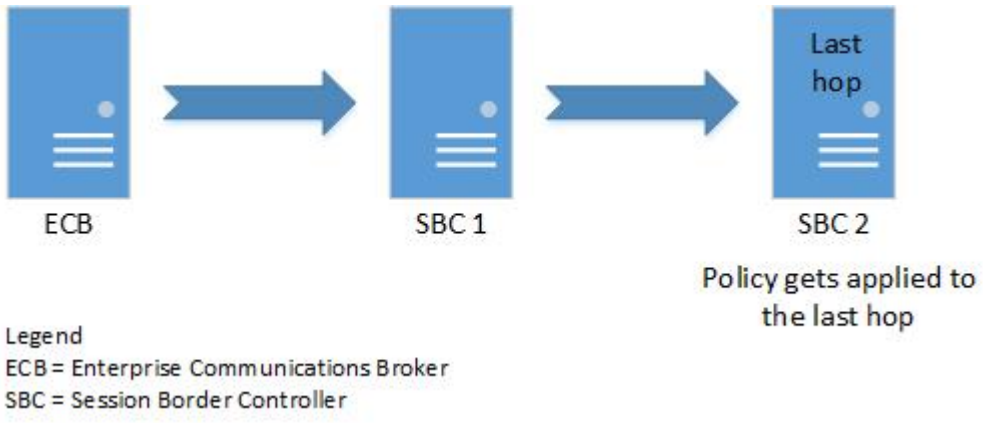
You can configure an Oracle Enterprise Communications Broker route Policy from the **Modify Routing entry** dialog. The dialog presents a list box titled **Policy**, with controls that allow you to **Add**, **Edit** or **Delete** policies to the route. When you click **Add**, the system displays a drop-down selection box displaying the names of all currently configured Policies. This list includes user-configured policies, as well as the system's pre-configured policies, Deny, Emergency and Stop-Recurse. You select the policy you want and click **OK**,

Runtime Routing with Policies in the User Table

You can apply a policy to the Oracle Enterprise Communications Broker user table for more granular control over where a policy gets applied to runtime routing decisions.

Applying a policy to the user table provides more granularity than the routing table can provide because user table entries allow individual phone numbers and phone number ranges.

The Communications Broker applies policies from the user table to the last hop on the route list specified in the routing table.



You can use policies from the user table and the routing table together. When used together, the system obeys the policy in the user table first and the routing table second.

When you configure a policy in the User database for a particular entry, the Communications Broker applies the policy to all agents for that entry including any Additional Target Groups among those agents. When the session agent is a Session Agent Group, the Communications Broker applies the policy to the session agent of that Session Agent Group.

To apply a policy to the user table:

1. Add the policy to the Communications Broker.
2. Add the policy to a user entry in the user table.

Applying a Policy to the Registrar

You configure the Oracle Enterprise Communications Broker Registrar with a Policy from the **Modify Registrar settings** dialog. The dialog displays a list box titled **Policy**, with controls that allow you to **Add**, **Edit** or **Delete** policies to the registrar. When you click **Add**, the system displays a drop-down selection box displaying the names of all currently configured Policies. This list includes user-configured policies, as well as the system's pre-configured policies, Deny, Emergency and Stop-Recurse. You select the policy you want and clicks **OK**.

Configurations Using Policy

Application examples using policy as a primary construct are presented below. These application examples are valid for both the Small and Large Enterprise models as base configurations.

Configuration examples include:

- Priority Call Handling, Using Constraint Policy Action
- Transcoding, Using Redirect Policy Action
- Call Recording, Using Redirect Policy Action
- Domestic vs. International Call, Using Translation Action
- Deny Routes, Using Routing Action
- Stop Recurse Routes, Using Routing Action
- Skip Routes, Using Routing Action

Priority Call Handling

The Oracle Enterprise Communications Broker provides support for ensuring that high priority calls, such as 911 calls, are not constrained by system utilization thresholds. Non-priority calls are still subject to system constraint configuration. The user configures priority call behavior by creating a policy with the constraint-action, then applying that policy to the applicable routes. The Oracle Enterprise Communications Broker applies this policy to the configured route, as well as every subsequent backup route, even if the backup routes are not configured with the policy.

The system ignores constraints, such as session and rate/burst limits, for calls that match the priority route. The **constraints-action** provides a built-in action called **IgnoreConstraints**. This action causes the policy to mark a call to ignore constraints. All conditions configured on the policy must be satisfied for the action to be applied. The constraints-action is special in that, when enabled, it sets the call to ignore constraints for ALL routes. This is necessary because backup routes may be utilized that are not specific to priority/emergency calls.

For example, an emergency call might first be routed to a 911 service, but on failure it would be sent to the PSTN. The call cannot be limited by constraints on the PSTN route.

The built-in **Emergency** policy is provided for user convenience, and uses the **IgnoreConstraints constraints-action**. This built-in policy can be edited, if desired.

Note that this setting does not affect the "Deny" and "StopRecurse" policies on routes. It is simply not known whether these policies are for constraining purposes or because the network infrastructure cannot support calls over those routes.

Priority Call Configurations

This example shows a policy configuration that excludes 911 calls from system constraints, ensuring that the system does not encumber the call delivery. The following configuration documents the built-in policy titled "Emergency" applied to a route for 911 calls.

Name	Emergency
Description	Built-in policy to ignore constraints for emergency or priority calls.
Condition	The default condition, which requires no configuration, is to always apply the policy.
Action	• Enter a name for this action. • Set Ignore Constraints.

Example Route

In addition to the preceding configuration, the system needs the following route.

Source Agent	Calling #	Destination Agent	Called #	Route	Cost	Policy
*	*	*	911	EmrSvc	0	Emergency

Transcoding and the Oracle Enterprise Communications Broker

Transcoding is the conversion of media streams' encoding between endpoints that use different codecs. The Oracle Enterprise Communications Broker allows the user to configure routing policies that identify session agents deployed for transcoding media sessions and redirect applicable signaling and media streams to those agents.

Transcoding Configurations

This example shows a policy configuration that identifies an offer that does not include PCMU or PCMA and routes the call to a Transcoding device, in this scenario an SBC, to ensure that the media uses either PCMU or PCMA.

Name	XcodePcmuPcma
Description	Send calls that need transcoding to PCMU or PCMA to a transcoding SBC
Condition	• Name—XcodePcmu • Contains codecs—Intentionally blank

	• PCMU
Action	• Name—XcodeRedirect • Redirect to Agent—XcodeSBC • Hairpin signaling—Set to enabled

Example Route

In addition to the preceding configurations, the system includes the following route.

Source Agent	Calling #	Destination Agent	Called #	Route	Cost	Policy
*	*	PBX	*	PBX	0	XcodePcmu

This configuration results in the system sending traffic that contains the PCMU codec to the PBX, and traffic without the PCMU codec to the XcodeSBC, back to the Oracle Enterprise Communications Broker and then to the PBX.

Multiple Outbound Translations

The Oracle Enterprise Communications Broker provides a means for refining outbound translation configurations using routing policy. Fixed outbound translation configuration is available at the agent level. Routing policy, however, includes an outbound translation action that takes precedence over agent and sip interface configuration and applies translation based on individual route matches.

Outbound Translation Configurations

This example shows an outbound translation configuration that configures E164 numbers for domestic calls, and E164-no-plus with 011 prepended for international calls. The example calls for two policies and two routes.

Name	Policy 1—InternationalDialNA Policy 2—DomesticDialNA
Description	Policy 1—North American international dialing Policy 2—North American domestic dialing
Condition	The default condition, which requires no configuration, is to always apply the policy.
Actions	Outbound Translation Action (InternationalDial) • Name—InternationalDial • Egress number translation mode—E164-no-plus • Number of digits for n digit dialing—0 • Prepend prefix on egress—011 Outbound Translation Action (DomesticDial)

- Name—DomesticDial
- Egress number translation mode—E164-no-plus
- Number of digits for n digit dialing—0
- Prepend prefix on egress—Intentionally blank

Routes

In addition to the dial patterns in this example, the system needs the following two new routes.

Route #	Source Agent	Calling #	Destination Agent	Called #	Route	Cost	Policy
1	*	*	PSTN-NA	11*	PSTN-NA	0	InternationalDialNA
2	*	*	PSTN-NA	1*	PSTN-NA	10	DomesticDialNA

Results

The configuration in this example provides the results shown in the following table.

From	Dial String	Transformation	Result
*	15551234	1555123415551234	Call is routed without dial string change
*	34555555	11345555550	Call is routed with dial string change

The system routes any call that includes a country code and does not begin with the digit "1" internationally, and with the applicable transformation.

Routing Action Configurations

The Oracle Enterprise Communications Broker includes three pre-configured **Policies** (based on the pre-configured **DenyAction**, **StopRecurseAction** and **IgnoreConstraints** routing actions) that the user can apply to routes. In addition, the pre-configured **Skip** routing action mode is available for easy Policy configuration, which the user can then apply to routes.

Modes for routing actions include:

- **Deny**—The Oracle Enterprise Communications Broker should not forward the call at all.
- **StopRecurse**—Stop trying backup routes if the current route fails.
- **Skip**—Do not try this route, based on condition; proceed to any backup options.

Deny Route Policy Configurations

This example shows a policy configuration that prevents the system from allowing the request's source to reach the destination. The configuration shown here documents the built-in policy titled "Deny" applied to an example route.

The **Deny** policy in any resultant SIP routes to a destination prevents the Oracle Enterprise Communications Broker (Communications Broker) from forwarding a SIP request to the

destination point in question. You can use the **Deny** policy to prevent sessions between two endpoints for policy or cost reasons.

The Communications Broker includes a pre-configured deny route policy that you can use without modification.

Name	Deny
Description	Built-in policy to deny the incoming session
Condition	The default condition, which requires no configuration, is to always apply the policy.
Action	Name—DenyAction Routing Mode—deny

Example Route

In addition to the configurations in this example, consider the following policy applied against this route.

Source Agent	Calling Number	Destination Agent	Called Number	Route	Cost	Policy
Class_B	*	Protect_Agnt	*	Transit_Agnt	0	Deny

This route entry is designed to prevent calls passing through the This route entry may be installed as part of one or multiple routes that could potentially reach `Protect_Agnt`. The routing engine recognizes the presence of this route entry and rejects the call.

In addition, the system applies this Deny regardless of the backup route on which it is specified.

Stop Recurse Route Policy Configurations

This example shows a policy configuration that prevents the system from recursing through ensuing backup routes if the route configured with this policy stops responding. The configuration shown here documents the built-in policy titled "StopRecurse" applied to an example route.

A Stop-Recurse hop stops further attempts to forward a SIP request to a destination after the system tries the route with the Stop-Recurse hop. You can use the Stop-Recurse hop to prevent calls from being forwarded on certain routes that are cost-prohibitive or might cause loops in SIP call flows.

The Oracle Enterprise Communications Broker (Communications Broker) includes a pre-configured stop recurse policy, with Routing Action applied.

Name	StopRecurse
Description	Built-in policy to prevent further backup route attempts.
Condition	The default condition, which requires no configuration, is to always apply the policy.
Action	Name—StopRecurse Routing Mode—stop-recurse

Example Route

To expand upon the example, consider the following route.

Source Agent	Calling Number	Destination Agent	Called Number	Route	Cost	Policy
Class_A	*	Protect_Agnt	*	Transit_Agnt	0	StopRecurse

This route allows traffic to reach `Protect_Agnt` as long as it originates by way of the agent named `Class_A`.

For example, consider the scenario where an endpoint can be reached using two routes.

- Route 1: Agent_1 to Agent_2 (**StopRecurse**) to PBX
- Route 2: Agent_1 to PSTN

Route 1 uses a **StopRecurse** policy defined on the hop between Agent_2 and PBX. The Communications Broker stops processing the call if Route 1 does not receive a successful response. This configuration can prevent calls initially targeted for Route 1 from using the PSTN.

Stop Recursion by SIP Response Code

The Stop Recurse parameter provides you with control over how the system allows or stops recursion on routing by allowing you to set the specific response codes that you want the system to act upon. You can control recursion by response code globally through the SIP Interface configuration or locally through an agent or agent group configuration. Valid response code values range from 300-599. You can enter individual response codes separated by a comma, such as 301,305 or a range such as 300-380. The system uses response codes 401 and 407 for the defaults.

Skip Route Policy Configurations

This example shows a policy configuration that prevents the system from using this route based on condition. This policy contrasts with the StopRecurse policy because the routing engine is free to recurse through other routing options after skipping.

The Oracle Enterprise Communications Broker includes a pre-configured Skip Action, which you can select to define a policy.

Name	MySkip
Description	Do not use this route, depending on condition.
Condition	The following Time Condition Fields activate the skip policy on weekdays from 9am to 5pm. Name—When to enforce MySkip Days—Monday, Tuesday, Wednesday, Thursday, Friday Start time—09:00:00 End time—16:59:59
Action	Name—SkipRoute

	Routing mode—skip

Example Route

In addition to the preceding configurations, this example applies the following policy against the target routes.

Source Agent	Calling Number	Destination Agent	Called Number	Route	Cost	Policy
Class_B	*	Protect_Agnt	*	Transit_Agnt	0	MySkip

The resulting configuration prevents the system from using this route to **Transit_Agnt** as a means of reaching **Protect_Agnt** on weekdays from 9 to 5.

9

Routing Configuration

The Oracle Enterprise Communications Broker (Communications Broker) performs session routing via its route configuration. Route configuration establishes hop-by-hop paths to signaling endpoints.

End stations may or may not be known by the Communications Broker; endpoints configured within the user database are known, all others are not. Whether or not they are known by the system, the last hop (agent) leading towards that end station is often known. For this reason, the Communications Broker builds its hop list by starting with the last agent it knows in the path and recursively adding hops (agents) needed to get to that hop. In the case where the last hop is not known, the system provides its last known hop with endpoint information and allows unknown hops to try to find the endpoint.

Communications Broker routing configuration allows the user to specify a route's cost to specify route preference. Cost may or may not be based on monetary considerations. But the reach of an enterprise's network often does allow the user to configure routes that keep session traffic within the enterprise infrastructure rather than incurring cost associated with a service provider.

The Communications Broker allows for a range of route preference criteria to differentiate between routing paths. Criteria includes source routing based on the agent or calling number. Target-oriented criteria is also available, allowing the enterprise to designate preferred paths for specific called numbers.

Routing Fields

The **Routing Table** configuration displays the route list.

To add routes, click **Add**. The Oracle Enterprise Communications Broker displays the **Routing Table** configuration.

> **Note:**
>
> The routing engine finds the optimum route based on the Source and Destination Agents. Oracle recommends avoiding wild cards in both Source and Destination Agents. When these values are configured as wild cards, the Communications Broker must check the entire routing table for a match and this negatively impacts performance.

The following list describes the fields available in the **Add** dialog and the corresponding configuration instructions.

Source Agent	Select the agent from which the traffic must come to match the route. The default is the wildcard *, meaning to match traffic from any agent.
Calling Number	Type in the number from which the traffic must come to match the route. A valid entry includes numeric characters or an FQDN resolvable through ENUM.

	Regular expressions and special character such as "~", “!”, “*”, “x”, "(x)", “[x]”. The default is the wildcard *, meaning to match traffic with any calling number.
Dest Agent	Select the agent to which the traffic must be targeted to match the route. The default is the wildcard *, meaning to match traffic to any agent.
Called Number	Type in the number to which the traffic is targeted to match the route. A valid entry includes numeric characters or an FQDN resolvable through ENUM. Regular expressions and special character such as "~", “!”, “*”, “x”, "(x)", “[x]”. The default is the wildcard *, meaning to match traffic with any called number.
Route	Select the agent that is the next hop in the route path.
Cost	Enter a cost associated with this route to specify route use preference when there are multiple routes to the same destination. A valid entry is numeric ranging from 0 to 100, with the lowest number (cost) being the preferred route.
Description	Enter any descriptive text you find helpful in identifying this route.
Tags	Enter one or more tags.
Policy	Policies that can alter processing of routes to a destination involving this hop.

Route Policy

The Oracle Enterprise Communications Broker (Communications Broker) includes a route configuration control called **Policy** that allows you to alter the behavior of routes. As the Communications Broker assembles hops together to create a complete route, it considers and acts on the policies configured on each route entry.

You configure a route **Policy** from the Policy Entries entry dialog. The parameter values include:

Policy [Deny | StopRecurse]

The default setting is empty, which imposes no policy on the route.

The **Deny** policy in any resultant SIP routes to a destination prevents the Communications Broker from forwarding a SIP request to the destination point in question. You can use the **Deny** policy to prevent sessions between two endpoints for policy or cost reasons. Consider this route entry.

Source Agent	Calling Number	Destination Agent	Called Number	Route	Cost	Policy
Class_B	*	Protect_Agnt	*	Transit_Agnt	0	Deny

The preceding route entry is designed to prevent calls passing through the agent named `Class_B` from reaching any endpoint behind `Protect_Agnt`. This route entry may be installed as part of one or multiple routes that could potentially reach `Protect_Agnt`. The routing engine recognizes the presence of this entry and rejects the call.

To expand upon the example, consider this route.

Source Agent	Calling Number	Destination Agent	Called Number	Route	Cost	Policy
Class_A	*	Protect_Agnt	*	Transit_Agnt	0	StopRecurse

The preceding route configuration allows traffic to reach `Protect_Agnt` as long as it originates througha the agent named `Class_A`.

The **StopRecurse** policy stops further attempts to forward a SIP request to a given destination if that route stops responding. The **StopRecurse** policy can prevent the system from forwarding calls on routes that are cost-prohibitive or might cause loops in SIP call flows.

For example, consider the scenario where an endpoint can be reached using two routes.

- Route 1: Agent_1 to Agent_2 (**StopRecurse**) to PBX
- Route 2: Agent_3 to PSTN

Route 1 uses a **StopRecurse** policy defined on the hop between Agent_2 and PBX. The Communications Broker stops processing the call if Route 1 does not receive a successful response. This configuration can prevent calls initially targeted for Route 1 from using the PSTN.

Deny Patterns in Route Parameter Syntax

The Oracle Enterprise Communications Broker (Communications Broker) allows you to configure routes that refine the routing engine's behavior, based on the called and calling number. Such configuration prevents the routing engine from using that route entry as a member of the applicable route sets.

You configure this behavior by creating a route that prepends the called and calling numbers with an exclamation point (!). Configure the digits following the exclamation point as the first digits in the string, not the entire string. The following table shows an example of a route entry using a deny pattern.

Source Agent	Calling Number	Destination Agent	Called Number	Route	Cost	Policy
Agent_1	!123	*	*	Agent_2	0	

In the preceding route example, traffic that includes a calling number starting with the digits 123 does not match this route. The Communications Broker does not use this route in any applicable route set.

You can also configure both calling and called numbers with the deny format to establish an "and" condition to the route.

Source Agent	Calling Number	Destination Agent	Called Number	Route	Cost	Policy
Agent_1	!123	*	!456	Agent_2	0	

In the preceding route example, traffic that includes a calling number starting with the digits 123 and a called number starting with the digits 456 does not match this route.

Conversely, the "and" condition allows the following traffic originating from `Agent_1` by way of `Agent_2` to match this route:

- From any calling number that does not start with the digits 123 to any called number.
- From any calling number that starts with the digits 123 to any called number other than one that starts with 456.
- To any called number that does not start with the digits 456.
- To any called number that starts with the digits 456 from any called number other than one that starts with 123.

Loop Sensing for PSTN Calls

When the Oracle Enterprise Communications Broker (Communications Broker) routing table contains no explicit route to the agent, the Communications Broker routing engine uses an implicit route that assumes the Communications Broker is directly available to the agent. Depending on your routing needs, you might choose to configure a default route to handle calls not routed by the implicit routes. For some calls, the default route will result in a loop back to the previous hop. Such looping can cause undesirable traffic in the network and result in calls that never reach the end point. To prevent call looping, use the **next-hop-policy-condition** parameter to construct a policy for the default route that prevents the Communications Broker from sending a call back to the previous hop.

Using the **next-hop-policy-condition** in a policy, in combination with a skip action, removes the default route as a choice for some calls and prevents the Communications Broker from forwarding a call:

- back to the agent that sent the call.
- to a group of agents when the agent that sent the call is a member of the group.
- back to the agent when the IP address matches the host in the from-uri of the received message.
- to a group of agents when any agent in the group uses an IP address and port that matches the host in the from-uri of the received message.

The **next-hop-policy-condition** compares the next hop in the route to either the **previous hop** or the **from-uri-host**. When either **previous hop** or **from-uri-host** resolves to true, the Communications Broker executes the **deny** action to stop call looping.

The process for configuring call loop prevention requires the following steps:

1. Set the **next-hop-policy-condition** in a policy, along with the comparison type and the deny action.
2. Add the policy to the target route.

Configure Loop Sensing for PSTN Calls

To prevent call looping, use the **next-hop-policy-condition** parameter to construct a policy for the default route that prevents the Oracle Enterprise Communications Broker (Communications Broker) from sending a call back to the previous hop when the agent does not respond.

1. Access the Policy Entries configuration object. **Configuration** tab, **Service Provisioning** section, **Policy Entries**.
2. On the Policy entries page, click **Add**, and do the following:

Name	Enter a name for this policy.
Description	(Optional) Enter a description of the policy.

Conditions	Click **Add**, select **next-hop-compare-condition**, and do the following: a. Enter a name for this condition. b. Select the next hop compare mode that you want from the drop-down list. c. Click **OK**.
Actions	Click **Add**, and do the following: a. Select **routing-action** from the drop down list. b. Enter a name for this routing action. c. Select **deny** from the drop-down list. d. Click **OK**.

3. Click **OK**.
4. Save the configuration.

- Apply the policy to the target route.

10

Registrar Configuration

When enabled, the Oracle Enterprise Communications Broker's registrar provides location service and registration authentication functions. The user must decide whether to use authentication and, if so, which authentication resource to utilize. Should the user decide to use a local authentication resource, they configure it via Local Subscriber Tables (LST), available from the registrar configuration. Configure external authentication resource configuration via the LDAP configuration dialogs.

Registrar Configuration Fields

To configure the Oracle Enterprise Communications Broker to act as a SIP Registrar:

1. Access the SIP Registrar configuration object.

   Configuration tab, **System Administration** section, **SIP Registrar**.

2. On the Add SIP Registrar page, do the following:

Name	Name of the SIP Registrar
State	Select to use this SIP registrar configuration element.
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.
Minimum Register Expiration	Enter the expire time in seconds to be used in the REGISTER. Default: 300 (5 minutes). Valid values: 0-999999999
Authentication Profile	Set the method for the registrar to use to authenticate incoming registrations.
Additional Target Group	List additional targets for the SIP registrar.
Fork Group	Select a number from 1-100 for the fork group registrar endpoints.
Policy	Enter one or more policies that you want to use to alter the processing of routes to a destination with registered endpoints.

3. Save the configuration.

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 Enterprise Communications Broker’s file system.

LSTs enable a standalone Oracle Enterprise Communications Broker node or high-availability (HA) pair to forego relying on an external user database. Thus the Oracle Enterprise Communications Broker 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 Configuration

To configure the Oracle Enterprise Communications Broker to use LSTs for authentication, you need to create a local subscriber table configuration element that identifies that LST. The LST must include users with minimum configuration of user name and password. Alternatively, an LST entry can include an AOR and a universal number. If there is no AOR, the username is assumed to be the AOR. The universal number field assigns a universal number to all contacts registered to the AOR.

You have the option of setting the registrar to authenticate. When messages requiring authentication are received and processed by the sip registrar, the Oracle Enterprise Communications Broker uses the identified LST for authentication.

In a local subscriber table configuration, you must define an object **name**. The Oracle Enterprise Communications Broker stores LST files in the /code/lst directory. Do not specify a path in the **name** field.

When the registrar configuration includes a reference to an LST, the registrar uses it as its user list. The configuration may or may not include digest authentication functionality, depending on user configuration. Additional registrar configuration includes setting the **digest realm** appropriately (this is required for authentication), and setting the hash secret. 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)
```

Configure the Registrar with an LST

Define the registrar for using the Local Subscriber Table (LST) for registration authentication as opposed to an external resource, or to accept registrations without authentication.

1. Access the SIP Registrar configuration object.

   Configuration tab, **System Administration** section, **SIP Registrar**.

2. On the **SIP Registrar** page, click **Add** to add the SIP Registrar:

Field	Description
LST File	Specify the LST file for this registrar. Choose an existing LST file from the drop-down list.

Field	Description
Manage LST (Use to create an LST file.)	Do the following: a. File Name—Enter a filename for your new LST XML file. The Communications Broker stores these files in the /code/lst directory. b. Digest Realm—Enter the name (Realm ID) of the host realm initiating the authentication challenge. This value defines the protected space in which the digest authentication is performed. Valid value is an alpha-numeric character string. c. Encrypt File—Select o cause the system to encrypt the file. d. Encryption Secret—Click **Set** to display the Set Encryption secret dialog. Enter and then confirm the secret used in encryption and decryption of the passwords in the XML file. Once saved, the system does not echo this value back to the screen in plaintext format. e. Click **OK**—This creates your LST file and allows you to add subscriber entries. f. Click the **Add** button. The system displays the Add Local Subscriber Entry dialog. Enter Username, Password and AoR for this subscriber.
Authentication Method	Select LST from the drop-down list.
Digest Realm	Enter the name (realm ID) of the host realm initiating the authentication challenge. This value defines the protected space in which the digest authentication is performed. Valid value is an alpha-numeric character string.
LST Hash Secret	Click the **Set** button to display the Set LST hash secret dialog. Enter and then confirm the secret used in encryption and decryption of the LST.

3. Save the configuration.

Add an LST File

Add local subscriber table file:

1. Click **Configuration**, **SIP Registrar** and then click **LST**.
2. In the Local Subscriber Table page, click the **Add**. The system displays the **Add Local Subscriber Table** page.
3. **Name**— Name of the local subscriber table.

   The value given in the username attribute must be the same as the username that will be sent in the Authorization Header in the Request message from the users. Refer to RFC 2617 Http Authentication for details.
4. **Filename**—Select a filename for your new LST XML file. The Communications Broker stores these files in the /code/lst directory.

5. Click **Manage LST** to display the Add or Edit Local subscriber table dialog, from which you can add, change, copy and delete users from the LST. In the Add Local Subscriber Table (LSTfilename.xml) page:.

   - **Username**— Enter a username for this subscriber. Optional configuration includes, password, universal number and AoR

   a. **Password**—Enter the password associated with the username of the client. This is required for all LOGIN attempts. The password displays while typing but is not saved in clear-text (i.e., *****). Valid value is an alphanumeric character string. Click **Show Password** to display the password.

   b. **Address of Record**—The Address of Record attribute is optional to specify the address of record for the subscriber if it is different than the username.

   c. **Universal number**—The user's number in a format compatible for use within the routing table.

6. Repeat the subscriber add process for as many subscribers as intended.
7. Save and activate your changes when finished.

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 Enterprise Communications Broker 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 GUI.
3. A 503 Response is returned to the UA that sent the initial REGISTER message to the Oracle Enterprise Communications Broker.

LST Redundancy for HA Systems

The Oracle Enterprise Communications Broker synchronizes LSTs between redundant nodes to ensure that the standby node contains identical LST files. This process occurs automatically when the user uploads an LST via the GUI and when the user saves a change to an LST file via the GUI.

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' standalone='yes'?>
<LocalSubscriberTable>
    <realm>aaa</realm>
```

```
    <encryption>disabled</encryption>

<secret>02:4B:20:99:60:D2:73:4A:7B:66:B0:62:AC:8D:B5:7D:67:5F:4B:5B:47:F2:2E:5
0:B5</secret>
    <subscriber username="alice" aor="alice@company.com"
universalNum="1231231234" hash="d9bfe1cac8e7fe6b79da42d03b03b96b"/>
    <subscriber username="bob" aor="bob@company.com"
universalNum="1231232345" hash="af586127536d20f4c6e88a2921780b18"/>
    <subscriber username="carol" aor="carol@vendor.com"
universalNum="1231233456" hash="b695bc18bef48e2141555e7736bd88ec"/>
</LocalSubscriberTable>
```

The LST file's elements are explained below.

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:

- realm—Specifies the name of digest realm.
- encryption—This indicates whether or not the hash in the XML file is encrypted (MD5). The key for this encryption will be a preshared key and is configurable in the local subscriber table configuration element with the secret parameter.
- secret—Included if encryption is used, this is the encrypted secret.

subscriber

This element has the subscriber information. And has the following 5 attributes:

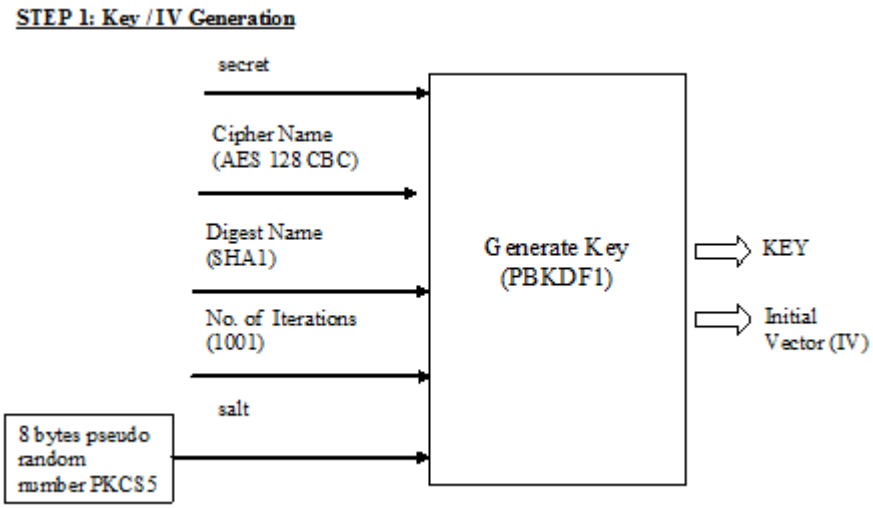
- username—The value given in the username attribute must be same as the username that will be sent in the Authorization header in the request message from the users. Refer RFC 2617 Http Authentication for details.
- aor—The aor attribute is optional to specify the address of record for the subscriber if it is different than the username.
- universalNum—The universalNum attribute is optional to specify the universal number (E.164) for the subscriber.
- 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)

LST Subscriber Hash and Encryption

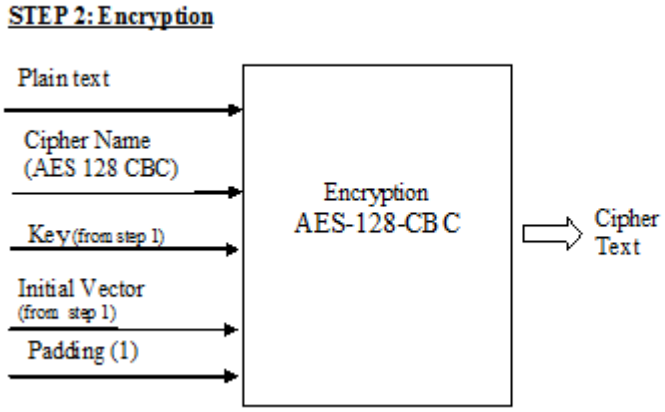
You can use AES-128 CBC to encrypt the hash in the subscriber element in the LST XML file. The PSK used for encryption is configured in the **secret** parameter and an 8-byte pseudo random number is used as the salt.

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.

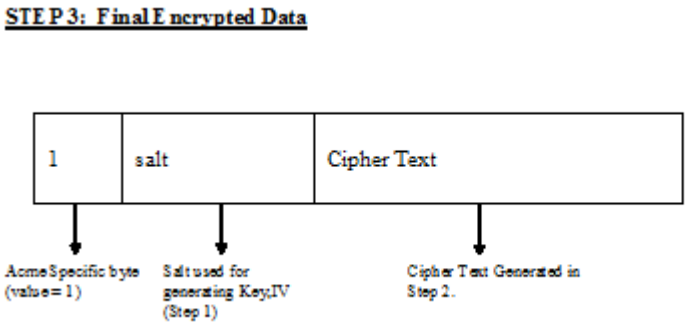
Key Initialization Vector



Encryption



Formatting Final Encryption



11

LDAP Client Configuration

The Oracle Enterprise Communications Broker (Communications Broker) supports LDAP as a communications mechanism for interaction with an LDAP server. For many enterprises, this means utilizing Active Directory, a common LDAP-based service, to request information used in SIP session routing and authentication. The Communications Broker's LDAP client requires configuration on the Communications Broker and the LDAP server.

Configuration aspects of LDAP client configuration include:

- LDAP server access—The user specifies LDAP server location and access preferences.
- Routing queries—The user specifies the conditions wherein the Communications Broker performs an LDAP dip to obtain location information (home agent) for FROM and REQUEST-URIs.
- AoR queries—Optionally searches for additional AoR matches in Active Directory so that it can create additional routes to target users that have contacts stored in separate records.
- SIP Authentication queries—As an optional registration authentication mechanism, LDAP client configuration can utilize domain authentication or customized authentication server configuration on the LDAP server, as follows:
  - The use of domain authentication requires an application be installed on the domain controller.
  - Customized authentication requires the specification of compatible authentication fields on both client and server.

> **Note:**
>
> The user must ensure that phone numbers in the LDAP database are unique. If the Communications Broker encounters multiple records with the same number, the lookup fails.

LDAP Configuration Options

The Oracle Enterprise Communications Broker provides options for LDAP configuration. These options provide the user with the flexibility to implement lookup precedence and preference on a per-agent and global basis.

An overall LDAP configuration on the Oracle Enterprise Communications Broker can include the following types of configurations.

- LDAP Config—A configuration that reaches one or multiple LDAP servers that refer to the same search base, use the same credentials, and typically service the same domain. The user can apply these configuration to agents.
- Global Config—A special LDAP configuration that the system uses as a default LDAP Config. When enabled, the system uses this LDAP Config for agents that are not configured with an LDAP Config or LDAP Group. The user cannot rename or delete the Global Config.

- LDAP Group—Multiple LDAP configs grouped together that allow the user to refine LDAP lookups across disparate LDAP domains.

The Oracle Enterprise Communications Broker's operational precedence for using your LDAP configuration on a per call basis is as follows:

1. If the applicable agent has an LDAP group applied, calls from that agent use the group configuration.
2. If the applicable agent does not have an LDAP group configuration, but does have an LDAP Config applied, calls from that agent use the LDAP Config.
3. If the call arrives on an agent without any LDAP configuration, calls from that agent use the Global LDAP configuration.

The Global LDAP configuration is disabled by default. This, in conjunction with the absence of any LDAP Configs, excludes the use of LDAP.

Making Settings

The Oracle Enterprise Communications Broker provides access to configuration fields via lists on the GUI.

When the user clicks the LDAP icon, the system displays a navigation pane on the left side of the screen, which includes the following links:

- LDAP
- Groups

When the LDAP link is selected, they system displays the list of currently configured LDAP Configs, including the Global Config. See the *Oracle® Enterprise Communications Broker User's Guide*, Release P-CZ2.0.0 for instructions on configuring LDAP Config fields.

Clicking the Groups link displays the LDAP Groups list, from which the user configures LDAP Groups.

Configure LDAP Server Access

Use the following procedure to configure Communications Broker to access one or more LDAP servers.

Points to be noted:

Table 11-1 Configuring LDAP Servers

Options in Configuring LDAP Servers	Follow-up Action by Communications Broker
Configure LDAP server with the port along with the FQDN name	Communications Broker: • Considers the configuration as an A-Record FQDN. • Sends out a DNS query when you save a configuration with FQDN as LDAP server. • Resolves the FQDN into an IP address using the media interface on saving.

Table 11-1 (Cont.) Configuring LDAP Servers

Options in Configuring LDAP Servers	Follow-up Action by Communications Broker
No port information provided along with server name.	Communications Broker considers the configuration as an SRV- Record. • Sends out a DNS query when you save a configuration with FQDN as LDAP server. • Resolves the FQDN into an IP address using the media interface on saving. • For SRV records FQDN, IP addresses are retrieved in the order of weights & priorities.
Configure LDAP server with an IP address without any port	Communications Broker assigns the default port.

1. Access the LDAP Configuration object.

   Configuration tab, **System Administration** section, **LDAP**, **LDAP Config**.

2. On the LDAP config page, click the **Add** button and do the following:

Name	Description
Name	Name of LDAP Config element
State	Select to enable the LDAP configuration.
LDAP Servers	Add the following to define the LDAP configuration: • Enter either an IP address(es), OR FQDNs. Only a single FQDN is supported. • Enter the IP addresses (IPv4) in the dotted decimal format (0.0.0.0). The order of the resolved IP addresses is the actual order of preference. **Note**: A combination of IP addresses and FQDN is not supported. • Optionally, add the port numbers for each LDAP Server (such as port 389, 636, 1025- 65535. Communications Broker performs an SRV query when no port is configured, and A query when a port is configured. • The first IP addresses listed is considered as the primary LDAP Server, and the remaining servers are considered the backup or secondary LDAP Servers. • Round-Robin strategy is used to determine the active LDAP Server. This is applicable to both static IP addresses and the resolved A - record FQDN IP addresses. SRV records, round-robin is not applied since it is based on weights and priority. The default strategy is Hunt strategy. • The default ports: 389 (for LDAP over TCP) and 636 (LDAP over TLS).
ldap-load-balance	Load balance A records or static IP addresses. SRV records support only Hunt strategy. • Default value: hunt • Available values: hunt or round-robin Note: This is required only when the LDAP server is configured as a FQDN.

Name	Description
Realm	Enter the name of the realm to receive requests on. Default: ecb. Communications Broker uses the Realm configured under the LDAP configuration object for DNS resolution. The selected Realm must have a link to the network-interface with DNS parameters configured. The same must be attached to physical interface.
Username	Enter the user name that the LDAP bind request uses for authentication before access is granted to the LDAP Server. Valid values are alpha-numeric characters. Default: blank.
Password	Enter the password to pair with the username attribute, that the LDAP bind request uses for authentication before access is granted to the LDAP Server. Valid values are alpha-numeric characters. Default: blank.
LDAP SearchBbase	Enter the base Directory Number you can use for LDAP search requests. Valid values are alpha-numeric characters. Default is blank.
Timeout Limit	Enter the maximum amount of time, in seconds, for which the Communications Broker waits for LDAP requests from the LDAP server before timing out. When an LDAP response is not received from the LDAP server within the time specified, the request is retried again based on the max-request-timeouts parameter value. Default: 15. Valid values:1 to 300 seconds.
Max Request Timeouts	Enter the maximum number of times that the LDAP Server is sent LDAP requests before the Communications Broker determines that the server is unreachable and terminates the TCP/TLS connection. When an LDAP response is not received within the time specified for the timeout-limit parameter value, the request is retried the number of times specified for this max-request-timeouts value. Default: 3. Valid values: 0-10.
TCP Keepalive	Specify whether or not the Communications Broker keeps the TCP connection to the LPAD Server alive. Default: Disabled. Valid values: Enabled | Disabled.
LDAP Sec Type	Select the LDAP security type to use when the Communications Broker accesses the LDAP server. This parameter enables the use of LDAP over TLS (LDAPS). If you set a value for this parameter, you must also specify an LDA TLS Profile value. Default: none Valid values: none (No LDAP security type specified.) | LDAPS (Method of securing LDAP communication using an SSL tunnel. This is denoted in LDAP URLs. The default port for LDAP over SSL is 636.)
Routing	• State—Select to enable routing. • Route Mode—Select how you want the Communications Broker to order routes. Valid values: match-only | match-first | attribute-order. • From Header Replacement—Enter any text you want replaced in the from header. • Lookup Queries—Click **Add**, set the values for lookup, and click **OK**. • Operation Type: LDAP Operation type. The default value is or. <and, or> • The LDAP Servers can have the msRTCSIP and msRTCSIP-OptionFlags.

Name	Description
Address of Record	• Lookup Number Attribute—Enter the name of the attribute to query. Default: sAMAccountName. • Lookup Number Format Type—Select a type of translation to apply to the number before the query. Default: None: Valid values: None-use the called number as-is. | E164-+14445551234 | E164-No-Plus-14445551234 | No Country code-4445551234 | Pattern Only-use a portion of a matching dial plan | Regular Expression-apply a regular expression. • Lookup Number regex pattern—Enter an expression. • Lookup Number regex result—Enter the format to create a telephone number or query from values captured in a regular expression. • AoR Attribute—Enter the name of an address of record attribute to return from the directory. • AoR Extraction Regex—Enter a regular expression to parse the address of record returned from the directory. • AoR Value Format—Enter the format to create the address of record from values captured in a regular expression.
SIP Authentication	• Username Attribute—Set the name of the attribute to query. Default: sAMAccountName. • Digest Has Attribute—Enter the name of the hash attribute to return from the directory. Default: orclDigestPwdAttribute.
TLS Profile	Select the name of the Transport Layer Security (TLS) profile that the Communications Broker uses when connecting to the LDAP Server. The ldap-sec-type must be set to **LDAPS** for this profile to apply. Valid values are alpha-numeric characters. Default is blank. See the *Oracle Enterprise Communications Broker Administrator's Guide* for instructions on how to create a TLS profile.

3. Click **Back**.
4. Save and activate the configuration.

LDAP Groups

LDAP groups on the Oracle Enterprise Communications Broker group **LDAP configs** together, allowing the user to refine lookups to multiple LDAP servers. The user configures **LDAP groups**, defines the matching criteria by which the system selects servers to query, and applies LDAP groups as profiles to agents.

When the system determines that it may find information it needs in the LDAP database, it checks to see if there is an **LDAP group** configured on the applicable agent. If there is no group, the system uses the LDAP configuration to control its lookups. This configuration can include a single LDAP Config configured on the agent or a Global LDAP config. If there is a group, the system:

1. Checks the matching criteria in the group to identify relevant LDAP servers, and
2. Performs lookups to relevant servers `using the order that the administrator has configured in the group.

If there is no match with any of the group's servers, the system does not perform an LDAP lookup and proceeds with the process sequence is uses to find information.

Matching Criteria in LDAP Groups

The Oracle Enterprise Communications Broker (Communications Broker) uses configurable **Matching Criteria** to determine whether or not to perform a lookup to each server in an **LDAP Group**. The Communications Broker supports regex expressions within matching criteria configuration.

The system evaluates matching criteria for all LDAP servers listed in the group. If there are no matches, the system proceeds without querying LDAP. If there are any matches, the system initiates lookups to servers in the order listed in the group. If there is no match, the system skips those servers.

Consider the example where the system performs a lookup for a phone number in LDAPGroup1. The **LDAP Group** selection is based on the agent configuration of the applicable end station. In this example, the LDAP1 agent configuration specifies LDAPGroup1 as its **LDAP group**.

In this example, the Administrator configured the first lookups within LDAPGroup1 to be directed to LDAP1. If the number matches a criteria entry, the system adds that server to the lookup list. If not, the system skips to the next servers in the group, evaluating their **Matching Criteria**.

Matching criteria for LDAPGroup1 could include the following entries.

Matching Criteria	LDAP Server
^\+1.*$	LDAP1
.*@Div1.com$	LDAP1
.*	LDAP2
^\+44.*$	LDAP3
^\+34555.*$	LDAP3

For LdapGroup1, the system queries LDAP1 only if the number starts with a +1 or has a host of Div1.com. The system queries LDAP2 in all cases based on its wildcard criteria. The system queries LDAP3 if the called number has a UK area code or has a Spain area code, then starts with 555.

Configure LDAP Groups

Configure LDAP before configuring LDAP groups.

Use the following procedure to create a new LDAP group:

1. Access the LDAP Configuration object.

   Configuration tab, **System Administration** section, **LDAP**, **LDAP Group**.

2. Click the **Add** button.
3. On the Add LDAP Group page, do the following:

Group Name	Enter a name for this group. The name must be between 1 and 128 alpha-numeric characters without spaces. The name can include the underscore, comma, period and dash characters, bot not as the first characters in the name.
Description	(Optional) Enter a description of this LDAP group.

State	Select to enable this LDAP group.
LDAP Agents	Click **Add** and do the following: a. Matching Criteria—Define the criteria the system must use to determine whether it should perform a lookup in the associated server for end station information. Regex is supported as a means of configuring matching strings. b. LDAP Config—Select the LDAP configuration that you want to use. c. You can add only the non-global LDAP Configurations to the group. Default “global” configuration can support operation-Type with single stage validation.

4. Click **OK**.

   The system displays the LDAP group configuration page, where can optionally add more agents.

5. After you add the necessary agents, click **OK**.

6. Save the configuration.

Apply your LDAP group to the applicable Agents from the **LDAP** drop-down list under the Agent controls.

Configure a Routing Query

To configure the Oracle Enterprise Communications Broker (Communications Broker) to query an LDAP database for the purpose of obtaining a call's routing information, per the Communications Broker processing sequence:

1. Access the LDAP Configuration object.

   Configuration tab, **System Administration** section, **LDAP**, **LDAP Config**.

2. On the LDAP Config page, expand **Routing** to expose the Routing parameters, and do the following:

State	Select to enable the use of routing queries for your configured LDAP servers.
Route Mode	Specify the route priority that the Communications Broker uses in the route list that determines which routes to create, and the priority of the routes within the route list. Default: Match-only. Valid values: Match-Only | Attribute Order | Match First. • Match-Only—When an exact match between the dialed telephone number and an LDAP attribute value in the search response entry occurs, a route is created corresponding to that LDAP attribute. When an exact match on multiple attributes occurs, the ordering of LDAP attributes in the LDAP configuration determines the priority for each route. For example, in an enterprise that uses the same phone number for both Teams and IPPBX phones, when the msRTCSIP-Line attribute is configured first, the corresponding next hop (Teams Server) is used to create the first route in the route list.

	• Attribute Order— The ordering of LDAP attributes in the LDAP configuration determines the priority for each route. When the msRTCSIP-Line attribute is configured first, the corresponding next hop (Teams Server) is used to create the first route in the route list. If there is a valid value present in the search response entry for a LDAP attribute, a route is created corresponding to that LDAP attribute. • Match First— If there is an exact match between the dialed telephone number and an LDAP attribute value in the search response entry, the corresponding route gets the highest priority in the route list. For the rest of the routes, the ordering of LDAP attributes in the LDAP configuration determines the priority for each route. So if the msRTCSIP-Line attribute is configured first, the corresponding next hop (Teams Server) would be used to create the second highest priority route in the route list. If there is a valid value present in the search response entry for an LDAP attribute, a route is created corresponding to that LDAP attribute. **Note:** The LDAP attribute must have a valid value in the response; a match is not necessary for that attribute. If an entry is returned in the search response, there must be a match on at least one other attribute. For example, the dialed telephone number could be +17813284392 (IP-PBX Phone#), and the msRTCSIP-Line in the response could be +17814307069 (Teams phone#). A route is created for the Teams phone#, even though the dialed telephone number is the PBX Phone#.
From Header Replacement	Enter the name of the LDAP lookup query to use as the From Header replacement.
Operation Type	Configure the Operation Type. By default “or” is selected. You can select "or" or "and" as the global condition for grouping of ldap-lookup query attribute elements. This allows you to change the logic of an LDAP query from "or" to "and", and also the ability to recurse through multiple LDAP look-up queries. The default “global” LDAP configuration can support Operation Type with single stage validation.

3. Click **Add** in the Lookup Query dialog, and do the following:

Each element identifies a lookup number attribute and dialed pattern with which the Communications Broker finds matches in the LDAP database and identifies contacts to which it builds routes. Multiple matches result in multiple targets to which the Communications Broker creates routes for call forking.

Lookup Number Attribute	Enter the Active Directory attribute name. Default: Telephone Number. Valid values: Alpha-numeric characters. Some examples of Active Directory attribute names include: • ipPhone and msRTCSIP-Line for Teams phone number

	• telephoneNumber for IP PBX phone number • mobile for Mobile phone number
Lookup Number Format Type	Select the expected attribute format from the drop down list. Default: None. Valid values: E164 | E614 No Plus | No Country Code | None | Pattern Only | Regular Expression.
Lookup Number Regex Pattern	Enter the regular expression pattern used to break down the string of digits in the phone number extracted from the request URI and the FROM of the SIP request. The variables extracted from the phone number can be used in the attribute-value-format parameter. Default regex: "^\+?1?(\d{2})(\d{3})(\d{4})$" Valid values: Alpha-numeric characters. (The default value assumes that the phone number is a North American phone number specified in the E.164 format.) It extracts the following variables from the phone number: $1 is the area code. | $2 and $3 are the next 3 and 4 digits in the phone number. Note that the system queries only for the home agent of the FROM if it has not already found it. The setting applies only when you set **Lookup number format type** to **regular-expression**.
Lookup Number Regex Result	**Lookup number regex result**—Enter the format for the attribute value. These format values are extracted from the phone number using the extraction-regex parameter. Default: "tel:+1$1$2$3". This value assumes that the phone number is a North American phone number specified in the E.164 format, and it recreates the phone number in E.164 format. Valid values: Alpha-numeric characters. In addition to the E.164 format, the Communications Broker uses other formats as well to store the phone numbers. You can customize the value specified for this parameter to enable successful queries for phone numbers in other formats. The setting applies only when you set **Lookup number format type** to **regular-expression**.
Home Agent Attribute	Enter the Active Directory attribute name for the agent field. Default: Blank. Valid values: Alpha-numeric characters. If created with the Oracle tools described in this document, the name would be orclAgentNameAttribute.
Home Agent Regex Pattern	Enter the regular expression pattern used to break down the agent name. Default: Blank.
Home Agent Regex Result	Enter the format of the regex result. Default: Blank.
Default Home Agent	Enter the name of the home agent to use for routing if the query does not return one.

4. Click **OK** to save the routing query.
5. Save the configuration.

Address of Record Configuration Fields

You can configure the Oracle Enterprise Communications Broker (Communications Broker) to query an LDAP database for the purpose of identifying additional Addresses of Record (AoR) that may apply to a call. The system identifies additional contacts from the AoRs and creates additional routes with which it can fork the calls.

1. Access the LDAP Configuration object.

   Configuration tab, **System Administration** section, **LDAP**, **LDAP Config**.

2. On the LDAP Config page, do the following:

Lookup Number Attribute	Enter the Active Directory attribute name. Default: sAMAccountName, which is the standard Active Directory username attribute. Valid values: Alpha-numeric characters.
Lookup Number Format	Select the expected phone number format from the drop down list. Default: None. Options include: E164 | E164 No Plus | No Country Code | None | Pattern Only | Regular Expression.
Lookup Number Regex Pattern	Enter the regular expression pattern used to break down the string of digits in the phone number extracted from the request URI of the SIP request. The variables extracted from the phone number can be used in the attribute-value-format parameter. Valid values: Alpha-numeric characters. Default: "^\+?1?(\d{2})(\d{3})(\d{4})$". (This value assumes that the phone number is a North American phone number specified in the E.164 format.) It extracts the following variables from the phone number: $1 is the area code | $2 and $3 are the next 3 and 4 digits in the phone number. Note that this setting applies only when you set **Lookup number format type** to **regular-expression**.
Lookup Number Regex Result	Enter the format for the attribute value. These format values are extracted from the phone number using the extraction-regex parameter. Default: "tel: +1$1$2$3". This value assumes that the phone number is a North American phone number specified in the E.164 format, and it recreates the phone number in E.164 format. Valid values: Alpha-numeric characters. In addition to the E.164 format, the Communications Broker uses other formats as well to store the phone numbers. You can customize the value specified for this parameter to enable successful queries for phone numbers in other formats. The setting applies only when you set **Lookup number format type** to **regular-expression**. In addition to the E.164 format, the Communications Broker uses other formats as well to store the phone numbers. You can customize the value specified for this parameter to enable successful queries for phone numbers in other formats.
AoR Attribute	Enter the Active Directory attribute name established to contain the AoR. Default: Blank. Valid values: Alpha-numeric characters.
AoR Extraction Regex	Enter the regular expression pattern used to break down the AoR. Valid values: Alpha-numeric characters.
AoR Value Format	Enter the format of the regex result. Default: Blank.

3. Click **OK** to save the routing query.
4. Save configuration.

SIP Authentication Query Configuration Fields

To configure the Oracle Enterprise Communications Broker to specify an alternate authentication field in a remote database for the purpose of authenticating registration attempts:

1. Access the LDAP Configuration object.

   Configuration tab, **System Administration** section, **LDAP**, **LDAP Config**.

2. On the LDAP Config page, expand **SIP Authentication** and do the following:

Username Attribute	Enter the name of the attribute where the digest username is stored in your LDAP database. Default: sAMAccountName. (The standard Active Directory username attribute.)
Digest Hash Attribute	Enter the name of the attribute where the digest authentication hash is stored in your LDAP database. Default: orclDigestPwdAttribute. (A custom field populated by the oidpwdcn password filter.)

3. Click **OK** to save the configuration.

Replacing the Calling Number in the FROM Header

The Oracle Enterprise Communications Broker (Communications Broker) provides for replacement of the calling number in SIP messages' FROM headers. Applicable messages include INVITEs that match the query, and all messages sent by the Communications Broker to those calls callees. An example application is allowing recipient UEs to display a caller ID that is recognized by the recipient, even during an enterprise's transition to new dialing schemes.

This calling number replacement function refers to LDAP resources as the source of the replacement calling number. You configure a lookup query from the **Modify LDAP config** dialog to specify this source. Configured lookup queries become available in the **FROM header replacement** drop-down list, from which you selects their query. This selection specifies and enables the replacement.

This feature piggybacks normal LDAP lookup procedures by collecting an additional value within the LDAP query request and response sequence. The Oracle Enterprise Communications Broker replaces the FROM header of the outgoing message with this value.

While processing this LDAP response for calling number, the Communications Broker stores the result of the query and uses it to create the FROM header user parts for applicable outgoing messages. For traffic in which there is no match to the calling number, the Communications Broker simply uses the original calling number.

You can disable this replacement function by clearing the lookup query attribute name from the **FROM header replacement** field.

12

Denial of Service Protection and ACLs

This section explains the Denial of Service (DoS) protection for the Oracle Enterprise Communications Broker (Communications Broker). The Communications Broker DoS protection functionality protects softswitches and gateways with overload protection, dynamic and static access control, and trusted device classification and separation at Layers 3-5. The Communications Broker itself is protected from signaling and media overload, but more importantly the feature allows legitimate, trusted devices to continue receiving service even during an attack. DoS protection prevents the Communications Broker host processor from being overwhelmed by a targeted DoS attack from the following:

- IP packets from an untrusted source as defined by provisioned or dynamic ACLs
- IP packets for unsupported or disabled protocols
- Nonconforming/malformed (garbage) packets to signaling ports
- Volume-based attack (flood) of valid or invalid call requests, signaling messages, and so on.
- Overload of valid or invalid call requests from legitimate, trusted sources

Levels of DoS Protection

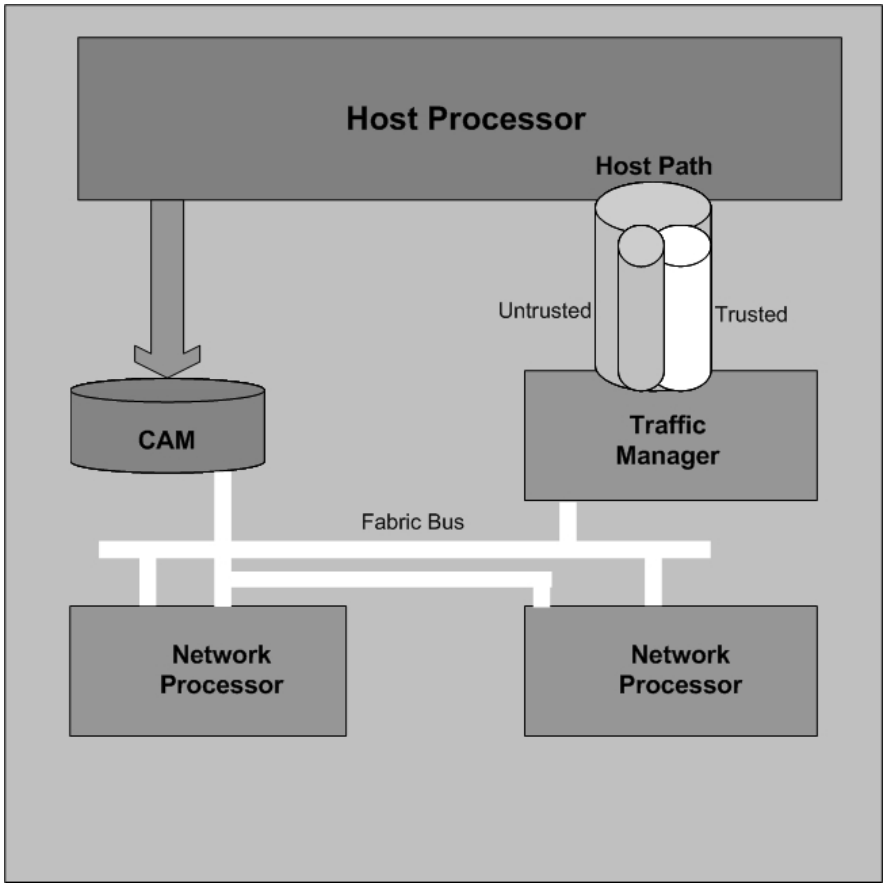
The multi-level Communications Broker DoS protection consists of the following strategies:

- Fast path filtering/access control—Access control for signaling packets destined for the Communications Broker host processor as well as media (RTP) packets. The Communications Broker performs media filtering by using the existing dynamic pinhole firewall capabilities. Fast path filtering packets destined for the host processor require the configuration and management of a trusted, untrusted and a deny list for each Communications Broker realm (although the actual devices can be dynamically trusted or denied by the Communications Broker based on configuration). You do not have to provision every endpoint/device on the Communications Broker, but instead retain the default values.
- Host path protection—Includes flow classification, host path policing and unique signaling flow policing. Fast path filtering alone cannot protect the Communications Broker host processor from being overwhelmed by a malicious attack from a trusted source. The host path and individual signaling flows must be policed to ensure that a volume-based attack will not overwhelm the Communications Broker's normal call processing; and subsequently not overwhelm systems beyond it.
  The Communications Broker must classify each source based on its ability to pass certain criteria that is signaling- and application-dependent—At first each source is considered untrusted with the possibility of being promoted to fully trusted. The Communications Broker maintains two host paths, one for each class of traffic (trusted and untrusted), with different policing characteristics to ensure that fully trusted traffic always gets precedence.
- Host-based malicious source detection and isolation/dynamic deny list—Malicious sources can be automatically detected in real-time and denied in the fast path to block them from reaching the host processor.

About the Process

DoS attacks are handled in the Communications Broker's host path. The Communications Broker uses NAT table entries to filter out undesirable IP addresses; creating a deny list. After a packet from an endpoint is accepted through NAT filtering, policing is implemented in the Traffic Manager subsystem based on the sender's IP address. NAT table entries distinguish signaling packets coming in from different sources for policing purposes.

The Traffic Manager has two pipes, trusted and untrusted, for the signaling path. Each signaling packet destined for the host CPU traverses one of these two pipes.



Trusted Path

Packets from trusted devices travel through the trusted pipe in their own individual queues. In the Trusted path, each trusted device flow has its own individual queue (or pipe). The Communications Broker can dynamically add device flows to the trusted list by promoting them from the Untrusted path based on behavior; or they can be statically provisioned.

Trusted traffic is put into its own queue and defined as a device flow based on the following:

- source IP address
- source UDP/TCP port number
- destination IP address

- destination UDP/TCP port (SIP interface to which it is sending)
- realm it belongs to, which inherits the Ethernet interface and VLAN it came in on

For example, SIP packets coming from 10.1.2.3 with UDP port 1234 to the Communications Broker SIP interface address 11.9.8.7 port 5060, on VLAN 3 of Ethernet interface 0:1, are in a separate Trusted queue and policed independently from SIP packets coming from 10.1.2.3 with UDP port 3456 to the same Communications Broker address, port and interface.

Data in this flow is policed according to the configured parameters for the specific device flow, if statically provisioned. Alternatively, the realm to which endpoints belong have a default policing value that every device flow will use. The defaults configured in the realm mean each device flow gets its own queue using the policing values. As shown in the previous example, if both device flows are from the same realm and the realm is configured to have an average rate limit of 10K bytes per second (10KBps), each device flow will have its own 10KBps queue. They are not aggregated into a 10KBps queue.

The individual flow queues and policing lets the Communications Broker provide each trusted device its own share of the signaling, separate the device's traffic from other trusted and untrusted traffic, and police its traffic so that it can't attack or overload the Communications Broker (therefore it is trusted, but not completely).

Address Resolution Protocol Flow

The Address Resolution Protocol (ARP) packets are given their own trusted flow with the bandwidth limitation of 8 Kbps. ARP packets are able to flow smoothly, even when a DoS attack is occurring.

Untrusted Path

Packets (fragmented and unfragmented) that are not part of the trusted or denied list travel through the untrusted pipe. In the untrusted path, traffic from each user/device goes into one of 2048 queues with other untrusted traffic. Packets from a single device flow always use the same queue of the 2048 untrusted queues, and 1/2048th of the untrusted population also uses that same queue. To prevent one untrusted endpoint from using all the pipe's bandwidth, the 2048 flows defined within the path are scheduled in a fair-access method. As soon as the Communications Broker decides the device flow is legitimate, it will promote it to its own trusted queue.

All 2048 untrusted queues have dynamic sizing ability, which allows one untrusted queue to grow in size, as long as other untrusted queues are not being used proportionally as much. This dynamic queue sizing allows one queue to use more than average when it is available. For example, in the case where one device flow represents a PBX or some other larger volume device. If the overall amount of untrusted packets grows too large, the queue sizes rebalance, so that a flood attack or DoS attack does not create excessive delay for other untrusted devices.

In the usual attack situations, the signaling processor detects the attack and dynamically demotes the device to denied in the hardware by adding it to the deny ACL list. Even if the Communications Broker does not detect an attack, the untrusted path gets serviced by the signaling processor in a fair access mechanism. An attack by an untrusted device will only impact 1/1000th of the overall population of untrusted devices, in the worst case. Even then there's a probability of users in the same 1/1000th percentile getting in and getting promoted to trusted.

IP Fragment Packet Flow

All fragment packets are sent through their own 1024 untrusted flows in the Traffic Manager. The first ten bits (LSB) of the source address are used to determine which fragment-flow the packet belongs to. These 1024 fragment flows share untrusted bandwidth with already existing untrusted-flows. In total, there are 2049 untrusted flows: 1024-non-fragment flows, 1024 fragment flows, and 1 control flow.

Fragmented ICMP packets are qualified as ICMP packets rather than fragment packets. Fragment and non-fragmented ICMP packets follow the trusted-ICMP-flow in the Traffic Manager, with a bandwidth limit of 8Kbs.

Host Path Protection Process

The Communications Broker Network Processors (NPs) check the deny and permit lists for received packets, and classify them as trusted, untrusted or denied (discard). Only packets to signaling ports and dynamically signaled media ports are permitted. All other packets sent to Communications Broker ports are filtered. Only packets from trusted and untrusted (unknown) sources are permitted; any packet from a denied source is dropped by the NP hardware. The Traffic Manager manages bandwidth policing for trusted and untrusted traffic, as described earlier. Malicious traffic is detected in the host processor and the offending device is dynamically added to denied list, which enables early discard by the NP. Devices become trusted based on behavior detected by the Signaling Processor, and dynamically added to the trusted list. This process enables the proper classification by the NP hardware. All other traffic is untrusted (unknown).

Access Control for Hosts

ACLs are supported for the SIP signaling protocol. The Communications Broker loads ACLs so they are applied when signaling ports are loaded. The following rules apply to static NAT entries based on your configuration:

- If there are no ACLs applied to a realm that have the same configured trust level as that realm, the Communications Broker adds a default NAT entry using the realm parameters.
- If you configure a realm with none as its trust level and you have configured ACLs, the Communications Broker only applies the ACLs.
- If you set a trust level for the ACL that is lower than the one you set for the realm, the Communications Broker will not add a separate NAT entry for the ACL.

ACLs provide access control based on destination addresses when you configure destination addresses as a way to filter traffic. You can set up a list of access control exceptions based on the source or the destination of the traffic.

For dynamic ACLs based on the promotion and demotion of endpoints, the rules of the matching ACL are applied.

Access Control Endpoint Classification Capacity and DoS

To view endpoint classification capacity limits for your current platform, use the **show platform limits** command. The output is dependent on the combination of hardware and software you are running.

Static and Dynamic ACL Entry Limits

The Communications Broker can simultaneously police a maximum of 250,000 trusted device flows, while at the same time denying an additional 32,000 attackers. If list space becomes full and additional device flows need to be added, the oldest entries in the list are removed and the new device flows are added.

Host Path Traffic Management

The host path traffic management consists of the dual host paths discussed earlier:

- Trusted path is for traffic classified by the system as trusted. You can initially define trusted traffic by ACLs, as well as by dynamically promoting it through successful SIP registration, or a successful call establishment. You can configure specific policing parameters per ACL, as well as define default policing values for dynamically-classified flows. Traffic for each trusted device flow is limited from exceeding the configured values in hardware. Even an attack from a trusted, or spoofed trusted, device cannot impact the system.
- Untrusted path is the default for all unknown traffic that has not been statically provisioned otherwise. For example, traffic from unregistered endpoints. Pre-configured bandwidth policing for all hosts in the untrusted path occurs on a per-queue and aggregate basis.

Traffic Promotion

Traffic is promoted from untrusted to trusted list when the following occurs:

- successful SIP registration for SIP endpoints
- successful session establishment for SIP calls

Malicious Source Blocking

Malicious source blocking consists of monitoring the following metrics for each source:

- SIP transaction rate (messages per second)
- SIP call rate (call attempts per second)
- Nonconformance/invalid signaling packet rate

Device flows that exceed the configured maximum signaling threshold, untrusted signaling threshold, invalid signaling threshold, or the configured valid signaling threshold within the configured time period are demoted, either from trusted to untrusted, or from untrusted to denied classification.

Blocking Actions

Blocking actions include the following:

- Dynamic deny entry added
- SNMP trap generated, identifying the malicious source (apSysMgmtExpDOSTrap—OID : 1.3.6.1.4.1.9148.3.2.8.0.2)

Dynamically added deny entries expire and are promoted back to untrusted after a configured default deny period time. You can also manually clear a dynamically added entry from the denied list.

13

ACL Configuration

The Oracle Enterprise Communications Broker (Communications Broker) provides for realm-based access control lists (ACLs) so you can explicitly permit or deny access to the realm on a per-packet basis. By default, the Communications Broker allows all packets to access the realm. You configure ACLs to filter packets by IPv4 address, application protocol (SIP) and transport protocol. You can also specify ranges of that IPv4 addressing for which you want to filter.

The Communications Broker allows ACL configuration via the GUI's System Administration controls. You set the Communications Broker to control access to the realm based on available parameters. You can also configure a free text description of each ACL to clarify their purpose.

In the absence of any ACLs, the Communications Broker allows access to all packets. In addition, ACLs support real-time configuration, meaning they do not require system reboot to become functional.

Configuration Overview

Configuring Communications Broker DoS protection includes masking source IP and port parameters to include more than one match and configuring guaranteed minimum bandwidth for trusted and untrusted signaling path. You can also configure signaling path policing parameters for individual source addresses. Policing parameters are defined as peak data rate (in bytes/sec), average data rate (in bytes/sec), and maximum burst size.

You can configure deny list rules based on the following:

- ingress realm
- source IP address
- source port
- transport protocol (TCP/UDP)
- application protocol (SIP)

Configure an ACL

You can add to specify how you want the Communications Broker to enforce realm access by configuring access control entries.

1. Access the Access Control configuration object

   Click the **Configuration** tab, **System Administration** section, **DoS**

2. On the Access Control page, do the following:

Realm ID	Enter the Name of the ingress realm to which this ACL applies.
Description	Enter a text description of the ACL for identification purposes.

Source Address	Enter the source IPv4 address and port number for the host in the following format:
	
	`<IP address>[/number of address bits>][:<port>][/<port bits>]`
	
	For example:
	
	`10.0.0.1/24:5000/14`
	`10.0.0.1/16`
	`10.0.0.1/24:5000`
	`10.0.0.1:5000`

Destination Address	(This is ignored if you configure an application protocol.) Enter the destination IPv4 address and port for the destination in the following format: `<IP address>[/number of address bits>][:<port>[/<port bits>]]` You do not need to specify the number of address bits if you want all 32 bits of the address to be matched. You also do not need to specify the port bits if you want the exact port number matched. If you do not set the port mask value or if you set it to 0, the exact port number will be used for matching. The default value is **0.0.0.0**.
Application Protocol	Enter the application protocol type for this ACL entry. The valid values are: • SIP • None **Note:** If application-protocol is set to none, the destination-address and port will be used. Ensure that your destination-address is set to a non-default value (0.0.0.0.)
Transport Protocol	Select the transport-layer protocol configured for this ACL entry. The default value is **ALL**. The only valid value is: • ALL
Access	Enter the access control type or trusted list based on the trust-level parameter configuration for this host. The default value is **permit**. The valid values are: • **permit**—Puts the entry into the untrusted list. • **deny**—Puts the entry in the deny list.
trust-level	Indicate the trust level for the host with the realm. The default value is **none**. The valid values are:

	• **none**—Host is always untrusted. It is never promoted to the trusted list or demoted to the deny list. • **low**—Host can be promoted to the trusted list or demoted to the deny list. • **medium**—Host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list. • **high**—Host is always trusted.
invalid-signal-threshold	Enter the number of invalid signaling messages that trigger host demotion. The value you enter here is only valid when the trust level is low or medium. Available values are: • Minimum—Zero (0) is disabled. • Maximum—999999999 If the number of invalid messages exceeds this value based on the tolerance window parameter, the host is demoted. The tolerance window default is 30 seconds. Bear in mind, however, that the system uses the same calculation it uses for specifying "recent" statistics in show commands to determine when the number of signaling messages exceeds this threshold. This calculation specifies a consistent start time for each time period to compensate for the fact that the event time, such as a user running a show command, almost never falls on a time-period's border. This provides more consistent periods of time for measuring event counts. The result is that this invalid signal count increments for two tolerance windows, 60 seconds by default, within which the system monitors whether or not to demote the host. The signal count for the current tolerance window is always added to the signal count of the previous tolerance window and compared against your setting.
maximum-signal-threshold	• Minimum—Zero (0) is disabled. • Maximum—999999999 If the number of messages received exceeds this value within the tolerance window, the host is demoted.
untrusted-signal-threshold	Set the maximum number of untrusted messages the host can send within the tolerance window. Use to configure different values for trusted and un-trusted endpoints for valid signaling message parameters. Also configurable per realm. The default value is **0**, disabling this parameter. The valid range is: • Minimum—Zero (0) is disabled. • Maximum—999999999
deny-period	Indicate the time period in seconds after which the entry for this host is removed from the deny list. The default value is **30**. The valid range is: • Minimum—Zero (0) is disabled. • Maximum—999999999

3. Click **OK**.
4. (Optional) Add another Access Control list.
5. Save the configuration.

Access Control for a Realm

Each host within a realm can be policed based on average rate, peak rate, and maximum burst size of signaling messages. These parameters take effect only when the host is trusted. You can also set the trust level for the host within the realm. All untrusted hosts share the bandwidth defined for the media manager: maximum untrusted bandwidth and minimum untrusted bandwidth.

To configure access control for a realm:

1. Access the Realm Config.

   Click the **Configuration** tab, **Network**, **Realm Config**
2. On the Realm Config page, do the following:

Access Control Trust Level	Indicate the trust level for the host with the realm. The default value is **none**. The valid values are: • **none**—Host is always untrusted. It is never promoted to the trusted list or demoted to the deny list. • **low**—Host can be promoted to the trusted list or demoted to the deny list. • **medium**—Host can be promoted to the trusted list but is only demoted to untrusted. It is never added to the deny list. • **high**—Host is always trusted.
Invalid Signal Threshold	Enter the number of invalid signaling messages that trigger host demotion. The value you enter here is only valid when the trust level is low or medium. Available values are: • Minimum—Zero (0) is disabled. • Maximum—999999999 If the number of invalid messages exceeds this value based on the tolerance window parameter, configured in the media manager, the host is demoted. The tolerance window default is 30 seconds. Bear in mind, however, that the system uses the same calculation it uses for specifying "recent" statistics in show commands to determine when the number of signaling messages exceeds this threshold. This calculation specifies a consistent start time for each time period to compensate for the fact that the event time, such as a user running a show command, almost never falls on a time-period's border. This provides more consistent periods of time for measuring event counts. The result is that this invalid signal count increments for two tolerance windows, 60 seconds by default, within which the system monitors whether or not to demote the host. The signal count for the current tolerance window is always added to the signal count of the previous tolerance window and compared against your setting.

Maximum Signal Threshold	• Minimum—Zero (0) is disabled. • Maximum—999999999 If the number of messages received exceeds this value within the tolerance window, the host is demoted.
Untrusted Signal Threshold	Set the maximum number of untrusted messages the host can send within the tolerance window. Use to configure different values for trusted and un-trusted endpoints for valid signaling message parameters. Also configurable per realm. The default value is **0**, disabling this parameter. The valid range is: • Minimum—Zero (0) is disabled. • Maximum—999999999
Deny Period	Indicate the time period in seconds after which the entry for this host is removed from the deny list. The default value is **30**. The valid range is: • Minimum—Zero (0) is disabled. • Maximum—999999999

3. Click **OK**.
4. Save the configuration.

Changing the Default Oracle Enterprise Communications Broker Behavior

The Oracle Enterprise Communications Broker automatically creates permit untrusted ACLs that let all sources (address prefix of 0.0.0.0/0) reach each configured realm's signaling interfaces, regardless of the realm's address prefix. To deny sources or classify them as trusted, you create static or dynamic ACLs, and the global permit untrusted ACL to specifically deny sources or classify them as trusted. Doing this creates a default permit-all policy with specific deny and permit ACLs based on the realm address prefix.

You can change that behavior by configuring static ACLs for realms with the same source prefix as the realm's address prefix; and with the trust level set to the same value as the realm. Doing this prevents the permit untrusted ACLs from being installed. You then have a default deny all ACL policy with specific static permit ACLs to allow packets into the system.

Example 1 Limiting Access to a Specific Address Prefix Range

The following example shows how to install a permit untrusted ACL of source 12.34.0.0/16 for each signalling interface/port of a realm called access. Only packets from within the source address prefix range 12.34.0.0/16, destined for the signaling interfaces/port of the realm named access, are allowed. The packets go into untrusted queues until they are dynamically demoted or promoted based on their behavior. All other packets are denied/dropped.

- Configure a realm called access and set the trust level to low and the address prefix to 12.34.0.0/16.
- Configure a static ACL with a source prefix of 12.34.0.0/16 with the trust level set to low for the realm named access.

Example 2 Classifying the Packets as Trusted

Building on Example 1, this example shows how to classify all packets from 12.34.0.0/16 to the realm signaling interfaces as trusted and place them in a trusted queue. All other packets from outside the prefix range destined to the realm's signaling interfaces are allowed and classified as untrusted; then promoted or demoted based on behavior.

You do this by adding a global permit untrusted ACL (source 0.0.0.0) for each signaling interface/port of the access realm. You configure a static ACL with a source prefix 12.34.0.0/16 and set the trust level to high.

Adding this ACL causes the Oracle Enterprise Communications Broker to also add a permit trusted ACL with a source prefix of 12.34.0.0/16 for each signaling interface/port of the access realm. This ACL is added because the trust level of the ACL you just added is high and the realm's trust level is set to low. The trust levels must match to remove the global permit trusted ACL.

Example 3 Installing Only Static ACLs

This example shows you how to prevent the Oracle Enterprise Communications Broker from installing the global permit (0.0.0.0) untrusted ACL.

- Configure a realm with a trust level of none.
- Configure static ACLs for that realm with the same source address prefix as the realm's address prefix, and set the trust level to any value.

The system installs only the static ACLs you configure.

14
ECB Sync

The Oracle Enterprise Communications Broker (Communications Broker) allows you to configure multiple Communications Brokers to interact with each other and share operational information. This functionality, called ECB Synchronization, works like layer-three routers dynamically exchanging routing information. This results in extensible Communications Broker deployments where any Communications Broker can use information from a peer Communications Broker to make a routing decision, including simply forwarding to that peer so that it can perform the routing.

ECB sync uses the Cluster Network Protocol (CNP). Communications Brokers configured for synchronization share the following information:

- User database
- Routes
- Dial plan
- Agents
- Policy

ECB sync shares information between Communications Brokers configured as pairs, where a transmitter sends to a receiver. The receiver updates its configuration with data from the transmitter. Categories of information exchanged include:

- Routing information - how to reach a destination.
- Context information - how to handle a given endpoint.

ECB sync configurations begin with establishing peers. Peers can operate as:

- Transmitter—Provides its peers with its information.
- Receiver—Uses the information received from a transmitter to extend its operational scope.
- Both Transmitter and Receiver.

Any Communications Broker can peer with up to 10 other Communications Brokers. A transmitter can provide its information to no more than 10 receivers. A receiver can obtain information from no more that 10 transmitters.

ECB Sync uses SIP SUBSCRIBE to establish relationships and exchange data. The system never displays the data in SIP SUBSCRIBE traffic in clear text, providing a layer of obfuscation. The system also uses SIP Digest authentication for authentication and authorization of peers. Optionally, you can configure TLS to secure ECB Sync traffic.

As soon as you add agents to the ECB Sync agent list and the configuration activated, the Communications Broker begins to send SUBSCRIBEs to its Sync agents. Subscriptions, subscription refreshes, and subscription termination all follow standard SIP procedures. Transmitters use NOTIFYs to send information to receivers. The process can use multiple, concurrent NOTIFY messages if the amount of information in the NOTIFY exceeds maximum payload. The transmitter also compresses information. The receiver un-compresses information.

Detail on transmitter/receiver interaction includes:

- The transmitter sends only its own configuration data, not data learned from another Communications Broker.
- The transmitter updates all of its receivers with full configuration information upon an activate if any applicable configuration information has changed. Changes, for example, to a network interface would not trigger an ECB Sync update.
- The receiver's subscription refresh interval is 1/2 the expires time (30 seconds). This timing is not configurable. If the transmitter does not receive any refresh within the expires time, it terminates the subscription.
- The transmitter does not send subsequent NOTIFY messages without positive confirmation of the previous NOTIFY by the receiver.
- The receiver re-sends unanswered and rejected SUBSCRIBE messages in 60 second intervals until it receives a response.
- The transmitter cancels all NOTIFY procedures upon receipt of an error message from the receiver. When this happens, the receiver restarts the NOTIFY process from the beginning. The transmitter makes this additional attempt to resend the configuration/user data only once if it receives such an error.
- When the receiver encounters an unrecoverable error, it discards all information previously learned from the transmitter, cancels the subscription, and initiates a new subscription.
- The receiver un-subscribes from a transmitter gracefully upon activation of a configuration change that removes that transmitter from the receiver's list.

The Communications Broker secures SUBSCRIBE and NOTIFY transactions using SIP Digest procedures. The transmitter challenges the receiver upon receipt of a subscription. Using SIP Digest, the receiver replies to the challenge with standard SIP Digest user, realm, and password hash for the transmitter to verify. ECB sync configuration includes specifying the secret to use for the password. You must configure all receivers and all transmitters to use the same secret for this purpose.

The receiver tracks ECB sync information, differentiating between its own configuration and that of each transmitter. Grouping sync information by its source allows the receiver to discard the correct information if a transmitter becomes unreliable or invalid. The receiver also uses this grouping to prioritize overlapping configuration information. Overlapping configuration objects include those using the identical key. The rules include:

- A receiver uses it's own object if it is in the running configuration.
- If multiple transmitters provide information on the same object, the receiver uses transmitter name alphabetical order to determine which information to use.

Receivers keep all objects from all sources in memory should the source in use become invalid.

Note that all transmitters and receivers must use the same CNP version to operate properly. If a receiver gets CNP data from a higher version, it returns a 489 "Bad Event" error and drops the payload.

ECB Sync and High Availability

A receiver updates its standby Communications Broker with ECB sync data and status. If an active Communications Broker stops responding, the standby becomes available for immediate use as a receiver. Conversely, if a transmitter transitions from active to standby, the receiver assumes its subscription is still valid until it refreshes and receives a 481 error message. Upon receiving the 481 message, the receiver terminates the existing subscription and starts a new one with the new active. Note that the receiver retains learned configuration information despite the subscription change.

Synchronizing the Registration Cache

ECB Sync data includes the Oracle Enterprise Communications Broker's registration cache. The user can enable registration cache sync via a checkbox within the Sync config settings under the ECB Sync icon.

When enabled, the Oracle Enterprise Communications Broker presents its registration cache to all ECB Sync agents every nine minutes. Each ECB Sync agent uses this data to create a separate, ECB Sync-only registration cache table that includes contacts and the ECB from which it learned the cache entry.

When a call comes for a contact found in the ECB Sync-only registration cache, the Oracle Enterprise Communications Broker receiving the call adds a URI parameter to the request URI of the TO header and forwards the message to the Oracle Enterprise Communications Broker in the table. This URI parameter informs the target Oracle Enterprise Communications Broker that is must only use its registration cache for routing this call. This parameter appears as follows.

```
TO sip:user2@server2.com;orcl-regonly=true
```

In these cases, both Oracle Enterprise Communications Brokers forward the call. The Oracle Enterprise Communications Broker receiving the call uses all other routing sources to route the call, including LDAP, LST and UserDB. The Sync agent Oracle Enterprise Communications Broker routes the call using its registration cache, and skips all other routing sources, including LDAP, LST and UserDB.

ECB Sync Operations

Add Sync Agents. and enable ECB Sync configuration settings when you want set up ECB Sync Operations.

For more information, see:

1. Add Sync Agent
2. Enable Sync Config Settings

Add Sync Agent

A Sync Agent requires you to specify at least one agent for the Communications Broker to peer with before the system can perform sync operations. Each Communications Broker can peer with up to ten Sync agents and each Sync agent can peer with up to 10 Communications Brokers.

Each Communications Broker requires a unique hostname configured in the system-config. For synchronization, the session agent must have the same name as the hostname assigned to the remote sync agent. Configure the agents that you want to add as Sync peers.

Use the optional step in the following procedure to add more agents to Sync.

1. Access the Sync Agents configuration object.

   Configuration tab, **System Administration** section, **Sync**, **Sync Agents**.
2. On the **Sync Agents** page, click **Add**.
3. On the **Add Sync Agent** page, select an agent from the drop-down list.

4. Click **OK**.
5. (Optional)—Repeat step 3 to add another sync agent.
6. Click **Back**.
7. Save the configuration.

Enable Sync Config Settings

Enable sync configuration when you want one Communications Broker to share information with another Communications Broker. For example, to share information about users, routes, dial plans, and agents.

To use Sync, you must enable the service and set the authentication secret. You can optionally enable the system to sync the configuration and the registration cache from one Communications Broker to another.

The secret that you enter in the Sync configuration must match the secret used by the Sync agents.

1. Access the Sync Config Settings configuration object.

   Configuration tab, **System Administration** section, **Sync**, **Sync Config Settings**.

2. On the **Modify Configuration Settings** page, do the following:

State	Select to enable Communications Broker sync operations.
Secret	Secret—Enter the secret that you use for authenticating with Communications Broker peers
Configuration	(Optional)—Select to enable the system to sync the configuration from one Communications Broker to another.
Registration	(Optional)—Select to enable the system to sync the registration cache from one Communications Broker to another.

3. Click **OK**.

Delete a Sync Agent

To delete or remove a Sync agent, use the **Sync Agents** option in the GUI.

Points to Note before deleting a Sync Agent

When you delete a Sync Agent:

- All learned data is removed. The existing data and the newly added data is transmitted to Peer ECB as since One -way Sync will be in place.
- Sync Agents are removed from the ECB Sync Peer status table
- Subscription expires when the ECB Sync agent is deleted or ECB Sync configuration is disabled.

To delete a Sync Agent

1. Click **Configuration** > **Sync** > **Sync Agents**.
2. In the **Sync Agent** page, select the Sync Agent and click **Delete**.

ECB Sync Monitoring

As described, the Oracle Enterprise Communications Broker keeps track of Communications Broker sync peers' status and data.

This information is visible via the GUI, as follows:

- Peer Status—The **ECB sync peer status** widget displays all peer relationships that apply to the current Oracle Enterprise Communications Broker. Information displayed includes:
  - Peer
  - Status—Indicating whether the peer relationship is In Service or Out of Service. This status indicates the reacheability. If you have not configured Sync Agents, the ECB peer status statistics does not display entries in both GUI and the ACLI.
  - Refresh time for producer, and consumer
  - Transmitter and Receiver Sync State—Verifying subscription status for both peers.
  - Transmitter and Receiver Uptime—Displaying the uptime of the subscription.
- Peer Data—The attributes listed below include a column titled **Learned From**, indicating the peer Communications Broker from which the data was synchronized:
  - User database
  - Routes
  - Dial plan
  - Agents
  - Policy

15

HMR Configuration

Header manipulation rules (HMRs) use Oracle Enterprise Communications Broker-specific controls and/or REGEX to identify information in signaling messages the user wants to change. These rules get applied to Oracle Enterprise Communications Broker agents, and operate globally on all applicable signaling traffic that reaches that agent. This section provides instructions on configuring HMR on the Oracle Enterprise Communications Broker GUI interface.

See the Header Manipulation Appendix in this guide for full explanation of how HMRs work. This appendix is provided for those with no prior experience with HMRs. HMR configuration errors can adversely impact all of an agent's traffic. Be fully confident about the intent of an HMR, and review your HMR configurations carefully before activating them.

SIP Header Manipulation

SIP header manipulation allows you to add, delete, or modify SIP message attributes on the Oracle Enterprise Communications Broker (Communications Broker). For example, SIP headers and SIP header elements.

The most common reason for manipulating SIP headers and SIP header elements is to fix an incompatibility problem between two SIP endpoints. For example, Softswitch - PSTN incompatibility or a SIP messaging problem between two different IP PBX platforms in a multi-site deployment where calls between the platforms are unsuccessful due to problems in the SIP messaging.

To enable the SIP header manipulation, create rule sets in which you specify header manipulation rules and, optionally, header element manipulation rules. SIP header elements are the sub-parts of the header, such as the header value, the header parameter, the URI parameter, and so on, excluding the header name. You can specify the actions that you want the system to perform for each header element.

After creating the header manipulation rule set, apply it to a session agent or SIP interface as "inbound" or "outbound."

Multi-Hop SIP Header Manipulation Rules

Oracle Enterprise Communications Broker (Communications Broker) Header Manipulation Rule (HMR) support allows you to specify a manipulation type depending on an agent's location (hop) in a route. Applicable hops include the next and last hop of a route. Applying an HMR when an agent is the last hop in a route is referred to as "multi-hop" HMR. You do not need to make any changes to the HMR configuration for it operate as a multi-hop HMR.

The default setting makes the system apply the outbound HMR only when the agent is the next hop in the route's path.

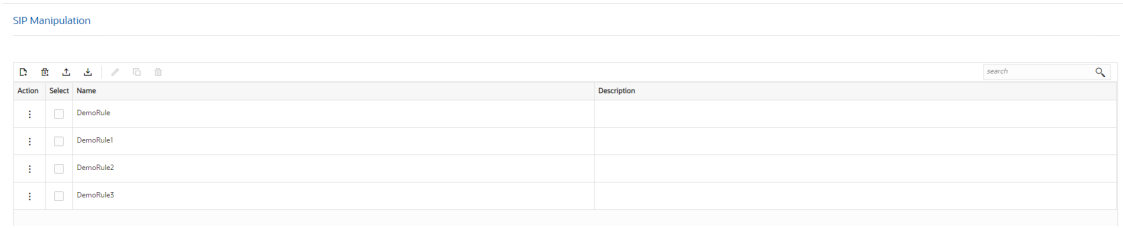
If there are multiple HMRs that the Communications Broker must apply for the route, it applies the HMR for the last hop first. If the same agent is both next and last hop for any given traffic, the Communications Broker applies the HMR only once regardless of the `Apply Outbound Manipulation On` setting.

Set the `Apply Outbound Manipulation On` parameter on the agent to specify when the system applies the agent's outbound HMR. Default: Next Hop Only. Valid values: Next Hop Only | Last Hop Only | Next and Last Hop.

SIP Header Manipulation Configuration Dialogs

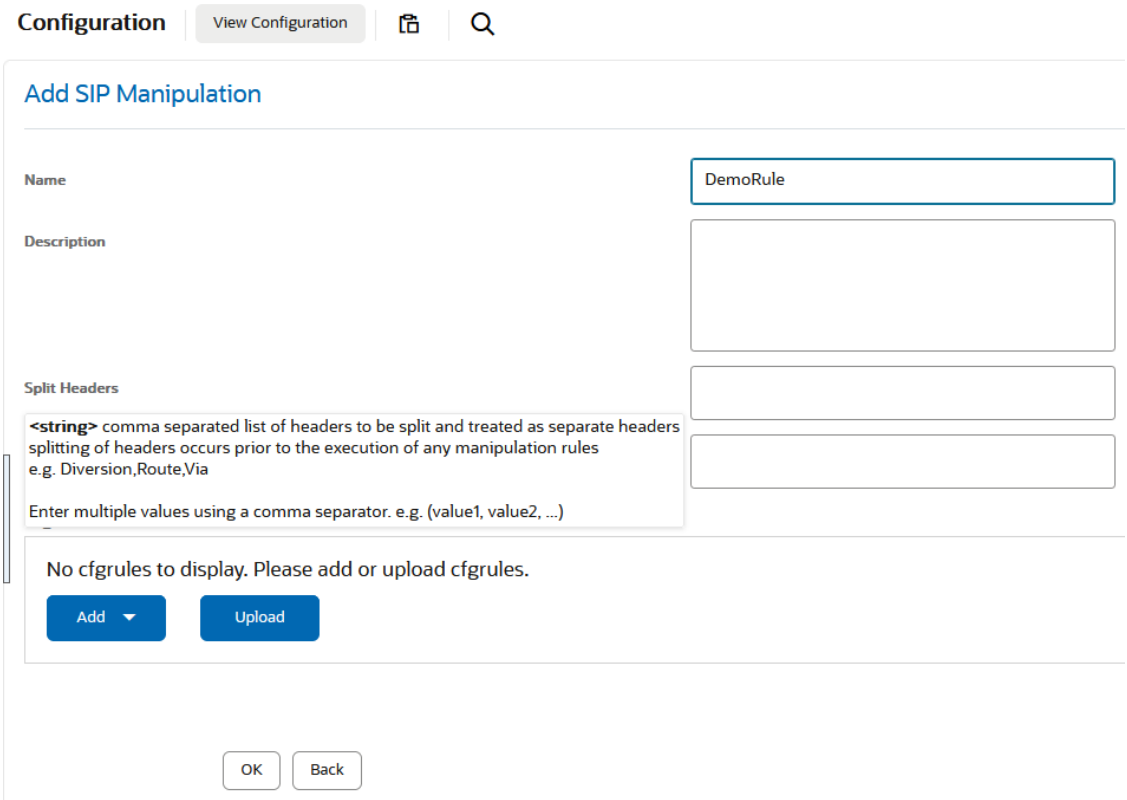
You configure Header Manipulation Rules (HMR) on the Oracle Enterprise Communications Broker (Communications Broker) by way of the SIP Manipulation page. In the configuration, you use the HMR dialogs to assign HMRs to agents, which defines where the system uses the rule.

The first HMR page, located at **Configuration** tab, **System Administration** section, **SIP Manipulation**, is the SIP Manipulation list. This list shows all manually configured HMRs available on the system. The dialog includes controls to add new HMRs, to edit, copy, and delete existing rules, and to upload and download pre-configured HMRs in to the system. The following screen capture shows the SIP manipulation page which dislpays any existing manually configured HMRs.

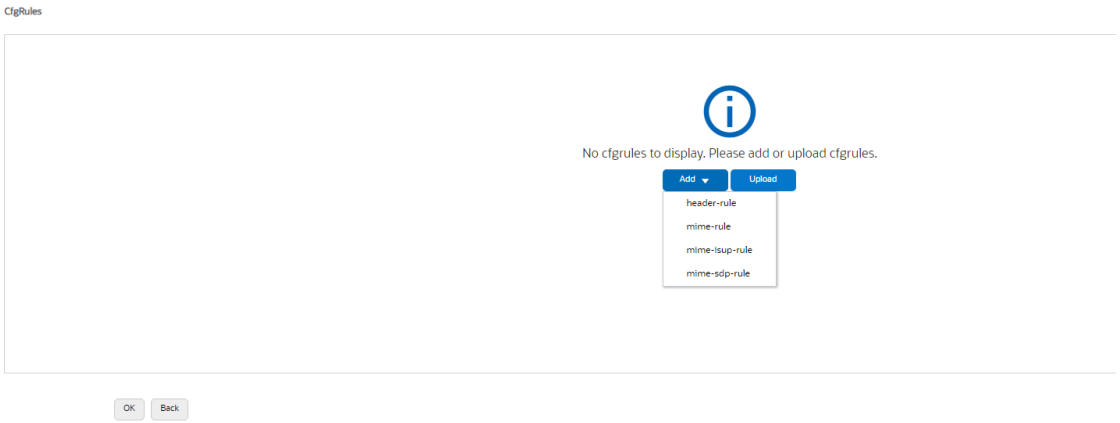


When you click the **Add**, the system displays the Add SIP Manipulation dialog where you create a new HMR. The following screen capture shows the Add SIP Manipulation page.

Figure 15-1 Add SIP Manipulation

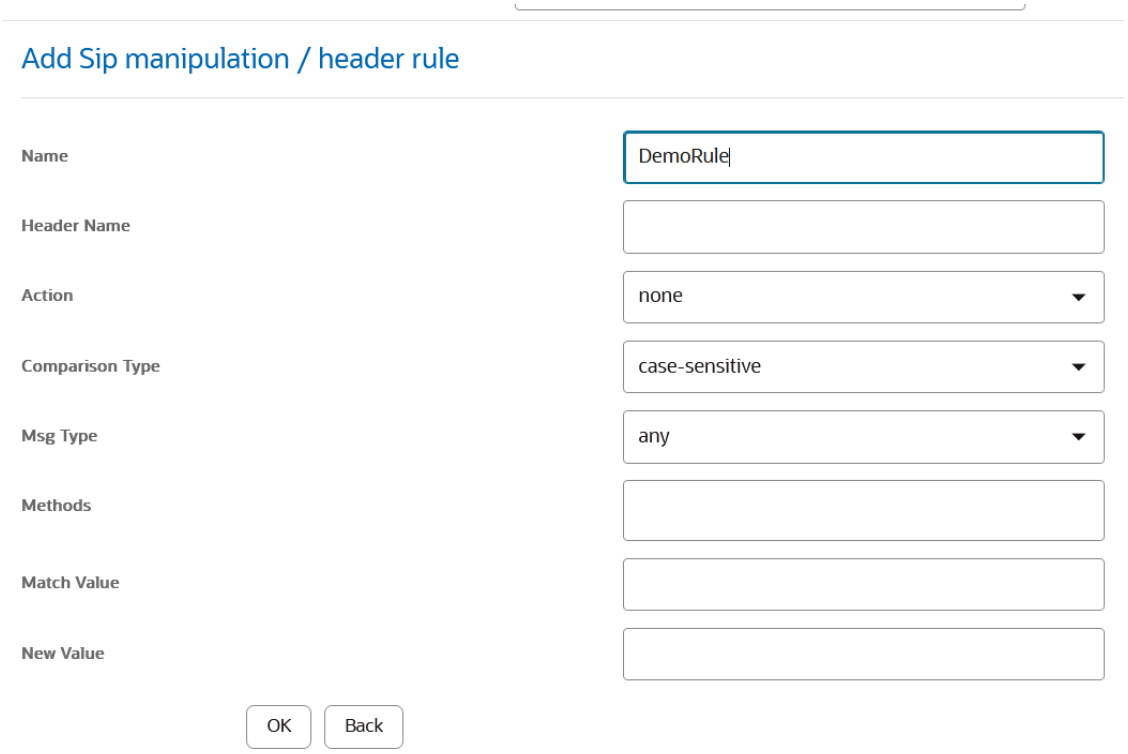


After naming and describing the HMR, scroll to the **CfgRules** section of the page, which displays the list of header or MIME rules that you can apply to this HMR. Note that header rule order of execution is critical when an HMR contains multiple rules. You can manage the order of HMR execution with the **Move** control in the menu that displays when you right-click a row in the table. The following screen capture shows the **CfgRules** drop-down list.



When you click a Configuration Rule, the system dislpays the configuration the particular configuration page for the rule type. The following illustration shows the **Add SIP Manipulation / Header Rule/ Element Rule** page as an example of a configuration for a Configuration Rule.

Figure 15-2 Add SIP Manipulation/Header rule dialog box



SIP Header Manipulation Rules Attributes and Values Reference

Refer to the following table for information about the attributes that you can configure for SIP header manipulation rules.

Attributes	Values and Descriptions
Action	• add—Adds a new header, if that header does not exist. • delete—Deletes the header, if it exists. • find-replace-all—Finds all matching headers and replaces with the header you specified for “Split” and "Join." • log—Logs the header. • manipulate—Manipulates the elements of this header to the element rules configured. • monitor—Monitors the header. • store—Stores the header. • none—(default) No action is taken. • reject—Rejects the header. • sip-manip—Manipulates the SIP elements of this header to the element rules configured. Default: None.
Comparison type	• boolean—Header is compared to header rule and must match exactly or it is rejected. • case-insensitive—Header is compared to header rule regardless of the case of the header. • case-sensitive—(default) Header is compared to the header rule and case must be exactly the same or it is rejected. • pattern-rule—Header is compared to the header rule and the pattern must be exactly the same or it is rejected. • refer-case-insensitive—Header is compared to the header rule regardless of the case in a REFER message. • refer-case-sensitive—Header is compared to the header rule and the case must be exactly the same as in the REFER message or it is rejected. Default: Case-sensitive.
Format	• ascii-string - A character-encoding scheme that represents text (128 ASCII codes, 7 bits). • binary-ascii - An encoding scheme where each byte of an ASCII character is used. Can use up to 256 bit patterns . • hex-ascii - An encoding scheme that uses a string of numbers (no spaces) to represent each ASCII character.
Header name	The name of the header to which the rule applies. Case-sensitive.
Match value	The value that you want to match against the element value for an action to be performed.

Attributes	Values and Descriptions
Match val type	The type of value to match to the match-field entry for the action to be performed. • any—(default) Element value in the SIP message is compared with the match-value field entry. If the match-value field is empty, all values are considered a match. • fqdn—Element value in the SIP message must be a valid FQDN to be compared to the match-value field entry. If the match-value field is empty, any valid FQDN is considered a match. If the element value is not a valid FQDN, it is not considered a match. • ip—Element value in the SIP message must be a valid IP address to be compared to the match-value field entry. If the match-value field is empty, any valid IP address is considered a match. If the element value is not a valid IP address, it is not considered a match.
Media type (SDP descriptor for SDP media rule)	• m—Media name and transport address • i—Media title • c—Connection information (optional when configured at the session level) • b—Zero or more bandwidth information lines • k—Encryption key • a—Zero or more media attribute lines • t—The session time is active • r—Zero or more repeat times
Methods	SIP method names to which you want to apply the header rule. For example, INVITE, ACK, BYE. When this field is empty, the system applies the MIME rule to all methods. Default: Blank.
Mime header	The parameter name to which the rule applies. The parameter name depends on the element name you entered. For uri-param, uri-user-param, and header-param it is the parameter name to be added, replaced, or deleted. For all other types, it serves to identify the element rule and any name can be used. Alpha-numeric characters. Default: blank.
Msg type	• any—(default) Requests, replies, and out-of-dialog messages • out-of-dialog—Out of dialog messages only. • reply—Reply messages only • request—Request messages only Default: Any.
Name	The name you want to use for the rule. Default: Blank.

Attributes	Values and Descriptions
New value	The value for a new element or replacement value for an existing element. You can enter an expression that includes a combination of absolute values, pre-defined parameters, and operators. • Absolute values—Use double quotes for clarity. You must escape all double quotes and back slashes that are part of an absolute value, and enclose the absolute value in double quotes. • Pre-defined values.—Pre-defined parameters always start with a $. For valid values, see the Pre-defined Parameters table. • Operators parameters—For valid values, see the Operators table. The following table describes the pre-defined parameters.

Pre-defined Parameter	Description
$ORIGINAL	Original value of the element is used.
$LOCAL_IP	IP address of the SIP interface on which the message was received for inbound manipulation; or sent on for outbound manipulation.
$REMOTE_IP	IP address the message was received from for inbound manipulation; or being sent to for outbound manipulation.
$REMOTE_VIA_HOST	Host from the top Via header of the message is used.
$TRUNK_GROUP	Trunk group is used.
$TRUNK_GROUP_CONTEXT	Trunk group context is used.

The following table describes the Operators.

Operator	Description
+	Append the value to the end. For example: acme”+”packet generates acmepacket
+^	Prepends the value. For example: acme”+^”packet generates packetacme
-	Subtract at the end. For example: 112311”-”11 generates 1123
-^	Subtract at the beginning. For example: 112311”-^”11 generates 2311

Attributes	Values and Descriptions
Parameter name	The parameter name to which the rule applies. The parameter name depends on the element name you entered. For uri-param, uri-user-param, and header-param it is the parameter name to be added, replaced, or deleted. For all other types, it serves to identify the element rule and any name can be used. Alpha-numeric characters. Default: Blank.
Type	The type of element on which to perform the action. Default: Blank. • header-param—Perform the action on the parameter portion of the header. • header-param-name—Perform the action on the header parameter name. • header-value—Perform the action on the header value. • mime—Perform the action on Multipurpose Internet Mail Extensions (MIME). • reason-phrase—Perform the action on reason phrases. • status-code—Perform the action on status codes. • teluri-param—Perform the action on a SIP telephone Uniform Resource Identifier (URI). • uri-display—Perform the action on the display of the SIP URI. • uri-header—Perform the action on a header included in a request constructed from the URI. • uri-header-name—Perform the action on a SIP URI header name. • uri-host—Perform the action on a Host portion of the SIP URI. • uri-param—Perform the action on the parameter included in the SIP URI. • uri-param-name—Perform the action on the name parameter of the SIP URI. • uri-phone-number-only—Perform the action on a SIP URI phone number only. • uri-port—Perform the action on the port number portion of the SIP URI. • uri-user—Perform the action on the user portion of the SIP URI. • uri-user-only—Perform the action on the user portion only of the SIP URI. • uri-user-param—Perform the action on the user parameter of the SIP URI.

Attributes	Values and Descriptions
Type (SDP descriptor for SDP line rule)	• v—Protocol version • o—Originator and session identifier • s—Session name • i—Session information • u—URI of description • e—Email address • p—Phone number • c—Connection information (not required when included in all media) • b—Zero or more bandwidth information lines or one or more time descriptions("t=" and "r=" lines) • z—Time zone adjustments • k—Encryption key • a—Zero or more session attribute lines or zero or more media descriptions • t—Time the session is active • r—Zero or more repeat times

SIP Header Manipulation Configuration

Configuring SIP manipulations from the Web GUI is a multi-faceted process performed through a series of nested dialogs that differ depending on the particular header and header element that you want to manipulate. It is not practical to document the entire SIP manipulations configuration process in one procedure. The documentation begins with the "Configure SIP Manipulation", topic where you can set the global parameters, if that is all you need. The documentation continues with procedures for each particular header and header element that you can manipulate. Each of those topics includes the global settings, so you can set or modify them there, as well.

header and header element that you can manipulate include the following:

- Configure MIME Rule—includes the mime-header-rule element.
- Configure MIME ISUP Rule—includes the mime-header-rule and isup-param-rule elements.
- Configure MIME SDP Rule—includes the mime-header-rule, sdp-session-rule, and the sdp-media-rule.

When you finish configuring SIP manipulations, apply the rules to a session agent or SIP interface as "inbound" or "outbound."

Configure SIP Manipulation

Descriptions of the applicable Oracle Enterprise Communications Broker configuration fields are provided below:

1. Access the SIP Manipulation configuration object.

   Configuration tab, **System Administration** section, **SIP Manipulation**.

2. On the SIP Manipulation page, do the following:

Table 15-1 SIP Manipulation

Fields	Description
Name	Enter the unique identifier for this SIP Manipulation. Use this name when applying this SIP manipulation. Default: Blank.
Description	Enter a text description for this SIP Manipulation.
Split Header	Enter a comma separated list of headers that the system separates prior to executing the manipulation. Example values: Allow,P-Asserted-Identity | Diversion,Allow.
Join Header	Enter a comma separated list of headers that the system joins together after the manipulation execution is complete. Example values: Allow,P-Asserted-Identity | Diversion,Allow.

3. (Optional) Configure header rules and element rules. See "Configure a Header Rule" and "Configure an Element Rule."

Configure a SIP Manipulation Header Rule

You can configure SIP header rules and element rules on the Oracle Enterprise Communications Broker (Communications Broker) from the "CfgRules" section of the "SIP Manipulations" page.

In the following procedure, you set the SIP Header Manipulation, Header Rule, and Element Rule parameters.

1. Access the SIP Manipulation configuration object.

   Configuration tab, **System Administration** section, **SIP Manipulation**.

2. On the SIP Manipulation configuration page, do one of the following:

   a. Select and existing SIP manipulation configuration from the table, right-click, and click **Edit**. (Subsequent SIP manipulation pages use "Modify" in the title.)

   b. Click the **Add** icon. (Subsequent SIP manipulation pages use "Add" in the title.)

3. On the Add or Modify SIP Manipulation page, do one of the following.

   a. If you chose **Add**, you must enter a name for this SIP Manipulation. (You can optionally complete the Description, Split Headers, and Join Headers parameters, at this time. See "Configure SIP Manipulation.") Proceed to the next step.

   b. If you chose to edit an existing configuration, proceed to the next step.

4. On the Add or Modify SIP Manipulation page under Cgf Rules, click the **Add** button and select **header-rule**.

5. On the Add SIP Manipulation / Header Rule page, do the following.

Name	Enter a unique name for this rule set. Valid values: Alpha-numeric.
Header Name	Enter the name of the header on which you want the Communications Broker to use this HMR. Set this parameter to @status-line, where the at-sign (@)—not allowed in SIP header names—to prevent undesired matches with header having the name status-code. Default: Blank.

Action	Select an action from the drop-down list for the header rule. Default: None. Valid values: Add | Delete | Find Replace All | Log | Manipulate | Monitor | None | Reject | SIP Manip | Store.
Comparison Type	Specify how the Communications Broker processes the match rules against the SIP header. Default: Refer Case Sensitive. Valid values: Boolean | Case Insensitive | Case Sensitive | Pattern Rule | Refer Case Insensitive | Refer Case Sensitive.
Msg Type	Specify the message type this rule applies to. Default: Any. Valid Values: Any | Out of Dialog | Reply | Request | Out of Dialog.
Methods	Enter the method type to use when this SIP HMR is used. Default: Blank. Valid values" ACK | CANCEL | INVITE. When you do not set the method, the Communications Broker applies the rule to all SIP methods.
Match Value	Enter the value to match against the header value in SIP packets; the Communications Broker matches these against the entire SIP header value. This is where you can enter values to match using regular expression values. Your entries can contain Boolean operators. When you configure HMR (using SIP manipulation rules and elements rules, you can use escape characters in the **match-value** parameter to support escaping Boolean and string manipulation operators. You can also escape the escape character itself, so that it is used as a literal string. For example, the Communications Broker treats the string \+1234 as +1234. The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “. You can also use the variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.
New Value	When you set the action parameter Add or to Manipulate, enter the new value that you want to substitute for the entire header value. This is where you can set stored regular expression values for the Communications Broker to use when it adds or manipulates SIP headers. When you configure HMR (using SIP manipulation rules and elements rules, you can use escape characters in the **match-value** parameter to support escaping Boolean and string manipulation operators. You can also escape the escape character itself, so that it is used as a literal string. For example, the Communications Broker treats the string \+1234 as +1234. The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “. You can also use the variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.
Cfg Rules	(Optional) Click **Add**, **element-rule**, and do the following. • Name—Enter a unique name for this header element rule. You can enter up to 128 alphanumeric characters with no spaces. The name can include the _, ., or - characters, cannot begin with either the . or the - characters. • Parameter name—Enter the parameter name to apply to the rule. • Type—Select the element type to which to apply this rule. • Action—Select an action from the drop-down list to apply to the element rule. Default: None.

- Match Val Type—Select a match value type that this rule applies to from the drop-down list. Default: Any.
- Comparison Type—Select an element type from the drop-down list to which to apply the rule. Default: Case Sensitive.
- Match Value—Enter the match value to compare against the current object. (To clear the value, enter and empty string.)
- New Value—Enter a new value for the object. Quoted display named must be escaped within quotes. For example, \"MyName\". (To clear the value, enter and empty string.)
- Click **OK**. The system displays the SIP Manipulation / Header Rule page.

Do one of the following:

- Add another Element Rule.
- Finish the Header Rule configuration by completing the following steps.

6. Click **OK**.
7. Click **OK**.
8. Save the configuration.

Configure a MIME Rule

You can configure Multi-Purpose Internet Mail Extensions (MIME) header rules and element rules on the Oracle Enterprise Communications Broker (Communications Broker) from the "CfgRules" section of the "SIP Manipulations" page.

In the following procedure, you set the SIP Header Manipulation, MIME Rule, and MIME Header Rule parameters.

1. Access the SIP Manipulation configuration object.

   Configuration tab, **System Administration** section, **SIP Manipulation**.
2. On the SIP Manipulation configuration page, do one of the following:
   a. Select and existing SIP manipulation configuration from the table, right-click, and click **Edit**. (Subsequent SIP manipulation pages use "Modify" in the title.)
   b. Click the **Add** icon. (Subsequent SIP manipulation pages use "Add" in the title.)
3. On the Add or Modify SIP Manipulation page, do one of the following.
   a. If you chose **Add**, you must enter a name for this SIP Manipulation. (You can optionally complete the Description, Split Headers, and Join Headers parameters, at this time. See "Configure SIP Manipulation.") Proceed to the next step.
   b. If you chose to edit an existing configuration, proceed to the next step.
4. On the Add or Modify SIP Manipulation page under Cfg Rules, click the **Add** button and select **mime-rule**.
5. On the Add or Modify SIP Manipulation / Mime Rule page, do the following.

Name	Enter a unique name for this rule set. Valid values: Alpha-numeric.
Content Type	Enter the name of the header on which you want the Communications Broker to use this HMR. Set this parameter to @status-line, where the at-

	sign (@)—not allowed in SIP header names—to prevent undesired matches with header having the name status-code. Default: Blank.
Msg Type	Specify the message type this rule applies to. Default: Any. Valid Values: Any | Out of Dialog | Reply | Request | Out of Dialog.
Methods	Enter the method type to use when this SIP HMR is used. Default: Blank. Valid values" ACK | CANCEL | INVITE. When you do not set the method, the Communications Broker applies the rule to all SIP methods.
Format	Select the encode - decode format from the drop-down list for the MIME content.
Action	Select an action from the drop-down list for the header rule. Default: None. Valid values: Add | Delete | Find Replace All | Log | Manipulate | Monitor | None | Reject | SIP Manip | Store.
Comparison Type	Specify how the Communications Broker processes the match rules against the SIP header. Default: Refer Case Sensitive. Valid values: Boolean | Case Insensitive | Case Sensitive | Pattern Rule | Refer Case Insensitive | Refer Case Sensitive.
Match Value	Enter the value to match against the header value in SIP packets; the Communications Broker matches these against the entire SIP header value. This is where you can enter values to match using regular expression values. Your entries can contain Boolean operators. When you configure HMR (using SIP manipulation rules and elements rules, you can use escape characters in the **match-value** parameter to support escaping Boolean and string manipulation operators. You can also escape the escape character itself, so that it is used as a literal string. For example, the Communications Broker treats the string \+1234 as +1234. The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “. You can also use the variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.
New Value	When you set the action parameter Add or to Manipulate, enter the new value that you want to substitute for the entire header value. This is where you can set stored regular expression values for the Communications Broker to use when it adds or manipulates SIP headers. When you configure HMR (using SIP manipulation rules and elements rules, you can use escape characters in the **match-value** parameter to support escaping Boolean and string manipulation operators. You can also escape the escape character itself, so that it is used as a literal string. For example, the Communications Broker treats the string \+1234 as +1234. The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “. You can also use the variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.
Cfg Rules	Click **Add**, **mime-header-rule**, and do the following. • Name—Enter a unique name for this header element rule. You can enter up to 128 alphanumeric characters with no spaces. The name can include the _, ., or - characters, but cannot begin with either the . or the - characters.

- Mime Header Name—Enter header name within the MIME part to which to apply the rule. Use headername@peramble to change the preamble of a SIP body. Use headername@epilogue to change the epilog of a SIP body.
- Action—Select an action from the drop-down list to apply to the element rule. Default: None.
- Comparison Type—Select the type of comparison from the drop-down list to use for the match value. Default: Match Value. (To clear the value, enter and empty string.)
- Match Value—Enter the match value to compare against the current object. (To clear the value, enter and empty string.)
- New Value—Enter a new value for the object. Quoted display named must be escaped within quotes. For example, \"MyName\". (To clear the value, enter and empty string.)
- Click **OK**. The system displays the SIP Manipulation / Mime Rule dialog.

Do one of the following:

- Add another mime-header-rule.
- Finish the MIME Rule configuration by completing the following steps.

6. Click **OK**.
7. Click **OK**.
8. Save the configuration.

Configure a MIME ISUP Rule

You can configure Multi-Purpose Internet Mail Extensions (MIME) header rules and element rules on the Oracle Enterprise Communications Broker (Communications Broker) from the "CfgRules" section of the "SIP Manipulations" page.

In the following procedure, you set the SIP Header Manipulation, MIME ISUP Rule, MIME Header Rule, and ISUP Param Rule parameters.

1. Access the SIP Manipulation configuration object.

   Configuration tab, **System Administration** section, **SIP Manipulation**.
2. On the SIP Manipulation configuration page, do one of the following:
   a. Select and existing SIP manipulation configuration from the table, right-click, and click **Edit**. (Subsequent SIP manipulation pages use "Modify" in the title.)
   b. Click the **Add** icon. (Subsequent SIP manipulation pages use "Add" in the title.)
3. On the Add or Modify SIP Manipulation page, do one of the following.
   a. If you chose **Add**, you must enter a name for this SIP Manipulation. (You can optionally complete the Description, Split Headers, and Join Headers parameters, at this time. See "Configure SIP Manipulation.") Proceed to the next step.
   b. If you chose to edit an existing configuration, proceed to the next step.
4. On the Add or Modify SIP Manipulation page under Cfg Rules, click the **Add** button and select **mime-header-rule**.

5. On the Add or Modify SIP Manipulation / Mime ISUP Rule page, do the following.

Name	Enter a unique name for this rule set. Valid values: Alpha-numeric.
Content Type	Enter the name of the header on which you want the Communications Broker to use this HMR. Set this parameter to @status-line, where the at-sign (@)—not allowed in SIP header names—to prevent undesired matches with header having the name status-code. Default: Blank.
Msg Type	Specify the message type this rule applies to. Default: Any. Valid Values: Any | Out of Dialog | Reply | Request | Out of Dialog.
Methods	Enter the method type to use when this SIP HMR is used. Default: Blank. Valid values" ACK | CANCEL | INVITE. When you do not set the method, the Communications Broker applies the rule to all SIP methods.
Format	Select the encode - decode format from the drop-down list for the MIME content.
Action	Select an action from the drop-down list for the header rule. Default: None. Valid values: Add | Delete | Find Replace All | Log | Manipulate | Monitor | None | Reject | SIP Manip | Store.
Comparison Type	Specify how the Communications Broker processes the match rules against the SIP header. Default: Refer Case Sensitive. Valid values: Boolean | Case Insensitive | Case Sensitive | Pattern Rule | Refer Case Insensitive | Refer Case Sensitive.
Match Value	Enter the value to match against the header value in SIP packets; the Communications Broker matches these against the entire SIP header value. This is where you can enter values to match using regular expression values. Your entries can contain Boolean operators. When you configure HMR (using SIP manipulation rules and elements rules, you can use escape characters in the **match-value** parameter to support escaping Boolean and string manipulation operators. You can also escape the escape character itself, so that it is used as a literal string. For example, the Communications Broker treats the string \+1234 as +1234. The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “. You can also use the variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.
New Value	When you set the action parameter Add or to Manipulate, enter the new value that you want to substitute for the entire header value. This is where you can set stored regular expression values for the Communications Broker to use when it adds or manipulates SIP headers. When you configure HMR (using SIP manipulation rules and elements rules, you can use escape characters in the **match-value** parameter to support escaping Boolean and string manipulation operators. You can also escape the escape character itself, so that it is used as a literal string. For example, the Communications Broker treats the string \+1234 as +1234. The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “. You can also use the variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.

Cfg Rules (instructions for configuring MIME HeaderRule)	Click **Add**, **MIME Header Rule**, and do the following. • Name—Enter a unique name for this header element rule. • Header Name—Enter header name within the MIME part to which to apply the rule. • Action—Select an action from the drop-down list to apply to the element rule. • Comparison Type—Select the type of comparison from the drop-down list to use for the match value. (To clear the value, enter and empty string.) • Match Value—Enter the match value to compare against the current object. (To clear the value, enter and empty string.) • New Value—Enter a new value for the object. Quoted display named must be escaped within quotes. For example, \"MyName\" . • Click **OK**. The system displays the SIP manipulation / Mime isup rule dialog. Do one of the following: • Add another MIME Header Rule. • Add an ISUP Param Rule, using the steps in the following table cell. • Finish the MIME ISUP rule configuration by completing steps 3-6.
Cfg Rules (instructions for configuring ISUP Param Rule)	Click **Add**, **isup-param-rule**, and do the following. • Name—Enter a unique name for this header element rule. • Type—Enter the parameter type that specifies the part of the isup body to manipulate. • Format—Select a format from the drop down list for the encode - decode mode of the binary body form string form-ascii. • Action—Select an action from the drop-down list to apply to the element rule. • Comparison Type—Select the type of comparison from the drop-down list to use for the match value. (To clear the value, enter and empty string.) • Match Value—Enter the match value to compare against the current object. (To clear the value, enter and empty string.) • New Value—Enter a new value for the object. Quoted display named must be escaped within quotes. For example, "\"MyName\" " . • Click **OK**. The system displays the SIP manipulation / Mime isup rule dialog. Do one of the following: • Add another ISUP Param Rule. • Finish the MIME ISUP Rule configuration by completing the following steps.

6. Click **OK**.
7. Click **OK**.

8. Save the configuration.

Configure a MIME SDP Rule

You can configure Multi-Purpose Internet Mail Extensions (MIME) header rules and element rules on the Oracle Enterprise Communications Broker (Communications Broker) from the "CfgRules" section of the "SIP Manipulations" page.

In the following procedure, you set the SIP Header Manipulation, MIME SDP Rule, MIME Header Rule, SDP Session Rule, and SDP Media Rule parameters.

1. Access the SIP Manipulation configuration object.

   Configuration tab, **System Administration** section, **SIP Manipulation**.

2. On the SIP Manipulation configuration page, do one of the following:

   a. Select and existing SIP manipulation configuration from the table, right-click, and click **Edit**. (Subsequent SIP manipulation pages use "Modify" in the title.)

   b. Click the **Add** icon. (Subsequent SIP manipulation pages use "Add" in the title.)

3. On the Add or Modify SIP Manipulation page, do one of the following.

   a. If you chose **Add**, you must enter a name for this SIP Manipulation. (You can optionally complete the Description, Split Headers, and Join Headers parameters, at this time. See "Configure SIP Manipulation.") Proceed to the next step.

   b. If you chose to edit an existing configuration, proceed to the next step.

4. On the Add or Modify SIP Manipulation page under Cfg Rules, click the **Add** button and select **mime-sdp-rule**.

5. In the Add or Modify SIP Manipulation / MIME SDP Rule page, do the following.

Name	Enter a unique name for this rule set. Valid values: Alpha-numeric.
Content Type	Enter the name of the header on which you want the Communications Broker to use this HMR. Set this parameter to @status-line, where the at-sign (@)—not allowed in SIP header names—to prevent undesired matches with header having the name status-code. Default: Blank.
Msg Type	Specify the message type this rule applies to. Default: Any. Valid Values: Any | Out of Dialog | Reply | Request | Out of Dialog.
Methods	Enter the method type to use when this SIP HMR is used. Default: Blank. Valid values" ACK | CANCEL | INVITE. When you do not set the method, the Communications Broker applies the rule to all SIP methods.
Format	Select the encode - decode format from the drop-down list for the MIME content.
Action	Select an action from the drop-down list for the header rule. Default: None. Valid values: Add | Delete | Find Replace All | Log | Manipulate | Monitor | None | Reject | SIP Manip | Store.
Comparison Type	Specify how the Communications Broker processes the match rules against the SIP header. Default: Refer Case Sensitive. Valid values: Boolean | Case Insensitive | Case Sensitive | Pattern Rule | Refer Case Insensitive | Refer Case Sensitive.
Match Value	Enter the value to match against the header value in SIP packets; the Communications Broker matches these against the entire SIP header

	value. This is where you can enter values to match using regular expression values. Your entries can contain Boolean operators. When you configure HMR (using SIP manipulation rules and elements rules, you can use escape characters in the **match-value** parameter to support escaping Boolean and string manipulation operators. You can also escape the escape character itself, so that it is used as a literal string. For example, the Communications Broker treats the string \+1234 as +1234. The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “. You can also use the variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.
New Value	When you set the action parameter Add or to Manipulate, enter the new value that you want to substitute for the entire header value. This is where you can set stored regular expression values for the Communications Broker to use when it adds or manipulates SIP headers. When you configure HMR (using SIP manipulation rules and elements rules, you can use escape characters in the **match-value** parameter to support escaping Boolean and string manipulation operators. You can also escape the escape character itself, so that it is used as a literal string. For example, the Communications Broker treats the string \+1234 as +1234. The following are escape characters: +, -, +^, -^, &, |, \, (,), ., $, ^, and “. You can also use the variables, $REMOTE_PORT and $LOCAL_PORT, which resolve respectively to the far-end and remote UDP or TCP port value.
Cfg Rules (instructions for configuring mime-header-rule)	(Optional) Click **Add**, **mime-header-rule**, and do the following. • Name—Enter a unique name for this header element rule. • Mime Header Name—Enter header name within the MIME part to which to apply the rule. • Action—Select an action from the drop-down list to apply to the element rule. • Comparison Type—Select the type of comparison from the drop-down list to use for the match value. • Match Value—Enter the match value to compare against the current object. • New Value—Enter a new value for the object. Quoted display named must be escaped within quotes. For example, \"MyName\" . • Click **OK**. The system displays the SIP manipulation / Mime sdp rule dialog. Do one of the following: • Add another mime-header-rule. • Configure the sdp-session-rule and sdp-media-rule options, using the steps in the following table cells. • Finish the MIME SDP rule configuration by completing steps 3-6.
Cfg Rules (instructions for configuring	(Optional) Click **Add**, **sdp-session-rule** , and do the following. • Name—Enter a unique name for this header element rule.

sdp-session-rule)	• Action—Select an action from the drop-down list to apply to the this rule. • Comparison Type—Select the type of comparison from the drop-down list to use for the match value. • Match Value—Enter the match value to compare against the current object. • New Value—Enter a new value for the object. Quoted display named must be escaped within quotes. For example, \"MyName\". • CfgRules—(Optional) Click **Add**, **sdp-line-rule**. • Name—Enter a unique name for this rule. • Type—Enter a descriptor type to specify the SDP line to manipulate. • Action—Select an action from the drop-down list to apply to this rule. • Comparison Type—Select the type of comparison from the drop-down list to use for the match value. • Match Value—Enter the match value to compare against the current object. • New Value—Enter a new value for the object. Quoted display named must be escaped within quotes. For example, \"MyName\". • Click **OK**. The system displays the SIP manipulation / Mime sdp rule / Sdp session rule dialog. • (Optional) Add another sdp-line-rule. • Click **OK**. The system displays the SIP manipulation / Mime sdp rule dialog. Do one of the following: • Add another sdp-session-rule. • Configure the mime-header-rule and sdp-media-rule options, using the steps in the corresponding table cells in this procedure. • Finish the MIME SDP rule configuration by completing steps 3-6.
Cfg Rules (instructions for configuring sdp-media-rule)	(Optional) Click **Add**, **sdp-media-rule**. • Name—Enter a unique name for this header element rule. • Media Type—Enter the media type to manipulate. For example, audio or video. • Action—Select an action from the drop-down list to apply to the element rule. • Comparison Type—Select the type of comparison from the drop-down list to use for the match value. • Match Value—Enter the match value to compare against the current object. • New Value—Enter a new value for the object. Quoted display named must be escaped within quotes. For example, "\"MyName\" " . • Click **OK**.

- CfgRules—(Optional) Click **Add**, **sdp-line-rule**.
- Name—Enter a unique name for this rule.
- Type—Enter a descriptor type to specify the SDP line to manipulate.
- Action—Select an action from the drop-down list to apply to this rule.
- Comparison type—Select the type of comparison from the drop-down list to use for the match value. (To clear the value, enter and empty string.)
- Match Value—Enter the match value to compare against the current object. (To clear the value, enter and empty string.)
- New Value—Enter a new value for the object. Quoted display named must be escaped within quotes. For example, \"MyName\".
- Click **OK**. The system displays the SIP manipulation / Mime sdp rule / Sdp media rule dialog.
- (Optional) Add another sdp-line-rule.
- Click **OK**. The system displays the SIP manipulation / Mime sdp rule dialog.

Do one of the following:

- Add another sdp-media-rule.
- Finish the MIME SDP rule configuration by completing the following steps.

6. Click **OK**.
7. Click **OK**.
8. Save the configuration.

16

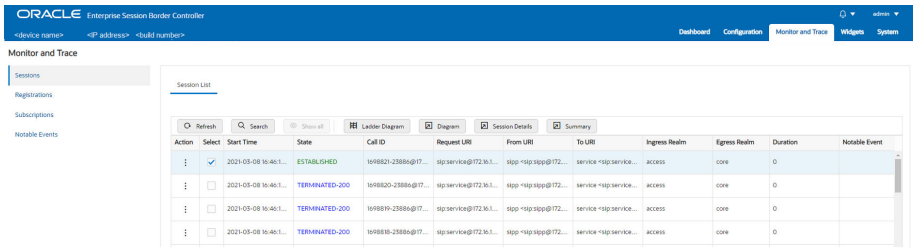
Monitor/Trace and Widgets Tab Operations

The Monitor and Trace tab displays links to tools that help you see the results of system and session data on the Oracle Enterprise Communications Broker (Communications Broker). The Monitor and Trace link provides summary reports that include session data, ladder diagrams of call and media flows, and Quality of Service statistics. The Widgets tab leads to dozens of Widgets where you can view graphical displays of Communications Broker activity.

Monitor and Trace

The Monitor and Trace tab displays the results of filtered SIP session data from the Oracle Enterprise Communications Broker in categorical lists. Each of the lists displays the results in a common log format for local viewing, which you can export as HTML and text files.

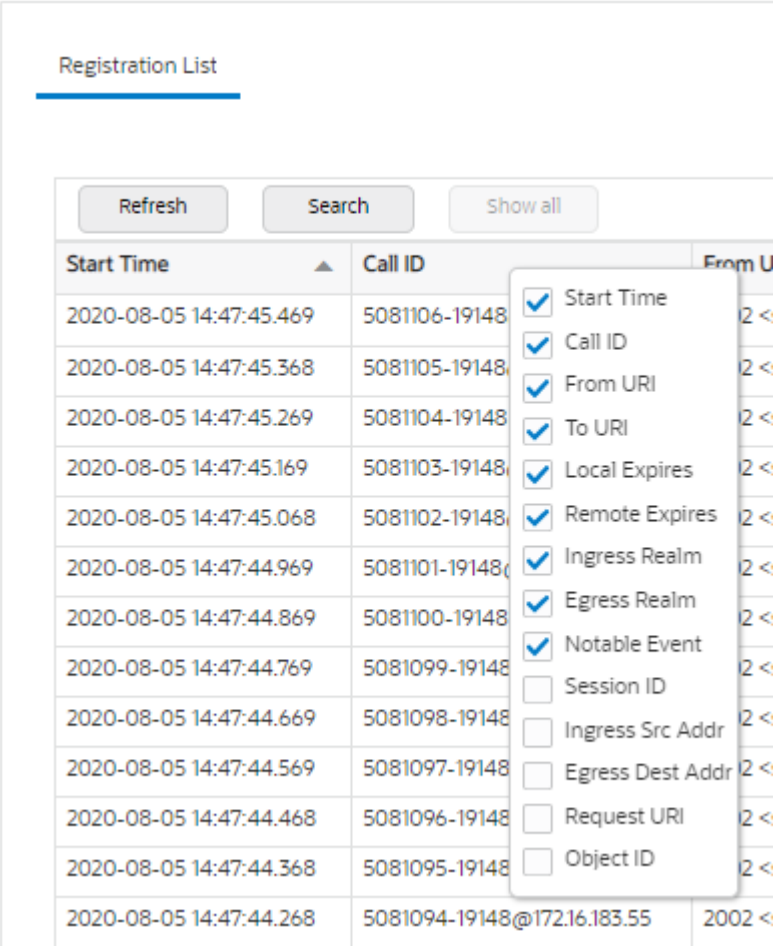
When you click the Monitor and Trace tab, the Web GUI displays links to the available lists in the navigation pane and defaults to Sessions. In the white space below the list, the GUI displays the total number of entries in the list and the numbers of the entries currently in view. The counter is dynamic and changes the enumeration as you scroll through the list. The following screen capture shows the Monitor and Trace default display.



Monitor and Trace supports the following summary reports.

- SIP Sessions Summary
- SIP Registrations Summary
- SIP Subscriptions Summary
- SIP Notable Events Summary

Each summary provides sorting, searching, paging, and exporting functionality, as well as a ladder diagram view where you can see a session summary, session details, and QoS statistics. You can choose which columns that you want each summary list to display by right clicking any column header and selecting column names from the pop-up list. The available column names vary according to the list type.



Monitor and Trace can store messages per session and it can store cumulative sessions across all report types. When the sessions maximum is reached, the system removes the oldest call and adds the newest call.

- On systems with less than 4GB of RAM, the system can store:
  - 50 messages
  - 2,000 sessions
- On systems with more than 4GB of RAM, the system can store:
  - 50 messages
  - 4,000 sessions

The call database does not persist when you change the number of rows in a Ladder Diagram and Save and Activate the change.

The system can perform live paging from Monitor and Trace tables.

> **Note:**
>
> Only one user at a time can view Monitor and Trace information. Monitor and Trace does not support multiple, simultaneous viewers.

SIP Notable Events Summary

The SIP Notable Events Summary contains all logged sessions that have a notable event on the Oracle Enterprise Communications Broker (Communications Broker) associated with the session. The columns that display on the Notable Events Summary page depend on the columns that you selected in the "Customizing the Page Display" procedure.

The following table describes the columns that Notable Events Summary page can display.

Start Time	Timestamp of the first SIP message in the call session.
State	Status of the call or media event session. Valid values: INITIAL—Session for which an INVITE or SUBSCRIBE was forwarded. EARLY—Session received the first provisional response (1xx other than 100). ESTABLISHED—Session for which a success (2xx) response was received. TERMINATED—Session that has ended by receiving or sending a BYE for an “Established” session or forwarding an error response for an “Initial” or Early session. The session remains in the terminated state until all the resources for the session are freed up. FAILED Session that has failed due to a 4xx or 5xx error code.
Call ID	Identification of the call source. Includes the phone number and source IP address.
Request URI	Uniform Resource Identifier (URI) formatted string that identifies a resource through a protocol, name, location, and any other applicable characteristic, and is sent by the Communications Broker in REQUEST headers.
From URI	URI formatted string that identifies the call source information.
To URI	URI formatted string that identifies the call destination information.
Ingress Realm	Name of the inbound realm.
Egress Realm	Name of the outbound realm.
Notable Event	Indicates if a notable event has occurred on the call session. Valid values: short session—Sessions that don’t meet a minimum configurable duration threshold; Session dialogue, captured media information and termination signalling; Any event flagged as a short session interesting event. local rejection—Sessions locally rejected at the Communications Broker for any reason (for example, Session Agent (SA) unavailable, no route found, SIP signalling error, etc.); Session dialogue, capture media information and termination signaling; Any event flagged as a local rejection interesting event.
Session ID	Identification assigned to the call session.
Ingress Src Addr	Source IP address of the incoming call or media event.
Ingress Src Port	Source port of the incoming call or media event.

Egress Dest Addr	Destination IP address of the outgoing call or media event.
Object ID	ID number of the object in a row. Use to aid troubleshooting.

Display a Notable Events Report

1. Access Notable Events: , **Monitor and Trace** tab, **Notable Events**.
2. Use the controls on the page to view information about the records in this report.

SIP Registrations Summary

The SIP Registrations Summary displays a summary of all logged SIP registrations sessions on the Oracle Enterprise Communications Broker (Communications Broker). The columns that display on the Registrations Summary page depend on the columns you selected in the "Customizing the Page Display" procedure.

The following table describes the columns available on the Registrations Summary page.

Start Time	Timestamp of the first SIP message in the call session.
Call ID	Identification of the call source. Includes the phone number and source IP address.
From URI	URI formatted string that identifies the call source information.
To URI	URI formatted string that identifies the call destination information.
Local Expires	The current setting for the expiration of a registration request sent from the Integrated Media Gateway (IMG) to a Remote SIP User Agent. Default: 3600 seconds.
Remote Expires	The current setting for the expiration of a registration request sent from the Remote SIP User Agent to the Integrated Media Gateway (IMG). Default: 3600 seconds.
Ingress Realm	Incoming realm name.
Egress Realm	Outgoing realm name.
Notable Event	Indicates a notable event that occurred on the call session. Valid value: local rejection - Sessions locally rejected at the E-SBC for any reason (for example, Session Agent (SA) unavailable, no route found, SIP signalling error, etc.); Session dialogue, capture media information and termination signalling; Any event flagged as a local rejection interesting event
Session ID	Identification assigned to the call session.
Ingress Src Addr	Source IP address of the incoming call or media event.
Egress Dest Addr	Destination IP address of the outgoing call or media event.
Request URI	Uniform Resource Identifier (URI) formatted string that identifies a resource with the protocol, name, location, and any other applicable characteristic. The SBC only sends the URI in REQUEST headers.
Object ID	ID number of the object in a row. Use to aid troubleshooting.

Display a Registrations Report

1. Access Registrations: **Monitor and Trace** tab, **Registrations**.
2. Use the controls on the page to view information about the records in this report.

SIP Sessions Summary

The SIP Sessions Summary summarizes all logged call sessions on the Oracle Enterprise Communications Broker (Communications Broker). The columns that display on the Sessions Summary page depend on the columns that you specified in the "Customizing the Page Display" procedure.

The following table describes the columns on the SIP Session Summary page.

Start Time	Timestamps of the first SIP message in the call session.
State	Status of the call or media session. Valid values are: INITIAL—Session for which an INVITE or SUBSCRIBE was forwarded. EARLY—Session that received the first provisional response (1xx other than 100). ESTABLISHED—Session for which a success (2xx) response was received. TERMINATED—Session that ended by receiving or sending a BYE for an “Established” session or forwarding an error response for an “Initial” or "Early" session. The session remains in the terminated state until all the resources for the session are freed up. FAILED—Session that failed due to a 4xx or 5xx error code.
Call ID	Identification of the call source. Includes the phone number and source IP address.
Request URI	Uniform Resource Identifier (URI) formatted string that identifies a resource by way of a protocol, name, location, and any other applicable characteristic that is sent by the Communications Broker in REQUEST headers.
From URI	URI formatted string that identifies the call source information.
To URI	URI formatted string that identifies the call destination information.
Ingress Realm	Name of the inbound realm.
Egress Realm	Name of the outbound realm.
Duration	Amount of time, in seconds, that the call or media event was active.
Notable Event	Indicates if a notable event has occurred on the call session. Valid values are: Short Session—Sessions that do not meet a minimum configurable duration threshold. Session dialogue, captured media information, and termination signaling. Any event flagged as a short session interesting event. Local Rejection—Sessions locally rejected at the Communications Broker for any reason, for example, Session Agent (SA) unavailable, no route

	found, SIP signaling error, and so on. Session dialogue, capture media information, and termination signaling. Any event flagged as a local rejection interesting event.
Session ID	Identification assigned to the call session.
Ingress Src Addr	Source IP address of the incoming call or media event.
Egress Dest Addr	Destination IP address of the outgoing call or media event.
Calling Pkts	Number of calling packets.
Called Pkts	Number of called packets.
Calling R	Calling packets R-Factor calculation.
Called R	Called packets R-Factor calculation.
Calling MOS	Calling packets Mean Opinion Score (MOS) calculation.
Called MOS	Called packets Mean Opinion Score (MOS) calculation.
Ingress Src Port	Source port of the incoming call or media event.
Object ID	ID number of the object in a row. Use to aid troubleshooting.

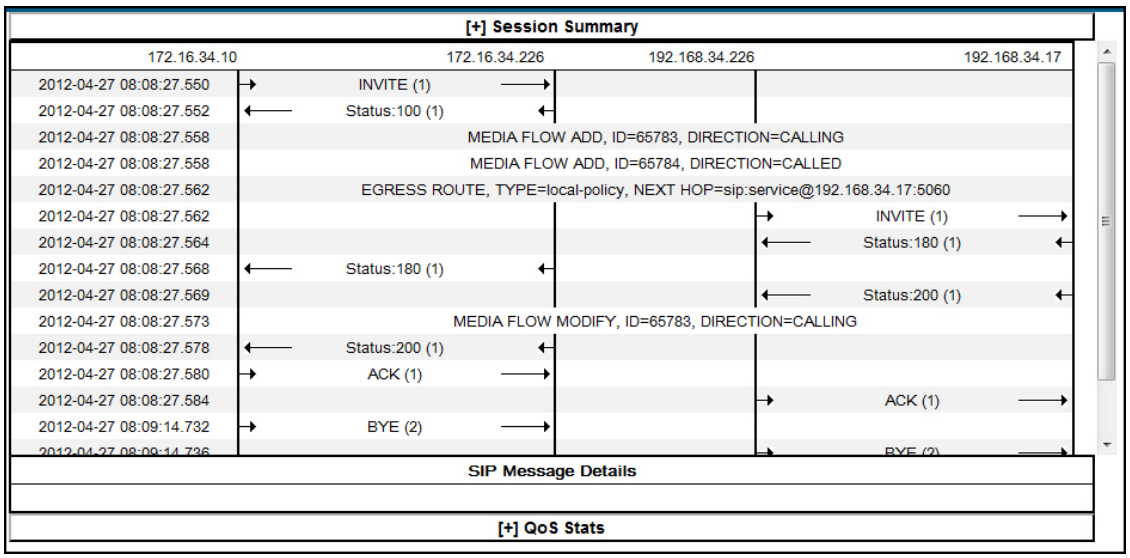
Display a Sessions Report

1. Access SIP Session Summary: **Monitor and Trace** tab, **Sessions**.

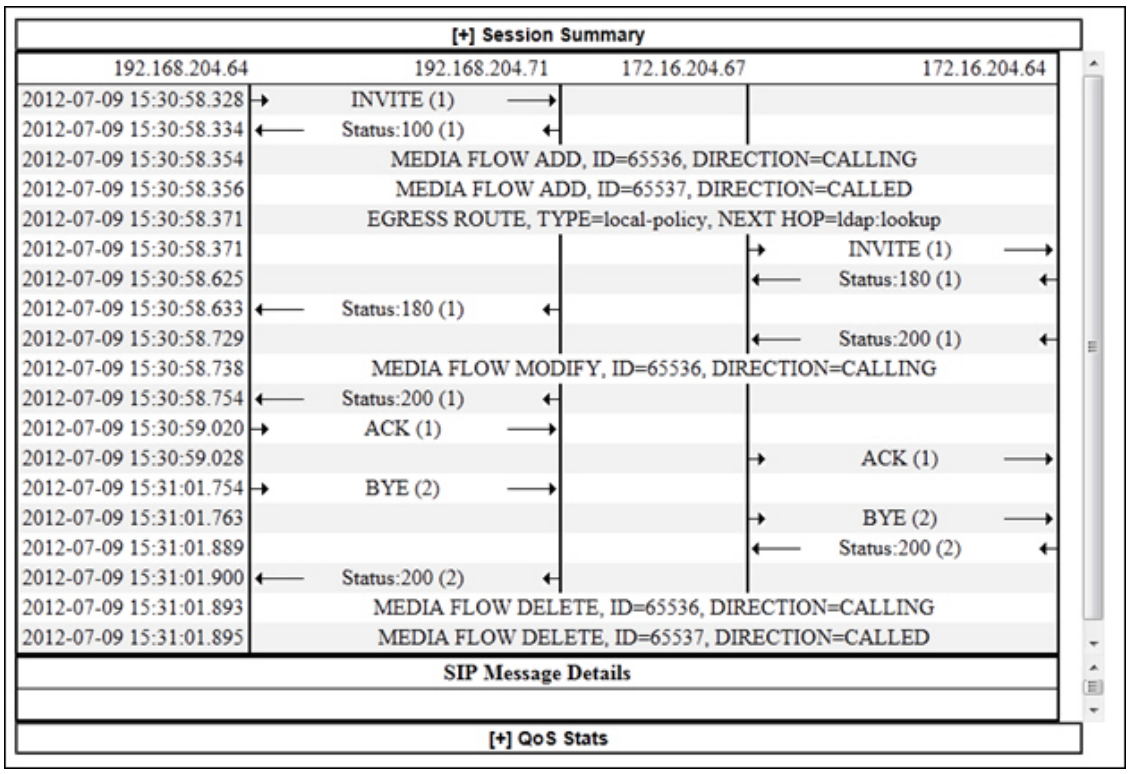
   The system displays the SIP Sessions Summary page.
2. Use the controls on the page to manage the records in the report.

SIP Message Details

The SIP Message Detail window displays detailed information and data flow (ingress and egress) about the call or media event.



When a session is routed using the a Lightweight Directory Access Protocol (LDAP) configuration (Active Directory) for the local policy, the LDAP information displays in the Session Summary window. The next hop value containing "enum:…" or "dns:…" displays. Similarly, the next hop value "ldap:…" displays for LDAP queries.



SIP Subscriptions Summary

The SIP Subscriptions Report displays a summary of all logged SIP subscription sessions on the Oracle Enterprise Communications Broker (Communications Broker). The columns that display on the Subscription Report page depend on the columns you selected in the "Customizing the Page Display" procedure.

The following table describes the columns on the Subscriptions Report page.

Start Time	Timestamps of the first SIP message in the call session.
Call ID	Identification of the call source. Includes the phone number and source IP address.
From URI	URI formatted string that identifies the call source information.
To URI	URI formatted string that identifies the call destination information.
Events	Specific subscribe event package that was sent from an endpoint to the destination endpoint. Applicable event packages include the following: Conference—Event package that allows users to subscribe to a conference Uniform Resource Identifier (URI). Consent—pending additions - Event package used by SIP relays to inform user agents about the consent-related status of the entries to be added to a resource list.

	Dialog—Event package that allows users to subscribe to another user, and receive notifications about the changes in the state of the INVITE-initiated dialogs in which the user is involved. Message—summary - Event package that carries message-waiting status and message summaries from a messaging system to an interested User Agent (UA). Presence—Event package that conveys the ability and willingness of a user to communicate across a set of devices. A presence protocol is a protocol for providing a presence service over the Internet or any IP network. Reg—Event package that provides a way to monitor the status of *all* the registrations for a particular Address of Record (AoR). Refer—Event package that provides a mechanism to allow the party sending the REFER to be notified of the outcome of a referenced request. Winfo—Event package for watcher information. It tracks the state of subscriptions to a resource in another package. Vvq-rtcpx—Event package that collects and reports the metrics that measure quality for RTP sessions.
Expires	The current setting for the expiration of a registration request. Default: 3600 sec.
Ingress Realm	Incoming realm name.
Egress Realm	Outgoing realm name.
Notable Event	Indicates that a notable event occurred on the call session. Valid value is: local rejection - Sessions locally rejected at the Communications Broker for any reason (for example, Session Agent (SA) unavailable, no route found, SIP signaling error); Session dialogue, capture media information and termination signaling; Any event flagged as a local rejection interesting event
Session ID	Identification assigned to the call session.
State	Status of the call or media session. Valid values are: INITIAL—Session for which an INVITE or SUBSCRIBE was forwarded. EARLY—Session that received the first provisional response (1xx other than 100). ESTABLISHED—Session for which a success (2xx) response was received. TERMINATED—Session that ended by receiving or sending a BYE for an “Established” session or forwarding an error response for an “Initial” or "Early" session. The session remains in the terminated state until all the resources for the session are freed up. FAILED—Session unsuccessful due to a 4xx or 5xx error code.
Ingress Src Addr	Source IP address of the incoming call or media event.

Egress Dest Addr	Destination IP address of the outgoing call or media event.
Request URI	Uniform Resource Identifier (URI) formatted string that identifies a resource by way of a protocol, name, location, and any other applicable characteristic, and is sent by the Communications Broker in REQUEST headers.
Object ID	ID number of the object in a row. Use to aid troubleshooting.

Display a Subscriptions Report

1. Access Subscriptions: **Monitor and Trace** tab, **Subscriptions**.
2. Use the controls on the page to view information about the records in this report.

Ladder Diagrams and Display Controls

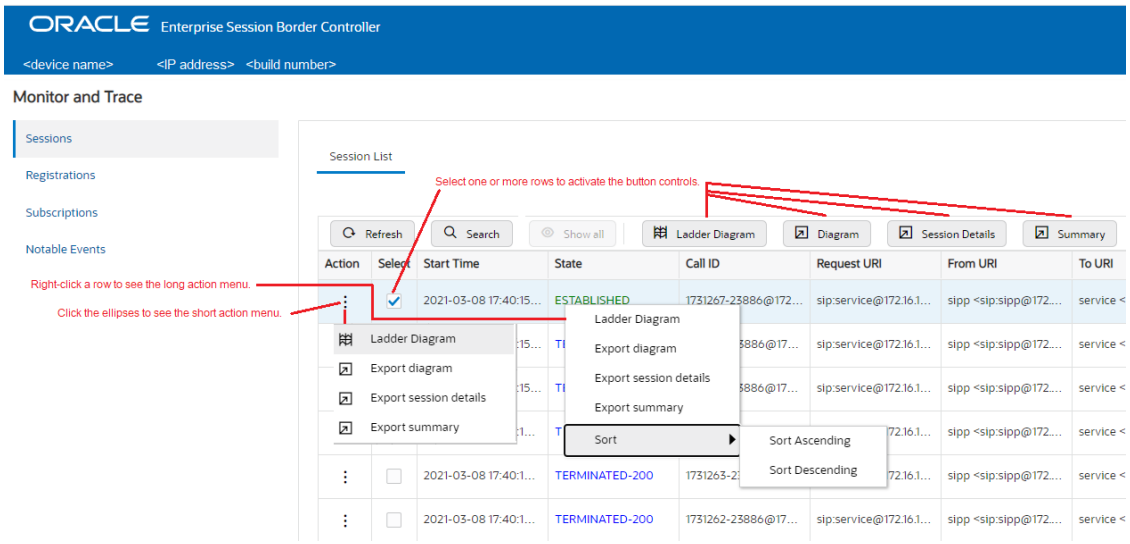
A ladder diagram is a graphical representation of the flow of call and media packets on ingress and egress routes through the Oracle Enterprise Communications Broker (Communications Broker). Viewing ladder diagrams can help you with troubleshooting and system monitoring. The Web GUI can display ladder diagrams for each of the summary reports available through Monitor and Trace.

The Communications Broker can store up to 4,000 calls on a system with more than 4GB of memory and up to 2,000 calls on a system with less than 4GB of memory. A ladder diagram can display an unlimited number of rows, but the relationship of calls to rows is inverse. The more rows you see per call, the fewer calls the Communications Broker can display. You cannot set the number of rows displayed.

To display one or more ladder diagrams, select one or more rows in the Session, Registrations, Subscriptions, or Notable Events lists and click **Ladder Diagram**. The Web GUI adds a tab to the top of the page for each ladder diagram. Click the tabs to see the ladder diagrams.

> **Note:**
>
> You can also right-click a row and click **Ladder Diagram** on the pop-up menu or click the ellipses at the beginning of a row and click **Ladder Diagram** on the pop-up menu.

The following illustration shows a sample ladder diagram with notations about the controls for viewing and exporting.



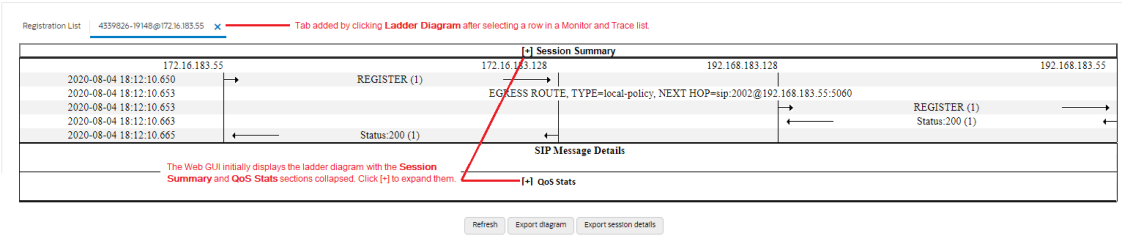
Ladder diagrams contain the following sections.

Session Summary—Displays session data. Use the [+] and [-] controls to toggle between show and hide. The default is hide.

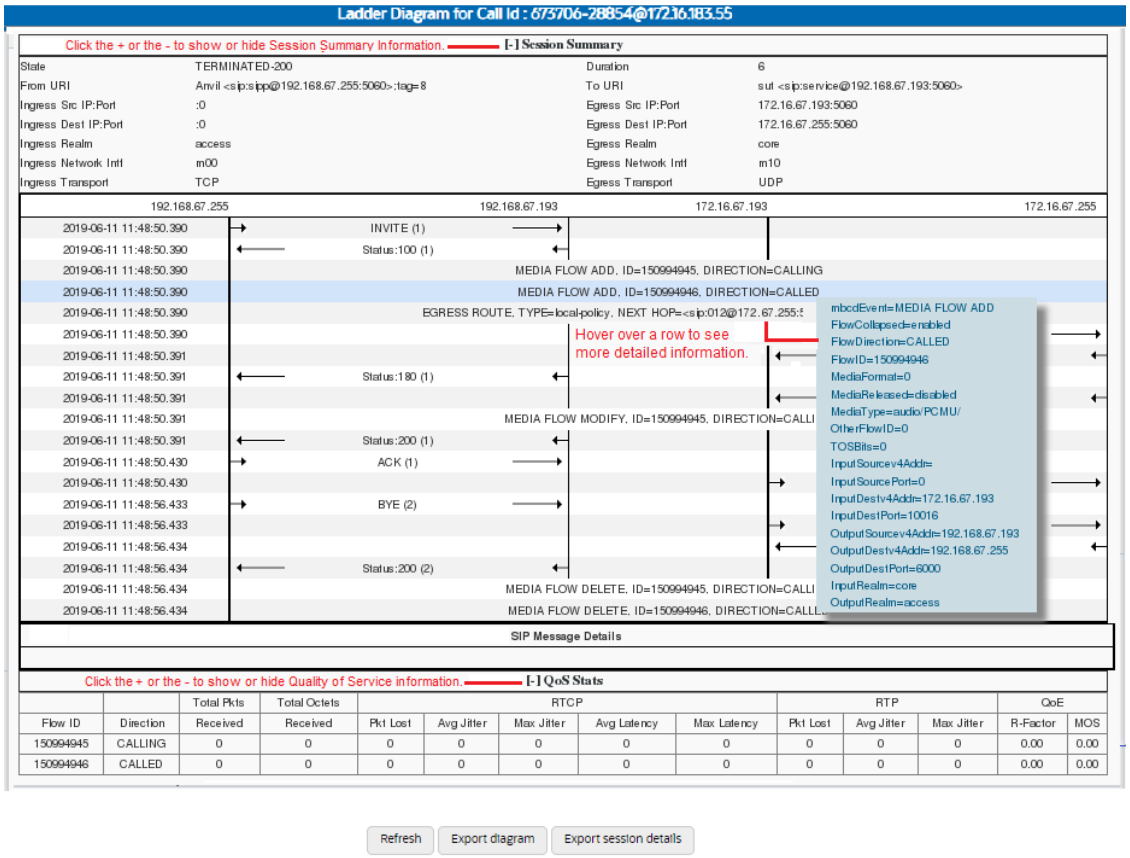
Ladder Diagram—Displays SIP message and call flow information. Hover over a line in the ladder diagram to see more information about the highlighted flow.

QoS Stats—Displays Quality of Service (QoS) information. Use the [+] and [-] controls to toggle between show and hide. The default is hide. You must enable Quality of Service (QoS) on your system to see the QoS Statistics information.

The following screen capture shows an example of a ladder diagram as initially displayed with the Session Summary and QoS Stats sections hidden.



The following screen capture shows an example of an Communications Broker ladder diagram with the Session Summary and QoS sections expanded. This screen capture also shows an example of the pop-up that displays detailed SIP Message Details when you hover over a row in the diagram.



> **Note:**
>
> The Communications Broker captures SIP messages, applies the Header Manipulation Rules (HMR) that you configured on the Communications Broker, and applies the Session Plug-in Language (SPL) to that message. When the Communications Broker sends the message, it applies the SPL, the HMR, and sends the captured SIP message. When viewing the session detail on a Ladder Diagram, the HMR and SPL information may be present.

Ladder Diagram Controls

Ladder diagrams display the following controls at the bottom of the page.

Refresh—Use to refresh the data in the ladder diagram.

Export diagram—Use to export the ladder diagram to an HTML file.

Export session details—Use to export the session details to a text file.

Close and Remove Ladder Diagram Tabs

To remove an individual tab you can either click the X on the tab or right-click and click **Remove** on the pop-up menu.

To remove all tabs, right-click a tab and click **Close All** on the pop-up menu.

Display a Ladder Diagram

You can display a ladder diagram of call and media flow data from the Sessions, Registrations, Subscriptions, and Notable events lists in Monitor and Trace. Each list page displays a table, where each row represents one session. You can view a ladder diagram for each session with the following procedure.

The **Monitor and Trace** tab can display multiple ladder diagrams in separate tabs. You can either select multiple rows from the list initially or repeat the procedure later to display more ladder diagram tabs.

1. Access the **Monitor and Trace** tab.
2. On the **Monitor and Trace** tab, select one of the following lists:
   - Sessions
   - Registrations
   - Subscriptions
   - Notable Events
3. On the list page, do one of the following :
   - Double click a row.
   - Select one or more rows and click **Ladder Diagram**.

   The Web GUI adds one or more tabs above the list that you can click to view the contents.
4. (Optional)—On the ladder diagram, click the [+] to expand the Session Summary and QoS Stats sections.
5. (Optional)—Refresh the diagram, export the diagram, or export the session details, using the buttons at the bottom of the page.

Session Summary

To see a Session Summary, open a ladder diagram from a record in a Summary report and click the [+] control that precedes "Session Summary" at the top of the page. Monitor and Trace displays the session data and statistics.

The following screen capture shows a sample Session Summary page generated from a selected item on a ladder diagram.

[-] Session Summary

State	TERMINATED-200	Duration	47
From URI	sipp <sip:sipp@100.10.30.10:5060>;tag=24	To URI	sut <sip:service@100.10.30.226:5060>;tag=99
Ingress Src IP:Port	100.10.30.10:5060	Egress Src IP:Port	200.20.40.226:5060
Ingress Dest IP:Port	100.10.30.226:5060	Egress Dest IP:Port	200.20.40.17:5060
Ingress Realm	access	Egress Realm	backbone
Ingress Network Intf	access	Egress Network Intf	backbone
Ingress Transport	UDP	Egress Transport	UDP

The following table describes each field in the Session Summary report.

State	Status of the call or media session. Valid values are: INITIAL—Session for which an INVITE or SUBSCRIBE was forwarded.

	EARLY—Session received the first provisional response (1xx other than 100). ESTABLISHED—Session for which a success (2xx) response was received. TERMINATED—Session that has ended by receiving or sending a BYE for an “Established” session or forwarding an error response for an “Initial” or Early session. The session remains in the terminated state until all the resources for the session are freed up. FAILED—Session that has failed due to a 4xx or 5xx error code.
Duration	Amount of time, in seconds, that the call or media session was active.
From URI	URI formatted string that identifies the call source information.
To URI	URI formatted string that identifies the call destination information.
Ingress Src IP:Port	Source IP address and port number of the incoming call or media session.
Egress Src IP: Port	Source IP address and port number of the outgoing call or media session.
Ingress Dest IP:Port	Destination IP address and port number of the incoming call or media session.
Egress Dest IP: Port	Destination IP address and port number of the outgoing call or media session.
Ingress Realm	Incoming realm name.
Egress Realm	Outgoing realm name.
Ingress Network Intf	Name of the incoming network interface on the Oracle Enterprise Communications Broker (Communications Broker).
Egress Network Intf	Name of the outgoing network interface on the Communications Broker.
Ingress Transport	Protocol type used on the incoming call or media session. Valid values are User Datagram Protocol (UDP) or Transport Control Protocol (TCP).
Egress Transport	Protocol type used on the outgoing call or media session. Valid values are User Datagram Protocol (UDP) or Transport Control Protocol (TCP).

Display a Session Summary

You can view data and statistics about a call or media session by displaying the Session Summary from a ladder diagram.

1. Under the **Monitor and Trace** tab, select one of the following summaries:
   - Sessions
   - Registrations
   - Subscriptions
   - Notable Events

2. On the Summary page, select a row in the table and right-click.
3. On the pop-up menu, click Ladder Diagram.
4. (Optional)—On the ladder diagram, click the [+] to expand the Session Summary section.

QoS Statistics

The Quality of Service (QoS) Stats section of the Session Summary displays information about the quality of the service for a selected call session or media event. To see the QoS statistics, open a ladder diagram from a record in a Summary report and click the [+] control that precedes "QoS Statistics" at the bottom of the page.

Expand QoS Stats section with the [+} control.

[-] QoS Stats

		Total Pkts	Total Octets	RTCP					RTP			QoE	
Flow ID	Direction	Received	Received	Pkt Lost	Avg Jitter	Max Jitter	Avg Latency	Max Latency	Pkt Lost	Avg Jitter	Max Jitter	R-Factor	MOS
100663297	CALLING	0	0	0	0	0	0	0	0	0	0	0.00	0.00
100663298	CALLED	0	0	0	0	0	0	0	0	0	0	0.00	0.00

The following table describes each column in the QoS Stats report.

Flow ID	ID number assigned to the call session or media event flow of data.
Direction	The direction of the call or media event flow. CALLING—egress direction CALLED—ingress direction
Total Pkts Received	Total number of data packets received on the interface during the active call session or media event.
Total Octets Received	Total number of octets received on the interface during the active call session or media event.
RTCP	Real-time Transport Control Protocol—used to send control packets to participants in a call.
Pkts Lost	Number of RTCP data packets lost on the interface during the active call session or media event.
Avg Jitter	Average measure of the variability, called jitter, over time of the RTCP packet latency across a network. A network with constant latency has no jitter. Jitter is referred to as Packet Delay Variation (PDV). It is the difference in the one-way end-to-end delay values for packets of a flow. Jitter is measured in terms of a time deviation from the nominal packet inter-arrival times for successive packets.
Max Jitter	Maximum measure of the variability, called jitter, over time of the RTCP packet latency across a network. A network with constant latency has no variation jitter.
Avg Latency	Average observed one-way signaling latency during the active window period. This is the average amount of time the signaling travels in one direction.

Max Latency	Maximum observed one-way signaling latency during the sliding window period. This is the maximum amount of time the signaling travels in one direction.
RTP	Real-Time Transport Protocol—a standard packet format for delivering audio and video over the internet.
Pkts Lost	Number of RTP data packets lost on the interface during the active call session or media event.
Avg Jitter	Average measure of the variability, called jitter, over time of the RTP packet latency across a network. A network with constant latency has no jitter. Jitter is referred to as Packet Delay Variation (PDV). It is the difference in the one-way end-to-end delay values for packets of a flow. Jitter is measured in terms of a time deviation from the nominal packet inter-arrival times for successive packets.
Max Jitter	Maximum measure of the variability, called jitter, over the time of the RTP packet latency across a network. A network with constant latency has no jitter.
QoE	Quality of Experience—measurement used to determine how well the network is satisfying the end user's requirements.
R-Factor	Rating Factor—An average Quality of Service (QoS) factor observed during the active window period. QoS shapes traffic to provide different priority and level of performance to different data flows. R-Factors are metrics in VoIP that use a formula to take into account both user perceptions and the cumulative effect of equipment impairments to arrive at a numeric expression of voice quality. This statistic defines the call or transmission quality, which is expressed as an R factor.
MOS	Mean Opinion Score (MOS) score—MOS is a measure of voice quality. MOS provides a numerical indication of the perceived quality of the media received after being transmitted and eventually compressed using Codecs.

Display QoS Statistics

When you want to view QoS statistics for a call or media flow session, you can do so from a ladder diagram in **Monitor and Trace**.

1. Under the **Monitor and Trace** tab, select one of the following summaries:
   - Sessions
   - Registrations
   - Subscriptions
   - Notable Events
2. On the Summary page, select a row in the table and right-click.
3. On the pop-up menu, click Ladder Diagram.
4. On the ladder diagram, click the [+] to expand the **QoS Stats** section.

SIP Monitor and Trace Filter Configuration

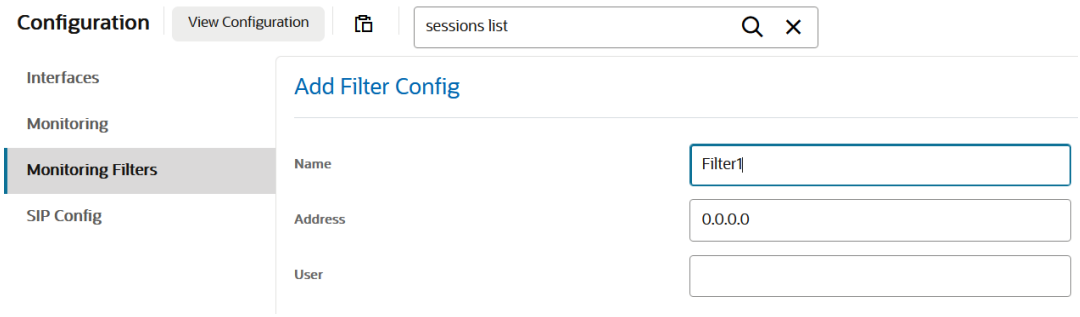
The SIP Monitor and Trace function allows you to monitor SIP sessions for notable events and display the results in the Oracle Enterprise Communications Broker (Communications Broker) SIP Notable Events summary. Such information may help you perform troubleshooting. For

more targeted monitoring, you can configure filters on particular users and addresses on the Communications Broker, and on a specific agent.

The Communications Broker Configuration page, located on the **Configuration** tab, **System Administration** section, **SIP Interface**, **Monitoring Filters**, includes the following objects for configuring SIP Monitoring filters:

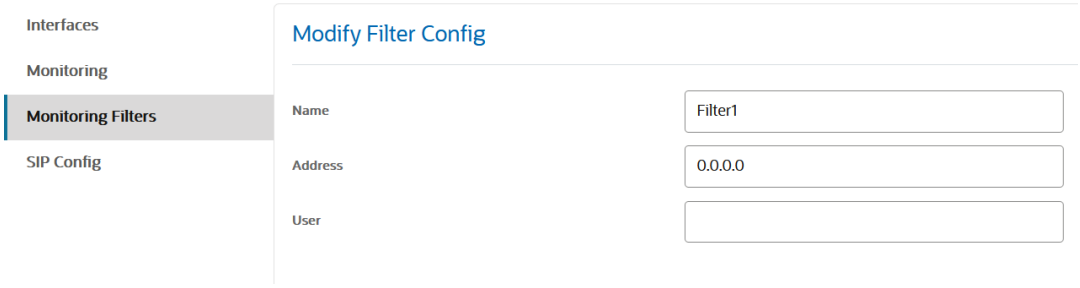
- The SIP Interface configuration page displays the **Monitoring Filters** object in the navigation pane, which you use to configure individual filters.

Figure 16-1 filter config dialog



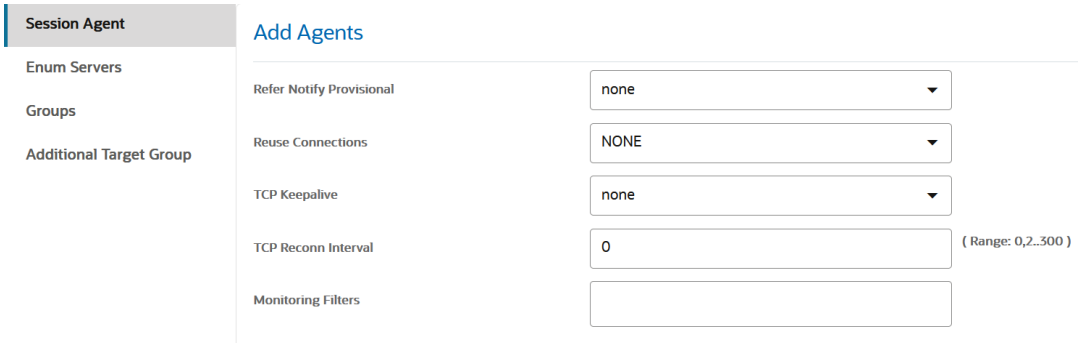
- The **Monitoring** object on the SIP interface configuration page displays the **Monitoring Filters** element in the dialog. Use it to apply filters to the Communications Broker.

Figure 16-2 Modify Filter Config



- The Add Agents configuration page displays the **Monitoring Filters** configuration element to the Advanced section. Use it to apply filters to an agent.

Figure 16-3 Monitoring Filters



- **Note:**

  After the P-CZ2.0.0m4 release, the system does not support the former "Enable SIP Monitor and Trace" setting. You must re-configure SNMP event traps through the dialogs described in this topic.

Use the following filter configuration process for both new installations and upgrades.

1. Create one or more filters in the Monitoring Filters object. You may use an asterisk character as a filter, if you want to monitor all session data.
2. Add one or more filters to the Monitoring object.
3. (Optional) Add one or more monitoring filters to an agent that you want to monitor.

Search for a Record

The **Search** button at the top of the report page allows you to find a specific record within a Monitor and Trace report. It also allows you to specify criteria on which to perform the search.

After defining a search criteria in the Search Filter dialog box, clicking Search automatically populates the report page with the records that match the specified criteria specified. The search performs the filtering process of criteria dependent on the report page from which you are running the search.

For example, performing a search from the Sessions report page displays only the reports pertaining to call sessions. If you perform a search on the Registration report page, only the reports pertaining to call registrations displays on the report page. The search string containing the criteria on which you performed the search, displays in the top left corner of the page.

Note:

A SIP Monitor and Trace global search can find items in the SIP headers, as well. The system saves the search criteria until you click **Reset** in the dialog box, or until you log out of the HTTP session.

Search for a Report Record

Use the Search button located above the report table to help you find a specific record in an Oracle Enterprise Communications Broker (Communications Broker) Monitor and Trace report. When you click Search, the Web GUI displays a dialog where you specify criteria for the search.

You can specify a value for any or all of the fields in the Search field. The search process searches for records with all of the values you specify and displays only the records with all those values. When you perform a Global Search and specify values in other fields, the search process searches the other specified fields first and then filters on the Global Search field.

- If you specify the “*” character in a search string, the search is performed on that exact string. For example, if you search for “123*45”, the search shows results for all strings containing “123*45”.
- You can use quotes (“ “) to specify a search. For example, you can enter Smith and the search finds all of the records that match Smith, such as: John **Smith** field<sip:sipp@192.168.1.70:5070>;tag=12260SIPpTag001.

- When you enter a space before or after a quotation mark, (for example, “Smith “), the search returns no data.

1. On any reports page, click **Search** and do the following in the Search dialog.

Global Search—Search all parameters in all records.	Enter the URI formatted string of the call source information you are searching. Valid values: alpha-numeric characters. For example, sipp<sip:sipp@172.16.34.10:5060;tag=24.
From URI—Search on the From-URI header.	Enter the URI formatted string that contains a protocol, name, location, or any other applicable characteristic, that is sent by the Communications Broker in the FROM header. Valid values: alpha-numeric characters. For example, sip:service@172.16.34.226:5060.
Request URI—Search on the Request-URI header.	Enter the URI formatted string that contains a protocol, name, location, or any other applicable characteristic, that is sent by the Communications Broker in the REQUEST header. Valid values: alpha-numeric characters. For example, sip:service@172.16.34.226:5060.
To URI—Search on the To-URI header.	Enter the URI formatted string of the call destination information you want to find. Valid values: alpha-numeric characters. For example, sut<sip:service@172.16.34.226:5060;tag=99.
Start Date— Search from messages that start at the specified date and time.	• Enter a starting date to search on in the first text box in the format YYYY-MM-DD (where Y =year, M=month, and D=day). For example, 2012-04-15 would search for all records ending on April 15, 2012. Valid values: numeric characters only. • Enter a start time to search on in the last three text boxes in the format HH mm ss (where H=hour, m=minutes, and s=seconds). For example, 01 30 45 would search for all records starting at 1:30 and 45 seconds. Valid values: numeric characters only.
End Date—Search from messages that end at the specified date and time.	• Enter an end date to search on in the first text box in the format YYYY-MM-DD (where Y =year, M=month, and D=day). For example, 2012-04-15 would search for all records starting on April 15, 2012. Valid values: numeric characters only. • Enter an end time to search on in the last three text boxes in the format HH mm ss (where H=hour, m=minutes, and s=seconds). For example, 01 30 45 would search for all records ending at 1:30 and 45 seconds. Valid values: numeric characters only.
Session ID—Search on the monitored SIP session ID, as shown in the Summary table.	Enter the ID of the call session you want to search. Valid values: alpha-numeric characters. For example, 22-3412@172.16.34.1.
In Call ID—Search on the SIP call ID of the initial received request.	Enter the ID of the incoming call (phone number and source IP address). Valid values: alpha-numeric characters. For example, 25-3412@172.16.34.10.

Out Call ID—Search on the SIP call ID of the first routed request.	Enter the ID of the outgoing call (phone number and IP address). Valid values: alpha-numeric characters. For example, 14-3412@172.14.54.6.
State (with result code)—Search on the state of the call, as shown in the Summary table.	Enter the status of the call session with the result code for which you want to search. Result codes can range from 1xx to 5xx. For example, terminated-200, or failed-400. Case-sensitive. Valid values: • INITIAL—<result code> • EARLY—<result code> • ESTABLISHED—<result code> • TERMINATED—<result code> • FAILED—<result code>
Notable Event—Search on a notable event type that you select from the drop-down list.	Select the notable event that you want to find. Valid values: • any event—Search displays any notable event that was stored in memory. • short session—Search displays only records that indicate a short-session duration occurred. • local rejection— Search displays only records that indicate a local-rejection occurred.
In Realm—Search the realm from which the initial request was received.	Enter the name of the realm to which the incoming call belongs. Valid values: alpha-numeric characters. For example, access.
Out realm—Search the realm to which the first routed request was sent.	Enter the name of the realm to which the outgoing call belongs. Valid values: alpha-numeric characters. For example, backbone.
Destination Agent—Search on the name of the session agent from which the initial request was received.	Enter the name of the session agent (SA) on the incoming call session. Valid values: alpha-numeric characters. For example, SA1.
Destination Session Agent—Search on the realm to which the first routed request was sent.	Enter the name of the session agent (SA) on the outgoing call session. Valid values: alpha-numeric characters. For example, SA2.
Ingress Destination Address—Search on the IP address from which the initial request was received.	Enter the source IP address of the SA that accepted the incoming call session. You must enter the IP Address in dotted decimal format (0.0.0.0). For example, 172.45.6.7.

Egress Destination Address—Search on the IP address to which the initial request was sent.	Enter the destination IP address of the SA that accepted the outgoing call session. You must enter the IP Address in dotted decimal format (0.0.0.0). For example, 172.64.56.7.
In Network Interface —Search on the IP address on which the initial*f* request was received.	Enter the incoming core network interface that connects the Communications Broker to your network. You must enter the IP Address in dotted decimal format (0.0.0.0). For example, 192.45.6.7.
Out Network Interface—Search on the IP address that was the source for the routed request.	Enter the outgoing network interface that connects theCommunications Broker to the outside network. You must enter the IP Address in dotted decimal format (0.0.0.0). For example, 192.45.6.8.

2. Click **Search**.

Specify Additional Identifiers

To specify additional identifiers:

1. In the Session Id field, enter the ID of the call session you want to search. Valid values are alpha-numeric characters. For example, 22-3412@172.16.34.1.
2. In the In Call ID field, enter the ID of the incoming call (phone number and source IP address). Valid values are alpha-numeric characters. For example, 25-3412@172.16.34.10.
3. In the Out Call ID field, enter the ID of the outgoing call (phone number and IP address). Valid values are alpha-numeric characters. For example, 14-3412@172.14.54.6.
4. In the State (with result code) field, enter the status of the call session with the result code for which you want to search. Valid values are (case-sensitive):
   - INITIAL-<result code>
   - EARLY-<result code>
   - ESTABLISHED-<result code>
   - TERMINATED-<result code>
   - FAILED-<result code>

   Result codes can range from 1xx to 5xx. For example, terminated-200, or failed-400.
5. In the Notable Event field, select the notable event for which you want to search. Valid values are:
   - any-event - search displays any notable event that was stored in memory.
   - short-session - search displays only records that indicate a short-session duration has occurred.
   - local-rejection - search displays only records that indicate a local-rejection has occurred.
6. To search on additional parameters, click on the Additional Search Options down arrow to expand the dialog box.

Specify Additional Search Options

To specify additional search options:

1. In the "**In Realm**" field, enter the name of the realm for which the incoming call belongs. Valid values are alpha-numeric characters. For example, access.
2. In the "**Out Realm**" field, enter the name of the realm for which the outgoing call belongs. Valid values are alpha-numeric characters. For example, backbone.
3. In the "**In SA**" field, enter the name of the session agent (SA) on the incoming call session. Valid values are alpha-numeric characters. For example, SA1.
4. In the "**Out SA**" field, enter the name of the session agent (SA) on the outgoing call session. Valid values are alpha-numeric characters. For example, SA2.
5. In the "**In Source Addr**" field, enter the source IP address of the SA that accepted the incoming call session. IP Address must be entered in dotted decimal format (0.0.0.0). For example, 172.45.6.7.
6. In the "**Out Dest Addr**" field, enter the destination IP address of the SA that accepted the outgoing call session. IP Address must be entered in dotted decimal format (0.0.0.0). For example, 172.64.56.7.
7. In the In Network Interface field, enter the incoming core network interface that connects the Communications Broker to your network. IP Address must be entered in dotted decimal format (0.0.0.0). For example, 192.45.6.7.
8. In the Out Network Interface field, enter the outgoing network interface that connects your Communications Broker to the outside network. IP Address must be entered in dotted decimal format (0.0.0.0). For example, 192.45.6.8.
9. Click <**Search**> to perform the search with the values you specified. A list of the records that the search process filtered, display in the window. The GUI saves the search specifications until you click <**Reset**> in the search dialog box, OR until you log out of the GUI.

Export Monitor and Trace Information to a Text File

Monitor and Trace allows you to export Monitor and Trace information to a text file from the Sessions, Registrations, Subscriptions, and Notable Events Reports.

To export report information, do one of the following:

- Select a row and click one of the export buttons displayed above the report list.
- Select a row, right click, and click an export operation on the menu.
- Click the ellipses on a row and click an export operation on the menu.

The following list describes the export commands.

Diagram	Exports all of the information in the Ladder Diagram (Session Summary, SIP Message Details, and QoS statistics), to an HTML file format.
Session Details	Exports detailed information about the SIP messages and media events in the Ladder Diagram associated with the selected session, to a file in text format.
Summary	Exports all logged session summary records to a file in text format.

Export Report Information to a Text File

To export information from a Monitor and Trace report to a text file:

> **Note:**
>
> The GUI exports Ladder Diagrams as HTML files.

1. Access the **Monitor and Trace** tab.
2. On the Monitor and Trace page, select a report type.
3. On the report Summary page for the selected report type, select a report from the list, right-click, and click one of the following:
   - Click **Export diagram**.
   - Click **Export session details**.
   - Click **Export summary**.

   The system downloads the report in a file at the bottom of your screen.

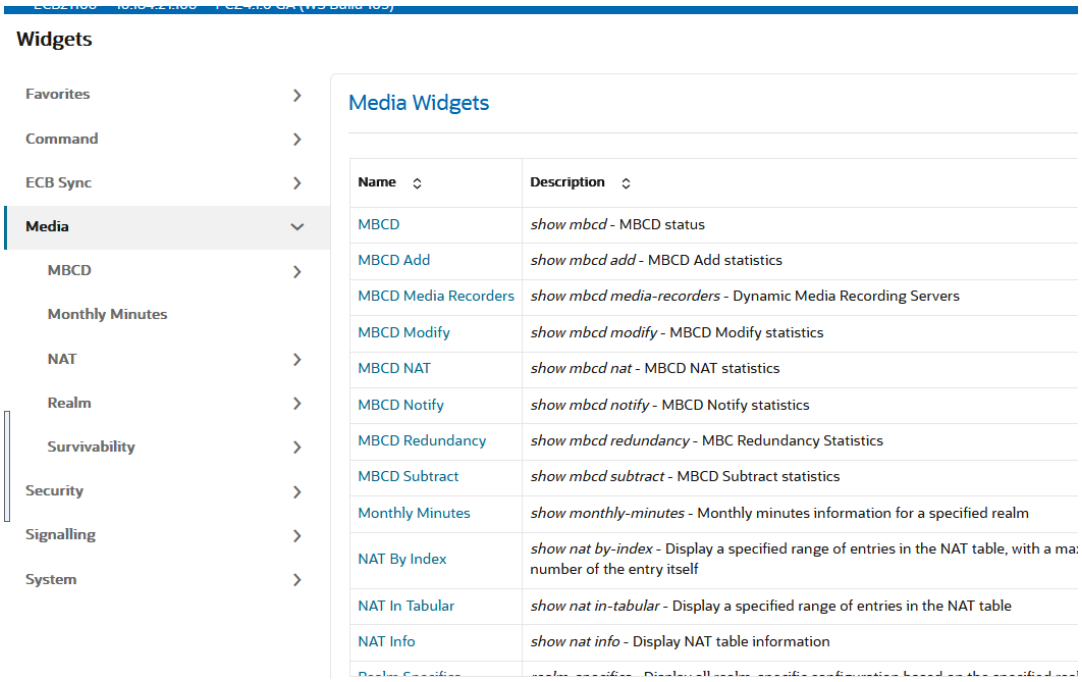
Widget Access and Behavior

In the navigation pane, the Widgets tab displays links to the Media, Security, Signaling, and System groups along with a link to your self-created Favorites list.

The Widgets tab provides two ways to navigate to the Widget you want to see.

- Click a group link in the navigation pane. The Web GUI lists all the Widgets in the group in the center pane with a link to each one, the corresponding ACLI show command, and a description. Scroll to see them all. When you click a link in the center pane, the Web GUI displays the data for the specified Widget. The following screen capture shows an example.

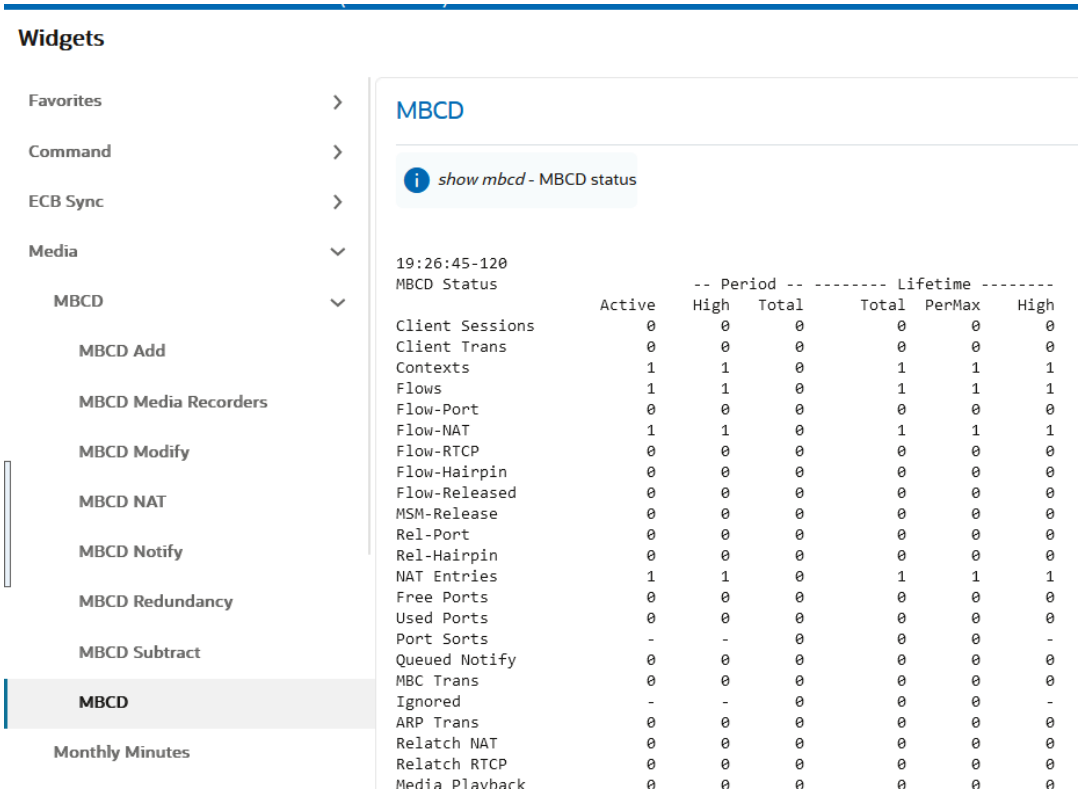
Figure 16-4 Widget Categories



- Click the twister to the right of the group name in the navigation pane. The Web GUI lists all the Widgets in the group beneath the group name in the navigation pane, with a link to each one. Scroll to see them all. When you click a link in the navigation pane, the Web GUI displays the data for the specified Widget. The following screen capture shows the Media group expanded.

> **Note:**
>
> The Web GUI also displays all the Widgets for the group in the center pane.

Figure 16-5 Widget Display



Each Widget displays management controls in the upper right corner of the Widget. The Web GUI displays only the controls that apply to the selected Widget.

Figure 16-6 Widget Control Buttons

To learn what each control does, hover over the control. The following list describes the Widget controls.

1. Refresh—Updates the data in the Widget.
2. Settings—Displays the Settings Page where you set the auto-refresh interval and any other parameters that apply to the particular Widget.
3. Add—Adds this Widget to the Dashboard.
4. Favorites—Adds this Widget to your Favorites list on the Widgets tab.
5. Show Information—the information icon "i" describes the data display. For example, in the Current Memory Usage Widget, the information icon says, "Pie graph displays current percentage of free and allocated memory."

17

Troubleshooting and Maintenance

Whereas the Oracle Enterprise Communications Broker Administrator's Essentials Guide provides system-based troubleshooting information, this chapter presents parallel information related to the SIP services the Oracle Enterprise Communications Broker (Communications Broker) provides.

To a large extent, the Communications Broker's widgets display status and quantitative information about SIP traffic. Many of these widgets provide information that is self-explanatory. This chapter provides descriptions and instructions on key widgets used to analyze service operations.

Maintenance consists of a variety of tasks, including managing system files. The Communications Broker's System tab provides access to file management controls, and is described in this chapter.

Refer to the Oracle Enterprise Communications Broker Administrator's Essentials Guide for further troubleshooting and maintenance information related to system administration.

Audit Logs

The Oracle Enterprise Communications Broker (Communications Broker) can record user actions in audit logs by way of the Web GUI. The audit logs record the creation, modification, and deletion of all user-accessible configuration elements, as well as attempted access to critical security data such as public keys. For each logged event, the system provides the associated user-id, date, time, event type, and success or failure data.

You can configure the system to record audit log information in either verbose mode or brief mode. Verbose mode captures the system configuration after every change, and displays both the previous settings and the new settings in addition to the event details. Brief mode displays only the event details. Although you can specify the recording mode, you cannot specify which actions the system records. The following list describes the actions that the system records.

Global	• Log on and log off. • Save a template configuration. • Click **Complete** in a System Operation dialog.
Home tab	• Add, reset, and save. • Change Widget settings.
Configuration tab	• Save and activate a configuration. • Discard a configuration. • Add, edit, delete, and copy configuration changes. • Run the generate and import certificate commands.

System tab	• Add audit entries to the system file management actions, such as upload, download, restore, backup, add, edit, and delete. • Force an HA switch over. • Run the Show Support Information command. • Run the Upgrade Software System Operation. • Download and view an audit log.
Monitoring tab	• Export the summary. • Export the session detail. • Export from a Widget. • Add a Widget to favorites. • Clear, clear all on alarm, add, and delete license.

The system writes audit log events in Comma Separated Values (CSV) lists in the following format:

```
{TimeStamp,
src-user@address:port,Category,EventType,Result,Resource,Prev,
Detail}
```

The following list describes each value written to an audit log event.

TimeStamp	Shows the time when the system wrote the event to the audit log.
src-user@address:port	Identifies the system that wrote the audit log line.
Category	Classifies the event as: • Configuration • Security • System
EventType	Identifies the action that caused the event as: • Activate-config • Acquire-config • Create • Data-access • Delete • Halt • Login • Logout • Modify • Reboot

	• Save-config
Result	Identifies the outcome of the event as: • Failure • Success
Resource	Describes the action within the event. Some of the numerous actions that the system can log include: • Authentication • Banner (Means that someone edited the log on banner text.) • Download <filename> • Generate public key • Reboot • Upload <filename>
Prev—(verbose mode)	Displays the setting prior to this change.
Details—(verbose mode)	Displays additional information about the change, depending on the following event types: • Create—displays “New = element added." • Data-access—displays “Element = accessed element.” • Delete—displays “Element = deleted element.” • Modify—displays “Previous = oldValue New = newValue.”

As the Communications Broker records audit log data, users with admin privileges can read, copy, and download that information from the Web GUI. No one can delete or edit the original log. You can View, Refresh, and Download audit logs by way of the System tab. Go to Audit Log under File Management.

You can configure the system to transfer audit log files to an SFTP server by way of secure FTP push, when conditions satisfy one of the following specifications.

- The specified amount of time since the last transfer elapsed.
- The size of the audit log reached the specified threshold. (Measured in Megabytes)
- The size of the audit log reached the specified percentage of the allocated storage space.

The Communications Broker transfers the audit logs to a designated directory on the target SFTP server. The audit log file is stored on the target SFTP server with a filename in the following format: **audit<timestamp>**. The timestamp is a 12-digit string the YYYYMMDDHHMM format.

Use the following process to configure transferring audit logs to an SFTP server.

1. Configure secure FTP push. See "Secure FTP Push Configuration."
2. Configure audit logging. See "Configure Audit Logging."

Secure FTP Push Configuration

You can configure the Oracle Enterprise Communications Broker (Communications Broker) to securely send audit log files to an SFTP push receiver for storage. Configure secure FTP push before you configure audit logging.

You can configure the Oracle Enterprise Communications Broker (Communications Broker) to log on to a push receiver using one of the following authentication methods to create a secure connection.

Password
Configure a username and password, and leave the **public-key** parameter blank. Note that you must also import the host key from the SFTP server to the Communications Broker for this type of authentication.

Public key
Set the **public-key** parameter to a configured public key record name including an account **username**, and configure the SFTP server with the public key pair from the Communications Broker.

It is also common for the SFTP server to run the Linux operating system. For Linux, the command ssh-keygen-e creates the public key that you need to import to the Communications Broker. The ssh-keygen-e command sequence requires you to specify the file export type, as follows.

```
[linux-vpn-1 ~]# ssh-keygen -e
Enter file in which the key is (/root/.ssh/id_rsa/): /etc/ssh/
ssh_host_rsa_key.pub
```

If you cannot access the SFTP server directly, but you can access it from another Linux host, use the ssh-keyscan command to get the key. An example command line follows.

```
root@server:~$ssh-keyscan -t dsa sftp.server.com
```

Configure Secure FTP Push with Public Key Authentication

For increased security when sending files from the Oracle Enterprise Communications Broker (Communications Broker) to an SFTP server, you can choose authentication by public key exchange rather than by password. To use a public key exchange, you must configure public key profiles on both devices and import the key from each device into the other.

The following list of tasks shows the process for configuring authentication by public key between the Communications Broker and an SFTP server. For each step in the process, see the corresponding topic for detailed instructions.

1. Generate an RSA public key on the Communications Broker. See "Generate an RSA Public Key."
2. Create a DSA public key on the SFTP server. See "Generate a DSA Public Key."
3. Import the DSA public key from the SFTP server into the Communications Broker using the **known-host** option in the Import Key dialog. See "Import a DSA Public Key."
4. Add the RSA public key to the authorized_keys file in the .ssh directory on the SFTP server. See "Copy the RSA Public Key to the SFTP Server."

Generate an RSA Public Key

Add a public key profile on the Oracle Enterprise Communications Broker (Communications Broker) and generate an RSA key. You will later import the RSA key into the SFTP server to enable authentication by way of public key exchange with the Communications Broker.

1. Access the Public Key configuration object: **Configuration** tab, **Security**, **Public key**.
2. On the Public Key page, click **Add**.
3. In the Add Public Key dialog, do the following:

Name	Enter the name of this profile.
Type	Select RSA.
Size	Enter one of the following: • 512 • 1024 • 2048 • 4096

4. Click **OK** to create the public key profile.

   The system displays the Public Key list box including the new profile.
5. Save and activate the configuration.
6. Select the newly created profile, and in the **Action** column, click **Generate**.

   The Communications Broker displays the key in the Generate Key text box for you to copy to the SFTP server.
7. Save the configuration.

- Generate a DSA public key.

Generate a DSA Public Key

Generate and save a DSA public key on the SFTP server. You will later import the DSA key into the Oracle Enterprise Communications Broker (Communications Broker) to enable authentication by way of public key exchange with the SFTP server.

1. Run the following command on the SFTP server:
   ssh-keygen -e -f /etc/ssh/ssh_host_dsa_key.pub | tee sftp_host_dsa_key.pub
2. Save the key to the authorized_keys file in the .ssh directory on the SFTP server.

- Import the DSA key into the Communications Broker.

Import a DSA Public Key

Import a DSA public key from the SFTP server into the Oracle Enterprise Communications Broker (Communications Broker).

- Generate and save a DSA public key on the SFTP server.

Perform the following procedure on the Communications Broker and select "known-host" for type.

1. Access the SSH file system on the SFTP server by way of a terminal emulation program.
2. On the SFTP server, copy the base64 encoded public file. Be sure to include the Begin and End markers, as specified by RFC 4716 *The Secure Shell (SSH) Public Key File Format*.

   For OpenSSH implementations host files are generally found at `/etc/ssh/ssh_host_dsa_key.pub`, or `/etc/ssh/sss_host_rsa.pub`. Other SSH implementations can differ.
3. On the Communications Broker, click the **Configuration** tab, **Security**, **Public Key**.
4. On the Public key page, click the **Import key** button, and do the following.

Type	Select known-host.
Name	Enter a name for your profile, which the Communications Broker displays in public key drop-down lists.
SSH Public Key	Paste the DSA public key from the SFTP server into the text box. Ensure that the text of the key ends with a semi-colon.

5. Click **Import**.

   The Communications Broker imports the key and makes it available for configuration as the public key on an external device.

Copy the RSA public key to the SFTP server.

Copy the RSA Public Key to the SFTP Server

Copy the RSA public key from the from the Oracle Enterprise Communications Broker (Communications Broker) to the authorized_keys file in the .ssh directory on the SFTP server.

- Confirm that the .ssh directory exists on the SFTP server.
- Confirm the following permissions: Chmod 700 for .ssh and Chmod 600 for authorized_keys.

When adding the RSA key to the authorized_keys file, ensure that no spaces occur inside the key. Insert one space between the ssh-rsa prefix and the key. Insert one space between the key and the suffix. For example, ssh-rsa <key> root@1.1.1.1.

1. Access the SSH file system on a configured SFTP server with a terminal emulation program.
2. Copy the RSA key to the SFTP server, using a text editor such as vi or emacs, and paste the RSA key to the end of the authorized_keys file.

Configure Audit Logging

The Oracle Enterprise Communications Broker (Communications Broker) provides a means of tracking user actions through Audit Logs. You can specify how the system records audit log information, and where to send the logs for archiving. You can configure the system to record in either brief or verbose mode. Verbose mode captures the system configuration after every change, and displays both the previous and new settings in addition to the event details. Brief mode displays only the event details.

- Configure one or more push receivers to receive the audit logs. See the documentation for the receiver.

- If you want to use public keys for authentication between the Communications Broker and the push receiver, configure public key profiles on both devices before configuring audit logging. See "Configure Secure File Transfer with Public Keys."

1. Access the Audit Logging configuration object: **Configuration** tab, **System Adminstration** section, **Security**, **Audit Logging**.
2. On the Audit Logging page, do the following:

State	Select to enable event recording in the audit log.
Detail Level	Select brief (default) or verbose output.
Audit Trail	Enables logging every command that is processed by the Communications Broker. • enabled: Logs all commands that the Communications Broker can process. • disabled: Logs only relevant information. • Default: disabled
Audit Record Output	Indicates how the Communications Broker logs audit records. • syslog: The Communications Broker logs audit records over syslog. • file: The Communications Broker logs audit records to a file. • both: The Communications Broker logs audit records over both syslog and to a file. • Default: file
File Transfer Time	Specify the amount of time, in hours, from the completion of the last transfer to the beginning of the next transfer. This determines when a file transfer occurs unless the Max storage space or Max file size triggers the transfer first. Default: 720. Range: 0-65535. **Note:** 0 disables this parameter.
Max Storage Space	Specify the maximum amount of space that the audit log can consume on the Communications Broker in MB. Default: 32. Range: 0-32.
Percentage Full	Use in conjunction with Max storage space to specify the percent of the Max storage space that triggers file transfer. This determines when a file transfer occurs unless the File transfer time or Max file size triggers the transfer first. Default: 75. Range: 0-99. **Note:** 0 disables this parameter.

Max File Size	Set the maximum size in Mega Bytes that the audit log can be before the system transfers the file. This determines when a file transfer occurs unless the Max storage space or Max file size triggers the transfer first. Default: 5. Range 0-10. **Note:** 0 disables this parameter.
Storage Path	Specifies the directory that houses the audit log. Default: /code/audit .
Push Receiver	Add a push receiver and configure the following parameters for sending audit log files from the Communications Broker to the receiver: • Server—Enter the IP address of the FTP/SFTP server to which you want the Communications Broker to push audit log files. Default: 0.0.0.0. • Port—Enter the port number on the FTP/SFTP server to which the Communications Broker will send audit log files. Default: 22 Range: 1-65535. • Remote Path—Enter the pathname to send the audit log files to the push receiver. Files are placed in this location on the FTP/SFTP server. Value: <string> remote pathname. • Filename Prefix—Enter the filename prefix to prepend to the audit log files that the Communications Broker sends to the push receiver. The Communications Broker does not rename local files. Values: <string> prefix for filenames. • Username—Enter the username the Communications Broker uses to connect to this push receiver. • Auth Type—Select the authentication methodology. Password (default) or public key. • Do one of the following: Password—When you set the Auth type to password, click **Set** to enter and confirm the password used to access this push receiver. Public Key—When you set the Auth type to public key, select the public key profile that you want from the drop-down list.

3. Click **OK**.
4. Save the configuration.

System File Management

You can manage Oracle Enterprise Communications Broker (Communications Broker) system and configuration files from the Web GUI on the **System tab** under **File Management**.

The following table lists and describes the file types that you can manage.

File Type	Format	Description
Backup Configuration	.gz	Contains a backup of the Communications Broker software configuration. You can apply this file to restore a previous configuration.
Configuration CSV	.csv	Contains Comma Separated Value configuration files that you can upload.
Local Route Table (LRT)	.xml, .gz	Contains the Local Routing Table (LRT) file that you can apply to theCommunications Broker. The LRT is an in-memory table that contains IP addresses that the local router recognizes. It calculates the destinations of messages it is responsible for forwarding.
Fraud Protection Table	.gz, .gzip, .xml	Contains fraud protection files that you can upload, download, delete, or open to modify.
Log	Text	Contains Log files with information about the various aspects of the Communications Broker. For example, information logged about the ACLI, SIP, or H323. Note: Only the Download and Delete functions are applicable to log files on the Communications Broker.
Audit Log	Text	Contains a list of files that log system events.
Playback Media	Any media format valid in an RTP audio stream	Contains call progress playback files. The Communications Broker can use these files in generated media streams. Note: The media files are raw binary files that contain data for the codec that you want played in the media stream. The Communications Broker plays the data on the first audio flow in the Session Description Protocol (SDP).
Software Image	.gz, .bz	Contains bootable images.
SPL Plug-in	.lua	Contains a Session Plug-in Language (SPL) file that you can apply to the Communications Broker to incorporate additional functionality. The SPL file contains a programming language capable of performing various tasks by utilizing APIs and callbacks in the Communications Broker.

The following table lists and describes the file management controls that display in the file type dialog, according to the supported behavior for the file type.

Control	Description
Refresh	Updates the screen to display the latest data.
Add (Fraud Protection files, only.)	Adds a new Fraud Protection file.
Upload	Uploads a file from your server or PC to the Communications Broker. The LRT, SPL, and backup configuration upload process provides the option of dynamically applying these files to the Communications Broker.
Download All (Log files, only.)	Downloads all Log files from the Communications Broker to your local server or PC (typically to the download directory on your system).
Backup	Creates a file that contains a backup of the device software configuration. You can apply this file to restore a previous configuration.
Restore (Backup configuration files, only.)	Restores and applies a Backup configuration file to the Communications Broker.
Delete	Deletes the file type from the Communications Broker.
Delete All (Log files, only.)	Deletes all Log files from the Communications Broker.

Upload a File

You can upload the following file types from your local server or PC to the Oracle Enterprise Communications Broker (Communications Broker).

> **Note:**
>
> The Log and Audit Log files do not support uploading.

File Type	File Format	Directory
Backup Configuration	.gz	/code/bkups
Configuration CSV	.csv	
Local Subscriber Table (LST)	.xml.gz	/code/gzConfig
Software image	.bz, .tar or no extension specified	/code/images
SPL Plug-on (SPL)	.lua, .spl	/code/spl

The file extension must be applicable to the file type you select. For example, an SPL Plug-in file requires the .lua extension

You can dynamically activate the Local route table and SPL Plug-in during the upload process.

You can immediately restore a backup configuration file after an upload is complete.

> **Note:**
>
> You cannot upload log files.

1. Access File Management: **System** tab, **File Management**.
2. On the File management page, select the type of file you want to upload.
3. In the Name column, select the file you want to upload.
4. Click **Upload**.
5. In the Upload file dialog, do the following:
   a. Click **Browse**.
   b. Select the file that you want to upload.
   c. Optional. For the Backup configuration file, select Restore the configuration after upload to apply a previous backed up configuration file immediately to the after the upload is complete.
   d. Optional. For the Local Subscriber Table file type, select Activate the LST file after upload to apply the LST upon upload.
   e. Optional. For the SPL Plug-in file type, select Activate the SPL file after upload to apply the SPL file upon upload.
   f. Click **Upload**.

Download a File

Use this procedure to download a file from the Oracle Enterprise Communications Broker (Communications Broker).

You can download any of the following file types from your local server or PC to the Communications Broker:

- Backup Configuration
- CSV File
- Local Subscriber Table (LRT)
- Log
- Software Image
- SPL Plug-in (SPL)

1. Access File Management: **System** tab, **File Management**.
2. On the File Management page, select the type of file you want to download from the File type list.
3. Place your cursor on the row of the file that you want to download.

> **Note:**
>
> For Log file types, you can select multiple log files to download, or place a checkmark in the box to the left of the Name column heading to select all log files to download. When downloading multiple log files, the File Management GUI compresses the files into one .tar file and downloads that file to your local server or PC.

4. Right-click the selected row, and click **Download**.
5. Do one of the following:

- Click **Open With** and select the application to open the file.
- Click **Save File** to save the file to your local server or PC.

6. Click **OK**.

   The system downloads the file to the folder on your local server or PC where your Browser sends all downloads (typically your “Download” folder) or opens (decompresses) the file type on your local server or PC (typically in the “Download” folder).

Delete a File

The following information describes the procedure and conditions for deleting a file from the Oracle Enterprise Communications Broker (Communications Broker).

You can delete any of the following file types from your local server or PC to the Communications Broker:

- Local Subscriber Table (LST)
- SPL Plug-in (SPL)
- Backup Configuration

> **Note:**
>
> You can select a single or multiple files to delete.

1. Access File Management: **System** tab, **File Management**.
2. From the File Management list, select the type of file that you want to delete.
3. Put your cursor in the row of the file you want to delete, and right-click.

   Note: If the GUI does not display a menu upon right-click, use the **Delete All** control located above the list of files.
4. Click **Delete**. The system displays following message.

   Are you sure you want to delete the file?
5. Click **Delete**.

Back Up a Configuration File

You can back up a configuration file from the Oracle Enterprise Communications Broker (Communications Broker) to your local server or PC. Back up allows you to save configurations that you can restore to the Communications Broker at a later time.

1. Access File Management: **System** tab, **File Management**.
2. Select a file from the list in the table.
3. Click **Backup**.
4. Click **OK**.

   The system adds the file to the Backup Configuration table.

Restore a Configuration File

You can restore a backed up configuration file to the Oracle Enterprise Communications Broker (Communications Broker).

1. Access File Management: **System** tab, **File Management**.
2. On the File Management page, select Backup Configuration.
3. Select a back up, and right-click.

> **Note:**
>
> **Restore** activates only when you select a back up file.

4. Click **Restore**.

   The GUI displays a confirmation dialog.
5. Click **Restore**.

   The system downloads the back up file to the Communications Broker. The Communications Broker re-boots and restores the configuration from the back up file.

CSV Configuration File Creation

The Comma Separated Values (CSV) file is a text format file supported by spreadsheet applications. You can import a CSV file into the Oracle Enterprise Communications Broker (Communications Broker) that contains its configuration, or you can export the current configuration on the Communications Broker to the CSV file. You can also upload parts of your Communications Broker configuration separately, such as users, dial plans, and routes. You can perform the upload manually or set the system to perform a periodic upload, automatically.

In the CSV file format, each row is defined on its own line and each column is separated by a comma.

You can create your own CSV configuration files, but be aware of the following rules for proper formatting.

- The Communications Broker ignores empty lines.
- If an entry contains a comma, enclose it in quotes to prevent it from being treated as a separator.
- The first non-empty line must be the keyword “object:”, followed by the configuration object name that is being configured (shown below as “sip-interface”).

  ```
  object:sip-interface
  ```

- The second non-empty line must state the operation to perform, which can be ADD, MODIFY, or DELETE.

  ```
  operation: ADD
  ```

- The third non-empty line must state the parameter name of the object to be configured, and each parameter name must be in its own column. This row defines the “labels” for each column for the subsequent rows. Only the attributes you want defined need to be

present. You can specify the parameter names in any order, but the data in subsequent rows must be consistent with the “labels” that you define in this row.

```
state,realm-id,description
```

- The fourth non-empty rows define instances (values) for the configuration object, each instance in its own column. In the following example, the third line defines a new sip-interface with state “enabled”, realm-id “public”, and description “public SIP interface”. These values are based on the “labels” defined in the second row.

```
enabled,public,public SIP interface
```

- On all subsequent rows, you can define any number of instances.
- The next row with an “object” keyword selects a new configuration object that is based on the previous object. You continue to input the data for this object according to the rules stated above. The following example shows a “sip-port” object added that is related to the sip-interface object.

```
object:sip-port
operation:ADD
address,port,transport-protocol
192.168.1.1,5060,UDP
192.168.1.1,5061,TCP
```

- In the preceding example, “sip-port” is a sub-object of “sip-interface” that creates new sip-ports from the last sip-interface instance (of realm-id “public”).
- Note that the Description field displays all text as one continuous line, unless you insert line breaks. When you want to insert line breaks in the Description field, for example between sentences that you want displayed on separate lines, do the following:
  - From the GUI, in the Description field of a Configuration object, add Line1 to the end of the line where you want the first break to occur. Add Line2 to the end of the next line where you want a break to occur, and so on.
  - In a CSV configuration file, add \010Line1 to the end of the line where you want the first break to occur. Add \010Line2 to the end of the next line where you want a break to occur, and so on.

> **Note:**
>
> After you create the initial CSV configuration file, you can set the Communications Broker to automatically upload updated versions of the file. See "Automatically Upload Updated CSV Configuration Files" and "Configure Automatic CSV File Uploads."

Caveats

- Files are written to the volatile directory of the file system on the system. For the Acme Packet 4500, this is the “/ramdrv/” directory. For the appliance and virtual machines, it is the “/var/” directory.
- Import and export occurs to and from the editing configuration.
- All error messages are printed to the screen, where the command was issued. Line numbers are provided with the error when possible.

- Objects and attributes cannot be set to instances (values) that are not allowed. For example, you cannot set an IP address to "enabled". Parsing continues normally after this error.
- If an object cannot be written (i.e. key field is missing), then that object is discarded and parsing continues as normal.
- The import is additive. Each object that is imported is expected to be new to the configuration. If there is already an object with the same key present, it generates an error 409 and is discarded. Parsing continues as normal after the error.

Create a CSV Configuration File

You can create a Comma Separated Values (CSV) file of the agents, dial plan, users, and routing configurations to make maintaining and updating such information easier. After you create the file, you upload it to the Oracle Enterprise Communications Broker (Communications Broker).

In the CSV file format, each row is defined on its own line and each column is separated by a comma. When preparing the CSV file, you must specify each object and operation that you want the Communications Broker to act on. For each object in the .csv file, the first three lines of the file format include the following information:

- object—Specify the object you want the Communications Broker to act on. Valid values: Agents, Dial Plan, Users, and Routing.
- operation—Specify the operation to perform. Valid values: ADD, MODIFY, or DELETE. If you do not specify an operation, the system defaults to ADD-MODIFY.
- parameters—A system provided list of all of the parameters for the object. The system can perform the specified operation on any of the parameters listed.

To create a CSV file that contains system configuration, do the following:

1. Open an application that supports a CSV file.
2. In the first row, first column, enter “object:” followed by a configuration object you want to import.`object:sip-interface`
3. In the second row, enter the operation to perform. `operation:ADD`.
4. In the third row, and each in its own column, enter the parameter names of the objects to be configured. state,realm-id,description`state,realm-id,description`.
5. In the fourth row, and each in its own column, enter the instances (values) for the configuration objects.`enabled,public,public SIP interface`.
6. In subsequent rows, define additional instances (values), as needed.
7. In the next empty row, first column, enter another object if needed, related to the first object (sip-interface).`object:sip-port`.
8. Repeat steps 3 through 6 for this object.
9. Save the file as a .csv.
10. Upload the configuration file using the upload button from the applicable dialog. (For example, upload a .csv file of users from the User database.)

- After you create the initial CSV configuration file, you can set the Communications Broker to automatically upload updated versions of the file. See "Automatically Upload Updated CSV Configuration Files" and "Configure Automatic CSV File Uploads."

Automatically Upload Updated CSV Configuration Files

When you want to automatically update selected configuration objects on the Oracle Enterprise Communications Broker (Communications Broker), you can enable the system to periodically check for updates and automatically upload the CSV configuration file. Within the CSV file, you can specify the operation that you want the Communications Broker to perform upon upload, such as add, modify, and delete data. The supported configuration objects include Agents, Dial Plan, Users, and Routing.

When enabled, the Communications Broker checks /code/csv every two minutes for a new file and uploads it to the Communications Broker. (The time interval is not configurable.)

When preparing the CSV file for upload, you specify each object and operation that you want the Communications Broker to act on.

For each object in the CSV file, the first three lines of the file format include the following information:

- object—Specify the object you want the Communications Broker to act on. Valid values: Agents, Dial Plan, Users, and Routing.
- operation—Specify the operation to perform. Valid values: ADD, MODIFY, or DELETE. If you do not specify an operation, the system defaults to ADD-MODIFY.
- parameters—A system provided list of all of the parameters for the object. The system can perform the specified operation on any of the parameters listed.

The following example shows the first three lines of the CSV file format:

```
object:user-number
operation:ADD/MODIFY/DELETE
AOR,number-or-pattern,description,dialing-context,agent,policy,tags
```

In the fourth line, you begin listing the parameter data to ADD, MODIFY, or DELETE. For example:

```
123@oracle.com,555[2000-3999],APAC,Server,
```

The Communications Broker uses only one CSV file at a time as the source file, but the file can contain multiple objects and multiple operations. The following example shows a CSV file with multiple objects, operations, and parameter data:

```
object:user-number
operation:MODIFY
AOR,number-or-pattern,description,Dialing-context,agent,policy,tags
123@oracle.com,555[2000-3999],APAC,Server,

object:dialing-context
operation:ADD
Name, geographic location, description, country code, outside line prefix,
Dial patterns, remove prefix, pattern, description, country code, replacement
prefix, replacement uri, go to context
APAC.Nepal,CALA.Uraguay,EMEA.Andorra

object:policy
operation:DELETE
```

```
Name, description, name, contains codecs, codec condition, time condition,
next hop compare condition, contains codecs, missing codecs, name, days,
start time, end time, name, next hop compare mode, name, routing mode, name
redirect to agent, hairpin signaling, name, egress number translation mode,
number of digits for n digit dialing, prepend calls on egress, name, ignore
constraints, name, header name, dialing ccontext, result store, new value,
name, display name
<policy name>, next hop compare condition, time condition

object:session-agent
operation:ADD
hostname,ip-address,port,state,RURI-with-Hostname,app-protocol,app-
type,transport-method,TLS-profile,realm-id,egress-realm-id,description,source-
context,egress-uri-mode,egress-number-translation-mode,number-of-digits-for-n-
digit-dialing,prepend-prefix-on-egress,outbound-translate-from-
number,tags,carriers,allow-next-hop-lp,associated-agents,stop-
recurse,constraints,max-sessions,max-inbound-sessions,max-outbound-
sessions,max-burst-rate,max-inbound-burst-rate,max-outbound-burst-rate,max-
sustain-rate,max-inbound-sustain-rate,max-outbound-sustain-rate,min-
seizures,min-asr,session-max-life-limit,time-to-resume,ttr-no-response,in-
service-period,burst-rate-window,sustain-rate-window,req-uri-carrier-
mode,proxy-mode,redirect-action,loose-routing,send-media-session,response-
map,ping-method,ping-interval,ping-send-mode,ping-all-addresses,ping-in-
service-response-codes,out-service-response-codes,load-balance-dns-
query,options,spl-options,media-profiles,in-translationid,out-
translationid,apply-outbound-manipulation-on,trust-me,request-uri-
headers,local-response-map,ping-to-user-part,ping-from-user-part,in-
manipulationid,out-manipulationid,manipulation-string,manipulation-pattern,p-
asserted-id,trunk-group,max-register-sustain-rate,early-media-
allow,invalidate-registrations,rfc2833-mode,rfc2833-payload,codec-
policy,enforcement-profile,early-media-inhibit,enable-OPTIONS-
ping,ldap,additional-target-group,fork-group,refer-call-transfer,refer-notify-
provisional,reuse-connections,tcp-keepalive,tcp-reconn-interval,max-register-
burst-rate,register-burst-window,sip-profile,sip-isup-profile,kpml-
interworking,precedence,monitoring-filters,session-recording-server,session-
recording-required,hold-refer-reinvite,send-tcp-fin,sip-recursion-policy,sm-
icsi-match-for-invite,sm-icsi-match-for-message
Client,1.1.1.1,1234,enabled,disabled,SIP,,UDP,,ecb,,,,no-
conversion,E164,4,,disabled,,,enabled,,,disabled,0,0,0,0,0,0,0,0,0,5,0,0,0,0,0
,0,0,None,,,enabled,enabled,,,0,keep-alive,disabled,,,hunt,,,,,,next-hop-
only,disabled,,,,,,,,,,,0,,disabled,none,0,,,disabled,disabled,,,1,disabled,no
ne,NONE,none,0,0,0,,,inherit,0,,,disabled,disabled,disabled,,,
```

> **Note:**
>
> The Communications Broker also supports the older CSV file format that does not provide the add, modify, delete, and automatic upload operations.

When the Communications Broker discovers a new CSV configuration file in /code/csv, it creates a log message and a temporary copy of the existing file before uploading the new file. Upon successful upload, the system creates an audit entry, deletes the temporary file, and moves the original file to an archive folder in the same location you specified for the source CSV file. The Communications Broker retains the last 10 successfully uploaded CSV files.

Should an error occur during the upload, the Communications Broker creates a message in the error log, moves the file to an error folder, generates a trap (SAVE_CSV_CONFIG_FAIL_TRAP), and raises an alarm in the Alarms dialog on the GUI. (Under Notifications.) The log notes errors for improperly formatted files and inaccessible files. The Communications Broker puts the error log in the same location as the CSV file (/code/csv). The Communications Broker retains the last 10 unsuccessfully uploaded CSV files.

> **Note:**
>
> The Communications Broker does not allow you to manually launch an upload during the automatic upload operation, and the system does not provide a way to queue a manual upload to launch when the automatic upload finishes. You must wait until the automatic upload finishes to launch a manual upload.

To enable automatic CSV file uploads, you enable the service in **system config**. See "Configure Automatic CSV Configuration File Uploads."

Enable Automatic CSV Configuration File Uploads

When you want the Oracle Enterprise Communications Broker (Communications Broker) to automatically update the .csv configuration file, you must enable the service.

- Prepare the .csv file for upload with any additions, modifications, or deletions.

1. Access the System Config configuration object.

   Configuration tab, **System Adminstration** section, **General Settings**, **System Config**.
2. On the System Config page, enable CSV Upload.
3. Click **OK**.
4. Save the configuration.

Upgrade Software

You can upgrade the Oracle Enterprise Communications Broker (Communications Broker) software from the System tab. The system requires a reboot after the upgrade.

1. Access Upgrade Software: **System** tab, **System Operations**, **Upgrade Software**.
2. In the Upgrade Software dialog, click **Verification**, and do the following:
   - View the health score.
   - Click **View Health Information**, and confirm that the system components are synchronized.
   - Click **View Configuration Version**, and note the Current Version and Running Version.
   - Click **View Disk Usage**, and confirm that the system has enough free space.
3. Select one of the following upload methods.

Local	Select a file from your system, and proceed to Step 4
Flash	Select a file already on the Communications Broker, and proceed to Step 4.

Network	Do the following: • Boot file. (Network) Enter the complete name of the boot file. • Host IP. (Network) Enter the IP address of the FTP server. • FTP username. (Network) Enter the user name to log onto the FTP server. • FTP password. (Network) Enter the password to log onto the FTP server. • Optional. Select `Reboot after upload`. Proceed to Step 6.

4. Select a file to upload.
5. (Optional) Select `Reboot after upload`.
6. Click `Complete`.
   - If you did not select `Reboot after upload`, the system displays a message stating that a reboot is required for the changes to take effect.
   - If you selected `Reboot after upload`, the system displays a message stating that it is about to reboot.
7. Click `OK`.

   If you selected `Reboot after upload`, the system reboots.

System Restart

You can manually restart the Oracle Enterprise Communications Broker (Communications Broker) from the Web GUI. If you have a High Availability (HA) deployment, connectivity to the standby Communications Broker stops until the restart completes.

When the restart completes, the active and standby systems each display the log on screen and you must manually log on to each system.

> **Note:**
>
> When you restart the system from the Web GUI, the Web GUI is unavailable until the restart completes.

When you perform a restart from the Web GUI	The system behaves
and no boot is in process and the system is not switching over to the standby system of an HA pair,	the GUI session closes and the system displays the Log On screen. You cannot log on to the Web GUI until the restart completes on the Communications Broker.
and a restart is already in progress,	the system displays a message stating that a restart cannot occur. The first restart must complete before another restart is initiated.
and the active system is currently switching over to the standby system in an HA environment,	the system displays a message stating that a restart cannot occur because the HA switch over is underway. The standby Communications Broker is updating and getting its configuration from the active Communications Broker.

Display Log Files

The Oracle Enterprise Communications Broker (Communications Broker) allows you to view log files without needing to download them.

1. Access the **File Management** page. Click **System** tab, **File Management**.
2. On the **File Management** page, select a Log file from list.
3. Expand a log file category and select a log file by selecting the check box by the file name.

   The Communications Broker enables the **View** control.
4. Click **View**.

   The Communications Broker displays the **Viewing log:[filename]** dialog with the log file's contents.

Display System Health

The Oracle Enterprise Communications Broker (Communications Broker) provides a widget that allows you to see the current health score and state of the Communications Broker.

- Access the **System Health** page. Click the **Widgets** tab, **System**, **System Health**.

  The GUI displays the **System Health Table**, where you can see the health score and state of the Communications Broker.

Obtain Support Information - Firefox Browser

You can manually generate a file by way of the Web GUI that contains troubleshooting information. You can save the file and send it to Oracle Customer Support.

1. Access Support Information: **System** tab, **Support Information**.
2. Click **Support Information**.

   The system displays the confirmation dialog.
3. Click **Confirm**.

   The system generates the file.
4. The browser asks you to do one of the following:

   The system generates and saves the file to the Download directory or to another location that you specify.

18
Active Directory Modifications

When using the Oracle Enterprise Communications Broker LDAP configuration to access authentication and routing information from Active Directory (AD), you must prepare AD to serve those functions. For authentication, you can add an Oracle-supplied DLL to the system to capture password hashes during password changes and store them for authentication.

The Oracle-supplied DLL, **oecbpwdcn.dll**, is an OSD DLL that provides the Windows-specific password hash capture function. When a user changes their password, the DLL intercepts the hash of the password and stores it for SIP authentication. The user's password is never visible in clear-text.

Related AD changes consist of the following, which can be done manually or by way of Oracle-provided scripts:

1. Create orclDigestRealmAttribute attribute (to store digest realm name) and associate it users.
2. Create orclDigestPwdAttribute attribute (to store hashed password) and associate with users.
3. Create orclAgentNameAttribute and associate it with users.

You can refer to **http://www.microsoft.com/technet/prodtechnol/windows2000serv/technologies/activedirectory/howto/adschema.mspx** for instructions on managing AD. You can manually add the following entries into AD:

```
dn: cn=orcldigestrealmattribute,cn=schema,cn=configuration,dc=example,dc=com
changetype: add
objectClass: top
objectClass: attributeSchema
cn: orclDigestRealmAttribute
instanceType: 4
attributeID:
1.2.840.113556.1.8000.2554.54362.52699.4250.17878.46369.10622351.7266019.1
attributeSyntax: 2.5.5.4
isSingleValued: TRUE
showInAdvancedViewOnly: TRUE
adminDisplayName: orclDigestRealmAttribute
adminDescription: Oracle ECB Digest Realm
oMSyntax: 20
lDAPDisplayName: orclDigestRealmAttribute
name: orclDigestRealmAttribute
```

This creates the attribute to which **oecbpwdcn.dll** stores password hashes.

```
dn: cn=orcldigestpwdattribute,cn=schema,cn=configuration,dc=example,dc=com
changetype: add
objectClass: top
objectClass: attributeSchema
cn: orclDigestPwdAttribute
instanceType: 4
```

```
attributeID:
1.2.840.113556.1.8000.2554.54362.52699.4250.17878.46369.10622351.7266019.2
attributeSyntax: 2.5.5.4
isSingleValued: TRUE
showInAdvancedViewOnly: TRUE
adminDisplayName: orclDigestPwdAttribute
adminDescription: Oracle ECB Digest Password
oMSyntax: 20
lDAPDisplayName: orclDigestPwdAttribute
name: orclDigestPwdAttribute
```

This creates an attribute that can be used for routing, specifically by providing a field for storing the users' Agent.

```
dn: cn=orclagentnameattribute,cn=schema,cn=configuration,dc=example,dc=com
changetype: add
objectClass: top
objectClass: attributeSchema
cn: orclAgentNameAttribute
instanceType: 4
attributeID:
1.2.840.113556.1.8000.2554.54362.52699.4250.17878.46369.10622351.7266019.3
attributeSyntax: 2.5.5.4
isSingleValued: TRUE
showInAdvancedViewOnly: TRUE
adminDisplayName: orclAgentNameAttribute
adminDescription: Oracle ECB Agent Name
oMSyntax: 20
lDAPDisplayName: orclAgentNameAttribute
name: orclAgentNameAttribute
```

> **Note:**
>
> You must replace %AD_DOMAN_NAME% with your AD domain name, such as dc=acme,dc=com.

For convenience, two LDIF files are provided to facilitate adding these two attributes. They are "addOrclECBAttribute.ldif" and "addUserObjClass.ldif". To add the two attributes automatically:

1. Make sure the Active Directory Schema Snap-In is installed by following the directions from:
   - **http://social.technet.microsoft.com/wiki/contents/articles/20319.how-to-create-a-custom-attribute-in-active-directory.aspx** or
   - **http://technet.microsoft.com/en-us/library/cc759633(v=ws.10).aspx**
2. Open the two files and replace %AD_DOMAN_NAME% with your actual AD domain name, such as dc=acme,dc=com.
3. Run the command "**ldifde –i –f addOrclECBAttribute.ldif –v**" to create the three attributes.
4. Then run the command "**ldifde –i –f addUserObjClass.ldif –v**" to associate the attributes to AD users.

5. Reload the AD schema or reboot AD.
6. Verify that the two attributes are present by checking users to see that attributes are available to them.

In addition to AD schema modification, follow the steps below to install **oecbpwdcn.dll**.

1. Install OID Password Change Notification (oecbpwdcn) DLL, by simply copying the oecbpwdcn.dll to your AD WINDOWS\system32
2. Using regedt32 to change the registry and enable the DLL. Invoke regedt32 and modify the registry setting HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Notification Packages. Add "oecbpwdcn" to the end of this list. Example registry entries, including oecbowdcn, could now include:
   - RASSFM
   - KDCSVC
   - WDIGEST
   - scecli
   - oecbpwdcn
3. Reboot AD.

Test your deployment as follows:

1. Assign a digest realm name to user's orclDigestRealmAttribute in AD. You can use script modifyUsersDigestRealmName.vbs to modify this attribute for all users. Right click on modifyUsersDigestRealmName.vbs and select "Run with Command Prompt"
2. Modify user password for any AD user (or reset the password)
3. Search against AD and look up the AD user and orclDigestPwdAttribute should have the generated hash value.

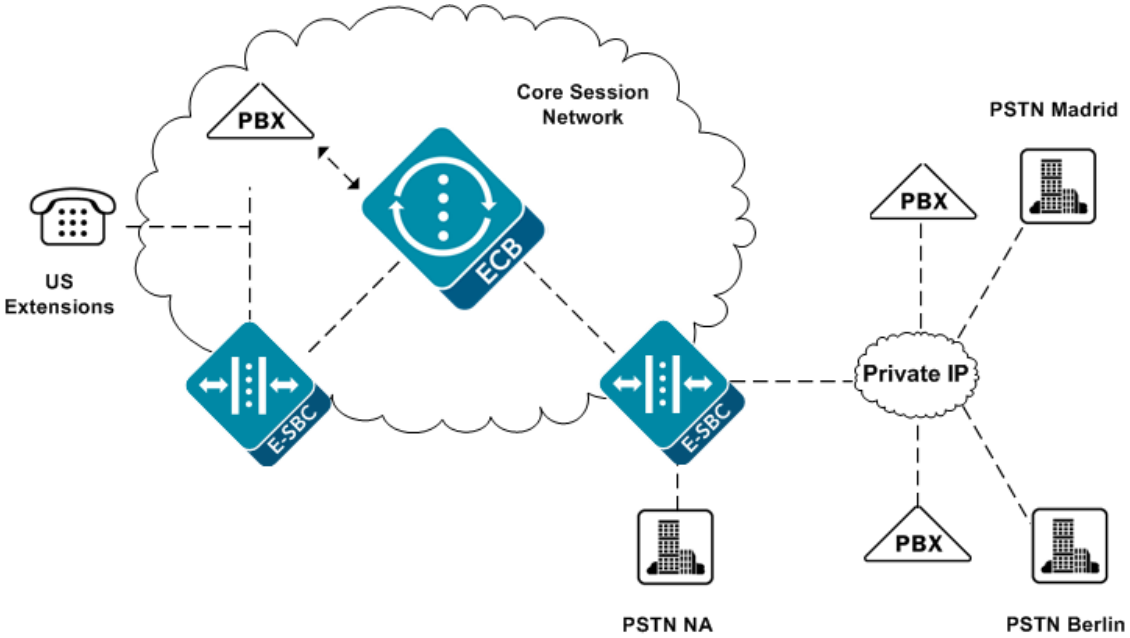
You can use a script named **displayUsersDigestRealmPassword.vbs** to display the values from all users. To do this, right-click**displayUsersDigestRealmPassword.vbs** and select "Run with Command Prompt."

19

Configuration Examples

The Oracle Enterprise Communications Broker (Communications Broker) is a flexible tool that provides a variety of configuration options that can achieve similar results.

In the following diagram an Communications Broker is deployed at the center of the session network to manage SIP signaling traffic for an enterprise headquartered in the U.S., with branch offices in the U.S., Spain, and Germany.



There are multiple E-SBCs, PBXs, and tie lines handling session services. The Communications Broker must normalize multiple signaling formats, integrate multiple vendor processes, provision varied session services, and handle an enterprise user database.

The following topics provide more examples for configuring the Communications Broker using sample configurations targeting specific environmental models. The information may help you to identify configuration options that track closely to the needs of your deployment.

Configuration Sequence

Small Enterprise Model

Large Enterprise Model - v2

Emergency Dial Configurations

Alternate Translation Modes

ENUM Example Configuration

Configuration Sequence

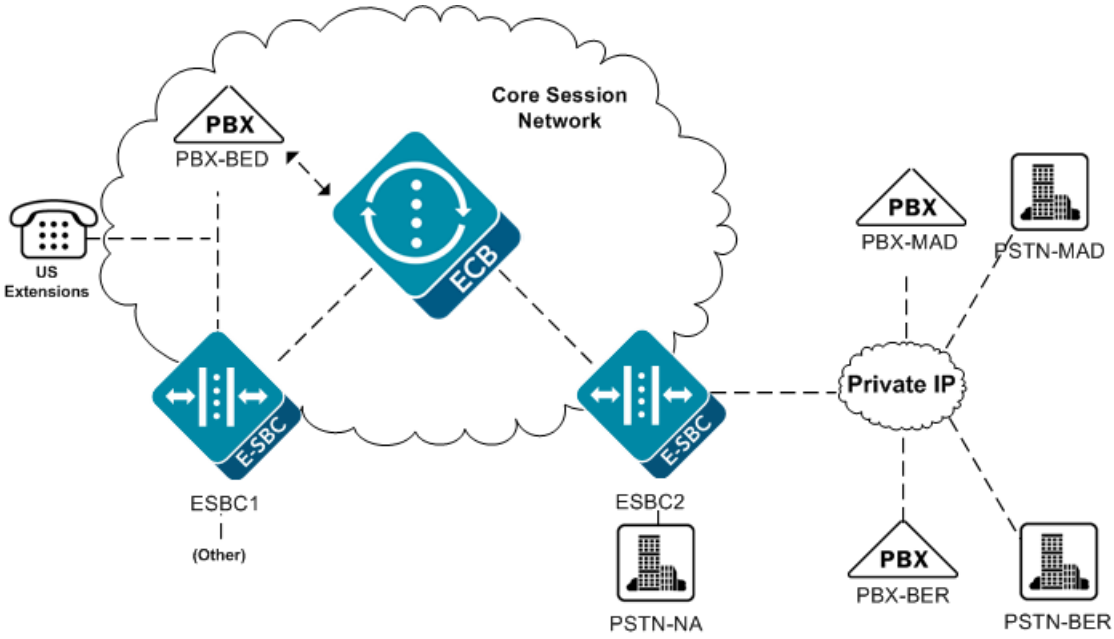
When first configuring the Oracle Enterprise Communications Broker for service, follow the sequence below to establish objects that you need in ensuing object configurations.

1. Agents - Agents help to segregate your network logically and geographically, providing Agent configuration also specifies the hops available to you on your network for routes. In addition Users often use Agents to help specify their contexts.
2. Dial Plans - Dial Plans specify all the contexts that you use in routes. Users may also be configured with contexts.
3. Users - It is often best to configure Users with existing Agents and Dial Plans to specify their locations and applicable policies.
4. Routes - Routes often use Agents to specify source and destinations.

You will find this sequence is of decreasing value over time. In addition, it is not required that you follow this sequence at any point. The flexibility of the Oracle Enterprise Communications Broker's configuration options allow you perform steps such as creating new contexts as you create users. The sequence is most useful for understanding how to piece together the elements of the Oracle Enterprise Communications Broker.

Initial Agent Configuration

The following diagram expands upon the network diagram in "Configuration Examples" by adding specific names for the agents required for this Oracle Enterprise Communications Broker deployment.



The following table shows agent configurations that correspond to the agents in the preceding diagram. This table provides examples from which you can learn the guidelines for configuring your own agents. If you were to perform the configuration for this example, you would configure each agent in the following table with IP address, Port, and Transport mode. Note that in this example, all Agents take the default Number Translation Mode, which is e.164. No other configuration is necessary to support the configuration in this example.

Hostname	Source Control	# Translation Mode	# of Digits	Prepend Prefix
PBX-BED		e.164		
PBX-MAD		e.164		
PBX-BER		e.164		

Hostname	Source Control	# Translation Mode	# of Digits	Prepend Prefix
ESBC1		e.164		
ESBC2		e.164		
PSTN-NA		e.164		
PSTN-MAD		e.164		
PSTN-BER		e.164		

> **Note:**
>
> In an actual deployment, you can add more agent configurations, as needed.

Dial Plan Strategies

Oracle Enterprise Communications Broker configuration design provides you with the flexibility to make the same settings, such as country code, on multiple elements. To the extent that child elements inherit properties from parent elements, endeavor to elegantly cover the basic requirements of your deployment with your initial configurations while preserving configuration options in child objects to meet the needs of exceptions and expansion.

The simplest way to approach dial plan configuration is to base your corporate contexts on your enterprise's branch offices. You configure dials plan and patterns comes within those contexts. The parent context establishes rules that you need enforced across the enterprise. Each branch office gets a child context that inherits from both the corporate parent and the rules associated with the country in which branch resides (geographic context).

Recall that the Oracle Enterprise Communications Broker comes pre-configured with the vast majority of geographic contexts you need. These contexts include the country code. By setting a geographic location to your branch office, you inherit country code. These contexts also include a description (of their location), which has no relevance to signaling processing.

The following corporate context configurations apply to the early enterprise configuration models presented in the ensuing sections.

Table 19-1 Dial Plan Configurations

Name	Geographic location	Country Code	Outside line prefix
acme	NA	NA	NA
acme.bedford	NA	NA	9
acme.madrid	EU.Spain	NA	NA
acme.berlin	EU.Germany	NA	NA

This appendix uses dial plans and dial patterns to establish differentiation between configuration models, with all using the same routes. For this reason, this appendix strays from the suggested configuration sequence and sets up routes next.

Route Strategies

Route configuration consists of mapping out an extensible strategy according to your deployment model and connecting agents together. Configure all agents as simply as possible

and create only as many routes as are necessary. Careful planning allows you to create routes that serve multiple purposes simultaneously.

You may recall that initial configuration procedures has user configuration preceding route configuration. This configurations design, however, covers all expected users without specific user configuration.

Table 19-2 Routes Configured

Route Number	Source Agent	Calling Number	Called Number	Dest Agent	Route	Cost
#1	*	*	34	*	PSTN-MAD	20
#2	*	*	*	PSTN-MAD	ESBC2	0
#3	*	*	*	PBX-MAD	ESBC2	0
#4	*	*	*	*	PSTN-BER	20
#5	*	*	*	PSTN-BER	ESBC2	0
#6	*	*	*	PBX-BER	ESBC2	0
#7	*	*	1*	*	PSTN-NA	20
#8	*	*	*	*	PSTN-NA	70

The table below explains the purpose of each route in the table above.

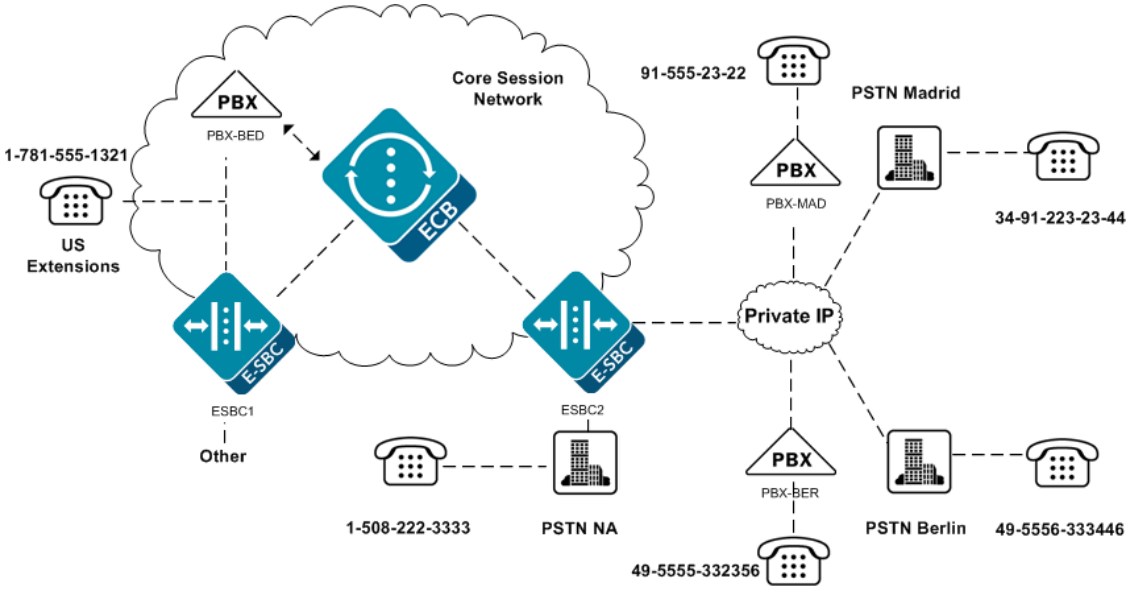
Route number	Description
#1	For traffic destined to Spain preceded with a "34" (Spain Country Code) and sourced anywhere, send to the PSTN agent in Madrid.
#2	For traffic destined to the PSTN in Madrid, use ESBC2.
#3	For traffic destined within the enterprise in Madrid, use ESBC2.
#4	For traffic destined to Germany preceded with a "49" (German Country Code) and sourced anywhere, send to the PSTN agent in Berlin.
#5	For traffic destined to the PSTN in Berlin, use ESBC2.
#6	For traffic destined within the enterprise in Berlin, use ESBC2.
#7	For traffic destined anywhere preceded with a "1" (US Country Code) and sourced anywhere, send to the North American PSTN agent.
#8	Default Route - If unable to determine any preferable route, use the most expensive route, which offloads traffic to the North American PSTN agent. Note the cost of 70, which is the highest cumulative cost of any other route set.

Note:

Routes 1, 4 and 7 are examples of tail hop routing, which keeps traffic within the enterprise network for as long as possible before issuing a call to Germany locally in Berlin.

Small Enterprise Model

The following diagram shows example extension numbers for the small enterprise model that must be reachable. The key characteristic for this model is the absence of overlap of dial patterns, called contexts, across branch locations.



The following tables define an example of the dial plan called `acme` for a small enterprise.

The following table shows dial patterns configured for the `acme` context.

Prefix	Pattern	Country Code	Replace Prefix	Replace URI	Go To Context
	1xxx	1	781555		
	2xxx	34	91555		
	3xxxxx	49	5555		

Applicable user database entries.

Number	Dialing Context	Agent
+17815551xxx		PBX-BED
+34915552xxx		PBX-MAD
+4955553xxxxx		PBX-BER

Child contexts inherit the rules of the corporate context, acme, and are also configured to inherit the geographic location contexts for the countries in which they reside.

When configuring dial patterns without using GoToContext, Communications Broker evaluates the pattern based on the configured sourceContext. Communications Broker first checks if the dial pattern matches the configured source context. If a match is found, the call will proceed using this context. If no match is found, it then checks the parent context, call will proceed using that context. This behavior aligns with Small Enterprise Model.

Regarding the configured routes, numbers preceded with the appropriate country code that do not match these dial patterns go to the PSTN of their respective countries.

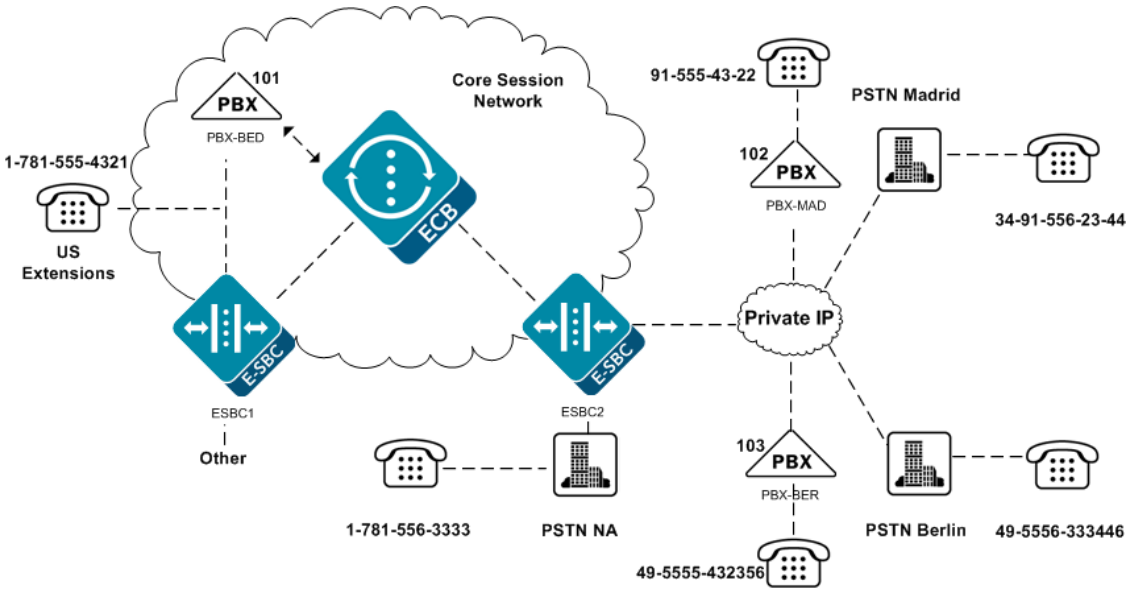
Reviewing the fields that do not require configuration is equally instructive. Based on the inheritance of rules from parent contexts, a means of basic access is available throughout the enterprise. In addition, based on the higher precedence of child context configurations, these fields are available to you to configure for special requirements at the separate branches.

This configuration generates the results shown in the following table.

From	Dial String	Transformation	Result
acme	1321	+17815551321	Call is directed to PBX-BED as e.164
acme	2322	+345552322	Call is directed to PBX-MAD as e.164
acme	332356	+495555332356	Call is directed to PBX-BER as e.164

Large Enterprise Model - v2

The following diagram shows example extension numbers for the medium to large enterprise model that must be reachable. The key characteristic for this model is the overlap of dial patterns, called contexts, across branch locations.



In this example, specific dial patterns are configured for the `acme` context that each child context inherits. These dial patterns provide the Oracle Enterprise Communications Broker with the means of distinguishing between branches, even though the extension patterns still begin with the same number.

Dial Patterns Configured for the `acme` Context

Prefix	Pattern	Country Code	Replace Prefix	Replace URI	Go To Context
101					acme.bedford
102					acme.madrid
103					acme.madrid
555				helpdesk@acme.com	

When configuring dial patterns with the GoToContext, Communications Broker determines the best match by evaluating the longest common parentage from sourceContext. This behavior aligns with the Large Enterprise Model - v2.

Each child context inherits these dial patterns from the acme context, providing you with a means of hierarchically assigning context.

In contrast to the small enterprise model, note the use of dial patterns specific to each branch. These contexts work in conjunction with the parent context's parents, delivering signaling with the enterprise's own prefixes to the branch's PBX.

Dial patterns configured for the `acme.bedford` context

Prefix	Pattern	Country Code	Replace Prefix	Replace URI	Go To Context
	4xxx		781555		

Dial Patterns Configured for the `acme.madrid` Context

Prefix	Pattern	Country Code	Replace Prefix	Replace URI	Go To Context
	4xxx		91555		

Dial Patterns Configured for the `acme.berlin` Context

Prefix	Pattern	Country Code	Replace Prefix	Replace URI	Go To Context
	4xxxxx		555		
111			berlinhelpdesk @acme.com		

Applicable user database entries

Number	Dialing Context	Agent
+17815551xxx		PBX-BED
+34915552xxx		PBX-MAD
+4955553xxxxx		PBX-BER

In contrast to the small enterprise model, note the use of dial patterns specific to each branch.

This configuration provides the results shown in the following table.

From	Dial String	Transformation	Result
acme.bedford	4321	+17815554321	Call is directed to PBX-BED as e.164
acme.bedford	1024322	+34915554322	Call is directed to PBX-MAD as e.164
acme.bedford	103432356	+495555432356	Call is directed to PBX-BER as e.164
acme.madrid	4322	+34915554322	Call is directed to PBX-MAD as e.164
acme.madrid	1014321	+17815554321	Call is directed to PBX-BED as e.164
acme.madrid	103432356	+495555432356	Call is directed to PBX-BER as e.164

From	Dial String	Transformation	Result
acme.berlin	432356	+495555432356	Call is directed to PBX-BER as e.164
acme.berlin	1014321	+17815554321	Call is directed to PBX-BED as e.164
acme.berlin	1024322	+34915554322	Call is directed to PBX-MAD as e.164

Emergency Dial Configurations

To dial emergency numbers, you add dial patterns to the corporate context child context for the emergency numbers. Configure these dial patterns with a Replace URI transformation that inserts the RFC 5031 compliant URN for emergency services. The following contexts and routes show examples of such a configuration. The last table in this topic shows the results of this configuration.

Dial Patterns Configured for the `acme.bedford` context

Prefix	Pattern	Country Code	Replace Prefix	Replace URI	Go To Context
	911			URN:service:sos	

Dial Patterns Configured for the `acme.madrid` context

Prefix	Pattern	Country Code	Replace Prefix	Replace URI	Go To Context
	112			URN:service:sos	

Dial Patterns Configured for the `acme.berlin` context

Prefix	Pattern	Country Code	Replace Prefix	Replace URI	Go To Context
	112			URN:service:sos	

In addition to the preceding dial patterns, the system needs three new routes.

Route #	Source Agent	Calling #	Called #	Dest Agent	Route	Cost
#1	*	1*	"service:sos"	*	PSTN-NA	0
#2	*	34*	"service:sos"	*	PSTN-MAD	0
#3	*	49*	"service:sos"	*	PSTN-BER	0

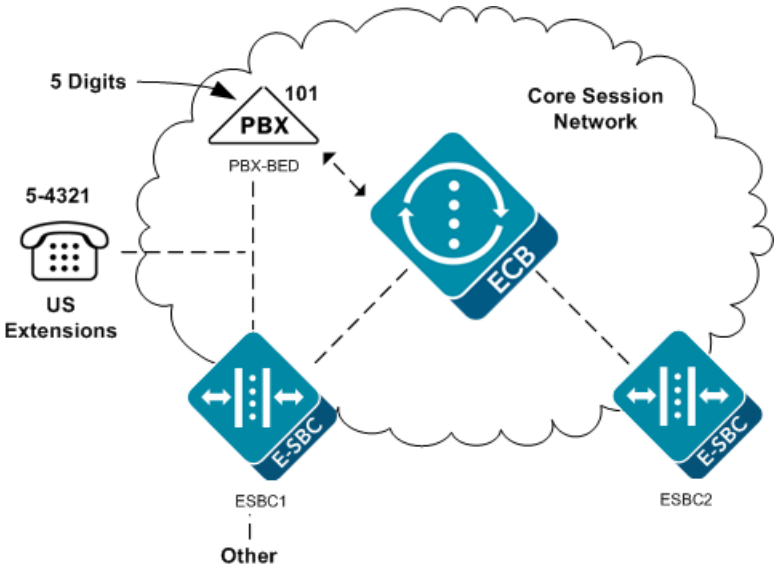
This configuration produces the following results.

From	Dial String	Transformations	Result
acme.bedford	911	URN:service:sos	Call is directed to PSTN-NA as emergency URN

From	Dial String	Transformations	Result
acme.madrid	112	URN:service:sos	Call is directed to PSTN-MAD as emergency URN
acme.berlin	112	URN:service:sos	Call is directed to PSTN-BER as emergency URN

Alternate Translation Modes

As described in agent configuration instructions, a translation mode specifies the format required by that agent. The configuration normally applies to a PBX when it is the last agent in the path to the user.



This example implements a different configuration for PBX-BED, as follows.

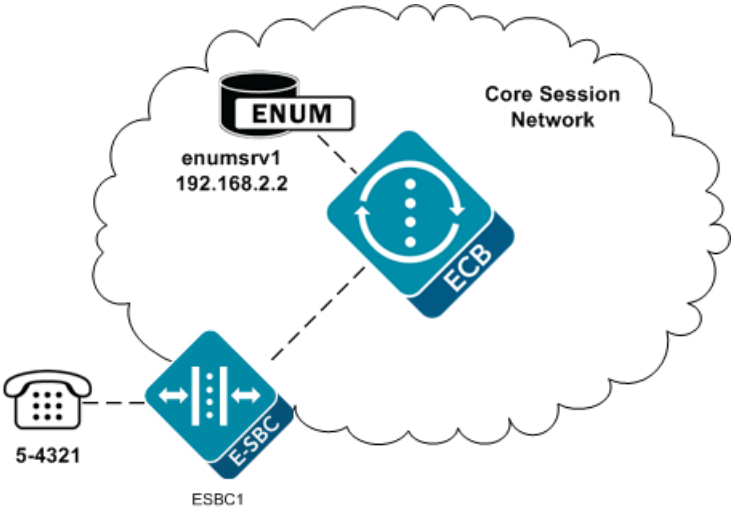
Hostname	Source Context	Number Translation Mode	Number of Digits	Prepend Prefix
PBX-BED		n-digit-dialing	5	

This configuration produces the following results.

From	Dial String	Transformation	Result
acme.bedford	54321	+17815554321	Call is directed to PBX-BED as a 5 digit number (54321)
acme.madrid	10154321	+17815554321	Call is directed to PBX-BED as a 5 digit number (54321)
acme.berlin	10154321	+17815554321	Call is directed to PBX-BED as a 5 digit number (54321)

ENUM Example Configuration

This example provides for the requirement that the signaling make an ENUM dip to resolve some element of the overall signaling path. The configuration must be able to provide ENUM resolutions to provide for the following deployment.



This configuration assumes the agent ESBC1 does not have a configured IP address. In addition, all numbers beginning with 5 are routed to ESBC1. To provide resolution for ESBC1, create an ENUM configuration as follows. In this example, the ENUM configuration includes only one server.

Hostname	Domain	Servers	Number Trans Mode	Number of Digits
enumsrv1	acme.com	192.168.2.2		

Configure a route for all 5-digit extensions beginning with a 5 to the agent named ENUM.

Route #	Source Agent	Calling number	Called number	Dest Agent	Route	Cost
#1		*	5xxxx	*	enum:enumsrv1	

The following table explains the purpose of the route in the preceding table.

Route #	Description
#1	For traffic destined to a five-digit number beginning with 5, go to ENUM server to resolve the address of ESBC1.

20

Format of Exported Text Files

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

Exported file examples include:

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

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

Exporting Files

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

You can export any of the following to a file:

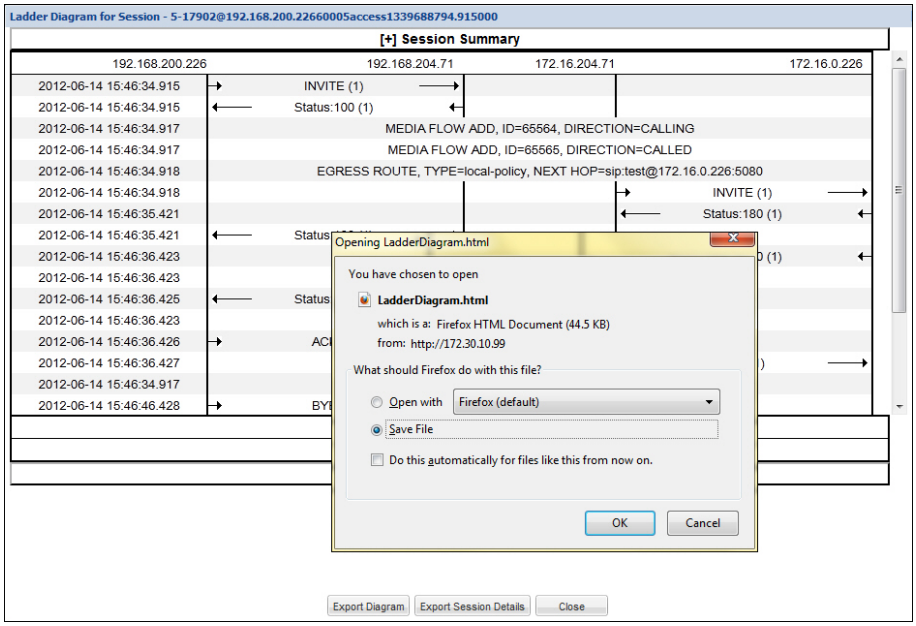
From the Sessions, Registrations, Subscriptions, and Notable Events Reports:

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

From the Ladder Diagram:

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

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



Session Summary Exported Text File

The following example shows a Session Summary text file exported from the GUI.

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

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

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

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

```
Igress Dest Address: 172.16.34.225
Igress Dest Port: 5060
Egress Source Address: 192.168.34.225
Egress Source Port: 5060
Egress Destination Address: 192.168.34.17
Egress Destination Port: 5060
Igress Realm: access
Egress Realm: backbone
Igress NetworkIf: access
Egress NetworkIf: backbone
```

Session Details Exported Text File

The following example shows a Session Details text file exported from the GUI.

```
Session Details:

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

INVITE sip:service@172.16.34.225:5060 SIP/2.0
Via: SIP/2.0/UDP 172.16.34.16:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>
Call-ID: 1-668@172.16.34.16
CSeq: 1 INVITE
Contact: sip:sipp@172.16.34.16:5060
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 135

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

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

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

----MBCD Evt
Nov 3 08:50:56.862 On 127.0.0.1:2945 sent to 127.0.0.1:2944
mbcdEvent=FLOW ADD
FlowCollapsed=enabled
```

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

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

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

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

v=0
o=user1 53655765 2353687637 IN IP4 192.168.34.225
s=-
```

```
c=IN IP4 192.168.34.225
t=0 0
m=audio 20004 RTP/AVP 0
a=rtpmap:0 PCMU/8000

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

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

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

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

----------------------------------------
Nov 3 08:50:56.872 On [1:0]192.168.34.225:5060 received from
192.168.34.17:5060
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.34.225:5060;branch=z9hG4bK4od0io20183g8ssv32f1.1
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 1 INVITE
Contact: <sip:192.168.34.17:5060;transport=UDP>
Content-Type: application/sdp
Content-Length: 137
v=0
o=user1 53655765 2353687637 IN IP4 192.168.34.17
s=-
c=IN IP4 192.168.34.17
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000

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

mbcdEvent=FLOW MODIFY
FlowCollapsed=enabled
FlowDirection=CALLING
```

```
FlowID=65541
MediaFormat=0
MediaReleased=disabled
MediaType=audio/PCMU/
OtherFlowID=0
TOSBits=0
InputSourcev4Addr=
InputSourcePort=0
InputDestv4Addr=172.16.34.225
InputDestPort=10004
OutputSourcev4Addr=192.168.34.225
OutputDestv4Addr=192.168.34.17
OutputDestPort=6000
InputRealm=access
OutputRealm=backbone

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

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

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

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

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

ACK sip:192.168.34.17:5060;transport=UDP SIP/2.0
Via: SIP/2.0/UDP 192.168.34.225:5060;branch=z9hG4bK48k3k1301ot00ssvf1v0.1
From: sipp <sip:sipp@172.16.34.16:5060>;tag=1
```

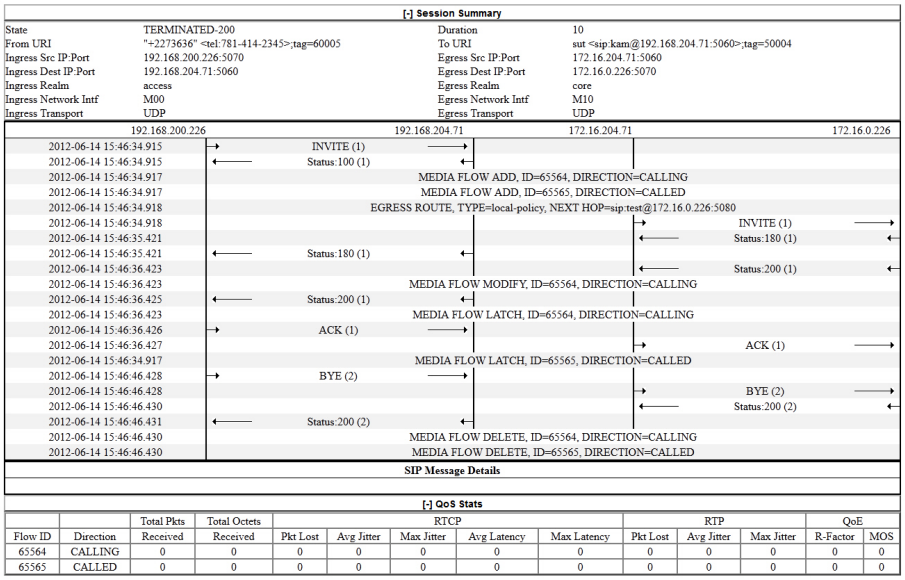
```
To: sut <sip:service@172.16.34.225:5060>;tag=2578
Call-ID: 1-668@172.16.34.16
CSeq: 1 ACK
Contact: <sip:sipp@192.168.34.225:5060;transport=udp>
Max-Forwards: 69
Subject: Performance Test
Content-Length: 0

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

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

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

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

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

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

```
Contact: <sip:service@172.16.34.225:5060;transport=udp>
Content-Length: 0

----MBCD Evt
Nov 3 08:51:01.891 On 127.0.0.1:2945 sent to 127.0.0.1:2944

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

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

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

State: TERMINATED-200
Duration: 5
From URI: sipp &lt;sip:sipp@172.16.34.16:5060&gt;;tag=1
To URI: sut &lt;sip:service@172.16.34.225:5060&gt;;tag=2578
Ingress Src Address: 172.16.34.16
Igress Src Port: 5060
Igress Dest Address: 172.16.34.225
Igress Dest Port: 5060
```

```
Egress Source Address: 192.168.34.225
Egress Source Port: 5060
Egress Destination Address: 192.168.34.17
Egress Destination Port: 5060
Igress Realm: access
Egress Realm: backbone
Igress NetworkIf: access
Egress NetworkIf: backbone
```

Ladder Diagram Exported HTML File

The following example shows a Ladder Diagram for a session, exported to an HTML file from the GUI.



[-] Session Summary

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

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

SIP Message Details

[-] QoS Stats

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

21

SIP HMR (Header Manipulation Rules)

SIP header manipulation can also be configured in a way that makes it possible to manipulate the headers in SIP messages both statically and dynamically. Using this feature, you can edit response headers or the Request-URI in a request, and change the status code or reason phrase in SIP responses.

Static SIP Header and Parameter Manipulation allows you to set up rules in your Oracle Enterprise Communications Broker configuration that remove and/or replace designated portions of specified SIP headers. SIP HMR allows you to set up dynamic header manipulation rules, meaning that the Oracle Enterprise Communications Broker has complete control over alterations to the header value. More specifically:

- The Oracle Enterprise Communications Broker can search header for dynamic content or patterns with the header value. It can search, for example, for all User parts of a URI that begin with 617 and end with 5555 (e.g., 617...5555).
- The Oracle Enterprise Communications Broker can manipulate any part of a patterns match with any part of a SIP header. For example, 617 123 5555 can become 617 231 5555 or 508 123 0000, or any combination of those.

To provide dynamic header manipulation, the Oracle Enterprise Communications Broker uses regular expressions to provide a high degree of flexibility for this feature. This allows you to search a specific URI when you do not know that value of the parameter, but want to use the matched parameter value as the header value. It also allows you to preserve matched sections of a pattern, and change what you want to change.

You can apply header manipulation to session agents, SIP interfaces, and realms. You do so by first setting up header manipulations rules, and then applying them in the configurations where they are needed. Within the header manipulation rules, there are sets of element rules that designate the actions that need to be performed on a given header.

Each header rule and each element rule (HMR) have a set of parameters that you configure to identify the header parts to be manipulated, and in what way the Oracle Enterprise Communications Broker is to manipulate them. These parameters are explained in detail, but the parameter that can take regular expression values is **match-value**. This is where you set groupings that you want to store, match against, and manipulate.

Generally, you set a header rule that will store what you want to match, and then you create subsequent rules that operate on this stored value. Because header rules and element rules are applied sequentially, it is key to note that a given rule performs its operations on the results of all the rules that you have entered before it. For example, if you want to delete a portion of a SIP header, you would create Rule 1 that stores the value for the purpose of matching, and then create Rule 2 that would delete the portion of the header you want removed. This prevents removing data that might be used in the other header rules.

Given that you are using regular expression in this type of configuration, this tightly sequential application of rules means that you must be aware of the results to be yielded from the application of the regular expressions you enter. When you set a regular expression match value for the first rule that you enter, the Oracle Enterprise Communications Broker takes the results of that match, and then a second rule might exist that tells the Oracle Enterprise Communications Broker to use a new value if it the second rule's match value finds a hit (and only 10 matches, 0-9, are permitted) for the results (yield) from applying the first rule.

Consider the example of the following regular expression entry made for a **match-value** parameter:'Trunk(.+)', which might be set as that match value in the first rule you configure. Given a SIP element rule called uri-param and the param-name tgid, it can yield two values:

- Grouping 0—The entire matching string (Trunk1)
- Grouping 1—The grouping (1)

In turn, these groupings can be referenced in an element rule by using this syntax:

```
$<header rule name >.$<element rule name.$<value>
```

Additional syntax options that can be used with this feature are:

- $headerName['['index']']
- $headerName['['index']'][.$index]
- $headerName['['index']'][.$elementName]
- $headerName['['index']'][.$elementName][.$index]

Guidelines for Header and Element Rules

Header rules and element rules share these guidelines:

- References to groupings that do not exist result in an empty string.
- References to element rule names alone result in a Boolean condition of whether the expression matched or not.
- A maximum of ten matches are allowed for a regular expression. Match 0 (grouping 0) is always the match of the entire matching string; subsequent numbers are the results for other groups that match.

Splitting and Joining Headers

To simplify header manipulation processes, the Oracle Enterprise Communications Broker provides a means of combining or breaking apart header strings that actually consist of multiple headers. An example application would be to separate headers that another SIP device joined together into a single string. This would allow header manipulation to work on each distinct header, after which the system could re-combine the headers, making the forwarded output consistent with the initial message.

Some SIP devices combine multiple headers into a single header, with each distinct header separated by a comma. This is not precluded by RFC 3261. The user can configure a header manipulation to separate these headers prior to performing the manipulation using the **split-headers** command. The user can also configure the system to join headers together after a manipulation is complete using the **join-headers** command. Split and join functions do not have to co-exist within a single header manipulation.

Precedence

The Oracle Enterprise Communications Broker applies SIP header rules in the order you have entered them. This guards against the Oracle Enterprise Communications Broker removing data that might be used in the other header rules.

This ordering also provides you with ways to use manipulations strategically. For example, you might want to use two rules if you want to store the values of a regular expression. The first

rule would store the value of a matched regular expression, and the second could delete the matched value.

In addition to taking note of the order in which header rules are configured, you now must also configure a given header rule prior to referencing it. For example, you must create Rule1 with the action store for the Contact header BEFORE you can create Rule2 which uses the stored value from the Contact header.

Duplicate Header Names

If more than one header exists for a configured header-name, the Communications Broker stores each value in an array whose index starts at 0. To reference those values, use the syntax `$<header-name>[<index>]`.

Add a trailing `[<index>]` value after the header-name parameter to represent the specific instance of the header on which to operate. Additional stored header values are indexed in the order in which they appear within the SIP message, and there is no limit to the index. The Communications Broker takes no action if the header does not exist.

Use index 0 to reference the first header. In addition to numerical values, possible index values are:

- `*` The Communications Broker references all headers.
- `^` The Communications Broker references the last stored header in the header rule.

Note that the header instance functionality has no impact on HMR's add action, and you cannot use this feature to insert headers into a specific location. Headers are added to the end of the list, except that Via headers are added to the top.

The order of stored headers differs from the order in which the Communications Broker adds headers on outgoing messages. When the header contains multiple values, the order of the values cannot be guaranteed. As a result, your regex patterns must not assume any specific order to the values of a header on an outgoing message.

Performing HMR on a Specific Header

HMR has been enhanced so that you can now operate on a specific instance of a given header. The syntax you use to accomplish this is similar to that you used to refer to a specific header rule stored value instance.

Using the header-name parameter, you can now add a trailing [<index>] value after the header name. This [<index>] is a numerical value representing the specific instance of the header on which to operate. However, the Oracle Enterprise Communications Broker takes no action if the header does not exist. You can also use the caret (^) to reference the last header of that type (if there are multiple instances)

The count for referencing is zero-based, meaning that the first instance of the header counts as 0.

Note that the header instance functionality has no impact on HMR's add action, and you cannot use this feature to insert headers into a specific location. Headers are added to the end of the list, except that Via headers are added to the top.

Multiple SIP HMR Sets

In general you use SIP HMR by configuring rules and then applying those rules to session agents, realms, or SIP interfaces in the inbound or outbound direction. In addition, the Oracle

Enterprise Communications Broker has a set method for how certain manipulation rules take precedence over others. For instance, inbound SIP manipulation rules defined in a session agent take precedence over any configured for a realm, and the rules for a realm take precedence over SIP interface manipulation rules.

The multiple SIP HMR feature gives you the ability to:

- Apply multiple inbound and outbound manipulations rules to a SIP message
- Provision the order in which the Oracle Enterprise Communications Broker applies manipulation rules

The **action** parameter in the header rules configuration now takes the value sip-manip. When you set the parameter to sip-manip, you then configure the **new-value** parameter with the name of a SIP manipulation rule that you want to invoke. The values for the **match-value**, **comparison-type**, and **methods** parameters for invoked rule are all supported. This means that the manipulation defined by the rules identified in the **new-value** parameter are carried out when the values for the **match-value**, **comparison-type**, and **methods** parameters are true.

The relationship between manipulation rules and manipulation rule sets is created once you load your configuration, meaning that the order in which you enter them does not matter. It also means that the Oracle Enterprise Communications Broker cannot dynamically perform validation as you enter rules, so you should use the ACLI **verify-config** command to confirm your manipulation rules contain neither invalid nor circular references. Invalid references are those that point to SIP manipulation rules that do not exist, and circular references are those that create endless loops of manipulation rules being carried out over and over. If you load a configuration exhibiting either of these issues, the Oracle Enterprise Communications Broker forces the action value for the rule to **none** and the rule will not be used.

MIME Support

Using the SIP HMR feature set, you can manipulate MIME types in SIP message bodies. While you can manipulate the body of SIP messages or a specific content type using other iterations of SIP HMR, this version gives you the power to change the MIME attachment of a specific type within the body by using regular expressions. To achieve this, you use the **find-replace-all** action type, which enables the search for a particular string and the replacement of all matches for that type. Although you use **find-replace-all** to manipulate MIME attachments, it can also be used to achieve other goals in SIP HMR.

Note that using **find-replace-all** might consume more system resources than other HMR types. Therefore this powerful action type should only be used when another type cannot perform the type of manipulation you require.

Manipulating MIME Attachments

Set the action type to find-replace-all to modify MIME attachments.

To manipulate a particular portion of the MIME attachment, for example when removing a certain attribute within the Content-Type of `application/sdp`, the Communications Broker needs to search the content multiple times because:

- SDP can have more than one media line
- The SIP message body can contain more than one `application/sdp`.

When the action type is find-replace-all, the Communications Broker treats the match-value as a regular expression and binds the comparison-type to pattern-rule, even if comparison-type is set to some other value. This type of action is both a comparison and action: for each regular

expression match within the supplied string, the Communications Broker substitutes the new value for that match.

Use subgroups to replace portions of the regular expression rather than the entire matched expression. The subgroup replacement syntax is formed by adding the string `[[:n:]]` to the end of the regular expression—where `n` is a number between 0 and 9. For example, setting the following parameters

```
action                  find-replace-all
match-value             sip:(user)@host[[:1:]]
new-value               bob
```

creates a new rule to replace only the user portion of the URI that searches for the regular expression and replaces all instances of the user subgroup with the value bob.

Setting the following parameters

```
action                  find-replace-all
match-value             0
new-value               1
```

creates a new rule to recursively replace all the 0 digits in a telephone number with 1. With this rule the user portion of a URI—or for any other string—with a value 1-781-308-4400 would be replaced as 1-781-318-4411.

If you leave the **new-value** parameter blank for **find-replace-all**, the Communications Broker replaces the matched sub-group with an empty string—an equivalent of deleting the sub-group match. You can also replace empty sub-groups, which is like inserting a value within the second sub-group match. For example, `user()@example.com[[:1:]]` with a configured new-value `_bob` yields `user_bob@host.com`.

Setting **find-replace-all** disables the following **parameter-type** values: **uri-param-name**, **uri-header-name**, and **header-param-name**. These values are unusable because the Communications Broker only uses case-sensitive matches for the match-value to find the parameter name within the URI. Since it can only be found by exact match, the Communications Broker does not support finding and replacing that parameter.

Escaped Characters

SIP HMR's support for escaped characters allows for searches for values you would be unable to enter yourself. Because they are necessary to MIME manipulation, support for escaped characters now includes:

- \f
- \n
- \r
- \t
- \v

New Reserved Word

To allow you to search for carriage returns an new lines, the SIP HMR MIME feature also adds support for the reserved word $CRLF. Because you can search for these value and replace

them, you also must be able to add them back in when necessary. Configuring $CRLF in the **new-value** parameter always resolves to /r/n, which you normally cannot otherwise enter through the ACLI.

About the MIME Value Type

To modify the MIME attachment, SIP HMR supports a **mime** value for the **type** parameter in the element rules configuration. Like the **status-code** and **reason-phrase** values, you can only use the **mime** type value against the Content-Type header.

When you set the element rule type to **mime**, you must also configure the **parameter-name** with a value. This step is a requirement because it sets the content-type the Oracle Enterprise Communications Broker (Communications Broker) manipulates in a specific part of the MIME attachment. You cannot leave this parameter blank because the Communications Broker does not let you save the configuration if you do. When you use the **store** action on a multi-part MIME attachment that has different attachment types, the Communications Broker stores the final instance of the content-type because it does not support storing multiple instances of element rule stored values.

If you do not know the specific content-type where the Communications Broker will find the **match-value**, you can use a wildcard in the **parameter-name** by setting with the asterisk (*) as a value. You cannot, set partial content-types (for example, application/*). So configured, the Communications Broker loops through the MIME attachment's content types.

You can set the additional **action** types listed in the following table with the described result:

Action Type	Description
delete-element	Removes the matched mime-type from the body. If this is the last mime-type within in message body, the Oracle Enterprise Communications Broker removes the Content-Type header.
delete-header	Removes all body content and removes the Content-Type header.
replace	Performs a complete replacement of the matched mime-type with the new-value you configure.
find-replace-all	Searches the specifies mime-type's contents and replaces all matching regular expressions with the new-value you configure
store	Stores the final instance of the content-type (if there are multi-part MIME attachments of various attachment types)
add	Not supported

MIME manipulation does not support manipulating headers in the individual MIME attachments. For example, the Communications Broker cannot modify the Content-Type given a portion of a message body such as the following one:

```
--boundary-1
Content-Type: application/sdp
v=0
o=use1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 10000 RTP/AVP 8
a=rtpmap:8 PCMA/8000/1
a=sendrecv
a=ptime:20
a=maxptime:200
```

Back Reference Syntax

You can use back reference syntax in the **new-value** parameter for header and element rules configurations. Denoted by the use of $1, $2, $3, etc. (where the number refers to the regular expression's stored value), you can reference the header and header rule's stored value without having to use the header rule's name. It instead refers to the stored value of this rule.

For example, when these settings are in place:

- header-rule=changeHeader
- action=manipulate
- match-value=(.+)([^;])

you can set the **new-value** as sip:$2 instead of **sip:$changeHeader.$2**.

You can use the back reference syntax for:

- Header rule **action**s **manipulate** and **find-replace-all**
- Element rule **action**s **replace** and **find-replace-all**

Using back reference syntax simplifies your configuration steps because you do not need to create a store rule and then manipulate rule; the manipulate rule itself performs the store action if the **comparison-type** is set to **pattern-rule**.

Notes on the Regular Expression Library

In the regular expression library, the dot (.) character does not match new lines or carriage returns. Conversely, the not-dot does match new lines and carriage returns. This provides a safety mechanism preventing egregious backtracking of the entire SIP message body when there are no matches. Thus, the Oracle Enterprise Communications Broker (Communications Broker)reduces backtracking to a single line within the body. The Communications Broker also supports:

Syntax	Description
\s	Whitespace
\S	Non-whitespace
\d	Digits
\D	Non-digits
\R	Any \r, \n, \r\n
\w	Word
\W	Non-word
\A	Beginning of buffer
\Z	End of buffer
\	Any character including newline, in the event that the dot (.) is not

In addition, you can also use:

- Escaped character shortcuts (\w\W\S\s\d\D\R) operating inside brackets [...]

SIP Message-Body Separator Normalization

The Oracle Enterprise Communications Broker supports SIP with Multipurpose Internet Mail Extension (MIME) attachments — up to a maximum payload size of 64KB — and has the

ability to allow more than the required two CRLFs between the SIP message headers and the multipart body's first boundary. The first two CRLFs that appear in all SIP messages signify the end of the SIP header and the separation of the header and body of the message, respectively. Sometimes additional extraneous CRLFs can appear within the preamble before any text.

The Oracle Enterprise Communications Broker works by forwarding received SIP messages regardless of whether they contain two or more CRLFs. Although three or more CRLFs are legal, some SIP devices do not accept more than two.

The solution to ensuring all SIP devices accept messages sent from the Oracle Enterprise Communications Broker is to strip all CRLFs located at the beginning of the preamble before the appearance of any text, ensuring that there are no more than two CRLFs between the end of the last header and the beginning of the body within a SIP message. You enable this feature by adding the new stripPreambleCrlf option to the global SIP configuration.

To enable the stripping of CRLFs in the preamble, add the **+stripPreambleCrlf** option to SIP Options.

SIP Header Pre-Processing HMR

By default, the Oracle Enterprise Communications Broker performs in-bound SIP manipulations after it carries out header validation. Adding the **inmanip-before-validate** option in the global SIP configuration allows the Oracle Enterprise Communications Broker to perform HMR on received requests prior to header validation. Because there are occaisional issues with other SIP implementations—causing invalid headers to be used in messages they send to the Oracle Enterprise Communications Broker—it can be beneficial to use HMR to remove or repair these faulty headers before the request bearing them are rejected.

When configured to do so, the Oracle Enterprise Communications Broker performs pre-validation header manipulation immediately after it executes the top via check. Inbound SIP manipulations are performed in order of increasing precedence: SIP interface, realm, and session agent.

The fact that the top via check happens right before the Oracle Enterprise Communications Broker carries out pre-validation header manipulations means that you cannot use this capability to repairs the first via header if it is indeed invalid. If pre-validation header manipulation were to take place at another time during processing, it would not be possible to use it for SIP session agents. The system learns of matching session agents after top via checking completes.

For logistical reasons, this capability does not extend to SIP responses. Inbound manipulation for responses cannot be performed any sooner that it does by default, a time already preceding any header validation.

To enable SIP header pre-processing, add the **+inmanip-before-validate** option to SIP Options.

Best Practices

The following list describes practices that Oracle recommends you follow for successful implementation of Header Manipulation Rules.

- Define all storage rules first.
  This recommendation is made because each subsequent header rule processes against the same SIP message, so each additional header rules works off of the results from the application of the rule that precedes it.

In general, you want to store values from the original SIP header rather than from the iteratively changed versions.

- Implement rules at the element rule rather than the header rule level. Header rules should only be a container for element rules.
- When you are creating rules to edit a header, add additional element rules to modify a single header rather than try to create multiple header rules each with one element rule. That is, create multiple element rules within a header rule rather than creating multiple header rules.
- Do not use header or element rule names that are all capital letters (for example, $IP_ADDRESS). Capital letters refer to predefined rules that are used as macros, and they might conflict with a name that uses capital letters.

About Regular Expressions

Two of the most fundamental ideas you need to know to work with regular expressions and with Header Manipulation Rules include:

- Regular expressions are a way of creating strings to match other string values.
- You can use groupings to create stored values on which you can then operate.

To learn more about regex, you can visit the following Web site, which has information and tutorials that can help to get you started:http://www.regular-expressions.info/.

Many of the characters you can type on your keyboard are literal, ordinary characters, which present their actual value in the pattern. Some characters have special meaning, and they instruct the regex function (or engine which interprets the expressions) to treat the characters in designated ways. The following table outlines these “special characters” or metacharacters.

Character	Name	Description
.	dot	Matches any one character, including a space; it will match one character, but there must be one character to match. Literally a . (dot) when bracketed ([]), or placed next to a \ (backslash).
*	star/asterisk	Matches one or more preceding character (0, 1, or any number), bracketed carrier class, or group in parentheses. Used for quantification. Typically used with a . (dot) in the format .* to indicate that a match for any character, 0 or more times. Literally an * (asterisk) when bracketed ([]).
+	plus	Matches one or more of the preceding character, bracketed carrier class, or group in parentheses. Used for quantification. Literally a + (plus sign) when bracketed ([]).
|	bar/vertical bar/pipe	Matches anything to the left or to the right; the bar separates the alternatives. Both sides are not always tried; if the left does not match, only then is the right attempted. Used for alternation.

Character	Name	Description
{	left brace	Begins an interval range, ended with } (right brace) to match; identifies how many times the previous singles character or group in parentheses must repeat. Interval ranges are entered as minimum and maximums ({minimum,maximum}) where the character/group must appear a minimum of times up to the maximum. You can also use these character to set magnitude, or exactly the number of times a character must appear; you can set this, for example, as the minimum value without the maximum ({minimum,}).
?	question mark	Signifies that the preceding character or group in parentheses is optional; the character or group can appear not at all or one time.
^	caret	Acts as an anchor to represent the beginning of a string.
$	dollar sign	Acts as an anchor to represent the end of a string.
[	left bracket	Acts as the start of a bracketed character class, ended with the] (right bracket). A character class is a list of character options; one and only on of the characters in the bracketed class must appear for a match. A - (dash) in between two character enclosed by brackets designates a range; for example [a-z] is the character range of the lower case twenty-six letters of the alphabet. Note that the] (right bracket) ends a bracketed character class unless it sits directly next to the [(left bracket) or the ^ (caret); in those two cases, it is the literal character.
(	left parenthesis	Creates a grouping when used with the) (right parenthesis). Groupings have two functions: They separate pattern strings so that a whole string can have special characters within it as if it were a single character. They allow the designated pattern to be stored and referenced later (so that other operations can be performed on it).

Expression Building Using Parentheses

You can now use parentheses (())when you use HMR to support order of operations and to simplify header manipulation rules that might otherwise prove complex. This means that expressions such as (sip + urp) - (u + rp) can now be evaluated to sip. Previously, the same expression would have evaluated to sipurprp. In addition, you previously would have been required to create several different manipulation rules to perform the same expression.

Configuration Examples

This section shows you several configuration examples for HMR. This section shows the configuration for the various rules that the Oracle Enterprise Communications Brokerapplied, and sample results of the manipulation. These examples present configurations as an entire list of fields and settings for each ruleset, nested header rules and nested element rules. If a

field does not have any operation within the set, the field is shown with the setting at the default or blank.

Example 1 Removing Headers

For this manipulation rule, the Oracle Enterprise Communications Broker removes the Custom header if it matches the pattern rule. It stores the defined pattern rule for the goodBye header. Finally, it removes the goodBye header if the pattern rule from above is a match.

This is a sample of the configuration:

```
sip-manipulation
        name                            removeHeader
        header-rule
                name                            removeCustom
                header-name                     Custom
                action                          delete
                comparison-type                 boolean
                match-value                     ^This is my.*
                msg-type                        request
                new-value
                methods                         INVITE
        header-rule
                name                            goodByeHeader
                header-name                     Goodbye
                action                          store
                comparison-type                 boolean
                match-value                     ^Remove (.+)
                msg-type                        request
                new-value
                methods                         INVITE
        header-rule
                name                            goodBye
                action                          delete
                comparison-type                 pattern-rule
                match-value                     $goodByeHeader
                msg-type                        request
                new-value
                methods                         INVITE
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK0g639r10fgc0aakk26s1.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDc1rm601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SDc1rm601-d01673bcacfcc112c053d95971330335-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060>
```

```
Max-Forwards: 69
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 140
```

Example 2 Manipulating the Request URI

For this manipulation rules, the Oracle Enterprise Communications Broker stores the URI parameter tgid in the Request URI. Then if the pattern rule matches, it adds a new header (x-customer-profile) with the a new header value tgid to the URI parameter in the request URI.

This is a sample of the configuration:

```
sip-manipulation
        name                            CustomerTgid
        header-rule
                name                            ruriRegex
                header-name                     request-uri
                action                          store
                comparison-type                 pattern-rule
                match-value
                msg-type                        request
new-value
                methods                         INVITE
                element-rule
                        name                            tgidParam
                        parameter-name                  tgid
                        type                            uri-param
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value
                        new-value
header-rule
                name                            addCustomer
                header-name                     X-Customer-Profile
                action                          add
                comparison-type                 pattern-rule
                match-value                     $ruriRegex.$tgidParam
                msg-type                        request
                new-value                       $ruriRegex.$tgidParam.$0
                methods                         INVITE
header-rule
                name                            delTgid
                header-name                     request-uri
                action                          manipulate
                comparison-type                 pattern-rule
                match-value                     $ruriRegex.$tgidParam
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            tgidParam
                        parameter-name                  tgid
                        type                            uri-param
```

```
                        action                          delete-element
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-
value                    $ruriRegex.$tgidParam.$0
                        new-value
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060 SIP/2.0
    Message Header
Via: SIP/2.0/UDP 192.168.200.61:5060;branch=z9hG4bK0g6plv3088h03acgh6c1.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDc1rg601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SDc1rg601-f125d8b0ec7985c378b04cab9f91cc09-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
        X-Customer-Profile: 123
```

Example 3 Manipulating a Header

For this manipulation rule, the Oracle Enterprise Communications Brokerstores the pattern matches for the Custom header, and replaces the value of the Custom header with a combination of the stored matches and new content.

This is a sample of the configuration:

```
sip-manipulation
        name                            modCustomHdr
        header-rule
                name                            customSearch
                header-name                     Custom
                action                          store
                comparison-type                 pattern-rule
                match-value                     (This is my )(.+)( header)
                msg-type                        request
                new-value
                methods                         INVITE
header-rule
                name                            customMod
                header-name                     Custom
                action                          manipulate
                comparison-type                 pattern-rule
                match-value                     $customSearch
                msg-type                        request
```

```
                new-value
methods                         INVITE
                element-rule
                        name                            hdrVal
                        parameter-name                  hdrVal
                        type                            header-value
                        action                          replace
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
new-value                       $customSearch.$1+edited+$customSearch.$3
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK20q2s820boghbacgs6o0.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDe1ra601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SDe1ra601-4bb668e7ec9eeb92c783c78fd5b26586-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my edited header
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
```

Example 4 Storing and Using URI Parameters

For this manipulation rule, the Oracle Enterprise Communications Broker stores the value of the URI parameter tag from the From header. It also creates a new header FromTag with the header value from the stored information resulting from the first rule.

This is a sample of the configuration:

```
sip-manipulation
        name                            storeElemParam
        header-rule
                name                            Frohmr
                header-name                     From
                action                          store
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            elementRule
```

```
                        parameter-name                  tag
                        type                            uri-param
                        action                          store
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value
header-rule
                name                            newHeader
                header-name                     FromTag
                action                          add
                comparison-type                 pattern-rule
                match-value                     $FromHR.$elementRule
                msg-type                        any
                new-value                       $FromHR.$elementRule.$0
                methods
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK4oda2e2050ih7acgh6c1.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDf1re601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SDf1re601-f85059e74e1b443499587dd2dee504c2-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
Content-Length: 140
        FromTag: 1
```

Example 5 Manipulating Display Names

For this manipulation rule, the Oracle Enterprise Communications Broker sores the display name from the Display header. It replaces the two middle characters of the original display name with a new string. Then is also replaces the From header's display name with "abc 123" if it matches sipp.

This is a sample of the configuration:

```
sip-manipulation
        name                            modDisplayParam
        header-rule
                name                            storeDisplay
                header-name                     Display
                action                          store
```

```
                comparison-type              case-sensitive
                match-value
                msg-type                     request
                new-value
                methods                      INVITE
                element-rule
                        name                         displayName
                        parameter-name               display
                        type                         uri-display
                        action                       store
                        match-val-type               any
comparison-type                pattern-rule
                        match-value                  (s)(ip)(p )
                        new-value
header-rule
                name                         modDisplay
                header-name                  Display
                action                       manipulate
                comparison-type              case-sensitive
                match-value
                msg-type                     request
                new-value
                methods                      INVITE
                element-rule
                        name                         modRule
                        parameter-name               display
                        type                         uri-display
                        action                       replace
                        match-val-type               any
                        comparison-type              pattern-rule
                        match-
value                    $storeDisplay.$displayName
                        new-
value                       $storeDisplay.$displayName.$1+lur+$storeDisplay.$di
splayName.$3
header-rule
                name                         modFrom
                header-name                  From
                action                       manipulate
                comparison-type              pattern-rule
                match-value
                msg-type                     request
                new-value
                methods                      INVITE
                element-rule
                        name                         fromDisplay
                        parameter-name
                        type                         uri-display
                        action                       replace
                        match-val-type               any
                        comparison-type              pattern-rule
                        match-value                  sipp
                        new-value                    "\"abc 123\" "
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK681kot109gp04acgs6o0.1
        From: "abc 123" <sip:sipp@192.168.1.60:5060>;tag=SD79ra601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SD79ra601-a487f1259e2370d3dbb558c742d3f8c4-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: slurp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
```

Example 6 Manipulating Element Parameters

For this more complex manipulation rule, the Oracle Enterprise Communications Broker:

- From the Display header, stores the display name, user name, URI parameter up, and header parameter hp
- Adds the header parameter display to the Params header, with the stored value of the display name from the first step
- Add the URI parameter user to the Params header, with the stored value of the display name from the first step
- If the URI parameter match succeeds in the first step, replaces the URI parameter up with the Display header with the value def
- If the header parameter match succeeds in the first step, deletes the header parameter hp from the Display header

This is a sample of the configuration:

```
sip-manipulation
        name                            elemParams
        header-rule
                name                            StoreDisplay
                header-name                     Display
                action                          store
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            displayName
                        parameter-name
                        type                            uri-display
```

```
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value
                        new-value
element-rule
                        name                            userName
                        parameter-name                  user
                        type                            uri-user
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value
                        new-value
element-rule
                        name                            uriParam
                        parameter-name                  up
                        type                            uri-param
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value
                        new-value
element-rule
                        name                            headerParam
                        parameter-name                  hp
                        type                            header-param
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value
                        new-value
        header-rule
                name                            EditParams
                header-name                     Params
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            addHeaderParam
                        parameter-name                  display
                        type                            header-param
                        action                          add
match-val-type                 any
                        comparison-type                 case-sensitive
                        match-value
                        new-
value                      $StoreDisplay.$displayName.$0
                element-rule
                        name                            addUriParam
                        parameter-name                  user
                        type                            uri-param
                        action                          add
```

```
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                new-value
$StoreDisplay.$userName.$0
        header-rule
                name                            EditDisplay
                header-name                     Display
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            replaceUriParam
                        parameter-name                  up
                        type                            uri-param
                        action                          replace
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     $StoreDisplay.$uriParam
                        new-value                       def
                element-rule
                        name                            delHeaderParam
                        parameter-name                  hp
                        type                            header-param
                        action                          delete-element
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value $StoreDisplay.$headerParam
                        new-value
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK7okvei0028jgdacgh6c1.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SD89rm601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SD89rm601-b5b746cef19d0154cb1f342cb04ec3cb-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: sipp <sip:user@192.168.1.60:5060;up=def>
        Params: sipp <sip:sipp1@192.168.1.60:5060;user=user>;display=sipp
        Params: sipp <sip:sipp2@192.168.1.60:5060;user=user>;display=sipp
        Edit: disp <sip:user@192.168.1.60:5060>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
```

Example 7 Accessing Data from Multiple Headers of the Same Type

For this manipulation rule, the Oracle Enterprise Communications Broker stores the user name from the Params header. It then adds the URI parameter c1 with the value stored from the first Params header. Finally, it adds the URI parameter c2 with the value stored from the second Params header.

This is a sample of the configuration:

```
sip-manipulation
        name                            Params
        header-rule
                name                            storeParams
                header-name                     Params
                action                          store
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            storeUserName
                        parameter-name                  user
                        type                            uri-user
                        action                          store
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value
header-rule
                name                            modEdit
                header-name                     Edit
                action                          manipulate
                comparison-type                 pattern-rule
                match-value
                msg-type                        request
                new-value
methods                         INVITE
                element-rule
                        name                            addParam1
                        parameter-name                  c1
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-
value                      $storeParams[0].$storeUserName.$0
                element-rule
                        name                            addParam2
                        parameter-name                  c2
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
```

```
                        match-value
                        new-
value                      $storeParams[1].$storeUserName.$0
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK9g855p30cos08acgs6o0.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SD99ri601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SD99ri601-6f5691f6461356f607b0737e4039caec-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp <sip:user@192.168.1.60:5060;c1=sipp1;c2=sipp2>
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
```

Example 8 Using Header Rule Special Characters

For this manipulation rule, the Oracle Enterprise Communications Broker:

- Stores the header value of the Params header with the given pattern rule, and stores both the user name of the Params header and the URI parameter abc
- Adds the URI parameter lpu with the value stored from the previous Params header
- If any of the Params headers match the pattern rule defined in the first step, adds the URI parameter apu with the value aup
- If all of the Params headers match the pattern rule defined in the first step, adds the URI parameter apu with the value apu
- If the first Params headers does not match the pattern rule for storing the URI parameter defined in the first step, adds the URI parameter not with the value 123
- If the first Params headers matches the pattern rule for storing the URI parameter defined in the first step, adds the URI parameter yes with the value 456

This is a sample of the configuration:

```
sip-manipulation
        name                            specialChar
        header-rule
                name                            searchParams
                header-name                     Params
                action                          store
                comparison-type                 pattern-rule
                match-value                     .*sip:(.+)@.*
                msg-type                        request
```

```
                    new-value
                    methods                         INVITE
                    element-rule
                            name                            userName
                            parameter-name
                            type                            uri-user
                            action                          store
                            match-val-type                  any
                            comparison-type                 case-sensitive
                            match-value
                            new-value
element-rule
                            name                            emptyUriParam
                            parameter-name                  abc
                            type                            uri-param
                            action                          store
                            match-val-type                  any
                            comparison-type                 pattern-rule
                            match-value
                            new-value
header-rule
                    name                            addUserLast
                    header-name                     Edit
                    action                          manipulate
                    comparison-type                 case-sensitive
                    match-value
                    msg-type                        request
                    new-value
                    methods                         INVITE
                    element-rule
                            name                            lastParamUser
                            parameter-name                  lpu
                            type                            uri-param
                            action                          add
                            match-val-type                  any
                            comparison-type                 case-sensitive
                            match-value
                            new-value $searchParams[^].$userName.$0
                    element-rule
                            name                            anyParamUser
                            parameter-name                  apu
                            type                            uri-param
                            action                          add
                            match-val-type                  any
                            comparison-type                 pattern-rule
                            match-value                     $searchParams[~]
                            new-value                       aup
                    element-rule
                            name                            allParamUser
                            parameter-name                  apu
                            type                            header-param
                            action                          add
                            match-val-type                  any
                            comparison-type                 pattern-rule
                            match-value                     $searchParams[*]
                            new-value                       apu
```

```
                element-rule
                        name                            notParamYes
                        parameter-name                  not
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-
value                    !$searchParams.$emptyUriParam
                        new-value                       123
                element-rule
                        name                            notParamNo
                        parameter-name                  yes
                        type                            uri-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-
value                    $searchParams.$emptyUriParam
                        new-value                       456
```

This is a sample of the result:

```
Request-Line: INVITE sip:service@192.168.200.60:5060;tgid=123 SIP/2.0
    Message Header
        Via: SIP/2.0/UDP
192.168.200.61:5060;branch=z9hG4bK681m9t30e0qh6akgj2s1.1
        From: sipp <sip:sipp@192.168.1.60:5060>;tag=SDchrc601-1
        To: sut <sip:service@192.168.1.61:5060>
        Call-ID: SDchrc601-fcf5660a56e2131fd27f12fcbd169fe8-06a3gu0
        CSeq: 1 INVITE
        Contact: <sip:sipp@192.168.200.61:5060;transport=udp>
        Goodbye: Remove Me
        Custom: This is my custom header
        Display: sipp <sip:user@192.168.1.60:5060;up=abc>;hp=123
        Params: sipp <sip:sipp1@192.168.1.60:5060>
        Params: sipp <sip:sipp2@192.168.1.60:5060>
        Edit: disp
<sip:user@192.168.1.60:5060;lpu=sipp2;apu=aup;not=123>;apu=apu
        Max-Forwards: 69
        Subject: Performance Test
        Content-Type: application/sdp
        Content-Length: 140
```

Example 9 Status-Line Manipulation

This section shows an HMR configuration set up for status-line manipulation.

Given that the object of this example is to drop the 183 Session Progress response when it does not have SDP, your SIP manipulation configuration needs to:

1. Search for the 183 Session Progress response
2. Determine if the identified 183 Session Progress responses contain SDP; the Oracle Enterprise Communications Broker searches the 183 Session Progress responses where the content length is zero

3. If the 183 Session Progress response does not contain SDP, change its status code to 699
4. Drop all 699 responses

```
sip-manipulation
        name                            manip
        description
        header-rule
                name                            IsContentLength0
                header-name                     Content-Length
                action                          store
                comparison-type                 pattern-rule
                match-value                     0
                msg-type                        reply
                new-value
                methods
        header-rule
                name                            is183
                header-name                     @status-line
                action                          store
                comparison-type                 pattern-rule
                match-value
                msg-type                        reply
                new-value
                methods
                element-rule
name                            is183Code
                        parameter-name
                        type                            status-code
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     183
                        new-value
        header-rule
                name                            change183
                header-name                     @status-line
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        reply
                new-value
                methods
                element-rule
                        name                            make199
                        parameter-name
                        type                            status-code
                        action                          replace
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     $IsContentLength0
& $is183.$is183Code
                        new-value                       199

sip-interface     options dropResponse=699
```

Example 10 Use of SIP HMR Sets

The following example shows the configuration for SIP HMR with one SIP manipulation configuration loading another SIP manipulation configuration. The goals of this configuration are to:

- Add a new header to an INVITE
- Store the user portion of the Request URI
- Remove all Route headers from the message only if the Request URI is from a specific user

```
sip-manipulation
        name                            deleteRoute
        description                     delete all Route Headers
        header-rule
                name                            deleteRoute
                header-name                     Route
                action                          delete
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
sip-manipulation
        name                            addAndDelete
        description                     Add a New header and delete Route
headers
        header-rule
                name                            addHeader
                header-name                     New
                action                          add
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value                       "Some Value"
                methods                         INVITE
        header-rule
                name                            storeRURI
                header-name                     request-uri
                action                          store
                comparison-type                 pattern-rule
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            storeUser
                        parameter-name
                        type                            uri-user
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     305.*
                        new-value
```

```
        header-rule
                name                            deleteHeader
                header-name                     request-uri
                action                          sip-manip
                comparison-type                 Boolean
                match-value                     $storeRURI.$storeUser
                msg-type                        request
                new-value                       deleteRoute
                methods                         INVITE
```

Example 11 Use of Remote and Local Port Information

The following example shows the configuration for remote and local port information. The goals of this configuration are to:

- Add LOCAL_PORT as a header parameter to the From header
- Add REMOTE_PORT as a header parameter to the From header

```
sip-manipulation
        name                            addOrigIp
        description
        header-rule
                name                            addIpParam
                header-name                     From
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            addIpParam
                        parameter-name                  newParam
                        type                            header-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value                       $LOCAL_IP
                element-rule
                        name                            addLocalPort
                        parameter-name                  lport
                        type                            header-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value
                        new-value                       $LOCAL_PORT
                element-rule
                        name                            addRemotePort
                        parameter-name                  rport
                        type                            header-param
                        action                          add
                        match-val-type                  any
                        comparison-type                 case-sensitive
```

```
                                match-value
                                new-value                       $REMOTE_PORT
```

Example 12 Response Status Processing

Given that the object of this example is to drop the 183 Session Progress response when it does not have SDP, your SIP manipulation configuration needs to:

1. Search for the 183 Session Progress response
2. Determine if the identified 183 Session Progress responses contain SDP; the Oracle Enterprise Communications Broker searches the 183 Session Progress responses where the content length is zero
3. If the 183 Session Progress response does not contain SDP, change its status code to 699
4. Drop all 699 responses

```
sip-manipulation
        name                            manip
        description
        header-rule
                name                            IsContentLength0
                header-name                     Content-Length
                action                          store
                comparison-type                 pattern-rule
                match-value                     0
                msg-type                        reply
                new-value
                methods
        header-rule
                name                            is183
                header-name                     @status-line
                action                          store
                comparison-type                 pattern-rule
                match-value
                msg-type                        reply
                new-value
                methods
                element-rule
                        name                            is183Code
                        parameter-name
                        type                            status-code
                        action                          store
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     183
                        new-value
        header-rule
                name                            change183
                header-name                     @status-line
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        reply
                new-value
                methods
```

```
                element-rule
                        name                            make699
                        parameter-name
                        type                            status-code
                        action                          replace
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     $IsContentLength0
& $is183.$is183Code
                        new-value                       699
sip-interface
    options dropResponse=699
```

The following four configuration examples are based on the this sample SIP INVITE:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed;boundary=boundary
Content-Length:  466
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 12345 RTP/AVP 18
a=rtpmap:8 G729/8000/1
a=fmtp:18 annexb=no
a=sendrecv
a=ptime:20
a=maxptime:200
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap:34 H263a/90000
a=ptime:30
--boundary--
```

Example 13 Remove a Line from SDP

In this example, the SIP manipulation is configured to remove all p-time attributes from the SDP.

```
sip-manipulation
        name                            removePtimeFromBody
        description                   removes ptime attribute from all bodies
        header-rule
                name                            CTypeManp
                header-name                     Content-Type
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            remPtime
                        parameter-name                  application/sdp
                        type                            mime
                        action                          find-replace-all
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value                     a=ptime:[0-9]{1,2}(\n|
\r\n)
                        new-value
```

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed;boundary=boundary
Content-Length:  466
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 12345 RTP/AVP 18
a=rtpmap:18 G729/8000/1
a=fmtp:18 annexb=no
a=sendrecv
a=maxptime:200
--boundary
```

```
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap:34 H263a/90000
--boundary-
```

Example 14 Back Reference Syntax

In this sample of back-reference syntax use, the goal is to change the To user. The SIP manipulation would be configured like the following:

```
sip-manipulation
        name                            changeToUser
        description                   change user in the To header
        header-rule
                name                            ChangeHeader
                header-name                     To
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        request
                new-value
                methods                         INVITE
                element-rule
                        name                            replaceValue
                        parameter-name
                        type                            header-value
                        action                          replace
                        match-val-type                  any
                        comparison-type                 pattern-rule
                        match-value                     (.+)(service)(.+)
                        new-value                       $1+Bob+$3
```

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:Bob@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed;boundary=boundary
Content-Length:  466
…
…
…
```

Example 15 Change and Remove Lines from SDP

In this sample of changing and removing lines from the SDP, the goal is to convert the G.729 codec to G.729a. The SIP manipulation would be configured like the following:

```
sip-manipulation
        name                            std2prop-codec-name
        description                     rule to translate standard to
proprietary codec name
        header-rule
                name                            CTypeManp
                header-name                     Content-Type
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        any
                new-value
                methods
                element-rule
                        name                            g729-annexb-no-std2prop
                        parameter-name                  application/sdp
                        type                            mime
                        action                          find-replace-all
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value                     a=rtpmap:[0-9]{1,3}
(G729/8000/1\r\na=fmtp:[0-9]{1,3} annexb=no)[[:1:]]
                        new-value                       G729a/8000/1
```

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed;boundary=boundary
Content-Length:  466
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 12345 RTP/AVP 8
a=rtpmap:18 G729a/8000/1
a=sendrecv
a=maxptime:200
--boundary
```

```
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap:34 H263a/90000
--boundary-
```

Example 16 Change and Add New Lines to the SDP

In this sample of changing and adding lines from the SDP, the goal is to convert non-standard codec H.263a to H.263. The SIP manipulation would be configured like the following:

```
sip-manipulation
        name                            prop2std-codec-name
        description                     rule to translate proprietary to
standard codec name
        header-rule
                name                            CodecManp
                header-name                     Content-Type
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                msg-type                        any
                new-value
                methods
                element-rule
                        name                            H263a-prop2std
                        parameter-name                  application/sdp
                        type                            mime
                        action                          find-replace-all
                        match-val-type                  any
                        comparison-type                 case-sensitive
                        match-value                     a=rtpmap:([0-9]{1,3})
H263a/.*\r\n
                        new-value                       a=rtpmap:+$1+"
H263/90000"+$CRLF+a=fmtp:+$1+" QCIF=4"+$CRLF
```

The result of manipulating the original SIP INVITE (shown above) with the configured SIP manipulation is:

```
INVITE sip:service@192.168.1.61:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.60:5060;branch=z9hG4bK-1-0
From: sipp <sip:sipp@192.168.1.60:5060>;tag=1
To: sut <sip:service@192.168.1.61:5060>
Call-ID: 1-15554@192.168.1.60
CSeq: 1 INVITE
Contact: <sip:sipp@192.168.1.60:5060;user=phone>
Max-Forwards: 70
Content-Type: multipart/mixed;boundary=boundary
Content-Length:  466
--boundary
```

```
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=audio 12345 RTP/AVP 8
a=rtpmap:18 G729/8000/1
a=fmtp:18 annexb=no
a=sendrecv
a=maxptime:200
--boundary
Content-Type: application/sdp
v=0
o=user1 53655765 2353687637 IN IP4 192.168.1.60
s=-
c=IN IP4 192.168.1.60
t=0 0
m=video 12345 RTP/AVP 34
a=rtpmap:34 H263/90000
a=fmtp:34 QCIF=4
--boundary-
```

Dialog-Matching Header Manipulation

The most common headers to manipulate using HMR are the To-URI and From-URI. Along with the to-tag, from-tag, and Call-ID values, these are also all headers that represent dialog-specific information that must match the UAC and UAS to be considered part of the same dialog. If these parameters are modified through HMR, the results can be that the UAC or UAS rejects messages.

While it is possible to ensure that dialog parameters match correctly using regular HMR, this feature offers a simpler and less error-prone method of doing so.

In addition, this section describes the addition of built-in SIP manipulations defined by Oracle best practices, and a new method of testing your SIP manipulations.

About Dialog-Matching Header Manipulations

The goal of this feature is to maintain proper dialog-matching through manipulation of dialog-specific information using HMR. Two fundamental challenges arise when looking at the issue of correctly parameters manipulating dialog-matching:

- Inbound HMR
- Outbound HMR

The new setting **out-of-dialog** (for the **msg-type** parameter) addresses these challenges by offering an intelligent more of dialog matching of messages for inbound and outbound HMR requests. This is a msg-type parameter, meaning that it becomes matching criteria for operations performed against a message. If you also specify methods (such as REGISTER) as matching criteria, then the rule is further limited to the designated method.

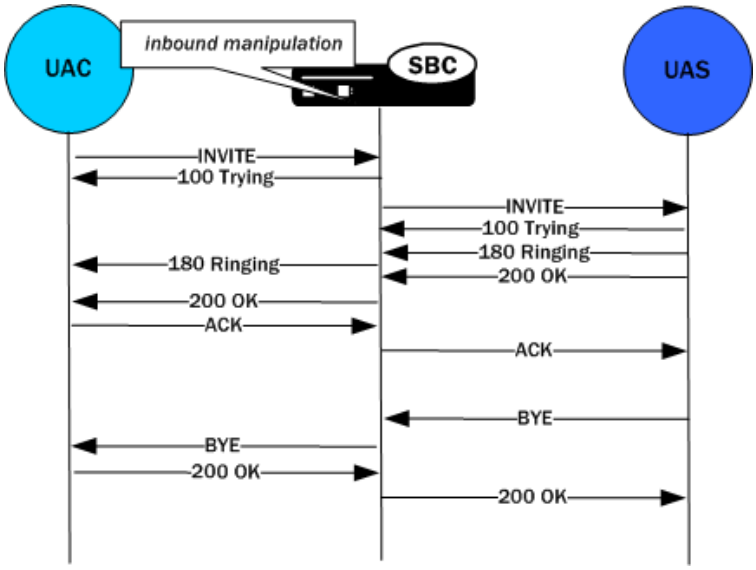
For both inbound and outbound manipulations, using the **out-of-dialog** setting means the message must be a request without a to-tag in order to perform the manipulation.

Inbound HMR Challenge

Because inbound manipulations take place before the message reaches the core of Oracle Enterprise Communications Broker (Communications Broker) SIP processing, the SIP proxy takes the manipulated header as directly received from the client. This can cause problems for requests leaving the Communications Broker for the UAC because the dialog does not match the initial request sent.

The unmodified header must be cached because for any subsequent request (For example, a BYE originating from the terminator. See the following diagram.) the Communications Broker might need to restore the original value, enabling the UAC to identify the message correctly as being part of the same dialog. For out-of-dialog requests (when the To, From, or Call-ID headers are modified) the original header is stored in the dialog when the **msg-type out-of-dialog** is used.

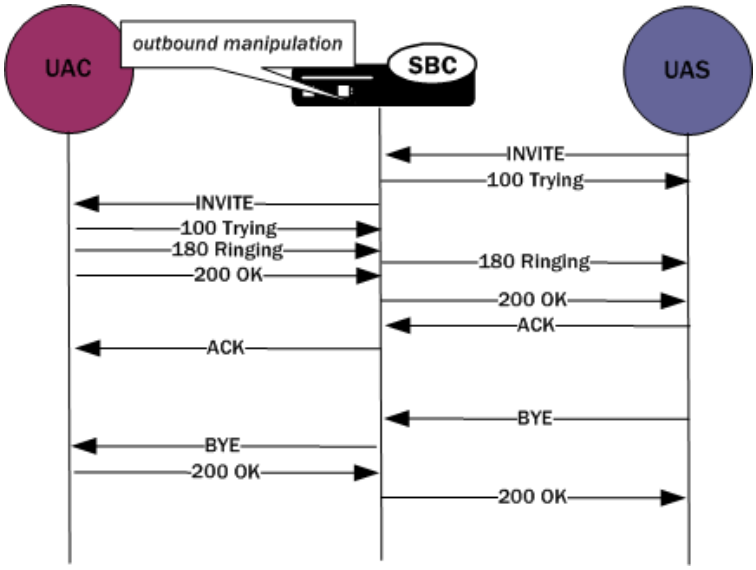
The Communications Broker performs the restoration of original headers outside of SIP manipulations. There are no manipulation rules to configure for restore the header to their original context. The Communications Broker recognizes that the headers are modified, and restores them to their original state prior to sending the message out. Restoration takes place prior to outbound manipulations so that any outbound manipulation can those headers after they are restored.



Outbound HMR Challenge

When you use the **out-of-dialog** setting for an outbound manipulation, the Oracle Enterprise Communications Broker executes this specific SIP header rule only if the same SIP header rule was executed against the initial dialog-creating request.

For example, if the INVITE's To header was not manipulated, it would not be correct to manipulate the To header in the BYE request. To do so would render the UAC unable to properly match the dialog. And this also means that the outbound manipulation should be carried out against a To, From, or Call-ID header in the BYE request if it was manipulated in the INVITE.



Built-In SIP Manipulations

In the course of HMR use, certain rules have become commonly used. Lengthy and complex, these rules do not include any customer-specific information and do they can be used widely. To make using them easier, they have been turned into built-in rules that you can reference in the **in-manipulationid** and **out-manipulationid** parameters that are part of the realm, session agent, and SIP interfaces configurations.

Built-in rules start with the prefix ACME_, so Oracle recommends you name your own rules in a different manner to avoid conflict.

While the number of built-in manipulation rules is expected to grow, one is supported at the present time: ACME_NAT_TO_FROM_IP. When performed outbound, this rule changes:

- The To-URI hostname to the logical $TARGET_IP and port to $TARGET_PORT
- The From-URI to the logical $REPLY_IP and port to be $REPLY_PORT

Unique HMR Regex Patterns and Other Changes

In addition to the HMR support it offers, the Oracle Enterprise Communications Broker can now be provisioned with unique regex patterns for each logical remote entity. This supplement to pre-existing HMR functionality saves you provisioning time and saves Oracle Enterprise Communications Broker resources in instances when it was previously necessary to define a unique SIP manipulation per PBX for a small number of customer-specific rules.

Manipulation Pattern Per Remote Entity

On the Oracle Enterprise Communications Broker, you can configure logical remote entities (session agents, realms, and SIP interfaces) with a manipulation pattern string that the system uses as a regular expression. Then the SIP manipulation references this regular expression using the reserved word $MANIP_PATTERN. At runtime, the Oracle Enterprise Communications Broker looks for the logical entity configured with a manipulation pattern string in this order of preference: session agent, realm, and finally SIP interface.

On finding the logical entity configured with the manipulation string, the Oracle Enterprise Communications Broker dynamically determines the expression. When there is an invalid reference to a manipulation pattern, the pattern-rule expression that results will turn out to be the default expression (which is \,+).

When the $MANIP_PATTERN is used in a manipulation rule's **new-value** parameter, it resolves to an empty string, equivalent of no value. Even though this process ends with no value, it still consumes system resources. And so Oraclerecommends you do not use $MANIP_PATTERN as a **new-value** value.

In the following example, the SIP manipulation references the regular expression from a realm configuration:

```
realm-config
        identifier                              net200
        description
        addr-prefix                             0.0.0.0
        network-interfaces                      public:0
        ...
        manipulation-pattern                    Lorem(.+)
sip-manipulation
        name                            manip
        description
        header-rules
                name                            headerRule
                header-name                     Subject
                action                          manipulate
                match-value                     $MANIP_PATTERN
                msg-type                        request
                comparison-type                 pattern-rule
                new-value                       Math
                methods                         INVITE
```

Reject Action

When you use this action type and a condition matching the manipulation rule arises, the Oracle Enterprise Communications Broker rejects the request (though does not drop responses) and increments a counter.

- If the **msg-type** parameter is set to **any** and the message is a response, the Oracle Enterprise Communications Broker increments a counter to show the intention to reject the message—but the message will continue to be processed.
- If the **msg-type** parameter is set to **any** and the message is a request, the Oracle Enterprise Communications Broker performs the rejection and increments the counter.

The **new-value** parameter is designed to supply the status code and reason phrase corresponding to the reject. You can use the following syntax to supply this information: status-code[:reason-phrase]. You do not have to supply the status code and reason phrase information; by default, the system uses 400:Bad Request.

If you do supply this information, then the status code must be a positive integer between 300 and 699. The Oracle Enterprise Communications Broker then provides the reason phrase corresponding to the status code. And if there is no reason phrase, the system uses the one for the applicable reason class.

You can also customize a reason phrase. To do so, you enter the status code followed by a colon (:), being sure to enclose the entire entry in quotation marks () if your reason code includes spaces.

When the Oracle Enterprise Communications Broker performs the **reject** action, the current SIP manipulation stops processing and does not act on any of the rules following the **reject** rule. This course of action is true for nested SIP manipulations that might have been constructed using the **sip-manip** action type.

SNMP Support

The Oracle Enterprise Communications Broker provides SNMP support for the Rejected Messages data, so you can access this information externally. The new MIB objects are:

```
apSysRejectedMessages    OBJECT-TYPE
        SYNTAX          Counter32
        MAX-ACCESS      read-only
        STATUS          current
        DESCRIPTION
                "Number of messages rejected by the SD due to matching
criteria."
        ::= { apSysMgmtMIBGeneralObjects 18 }
apSysMgmtRejectedMesagesThresholdExeededTrap        NOTIFICATION-TYPE
        OBJECTS         { apSysRejectedMessages }
        STATUS          current
        DESCRIPTION
        " The trap will be generated when the number of rejected messages
exceed the configured threshold within the configured window."
        ::= { apSystemManagementMonitors 57 }
apSysMgmtRejectedMessagesGroup  OBJECT-GROUP
        OBJECTS {
            apSysRejectedMessages
        }
        STATUS          current
        DESCRIPTION
                "Objects to track the number of messages rejected by the SD."
        ::= {  apSystemManagementGroups 18 }
apSysMgmtRejectedMessagesNotificationsGroup NOTIFICATION-GROUP
      NOTIFICATIONS {
                        apSysMgmtRejectedMesagesThresholdExeededTrap
                      }
                STATUS          current
                DESCRIPTION
                "Traps used for notification of rejected messages"
      ::= {  apSystemManagementNotificationsGroups 26 }
apSmgmtRejectedMessagesCap
            AGENT-CAPABILITIES
            PRODUCT-RELEASE     "Acme Packet SD"
            STATUS               current
            DESCRIPTION         "Acme Packet Agent Capability for enterprise
                                 system management MIB."
            SUPPORTS            APSYSMGMT-MIB
                INCLUDES        {
                                  apSysMgmtRejectedMessagesGroup,
                                  apSysMgmtRejectedMessagesNotificationsGroup
```

```
                                                }
::= { apSmgmtMibCapabilities 37 }
```

Log Action

When you use this action type and a condition matching the manipulation rule arises, the Oracle Enterprise Communications Broker logs information about the current message to a separate log file. This log files will be located on the same core in which the SIP manipulation occurred. On the core where sipt runs, a logfile called matched.log will appear when this action type is executed.

The matched.log file contains a timestamp, received and sent Oracle Enterprise Communications Broker network interface, sent or received IP address:port information, and the peer IP address:port information. It also specifies the rule that triggered the log action in this syntax: rule-type[rule:name]. The request URI, Contact header, To Header, and From header are also present.

```
---------------------------------------
Apr 17 14:17:54.526 On [0:0]192.168.1.84:5060 sent to 192.168.1.60:5060
element-rule[checkRURIPort]
INVITE sip:service@192.168.1.84:5060 SIP/2.0
From: sipp <sip:+2125551212@192.168.1.60:5060>;tag=3035SIPpTag001
To: sut <sip:service@192.168.1.84>
Contact: sip:sipp@192.168.1.60:5060
```

Name Restrictions for Manipulation Rules

Historically, you have been allowed to configure any value for the name parameter within a manipulation rule. This method of naming caused confusion when referencing rules, so now manipulation rules name must follow a specific syntax. They must match the expression ^[[alpha:]][[:alnum:]_]+$ and contain at least one lower case letter.

In other words, the name must:

- Start with a letter, and then it can contain any number of letters, numbers, or underscores
- Contain at least one lower case letter

All pre-existing configurations will continue to function normally. If you want to change a manipulation rule, however, you are required to change its name if it does not follow the new format.

The ACLI **verify-config** command warns you if the system has loaded a configuration containing illegal naming syntax.

Please note that the software allows you to make changes to HMRs, including configuring new functionality to existing rules, as long as you do not change the rule name. This results in an important consideration surrounding HMRs with hyphens in previously configured rule names.

- You can reference stored values in new value names. (Recall that stored values may be rule names.)
- You can perform subtraction in new value names.

If you use a rule names with hyphens within the REGEX of new value names, the system cannot determine whether the hyphen is part of the rule name or is intended to invoke subtraction within the REGEX. For this reason, you need to use great care with legacy HMR naming that includes hyphens.

As a general rule, create new rule names that follow the new rule naming guidelines if you intend to use new functionality in those rules.

New Value Restrictions

To simplify configuration and remove possible ambiguity, the use of boolean and equality operators (==, <=, <, etc.) for **new-value** parameter values has been banned. Since there was no specific functionality tied to their use, their ceasing to be use will have no impact to normal SIP manipulation operations.

Header Manipulation Rules for SDP

The Oracle Enterprise Communications Broker supports SIP header and parameter manipulation rules for four types of SIP message contents:

- headers
- elements within headers
- ASCII-encoded Multipurpose Internet Mail Extensions (MIME) bodies
- binary-encoded MIME ISDN User Part (ISUP) bodies

While Session Description Protocol (SDP) offers and answers can be manipulated in a fashion similar to ASCII-encoded MIME, such manipulation is primitive in that it lacks the ability to operate at the SDP session- and media-levels.

In addition, the system supports a variant of Header Manipulation Rules (HMR) operating on ASCII-encoded SDP bodies, with specific element types for descriptors at both the session-level and media-level, and the ability to apply similar logic to SDP message parts as is done for SIP header elements.

The configuration object, mime-sdp-rules, under sip-manipulation specifically addresses the manipulation of SDP parts in SIP messages. Just as existing header-rules are used to manipulate specific headers of a SIP message, mime-sdp-rules will be used to manipulate the SDP specific mime-attachment of a SIP message.

SDP Manipulation

mime-sdp-rules function in a similar fashion as header-rules. They provide

- parameters used to match against specific SIP methods and/or message types
- parameters used to match and manipulate all or specified parts of an SDP offer or answer
- a means of comparing search strings or expressions against the entire SDP
- different action types to allow varying forms of manipulation

Since only a single SDP can exist within a SIP message, users need not specify a content-type parameter as is necessary for a mime-rule. A mime-sdp-rule operates on the single SDP within the SIP message. If no SDP exists with the message, one can be added. If the message already contains a mime attachment, adding SDP results in a multipart message.

All manipulations performed against all or parts of the SDP are treated as UTF-8 ASCII encoded text. At the parent-level (mime-sdp-rule) the **match-value** and **new-value** parameters execute against the entire SDP as a single string.

An add action only succeeds in the absence of SDP because a message is allowed only a single SDP offer or answer. A delete operation at the mime-sdp-rule level will remove the SDP entirely.

Note that on an inbound sip-manipulation, SDP manipulations interact with the Oracle Enterprise Communications Broker codec-policy. SDP manipulations also interact with codec reordering and media setup. It is very possible to make changes to the SDP such that the call can not be setup due to invalid media parameters, or settings that will affect the ability to transcode the call. Consequently, user manipulation of the SDP can prove risky, and should be approached with appropriate caution.

Three configuration-objects, sdp-session-rule, sdp-media-rule, and mime-header-rule, exist under the mime-sdp-rule. These objects provide finer grained control of manipulating parts of the SDP.

sdp-session-rule

An sdp-session-rule groups all SDP descriptors, up until the first media line, into a single entity, thus allowing the user to perform manipulation operations on a session-specific portion of the SDP.

Like the mime-sdp-rule, all match-value and new-value operations performed at this level are executed against the entire session group as a complete string. Given the sample SDP below, if an sdp-session-rule is configured, the match-value and new-values operate only on the designated portion.

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

Nested under the sdp-session-rule configuration object is an sdp-line-rule object, the object that identifies individual descriptors within the SDP. The types of descriptors used at the sdp-session-rule level are v, o, s, i, u, e, p, c, b, t, r, z, k, and a, the descriptors specific to the entire session description.

This level of granularity affords the user a very simple way to making subtle changes to the session portion of the SDP. For instance, it is very common to have to change the connection line at the session level.

The add and delete actions perform no operation at the sdp-session-rule level.

sdp-media-rule

An sdp-media-rule groups all of the descriptors that are associated with a specific media-type into single entity, thus allowing the user to perform manipulation operations on a media-specific portion of the SDP. For example, a user can construct an sdp-media-rule to change an attribute of the audio media type.

Like a mime-sdp-rule, all match-value and new-value operations performed at this level are executed against the entire media-group as a complete string. Given the sample SDP below, if a media-level-descriptor is configured to operate against the application group, the match-value and new-values would operate only on designated portion.

```
        v=0
        o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
        s=SDP Seminar
        i=A Seminar on the session description protocol
        u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
        e=mjh@isi.edu (Mark Handley)
        c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
        a=recvonly
        m=audio 49170 RTP/AVP 0
        m=video 51372 RTP/AVP 31
        m=application 32416 udp wb
        a=orient:portrait
```

A configuration parameter **media-type** is used to specify the media group on which to operate. It contains all of the descriptors including the m-line up to the next m-line. This parameter is a string field and must match the media-type exactly as it appears within the SDP. The special media-type media can be used to refer to all media types. This is particularly useful when performing an add operation, when the user wants to add a media section between the first and second medias, but does not know what media type they are. Otherwise, during an add operation, the media section would be added before the specified media-type (if no index parameter was provided).

The types of descriptors used at the sdp-media-rule level are m, i, c, b, k, and a, the descriptors specific to the media description.

This level of granularity affords the user a very simple way to making subtle changes to the media portion of the SDP. For instance, it is very common to have to change the name of an audio format (for example G729 converted to g729b), or to add attributes specific to a certain media-type.

The index operator is supported for the media-type parameter (for example, media-type audio[1]). Like header rules, if no index is supplied, this means operate on all media-types that match the given name. For specifying specific media-types, the non-discrete indices are also supported (for example, ^ - last). Adding a media-type, without any index supplied indicates that the media should be added at the beginning. The special media-type media uses the index as an absolute index to all media sections, while a specific media-type will index relative to that given media type.

For sdp-media-rules set to an action of add where the media-type is set to media, the actual media type is obtained from the new-value, or more specifically, the string after m= and before the first space.

Given the following SDP:

```
        v=0
        o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
        c=IN IP4 224.2.17.12/127
        t=2873397496 2873404696
        m=audio 49170 RTP/AVP 0
```

```
        m=audio 48324 RTP/AVP 8
        m=video 51372 RTP/AVP 31
```

With the sdp-media-rule:

```
sdp-media-rule
        name                        smr
        media-type                  audio[1]
        action                      manipulate
        comparison-type             case-sensitive
        match-value
        new-value                   "m=audio 1234 RTP/AVP 8 16"
```

This rule operates on the 2nd audio line, changing the port and adding another codec, resulting in the SDP:

```
        v=0
        o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
        c=IN IP4 224.2.17.12/127
        t=2873397496 2873404696
        m=audio 49170 RTP/AVP 0
        m=audio 1234 RTP/AVP 8 16
        m=video 51372 RTP/AVP 31
```

The following rule, however:

```
sdp-media-rule
        name                        smr
        media-type                  media[1]
        action                      add
        comparison-type             case-sensitive
        match-value
        new-value                   "m=video 1234 RTP/AVP 45"
```

adds a new video media-type at the 2nd position of all media-lines, resulting in the SDP:

```
        v=0
        o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
        c=IN IP4 224.2.17.12/127
        t=2873397496 2873404696
        m=audio 49170 RTP/AVP 0
        m=video 1234 RTP/AVP 45
        m=audio 48324 RTP/AVP 8
        m=video 51372 RTP/AVP 31
```

sdp-line-rule

Unlike header-rules, sdp descriptors are not added in the order in which they are configured. Instead they are added to the SDP adhering to the grammar defined by RFC 4566 (as is shown below).

```
Session description
   v=  (protocol version)
   o=  (originator and session identifier)
   s=  (session name)
   i=* (session information)
   u=* (URI of description)
   e=* (email address)
   p=* (phone number)
      c=* (connection information -- not required if included in
        all media)
   b=* (zero or more bandwidth information lines)
   One or more time descriptions ("t=" and "r=" lines; see
        below)
   z=* (time zone adjustments)
   k=* (encryption key)
   a=* (zero or more session attribute lines)
   Zero or more media descriptions (see below)

Time description
   t=  (time the session is active)
   r=* (zero or more repeat times)

Media description, if present
   m=  (media name and transport address)
   i=* (media title)
   c=* (connection information -- optional if included at
        session level)
   b=* (zero or more bandwidth information lines)
   k=* (encryption key)
   a=* (zero or more media attribute lines)
```

* after the equal sign denotes an optional descriptor.

This hierarchy is enforced meaning that if you configure a rule which adds a session name descriptor followed by a rule which adds a version descriptor, the SDP will be created with the version descriptor first, followed by the session name.

The only validation that will occur is the prevention of adding duplicate values. In much the same way that header-rules prevents the user from adding multiple To headers, the descriptor rule will not allow the user to add multiple descriptors; unless multiple descriptors are allowed, as is in the case of b, t, r and a.

There exists a parameter **type** under the sdp-line-rule object that allows the user to specify the specific line on which to perform the operation. For example: v, o, s, i, u, e, p, c, b, t, r, z, k, a, and m. Details on these types can be found in RFC 4566.

For those descriptors, of which there may exist zero or more (b, t, r, and a) entries, indexing grammar may be used to reference the specific instance of that attribute. This indexing

grammar is consistent with that of header-rules for referring to multiple headers of the same type.

Given the example SDP below:

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
r=604800 3600 0 90000
r=7d 1h 0 25h
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

and the following sdp-line-rule:

```
sdp-line-rule
   name                 removeRepeatInterval
   type                 r[1]
   action               delete
```

The rule removeRepeatInterval removes the second repeat interval descriptor within the SDP.

The behavior of all SDP rules follow the same behavior of all manipulation rules in that they are executed in the order in which they are configured and that each rule executes on the resultant of the previous rule.

Each descriptor follows its own grammar and rules depending on the type specified. The values of the descriptor are evaluated at runtime since the new-values themselves are evaluated at runtime. At this time no validation of the grammar for each of the types is performed. The user is responsible for properly formatting each of the descriptors according to their specifications.

For instance, the version (v) descriptor can be removed from the SDP but leaving all descriptors for that SDP, causing the SDP to become invalid. This is consistent with the way header-rules operate, in that there is no validation for the specific headers once they have been manipulated through HMR.

Regular Expression Interpolation

An interpolated regular expression is a regular expression that is compiled and evaluated at runtime. Today all regular expressions are compiled at configuration time in order to improve performance. There are cases where a regular expression is determined dynamically from data within a SIP message. In these circumstances the regular expression is unknown until the time of execution.

In order to have a regular expression be interpolated at runtime, it must be contained within a set of {}. An interpolated expression can have any number of regular expressions and strings appended together. Any characters to the left or right of the curly braces will be appended to

the value within the curly braces. The curly braces are effectively two operators treated as one (interpolate the value contained within and then concatenate the values to the left and right of the curly braces). If the comparison-type is set to pattern-rule and the match-value contains a value that matches the grammar below, then it will be treated as an interpolated expression.

```
([^\\]|^)\{\$[^0-9]+[^}]*\}
```

The example below demonstrates using a user defined variable within a regular expression of another rule at runtime.

```
element-rule

        name                            someRule
        type                            header-value
        action                          replace
        comparison-type                 pattern-rule
        match-value                     ^sip:{$rule1.$0}@(.+)$
        new-value                       sip:bob@example.com
```

If the value of $rule1.$0 evaluates to alice then it will successfully match against the string sip:alice@comcast.net. An interpolated expression can be as simple as “{$rule1.$0}” or as complex as ^sip:{rule1.$0}@{$rule2[1].$2}$. It is not possible to interpolate a normal regular expression since the grammar will not allow the user to enter such an expression. Only variables can be contained with the curly braces.

The resultant of interpolated expressions can be stored in user defined variables. Given the same example from above, if the rule someRule was referenced by another rule, the value of sip:alice@comcast.net would be stored within that rule.

Interpolation only makes a single pass at interpolation, but does so every time the Rule executes. In other words, if the Rule is applied to the Route header, it will interpolate again for each Route header instance. What this means is that the value within the curly braces will only be evaluated once. For instance, if the value {$someRule.$1} evaluates to {$foobar.$2} the Communications Broker will treat $foobar.$2 as a literal string which it will compile as a regular expression. The Communications Broker will not recursively attempt to evaluate $foobar.$2, even if it was a valid user defined variable.

Interpolated regular expressions will evaluate to TRUE if and only if both the regular expression itself can be compiled and it successfully matches against the compared string.

You cannot use both interpolated expressions and number quantifiers like `{3,5}` in the same match-value. When interpolated expressions are evaluated, the brackets around the number quantifiers will be removed, leaving the literal string `3,5`. For example, if `$someRule.$1` resolves to a literal string 101, then a match-value of `^[0-9]{3,5} RTP.* {$someRule.$1}` will resolve to `^[0-9]3,5 RTP.* 101`, which will not match any number 3 to 5 times.

Regular Expressions as Boolean Expressions

Regular expressions can be used as boolean expressions today if they are the only value being compared against a string, as is shown in the case below.

```
mime-rule
        name                            someMimeRule
        content-type                    application/text
        action                          replace
```

```
        comparison-type                 pattern-rule
        match-value                     ^every good boy .*
        new-value                       every good girl does fine
However, regular expressions can not be used in conjunction with other
boolean expressions to form more complex boolean expressions, as is shown
below.

mime-rule
     name                       someMimeRule
     content-type               application/text
     action                     replace
     comparison-type            boolean
     match-value                $someRule & ^every good boy .*
     new-value                  every good girl does fine
```

There are many cases where the user has the need to compare some value as a regular expression in conjunction with another stored value. It is possible to perform this behavior today, however it requires an extra step in first storing the value with the regular expression, followed by another Manipulation Rule which compares the two boolean expressions together (e.g. $someRule & $someMimeRule).

In order to simplify the configuration of some sip-manipulations and to make them more efficient this functionality is being added.

Unfortunately, it is not possible to just use the example as is shown above. The problem is there are many characters that are commonly used in regular expressions that would confuse the HMR expression parser (such as $, and +). Therefore delimiting characters need to be used to separate the regular expression from the other parts of the expression.

To treat a regular expression as a boolean expression, it needs to be enclosed within the value $REGEX(<expression>,<compare_string>=$ORIGINAL); where <expression> is the regular expression to be evaluated. <compare_string> is the string to compare against the regular expression. This second argument to the function is defaulted to $ORIGINAL which is the value of the of the specific Manipulation Rule object. It can be overridden to be any other value the user desires.

The proper configuration for the example above to use regular expressions as boolean expressions is

```
mime-rule
        name              someMimeRule
        content-type      application/text
        action            replace
        comparison-type   boolean
        match-value       $someRule & $REGEX("^every good boy .*")
        new-value         every good girl does fine
```

It is also possible to use expressions as arguments to the $REGEX function. These expressions will in turn be evaluated prior to executing the $REGEX function. A more complex example is illustrated below.

```
header-rule
        name              checkPAU
        header-name       request-uri
        action            reject
        comparison-type   boolean
```

```
        match-value         (!$REGEX($rule1[0],$FROM_USER))&
                            (!$REGEX($rule2[0],$PAI_USER))
        msg-type            request
        new-value           403:Forbidden
        methods             INVITE,SUBSCRIBE,MESSAGE,PUBLISH,
                            OPTIONS, REFER
```

It should be noted that when using $REGEX() in a boolean expression, the result of that expression is not stored in the user variable. The comparison-type must be set to pattern-rule in order to store the result of a regular expression.

The arguments to the $REGEX() function are interpolated by default. This is the case since the arguments themselves must be evaluated at runtime. The following example is also valid.

```
mime-rule
        name                  someMimeRule
        content-type          application/text
        action                replace
        comparison-type       boolean
        match-value           $someRule & $REGEX("^every good
                              {$rule1[0].$0} .*")
```

Moving Manipulation Rules

You can move rules within any manipulation-rule container. Any manipulation rule that contains sub-rules offers the ACLI command **move** <from index> <to index>. For example, given the order and list of rules below:

1. rule1
2. rule2
3. rule3
4. rule4

You can move rule3 to position 1 by executing **move 3 1**. The resulting order is: rule3, rule1, rule2, rule4. A move operation causes a shift (or insert before) for all other rules. When you move a rule from the top or middle to the bottom, the system shifts all rules above the bottom up to the position of the rule that you moved. When you move a rule from the bottom or middle to the top, the system shifts all rules below down to the position of the rule that you moved. Positions start from 1.

A valid from-index and to-index are required to be supplied as arguments to the move action. If you enter a range that is out of bounds for either the from-index or to-index, the ACLI informs you that the command did not execute and the reason.

If you create an invalid sip-manipulation by incorrectly ordering the manipulation rules, the Oracle Enterprise Communications Broker validates the rules at configuration time and treats them as invalid prior to runtime. This may or may not affect the outcome of the sip-manipulation as a configured rule may not perform any operation if it refers to a rule that has yet to be executed. It is your responsibility to reorder the remaining rules in order to make the sip-manipulation valid again.

Note that rules of a different type at the same level are all part of the same list. Header-rules, mime-rules, mime-isup-rules, and mime-sdp-rules all share the same configuration level under sip-manipulation. When selecting a move from-index and to-index for a header-rule, you must

take into consideration the location of all other rules at the same level because the move is relative to all rules at that level. The move is not relative to the particular rule you selected (for example, the header-rule).

Because the list of rules at any one level can be lengthy, you can issue the **move** command one argument at a time, providing you with the ability to select indices. For example, typing **move** without any arguments displays the list of all the rules at that level. After selecting an appropriate index, the system prompts you with a to-index location based on the same list provided.

For Example:

```
ORACLE(sip-mime-sdp-rules)# move
select a rule to move
```

1: msr sdp-type=any; action=none; match-value=; msg-type=any

2: addFoo header-name=Foo; action=none; match-value=; msg-type=any

3: addBar header-name=Bar; action=none; match-value=; msg-type=any

```
selection: 2
destination: 1
Rule moved from position 2 to position 1
ACMEPACKET(sip-mime-sdp-rules)#
```

Rule Nesting and Management

There will be cases where the user wants to take a stored value from the SDP and place it in a SIP header, and vice-versa. All header-rules, element-rules, mime-rules, mime-isup-rules, isup-param-rules, mime-header-rules and mime-sdp-rules are inherited from a Manipulation Rule. A Sip Manipulation is of type Manipulation which contains a list of Manipulation Rules. Each Manipulation Rule can itself contain a list of Manipulation Rules. Therefore when configuring manipulation rules, they will be saved in the order which they have been configured. This is different from the way other configuration objects are configured. Essentially, the user has the option of configuring which type of object they want and when they are done, it gets added to the end of the sip-manipulation, such that order is preserved. This will mean that any Manipulation Rule at the same level can not share the same name. For example, names of header-rules can't be the same as any of the mime-sdp-rule ones or mime-isup-rule. This allows the user to reference stored values from one rule type in another at the same level.

ACLI Configuration Examples

The following eight sections provide sample SDP manipulations.

Remove SDP

```
sip-manipulation
        name                    stripSdp
        description             remove SDP from SIP message
        mime-sdp-rule
                name                    sdpStrip
                msg-type                request
```

```
                methods                 INVITE
                action                  delete
                comparison-type         case-sensitive
                match-value
                new-value
```

Remove Video from SDP

```
sip-manipulation
        name                        stripVideo
        description                 strip video codecs from SIP
                                    message
        mime-sdp-rule
                name                    stripVideo
                msg-type                request
                methods                 INVITE
                action                  manipulate
                comparison-type         case-sensitive
                match-value
                new-value
                sdp-media-rule
                        name              removeVideo
                        media-type        video
                        action            delete
                        comparison-type   case-sensitive
match-value
                        new-value
```

Add SDP

```
sip-manipulation
        name                    addSdp
        description             add an entire SDP if one does
                                not exist
        mime-sdp-rule
                name                    addSdp
                msg-type                request
                methods                 INVITE
                action                  add
                comparison-type         case-sensitive
                match-value
                new-value               "v=0\r\no=mhandley
2890844526 2890842807 IN IP4 "+$LOCAL_IP+"\r\ns=SDP Seminar\r\ni=A
Seminar on the session description protocol\r\u=http:
//www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps\r\e=mjh@isi.edu
(Mark Handley)\r\nc=IN IP4 "+$LOCAL_IP+"\r\nt=2873397496
2873404696\r\na=recvonly\r\nm=audio 49170 RTP/AVP 0\r\n"
```

Manipulate Contacts

This rule changes the contact in the SDP to the value contained in the Contact header.

```
sip-manipulation
        name                            changeSdpContact
        description                     changes the contact in the SDP to the
value of the contact header
        header-rule
                name                            storeContact
                header-name                     Contact
                action                          store
                comparison-type                 pattern-rule
                msg-type                        request
                methods                         INVITE
                match-value
                new-value
                element-rule
                        name                            storeHost
                        parameter-name
                        type                            uri-host
                        action                          store
                        match-val-type                  ip
                        comparison-type                 pattern-rule
                        match-value
                        new-value
        mime-sdp-rule
                name                            changeConnection
                msg-type                        request
                methods                         INVITE
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                new-value
                sdp-session-rule
                        name                    changeCLine
                        action                  manipulate
                        comparison-type         case-sensitive
                        match-value
                        new-value
                        sdp-line-rule
                           name                 updateConnection
                           type                 c
                           action               replace
                           comparison-type      case-sensitive
                           match-value          $storeContact.$storeHost
                           new-value            $storeContact.$storeHost.$0
```

Remove a Codec

This rule changes the contact in the SDP to the value contained in the Contact header.

```
sip-manipulation
        name                            removeCodec
```

```
        description                    remove G711 codec if it exists
        mime-sdp-rule
                name                            removeCodec
                msg-type                        request
                methods                         INVITE
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                new-value
                sdp-media-rule
                        name                        removeG711
                        media-type                  audio
                        action                      manipulate
                        comparison-type             case-sensitive
                        match-value
                        new-value
                        sdp-line-rule
                                name                    remove711
                                type                    m
                                action                  replace
                                comparison-type         pattern-rule
                                match-value             ^(audio [0-9]
                                                        {1,5} RTP.*)( [07]
                                                        \b)(.*)$
                                new-value               $1+$3
                        sdp-line-rule
                                name                    stripAttr
                                type                    a
                                action                  delete
                                comparison-type         pattern-rule
                                match-value             ^(rtpmap|fmtp):
                                                        [07]\b$
                                new-value
```

Change Codec

```
sip-manipulation
        name                           convertCodec
        description                    changeG711toG729
        mime-sdp-rule
                name                            changeCodec
                msg-type                        request
                methods                         INVITE
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                new-value
                sdp-media-rule
                        name                      change711to729
                        media-type                audio
                        action                    manipulate
                        comparison-type           case-sensitive
                        match-value
                        new-value
                        sdp-line-rule
```

```
                                name                    change711
                                type                    m
                                action                  replace
                                comparison-type         pattern-rule
                                match-value             ^(audio [0-9]{4,5}
                                                        RTP/AVP.*)( 0)(.*)$
new-value               $1+" 18"+$3
                        sdp-line-rule
                                name                    stripAttr
                                type                    a
                                action                  delete
                                comparison-type         pattern-rule
                                match-value             ^rtpmap:0 PCMU/
                                                        .+$
                                new-value
                        sdp-line-rule
                                name                    addAttr
                                type                    a
                                action                  add
                                comparison-type         boolean
                                match-value             $change711to729.
                                                        $stripAttr
                                new-value               rtpmap:18 G729/8000
```

Remove Last Codec and Change Port

```
sip-manipulation
        name                            removeLastCodec
        description                     remove the last codec
        mime-sdp-rule
                name                            removeLastCodec
                msg-type                        request
                methods                         INVITE
                action                          manipulate
                comparison-type                 case-sensitive
                match-value
                new-value
                sdp-media-rule
                        name                        removeLast
                        media-type                  audio
                        action                      manipulate
                        comparison-type             case-sensitive
                        match-value
                        new-value
                        sdp-line-rule
                                name                    isLastCodec
                                type                    m
                                action                  store
                                comparison-type         pattern-rule
                                match-value             ^(audio )([0-9]{4,
                                                        5})( RTP/AVP
                                                        [0-9]{1-3})$
new-value
                        sdp-line-rule
                                name                    changePort
```

```
                                type                    m
                                action                  replace
                                comparison-type         boolean
                                match-value             $removeLastCodec.
$removeLast.$isLastCodec
                                new-value               $removeLastCodec.
$removeLast.$isLastCodec.$1+0+$removeLastCodec.$removeLast.
$isLastCodec.$3
```

Remove Codec with Dynamic Payload

```
sip-manipulation
    name                        removeAMR
    description                 remove the AMR and AMR-WB dynamic codecs
    split-headers
    join-headers
    mime-sdp-rule
        name                        sdpAMR
        msg-type                    request
        methods                     INVITE
        action                      manipulate
        comparison-type             case-sensitive
        match-value
        new-value
        sdp-media-rule
            name                        mediaAMR
            media-type                  audio
            action                      manipulate
            comparison-type             case-sensitive
            match-value
            new-value
            sdp-line-rule
                name                        isAMR
                type                        a
                action                      delete
                comparison-type             pattern-rule
                match-value                 ^rtpmap:([0-9]{2,3}) AMR\/
                new-value
        sdp-media-rule
            name                        mediaIsAMR
            media-type                  audio
            action                      manipulate
            comparison-type             boolean
            match-value                 $sdpAMR.$mediaAMR.$isAMR[~]
            new-value
            sdp-line-rule
                name                        delFmtpAMR
                type                        a
                action                      delete
                comparison-type             pattern-rule
                match-value                 ^fmtp:
({$sdpAMR.$mediaAMR.$isAMR[~].$1})
                new-value
            sdp-line-rule
                name                        delAMRcodec
```

```
                type                        m
                action                      find-replace-all
                comparison-type             pattern-rule
                match-value                 ^(audio [0-9]+ RTP.*)
{$sdpAMR.$mediaAMR.$isAMR[~].$1}(.*)$
                new-value                   $1+$2
```

HMR Import-Export

Due to the complexity of SIP manipulations rules and the deep understanding of system syntax they require, it is often difficult to configure reliable rules. This feature provides support for importing and exporting pieces of SIP manipulation configuration in a reliable way so that they can be reused.

To Import HMRs, use the **Upload** link on the sip-manipulation list dialog, which is the first dialog displayed after clicking the HMR icon. To export, use **Download**.

Exporting

The SIP manipulation configuration contains an **export** command which sends the previously selected configuration to the designated file. The syntax is **export [FILENAME]**. The system compresses the file with gzip and writes it to the `/code/imports` directory.

> **Note:**
>
> SIP manipulation configurations can only be exported one at a time.

Exported data will look like this:

```
<?xml version='1.0' standalone='yes'?>
<sipManipulation
        name='manip'
        description=''
        lastModifiedBy='admin@console'
        lastModifiedDate='2009-10-16 14:16:29'>
        <headerRule
                headerName='Foo'
                msgType='any'
                name='headerRule'
                action='manipulate'
                cmpType='boolean'
                matchValue='$REGEX("[bB][A-Za-z]{2}")'
                newValue='foo'
                methods='INVITE'>
        </headerRule>
</sipManipulation>
```

To avoid conflicts when importing, the key and object ID are not included as part of the exported XML.

Importing

The **import** command imports data from a previously exported file into the currently-selected configuration. If no configuration was selected, a new one is created. The syntax is **import [FILENAME]**. Include the .gz extension in the filename. After importing, type **done** to save the configuration.

Importing a configuration with the same key as one that already exists returns an error. In this case:

- Delete the object with the same key and re-import.
- Select the object with the same key and perform an import that will overwrite it with new data.

Using SFTP to Move Files

After exporting a configuration, use SFTP to copy the file to other Oracle Enterprise Communications Brokers. Place the file in the `/code/imports` directory before using the **import** command on the second Communications Broker.