

Oracle® Enterprise Communications Broker Administrator's Guide



Release P-Cz4.2.0

F87371-01

July 2024

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2014, 2024, Oracle and/or its affiliates.

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

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

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

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

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

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

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

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

Contents

About This Guide

My Oracle Support ix

1 Applicable Platforms

Software Packaging	1-1
Caveats and Limitations for X9-2 Support	1-2
Oracle Server X9-2 Platform Preparation	1-2
Available Connections	1-3
X9-2 Back Panel Connectors and Ports	1-4
Cable the Oracle Server X9-2	1-6
Cable the Local Console	1-8
Connect ILOM to the Oracle Server X9-2	1-9
Install the Software on the Oracle Server X9-2	1-9
Next Steps After the Software Installation	1-10
Oracle Server X8-2 Platform Preparation	1-11
Available Connections	1-18
Cable the Oracle X8-2	1-19
Cable the Local Console	1-21
Connect ILOM to the Oracle X8-2	1-22
Install the Software on the X8-2	1-22
Next Steps After the Software Installation	1-23
Cable a Netra Server X8-2 for Acme Packet HA Node	1-23
HA Cabling	1-24
Rear Panel Cabling for HA	1-25
Cable a Netra Server X8-2 for Acme Packet HA Node	1-26
Configure the BIOS Setting	1-27
Virtual Systems	1-29
Log On to the System	1-29

2 Hardware Installation Summary

Connecting to The Oracle Enterprise Communications Broker	2-1
Local Connections and Time-outs	2-2

SSH Connections and Time-outs	2-3
Initiate SSH without Username and Password	2-3
SSH with Username and Password	2-4
GUI Access	2-4
Setting Your Login Banner	2-5
System Boot	2-5
Oracle Enterprise Communications Broker Boot Parameters	2-5
Upload the Stage 3 Boot Loader and System Image	2-6
Boot Parameter Changes	2-7
Set Boot Parameters	2-7
Set Boot Parameters from the ACLI	2-8
Change Boot Parameters by Interrupting a Boot in Progress	2-9
Set Management IP Address	2-10
Format Hard Drive	2-11
System Image Filename	2-11
Setting Up System Basics	2-12
New User and Superuser Passwords	2-12
New System Prompt	2-12
Setting the Initial Configuration	2-12
Initializing with Run Setup	2-13
Set Up High Availability Mode	2-14
Initialize the System from the GUI	2-17
Add a License with the Set License Function	2-18

3 System Administration

Save and Activate	3-1
General Settings and System Config Settings	3-2
Configure an NTP Server	3-3
Redundancy Config / High Availability Settings	3-3
Overview	3-4
Establishing Active and Standby Roles	3-4
Configure Redundancy Config	3-5
Force an HA Switchover	3-5
Configure System Config	3-6
SNMP Configuration	3-7
Configure SNMP Settings	3-7
Logging (Syslog)	3-8
Overview	3-8
Process Log Messages	3-8
Add a Syslog Server	3-9
Configure Syslog Settings	3-9

Enterprise Operations Monitor	3-9
Add a Monitor Collector	3-10
Configure Communications Monitoring Probe Settings	3-11
Support for Multiple VLANs	3-12
Add Multiple VLANs	3-12
Accounting Settings	3-12
Configure an Accounting Server	3-12
Configure Accounting	3-13
FTP Push	3-15
Multiple Push Receivers	3-16
Secure FTP Push Configuration	3-16
Add an FTP Push Receiver	3-17
Network Interface Configuration	3-18
Configure a Network Interface	3-19
Enable ICMP	3-20
Configure the Network Interface for High Availability Operations	3-20
Virtual MAC Addresses	3-21
Configure a Realm	3-21
Configurable TCP Timers	3-24
Configuring TCP Data Retransmission	3-24
Timer for Idle Connections	3-25
Configure the Network Parameters	3-25
DNS on the Communications Broker	3-26
DNS Functions on the Communications Broker	3-27
Retransmission Logic	3-29
DNS Server Status via SNMP	3-31
Configure DNS on the Network Interface	3-31
Security Settings	3-32
SHA 2 Support	3-32
Add a Certificate Record	3-33
TLS Profile Configuration	3-34
TLS Global Configuration	3-35
Generate a Certificate Request	3-35
Import a Certificate	3-36
RADIUS Authentication	3-36
Management Protocol Behavior	3-38
RADIUS Authentication Configuration	3-38
TACACS+ Overview	3-40
TACACS+ Authentication	3-41
TACACS+ Authorization	3-52
TACACS+ Accounting	3-59
Managing TACACS+ Operations	3-68

TACACS+ Configuration	3-70
SIP Interface Settings	3-71
Proxy Registration	3-72
Global SIP Timers	3-72
Overview	3-72
SIP Timers Discreet Configuration	3-72
Timer to Tear Down Long Duration Calls	3-73
Add a SIP Interface	3-74
Configure SIP Config	3-76
Restricting Session Initiation	3-78
Configure a SIP Interface Port	3-78
SIP Monitor and Trace Filter Configuration	3-79
SIP REFER	3-80
SIP REFER Method Call Transfer for Communications Broker	3-81
Dynamic REFER Support	3-84
180 and 100 NOTIFY in REFER Call Transfers for the Communications Broker	3-86
SNMP	3-90
Overview	3-90
Basic SNMP Parameters	3-90
SNMP Community	3-90
Trap Receivers	3-91
SNMP Community Settings	3-91
Set Trap Receiver Settings	3-91
HTTP Server Settings	3-92
Configuring the Communications Broker for SDM	3-93
Admin Security - Feature Set	3-94
Enabling the Admin Security Feature	3-95
Login Policy	3-95
Login Banner	3-96
Password Policy	3-97
Configuring Password Policy Properties	3-97
Changing a Password	3-99

4 Maintenance and Debugging

Your Oracle Enterprise Communications Broker Image	4-1
Obtain a New Image	4-1
Upgrade Software - Web GUI System Tab	4-2
Display Log Files	4-3
Display System Health	4-3
Obtain Support Information	4-3

5	Procedure to Avoid the activate_config Error	
6	Deploying Communications Broker on Oracle Cloud Infrastructure (OCI)	
	Deploying Communications Broker on OCI - Important Points to Note	6-1
	Standard Shapes Supported for Communications Broker Deployment	6-1
7	Deploying Communications Broker on AWS	
	Standard Shapes Supported for AWS Deployment	7-1
8	Deploying Communications Broker on Azure	
	Standard Shapes Supported for Azure Deployment	8-1

About This Guide

The *Oracle® Enterprise Communications Broker Administrator's Guide* provides the following information about the (Communications Broker) hardware and software.

- Supported platforms
- How to get the system operational
- Initial configuration
- 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
Release Notes	Contains information about the current release, including specifications, requirements, new features, enhancements, inherited features, known issues, caveats, and limitations.
Administrator's Guide	Describes how to deploy the system.
User's Guide	Describes how to configure SIP signaling management and how to tailor the system to specific needs.
Embedded Help system	Contains task-oriented topics for configuring, administering, maintaining, and troubleshooting the Communications Broker hardware and software.
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.

Related Documentation

The following table describes related documentation for the Communications Broker.

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.
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
Accounting Guide	Contains information about accounting support, including details about RADIUS and Diameter accounting and FTP push.

Revision History

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

Date	Description
July 2024	Initial Release Oracle Enterprise Communications Broker 4.2.0.

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

Applicable Platforms

Oracle provides the Oracle Enterprise Communications Broker (referred to as Communications Broker in this document) as an appliance and as an application for operation on virtual machines. When running as an appliance, Oracle packages the Communications Broker software with the Oracle Server X9-2 and X8-2 for Communications. When running as a virtual application, you can deploy the Communications Broker software on any third-party Common Off The Shelf (COTS) hardware that meets the specified guidelines.

Instructions for installation and maintenance of the Oracle Server X9-2 and X8-2 for Communications are common to the Session Border Controller, Session Router, and other appliance applications.

The Oracle Server X9-2 and X8-2 for Communications documentation identifies all of the hardware interfaces. For cabling the applicable Communications Broker interfaces, as named in the hardware documentation, use:

- s0p0—Service access
- wancom0—Management access
- wancom1/wancom2 —High Availability (HA) access
- SER MGT(COM1)—Serial management access

Run the application as a virtual machine over a VM system, such as Oracle VM Server, and use VM management software, such as Oracle VM Manager, to create and maintain your virtual machines.

For Virtual Machine installation instructions, see the "Platforms" chapter of the *Oracle Enterprise Session Border Controller ACLI Configuration Guide*. If you use COTS hardware, see the applicable documentation provided by your hardware vendor.

Software Packaging

The P-CZ 4.2.0 build image is labeled nnPCZ420.bz. The image is compressed by the zlib software library and includes all software components needed to install and operate the Oracle Enterprise Communications Broker (Communications Broker).

Note:

Note that you must obtain a license if you want to use TLS for media and signaling. You do not need a TLS license for SSH, SFTP, and HTTP operations. See "Add a License with the Set License Function."

Communications Broker software delivered for virtual machines includes the following packages:

Image Name	Description
nnPCZ420.bz	Standalone compressed image - This .bz image package is primarily used to load and operate the Communications Broker software as an appliance. You can also use the .bz image for existing virtual machines. Create your virtual machine according to specifications. Then copy this image to your machine (/code), and point your boot parameters to it.
nnPCz420-img-bin.ova	Virtual Machine Template - Import to a virtual machine hypervisor to create the entire machine.

Caveats and Limitations for X9-2 Support

Following are the caveats and limitations for X9-2 support in Communications Broker 4.2.0 Release.

- NVME is not supported in the Communications Broker 4.2.0 Release.
- Storage partitioning to Boot disk and system disk is not supported. This also should not affect any functionality, as from a product perspective, we do not change the /boot , / code partitions.
- VMWare is not supported on X9-2 in the Communications Broker 4.2.0 Release.

Oracle Server X9-2 Platform Preparation

Oracle Communications produces a variety of software products that run on the Oracle Server X9-2 platform, including Oracle session delivery applications.

Use your Hardware documentation to install and establish system management by way of Oracle Integrated Lights Out Manager (ILOM). Then use the steps below to prepare the Oracle X9-2 for session delivery software installation.

1. Confirm applicable firmware on the server.
 - To check the firmware versions installed in the server, go to the ILOM web interface, and navigate to **System Information, Firmware**.
 - Software and firmware versions qualified for use with Oracle Session Delivery products include:
 - ILOM — v5.1.0.20
 - BIOS — 61060500
2. Upgrade or downgrade the firmware on the server as necessary. See the [ILOM documentation](#) for ILOM upgrade instructions.
3. Configure the BIOS settings. (Settings navigation may differ based on the BIOS version.)
 - a. Observe the boot procedure, logged to the console during bootup, and use the documented key sequence to interrupt the boot and display the BIOS configuration dialogs. For example, pressing the F2 key is a common way to enter BIOS configuration from a terminal application that supports function keys.
 - b. Navigate to the Boot menu and, depending on the software distribution you are using, set the USB or CD as the first device followed by the disk controller. (Navigation: Boot)

- c. Disable Hyper-Threading. (Navigation: Advanced, Processor Configuration, Hyper-Threading)
- d. Disable CPU power limit. (Navigation: Advanced / CPU Power Management Configuration)
- e. Disable C6 Reporting. (Navigation: Advanced / CPU Power Management Configuration, CPU C6 report)
- f. Change Energy Performance to Performance. For example, set "ENERY_PERF_BIAS_CFG" mode to "PERF". (Navigation: Advanced / CPU Power Management Configuration, Energy Performance)
- g. To decrease boot up time, Oracle recommends disabling Intel PXE Boot Agent for both onboard and NIC ethernet ports. Press F2 and navigate to Advanced, Network Stack Configuration. Then disable IPv4 PXE support.

 **Note:**

PXE boot is not supported in this release.

- h. Reboot the server.
4. Perform a cold shutdown by removing all system power.

Available Connections

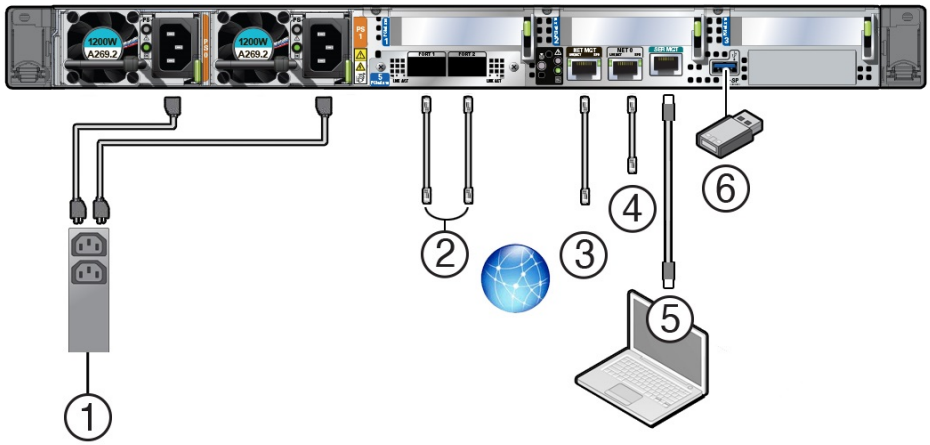
Please read all of the information for each of the available connections prior to cabling the Oracle Server X9-2.

Port	Description	You Need:
NET MGT	Provides a 10/100/1000 BASE-T Ethernet connection to the Service Processor (SP) through an RJ-45 connector. The NET MGT port provides support connections to the SP using the Oracle Integrated Lights Out Manager (ILOM) CLI and Web interface. By default, the NET MGT port is configured to use DHCP to automatically obtain an IP address. Alternatively, you can assign a static IP address to the NET MGT port. To use the NET MGT port, you must configure its network settings. When configured, use the NET MGT port IP address to log on to the device using a browser or secure shell.	Category 6 (or better) Ethernet cable to connect the NET MGT port to your network IP address for this port (required from DHCP or a static address)
NET 0	The 1 Gbps host management RJ-45 connector port enables you to connect the Oracle Server X9-2 to your network.	A Category 6 (or better) Ethernet cable to connect to the NET 0 port to your network Network parameters such as an IP address (can be provided by DHCP services or assigned a static address in the OS)

Port	Description	You Need:
SER MGT (COM1)	<p>Provides a TIA/EIA-232 serial Oracle/Cisco standard connection to the SP through an RJ-45 connector.</p> <p>SER MGT (COM1) connects to either Service Processor by default, but can be redirected to the host.</p> <p>Default settings:</p> <ul style="list-style-type: none"> • 8N1: eight data bits, no parity, one stop bit • 115200 baud 	<p>A terminal device (For example, terminal, connection to a terminal server, or computer such as a laptop running terminal emulation software)</p> <p>A cable to connect the terminal device to the SER MGT (COM1) port</p>
USB	<p>Provides USB3.0 connection to the computer. You can connect and disconnect USB cables to the USB port without affecting server operations.</p>	<p>Installation media</p> <p>Note: Maximum USB cable length: 5 meters</p>

X9-2 Back Panel Connectors and Ports

The following figure shows the locations of cable connectors and ports on the back of Oracle Server X9-2 and the cables and devices that you connect to them.



Call Out	Cable Port or Expansion Slot	Description
1	Power supply 0 input power Power supply 1 input power	<p>The server has two power supply connectors, one for each power supply, labeled PS0 and PS1. Power supply 0 input power and Power supply 1 input power both connect to a rack power distribution unit (PDU).</p> <p>Do not attach power cables to the power supplies until you finish connecting the data cables to the server. The server goes into Standby power mode, and the Oracle ILOM service processor initializes when the AC power cables are connected to the power source. System messages might be lost after 60 seconds if the server is not connected to a terminal, PC, or workstation.</p> <p>Oracle ILOM signals a fault on any installed power supply that is not connected to an AC power source, which might indicate a loss of redundancy.</p>
2	OCP-V3 NIC QSFP	<p>(Optional) 10/25/50/100/200 Gbs Open Compute Project (OCP) Version 3.0 (V3) Network Interface Card (NIC) with two QSFP ports (PORT 1 and PORT 2)</p> <p>Two QSFP 28/56 GbE Ethernet connectors for the Ethernet controller.</p> <p>Note: These ports are not used for the Session Router or Enterprise Session Router.</p>
3	Network management port (NET MGT)	<p>The service processor NET MGT port is the optional connection to the Oracle ILOM service processor. The service processor NET MGT port uses an RJ-45 cable for a 100/1000BASE-T connection.</p> <p>The NET MGT port is configured by default to use Dynamic Host Configuration Protocol (DHCP).</p>
4	Host Management Ethernet port (NET 0)	<p>NET 0: 1 Gbps Host Management RJ-45 connector port</p> <p>The host management Ethernet port enables you to connect the system to the network. The Ethernet port uses an RJ-45 cable for a 1 Gbps Host Management connection.</p>

Call Out	Cable Port or Expansion Slot	Description
5	Serial management port (SER MGT)	The service processor SER MGT port uses an RJ-45 cable and terminal (or emulator) to provide access to the Oracle ILOM command-line interface (CLI). Using Oracle ILOM, you can configure it to connect to the system console. This port does not support network connections.
6	USB port	The USB port supports hot-plugging. You can connect and disconnect a USB cable or a peripheral device while the server is running without affecting system operations.

Cable the Oracle Server X9-2

After mounting the Oracle Server X9-2 in an equipment rack and installing all components, use the following instructions to connect all appropriate data cables to the ports before powering the system up and beginning the configuration.

Oracle qualified the following configurations of the Oracle Server X9-2.

- Configuration A: A four-port 10GBASE-T Ethernet NIC
- Configuration B: A four-port 10-Gigabit QSFP+ NIC
- Configuration C: A two-port 40-Gigabit QSFP+ NIC

The quad-port NICs use Intel XL710 series cards.

On board interfaces for all configurations include:

- One 100/1000 BASE-T RJ-45 Oracle Integrated Lights Out Manager (ILOM) service processor (SP) network management (NET MGT) port
- One 1 Gbps Host Management RJ-45 connector port, labeled NET 0
- One RJ-45 serial management (SER MGT) port

Figure 1-1 Oracle Server X9-2 Configuration A: four-port 10GBASE-T Ethernet NIC

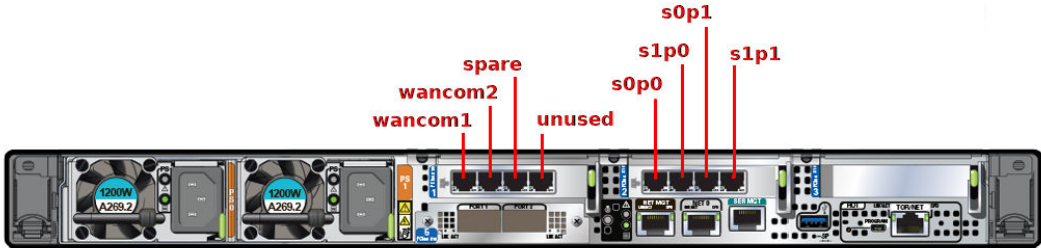
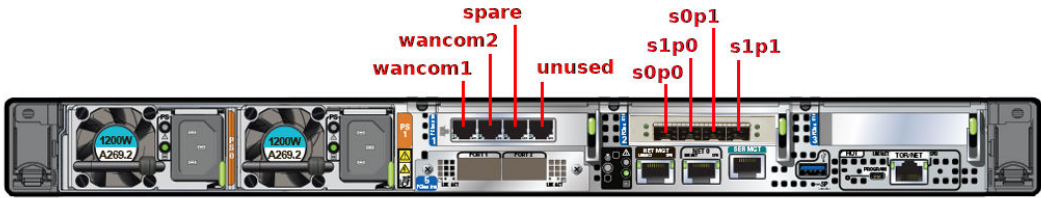
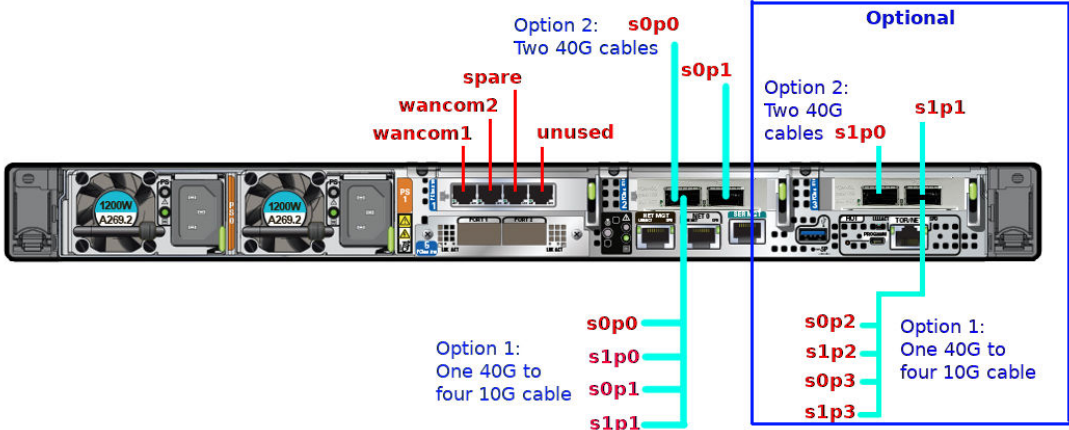


Figure 1-2 Oracle Server X9-2 Configuration B: four-port 10-Gigabit QSFP+ NIC



 **Note:**
The **spare** port can be used for the s0p0 Interface.

Figure 1-3 Oracle Server X9-2 Configuration C: two-port 40-Gigabit QSFP+



Oracle recommends using Category 6 (or better) for all Ethernet connections.

You do not need to use every port for proper operation.

Cable the Local Console

You can connect the Administration console to the local SER MGT (COM1) serial console port. You can also operate the console using serial emulation over ILOM.

To cable a console connection:

- Connect a serial console cable with an RJ-45 connector to the SER MGT port.
- Connect ethernet to the ILOM port and use serial emulation.

When configuring boot loader parameters, set the **Console Device** to COM1. Never use COM2 or VGA. The Oracle Server X9-2 server cannot boot the system when set to the default of VGA. You must change this bootparameter when deploying over this platform.

Refer to the section "Change Boot Parameters by Interrupting a Boot in Progress" within the *Installation and Platform Preparation Guide* to learn how to set your **Console Device** bootparameter to "COM1". Refer to [Set Up a Remote Console](#) to learn how to run an SSH session via iLOM using Virtual Serial Port Emulation.

Follow this procedure to cable your console:

1. Locate the appropriate cables to connect to the Oracle Server X9-2.
2. To cable a serial connection, insert the serial console cable into the SER MGT (COM1) port.

Figure 1-4 Connecting to USB and SER MGT (COM1) Ports



 **Note:**

Refer to the Oracle Server X9-2 hardware documentation for information on how to configure the terminal application to connect to the console, and how to establish communications with the Oracle Server X9-2.

3. For installation procedures, insert the USB stick in the USB port.
4. Lead the cables neatly away from the rear panel.
5. Plug in the cables to their respective destination components.

Connect ILOM to the Oracle Server X9-2

Use the following procedure to make a connection to the Oracle Server X9-2 Oracle Integrated Lights Out Manager (ILOM) port. For a remote permanent connection to the Service Processor over the ILOM connection, use the rear panel NET MGT port.

 **Note:**

Keep Ethernet cables separated from power cables by at least 60mm where possible and never run them in the same channel of the rack without segregation.

- Category 6 (or better) Ethernet

1. Locate the cable to connect to the Oracle Server X9-2 for Communications.
2. Plug the RJ-45 connector into the ILOM port.

Figure 1-5 Connecting to ILOM over the Network



3. Lead the cable neatly away from the rear panel.
 4. Connect the other end of the cable to the LAN.
- Refer to the [Oracle Server X9-2 hardware documentation](#) for information about how to configure the Web browser application to connect to the console, and how to establish communications with the Oracle Server X9-2.

Install the Software on the Oracle Server X9-2

Communications Broker 4.2.0 requires software installation when deployed on the Oracle Server X9-2.

Software installation to Oracle Server X9-2 includes the following high-level steps:

1. Insert your installation media into the USB slot or connect the ISO image by way of Oracle Integrated Lights Out Manager (ILOM) virtual media.

2. Power on the Oracle Server X9-2.
3. Observe the startup process, and press F8 to enter the boot menu when it becomes available.
4. Select the bootable USB or ISO setting.

 **Note:**

You may need to scroll through the list to reach the ISO setting.

 **Note:**

If you are performing ISO installation for the 2nd, or the 3rd time onwards (not the 1st time.)

- a. Press 'e' when the screen prompts.
- b. Type `factory-default` at the end of the line.
- c. Press Ctrl-x.

5. Save and exit the boot menu.
The Oracle Server X9-2 starts the Communications Broker installation.
6. Change the Console Device boot parameter to COM1 during installation. If you miss this change during the installation, power on and off the device or catch the boot parameter interrupt and change as soon as possible.
7. Remove the USB media when prompted by the Oracle Server X9-2.
8. Allow the Oracle Server X9-2 complete the installation process and boot to the newly installed Communications Broker software.

Next Steps After the Software Installation

Oracle recommends the following steps after installation on the Oracle Server X9-2 platform on Communications Broker 4.2.0.

1. Execute the **format hard-disk** command, per your requirements. See the "Formatting the Disk Volume" for reference and instructions.
2. Turn off the Communications Broker using the **Halt** command. This provides a graceful software shutdown, after which the hardware is still powered on.
3. Power cycle the hardware using the power switch, a power controller, or by physically disconnecting and reconnecting the power cable.

To configure Communications Broker, refer to the *ACLI Configuration Guide*.

Boot parameter changes to consider prior to service configuration include:

- Set the **Target Name** to your preferred Communications Broker name.
- Set the **Console Device** to COM1 (serial).
- Set the **IP Address** to your preferred management port IP address.
- Set the **Netmask** for your management port IP address.

- Set the **Gateway** address for your management port IP address.

 **Note:**

The boot parameters default Boot File is “/boot/bzImage”. Be aware that upgrading code includes obtaining images with, for example, an PCz prefix and the .bz file extension.

Oracle Server X8-2 Platform Preparation

Oracle Communications produces a variety of software products that run on the Oracle Server X8-2 platform. See the Release Notes for which Oracle Communications applications run on the X8-2.

Use your Hardware documentation to install and establish system management by way of Oracle Integrated Lights Out Manager (ILOM). Then use the steps below to prepare the Oracle X8-2 for session delivery software installation.

 **Note:**

The [ILOM Cable Connection procedure](#) also displays ILOM cabling.

1. Confirm applicable firmware on the server.
 - To check the firmware versions installed in the server, go to the ILOM web interface, and navigate to **System Information, Firmware**.
 - Software and firmware versions qualified for use with Oracle Session Delivery products include:
 - ILOM—v4.0.3.34
 - BIOS— 51.01.01.00
2. Upgrade or downgrade the firmware on the server as necessary. Go to https://docs.oracle.com/cd/E81115_01/index.html for ILOM upgrade instructions.
3. Configure the BIOS settings. (Settings navigation may differ based on the BIOS version.)
 - a. Observe the boot procedure, logged to the console during bootup, and use the documented key sequence to interrupt the boot and display the BIOS configuration dialogs. For example, pressing the F2 key is a common way to enter BIOS configuration from a terminal application that supports function keys.
 - b. Navigate to the Boot menu and, depending on the software distribution you are using, set the USB or CD as the first device followed by the disk controller. (Navigation: Boot)
 - c. Disable Hyper-Threading. (Navigation: Advanced, Processor Configuration, Hyper-Threading)

 **Note:**

Refer to "Hyperthreading and CPU Affinity" in the Session Border Controller Installation Guide for Oracle guidelines on the use of Hyper-threading.

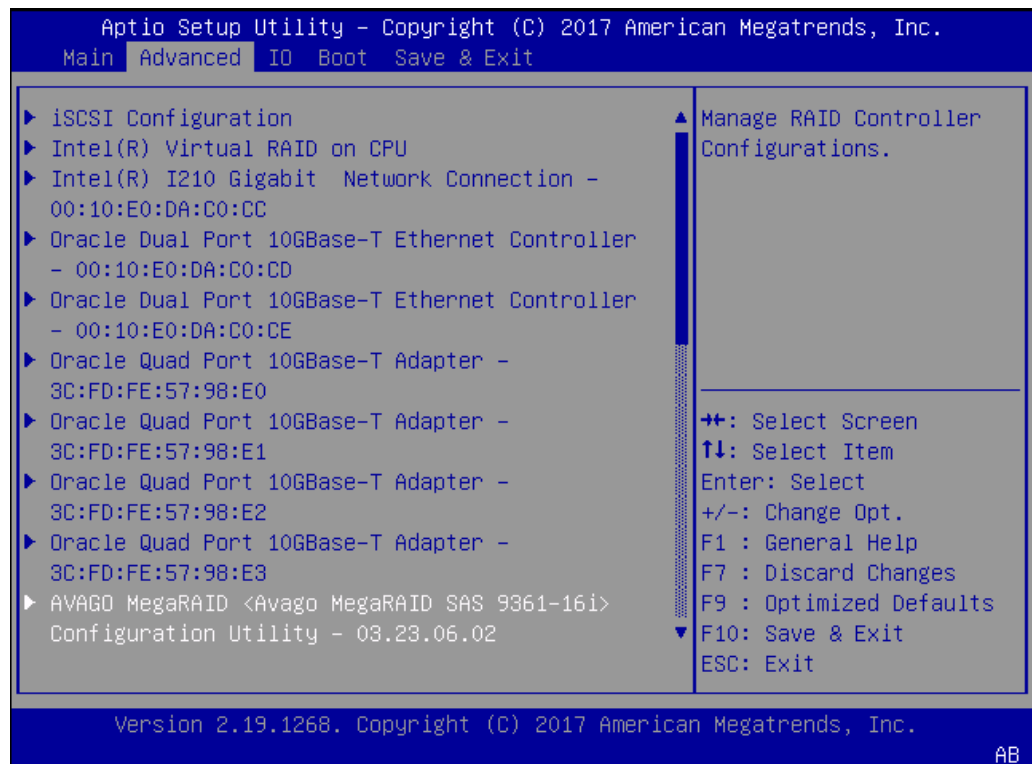
- d. Disable CPU power limit. (Navigation: Advanced / CPU Power Management Configuration)
- e. Disable C6 Reporting. (Navigation: Advanced / CPU Power Management Configuration, CPU C6 report)
- f. Change Energy Performance to Performance. For example, set "ENERY_PERF_BIAS_CFG" mode to "PERF". (Navigation: Advanced / CPU Power Management Configuration, Energy Performance)
- g. To decrease boot up time, Oracle recommends disabling Intel PXE Boot Agent for both onboard and NIC ethernet ports. Press F2 and navigate to Advanced, Network Stack Configuration. Then disable IPv4 PXE support.

 **Note:**

PXE boot is not supported in this release.

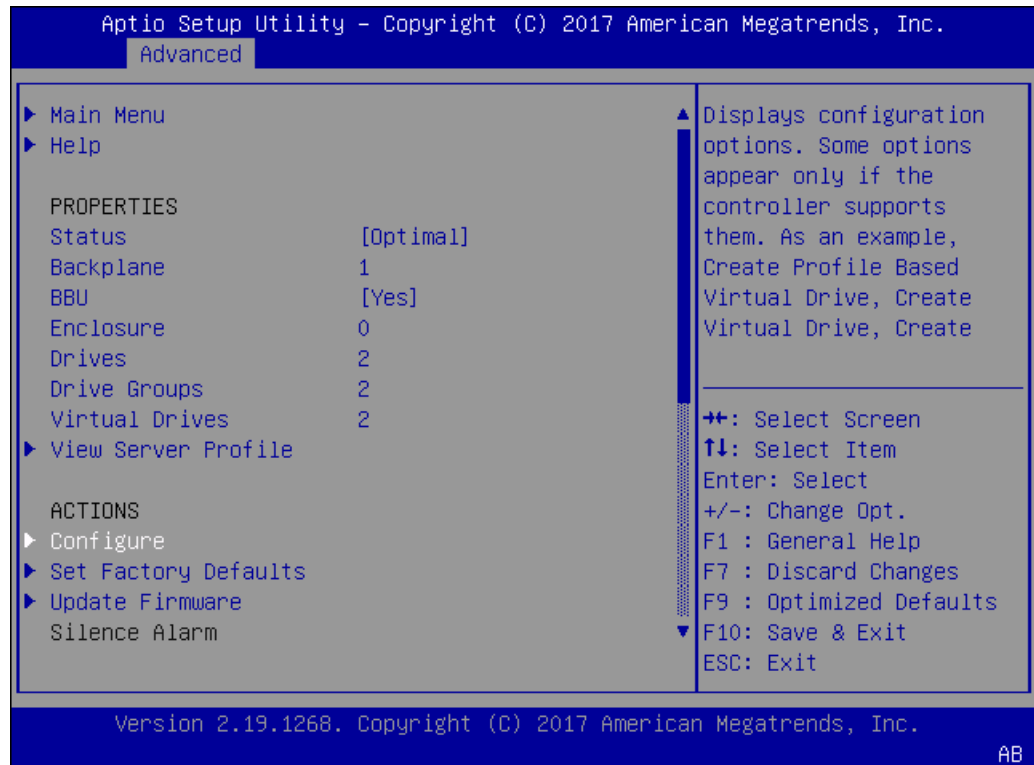
- h. Reboot the server.
4. Initialize the Hard Disk Drive.
- a. Open the ILOM remote system console to observe the system's boot cycle, and interrupt the boot cycle to enter the MegaRAID configuration utility.

Figure 1-6 Selecting RAID Configuration



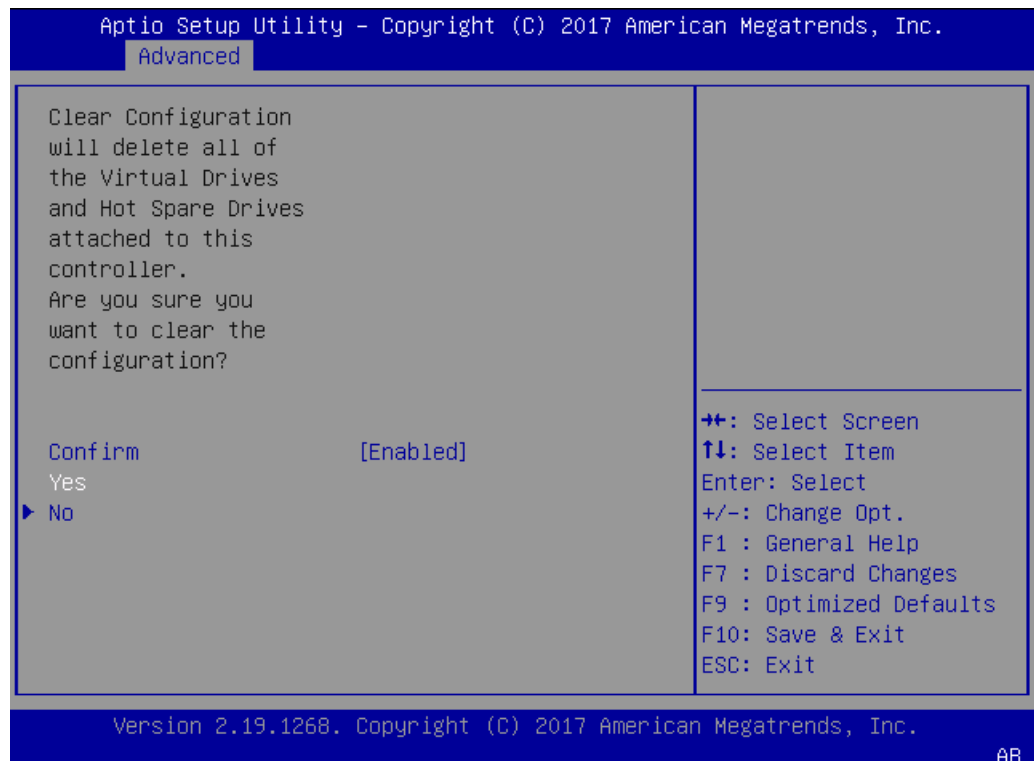
- b. Navigate the utility to establish your virtual drive's operation, initially including the **Configure** action.

Figure 1-7 Begin RAID Configuration



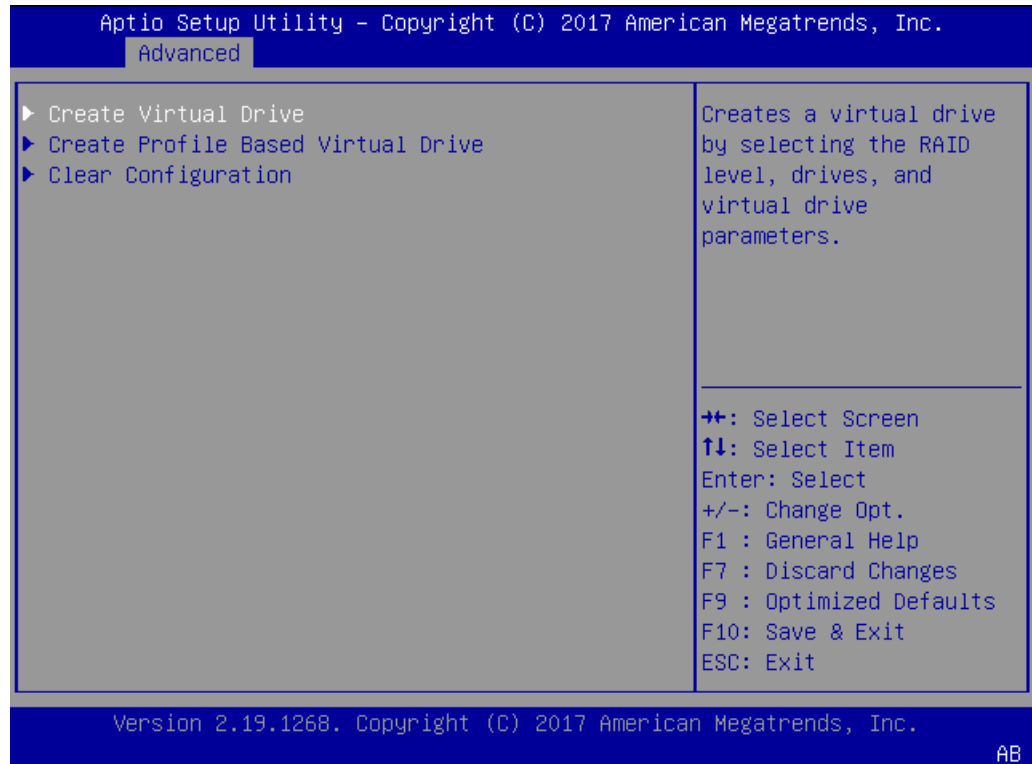
- c. Clear the configuration, regardless of the initial state.

Figure 1-8 Clear Any Existing RAID Configuration



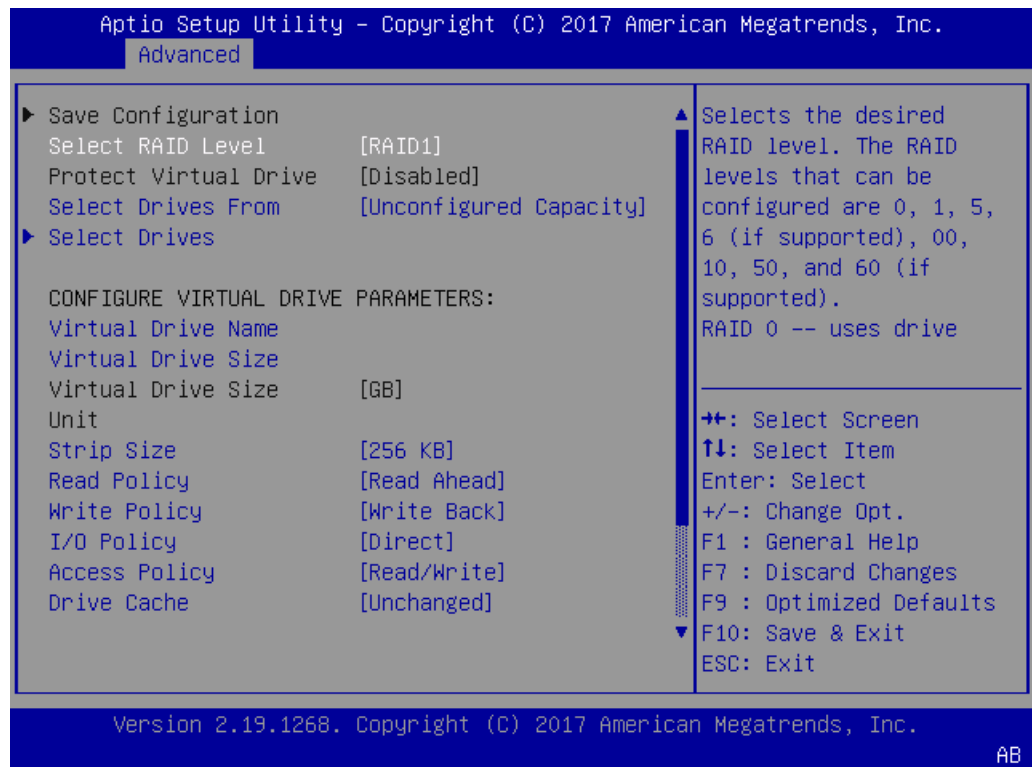
- d. Access the menu from which you create a virtual drive.

Figure 1-9 RAID - Create Virtual Drive



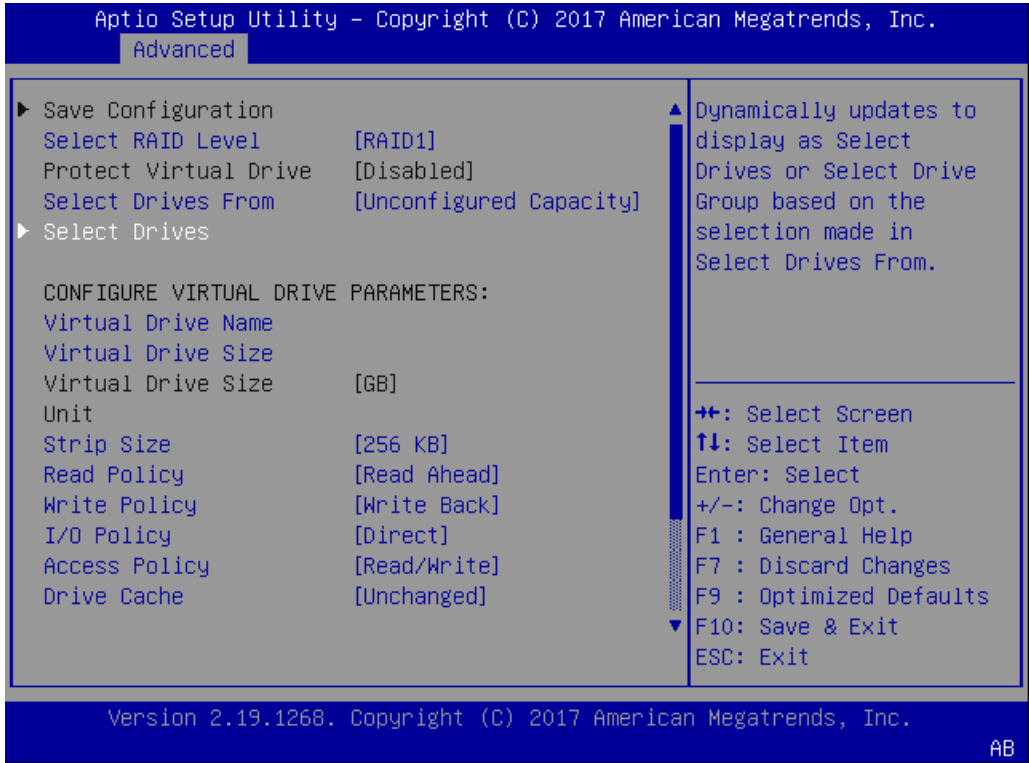
- e. Set the RAID level to RAID-1.

Figure 1-10 Set Drive to RAID1



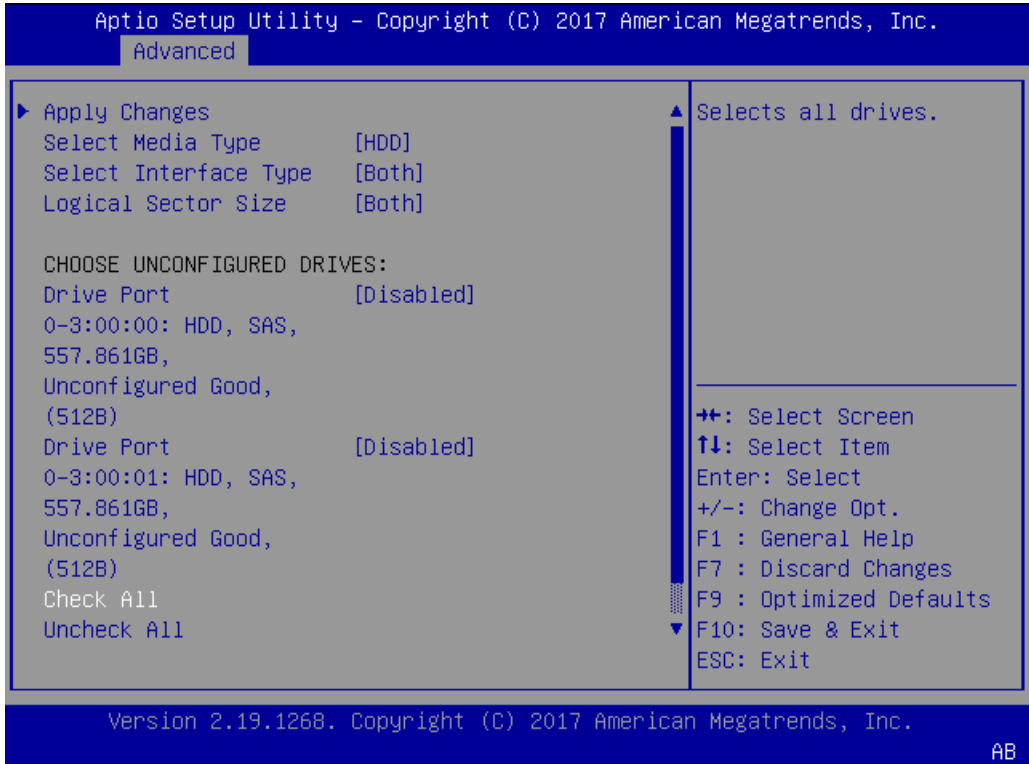
- f. Select your drives.

Figure 1-11 RAID - Select Drives



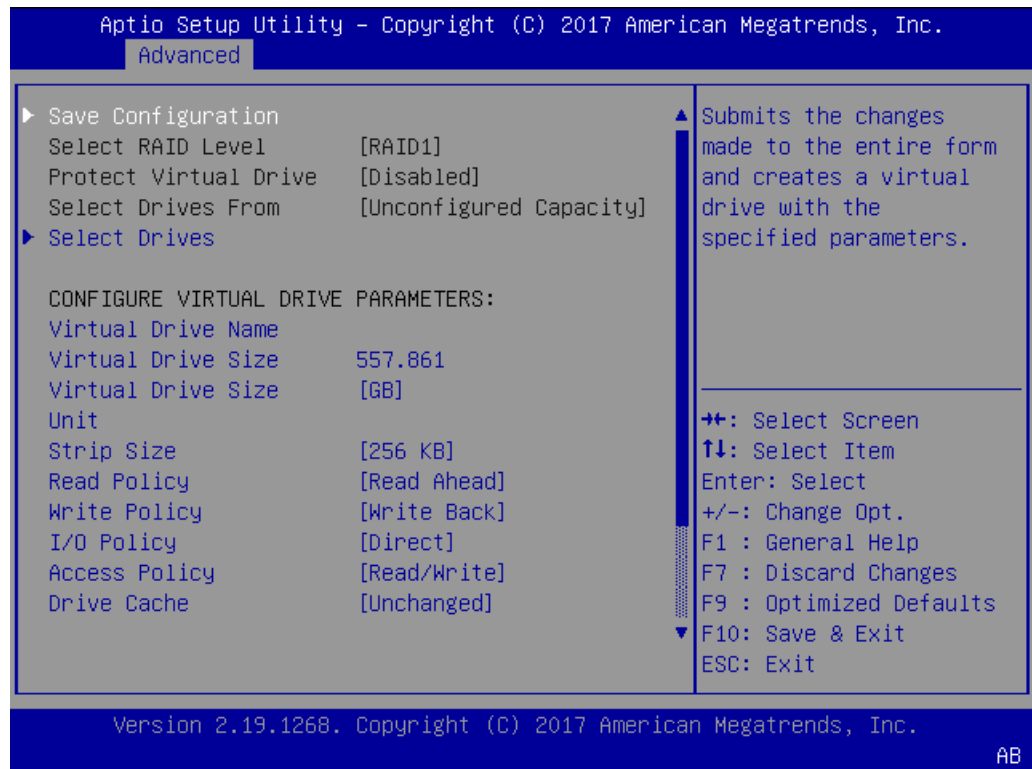
- g. It is common to select all drives at this point.

Figure 1-12 Select All Drives



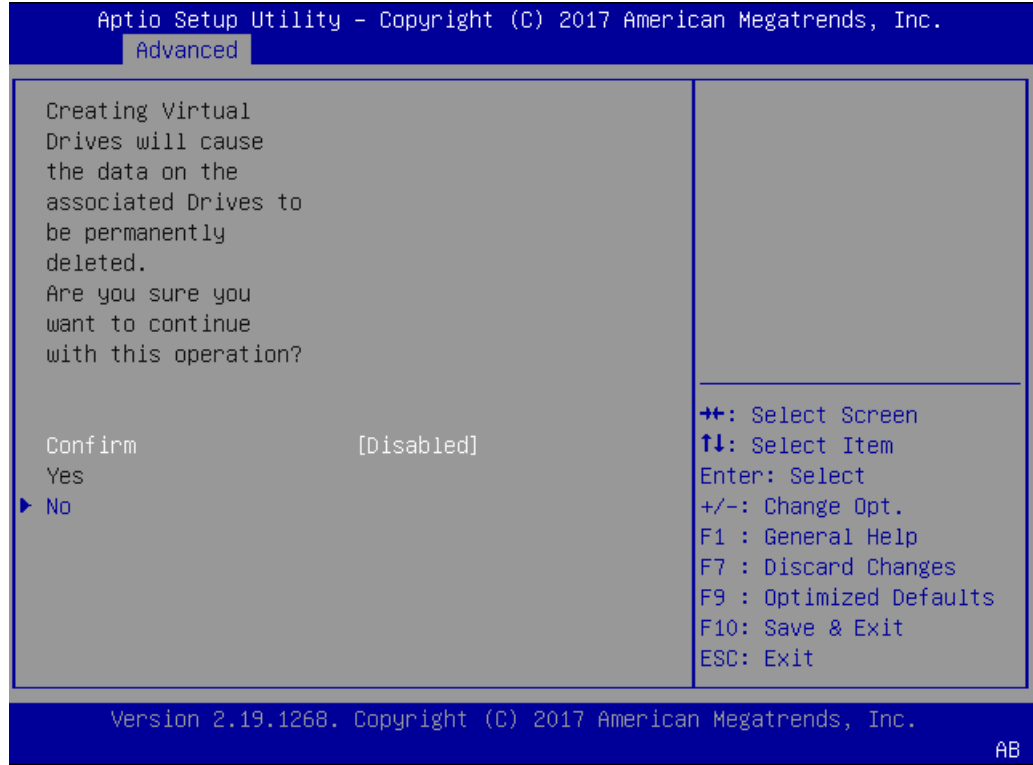
- h. Save the RAID configuration.

Figure 1-13 Save RAID Configuration



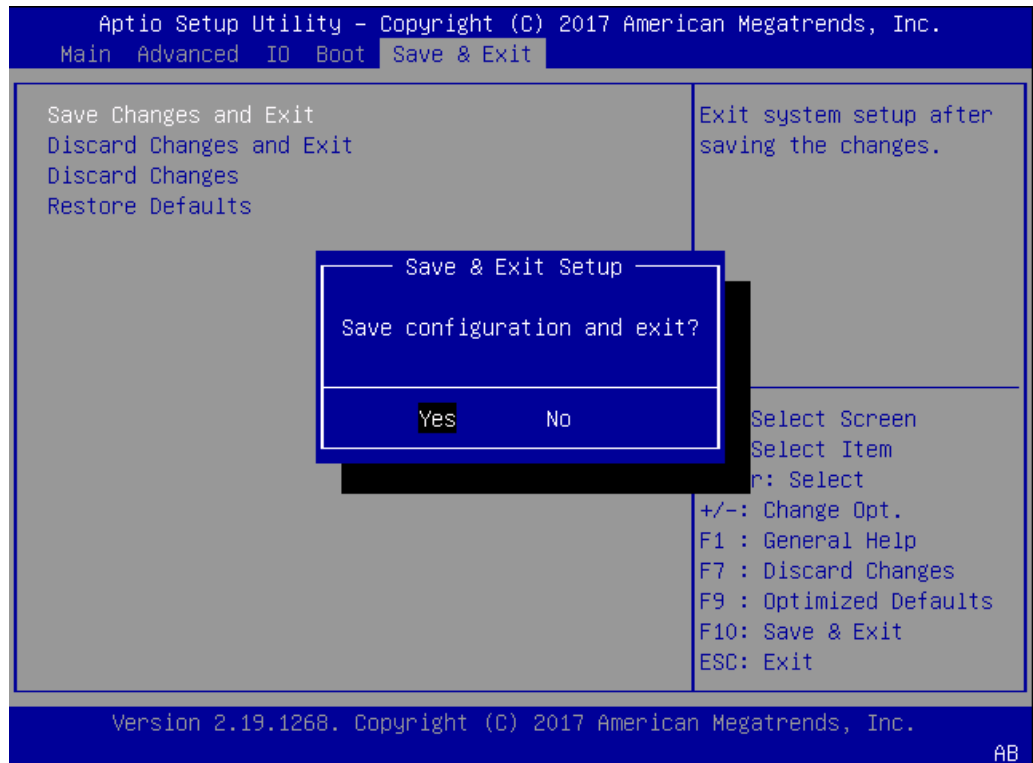
- i. The system allows you to Confirm your configuration and continue with initialization.

Figure 1-14 Initialize RAID Configuration



- j. After the initialization completes, return to the Main Menu to Save and Exit.

Figure 1-15 Exit RAID Configuration



5. Perform a cold shutdown by removing all system power.

Available Connections

Please read all of the information for each of the available connections prior to cabling the Oracle X8-2.

Port	Description	You Need:
NET (0-2)	<p>From left to right:</p> <ul style="list-style-type: none"> • 1 GigE ports - Net 0 • 10 GigE ports - Net 1, Net 2 <p>Enables you to connect the X8-2 to your network.</p>	<p>A Category 6 (or better) Ethernet cable to connect to the NET 0 port to your network</p> <p>Network parameters such as an IP address (can be provided by DHCP services or assigned a static address in the OS)</p> <p>Additional Category 6 (or better) Ethernet cables and Ethernet addresses as needed for additional connections to NET 0, 1 and 2.</p>
NET MGT	<p>Provides a 10/100/1000 BASE-T Ethernet connection to the Service Processor (SP) through an RJ-45 connector. The NET MGT port provides support connections to the SP using the Oracle Integrated Lights Out Manager (ILOM) CLI and Web interface. By default, the NET MGT port is configured to use DHCP to automatically obtain an IP address. Alternatively, you can assign a static IP address to the NET MGT port. To use the NET MGT port, you must configure its network settings. When configured, use the NET MGT port IP address to log on to the device using a browser or secure shell.</p>	<p>Category 6 (or better) Ethernet cable to connect the NET MGT port to your network</p> <p>IP address for this port (required from DHCP or a static address)</p>
SER MGT (COM1)	<p>Provides a TIA/EIA-232 serial Oracle/Cisco standard connection to the SP through an RJ-45 connector.</p> <p>SER MGT (COM1) connects to either Service Processor by default, but can be redirected to the host.</p> <p>Default settings:</p> <ul style="list-style-type: none"> • 8N1: eight data bits, no parity, one stop bit • 9600 baud (change to 115200 baud) • Disable hardware flow control (CTS/RTS) • Disable software flow control (XON/XOFF) 	<p>A terminal device (For example, terminal, connection to a terminal server, or computer such as a laptop running terminal emulation software)</p> <p>A cable to connect the terminal device to the SER MGT (COM1) port</p>

Port	Description	You Need:
USB	Provides USB3.0 connection to the computer. You can connect and disconnect USB cables to the USB port without affecting server operations.	Installation media Note: Maximum USB cable length: 5 meters

Cable the Oracle X8-2

After mounting the Oracle X8-2 in an equipment rack and installing all components, use the following instructions to connect all appropriate data cables to the ports before powering the system up and beginning the configuration.

Oracle qualified the following configurations of the Oracle X8-2.

- Configuration A: One Four-port 10 GigE NIC
- Configuration B: Two Four-port 10 GigE NICs (each of the three slots are qualified)
- Configuration C: One QSFP NIC (in quad port mode only) and ONE Four-port 10 GigE NIC

 **Note:**

The X8-2 does not support the 40G interface speed.

On board interfaces for all configurations include:

- One RJ-45 serial management (SER MGT) port
- One 10/100/1000BASE-T RJ-45 Oracle Integrated Lights Out Manager (ILOM) service processor (SP) network management (NET MGT) port
- One 1000BASE-T RJ-45 Gigabit Ethernet (GbE) port, labeled NET 0
- Two 10/25GbE SFP+ Ethernet ports, labeled NET 1 and NET 2
- Two 10GBASE-T RJ-45 Gigabit Ethernet (GbE) ports, labeled NET 1 and NET 2

 **Note:**

The 10/25GbE SFP+ Ethernet NET 1 port is the HA port. When using an SFP+ port, network connectivity is disabled on the 10GBASE-T RJ-45 GbE (NET 1) Ethernet port.

Figure 1-16 Oracle X8-2 Configuration A (4x10 GigE NIC)

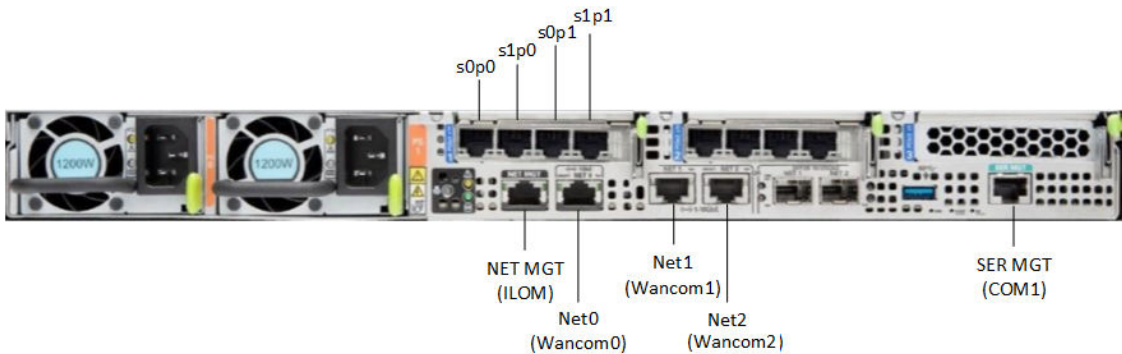


Figure 1-17 Oracle X8-2 Configuration B (Two 4x10 GigE NICs)

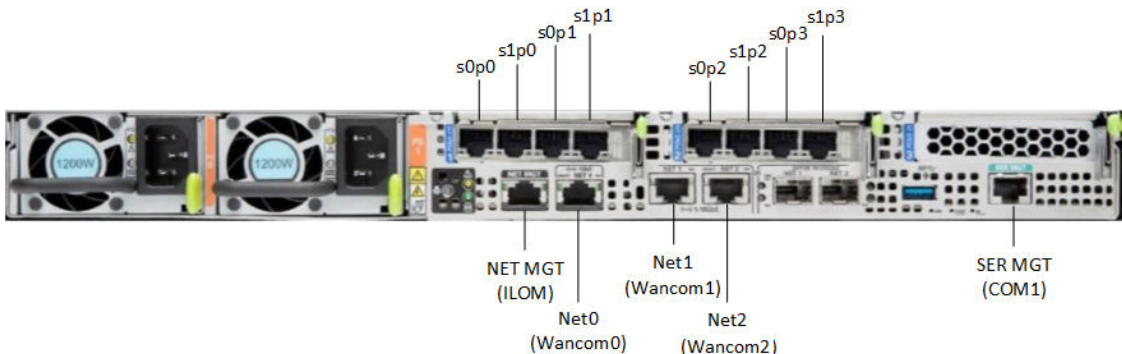
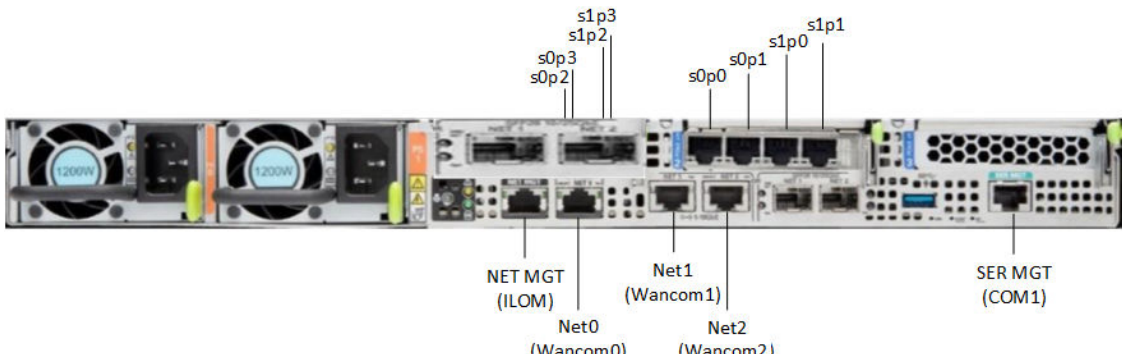


Figure 1-18 Oracle X8-2 Configuration B (One QSFP and One 4x10 GigE NICs)



Caution:

Please review your Oracle X8-2 Product Notes. Notes for release 1.1.2 describes physical issues with some optical transceivers installed into an SFP28 port.

Oracle recommends using Category 6 (or better) for all Ethernet connections.

You do not need to use every port for proper operation.

Cable the Local Console

You can connect the Administration console to the local SER MGT (COM1) serial console port. You can also operate the console using serial emulation over ILOM.

To cable a console connection:

- Connect a serial console cable with an RJ-45 connector to the SER MGT port.
- Connect ethernet to the ILOM port and use serial emulation.

When configuring boot loader parameters, set the **Console Device** to COM1. Never use COM2 or VGA. The Oracle X8-2 server cannot boot the system when set to the default of VGA. You must change this bootparameter when deploying over this platform.

Refer to the section "Change Boot Parameters by Interrupting a Boot in Progress" within the *Installation and Platform Preparation Guide* to learn how to set your **Console Device** bootparameter to "COM1". Refer to the section [Set Up a Remote Console \(SSH\)](#) to learn how to run an SSH session via iLOM using Virtual Serial Port Emulation.

Follow this procedure to cable your console:

1. Locate the appropriate cables to connect to the Oracle X8-2.
2. To cable a serial connection, insert the serial console cable into the SER MGT (COM1) port.

Figure 1-19 Connecting to USB and SER MGT (COM1) Ports



Note:

Refer to the Oracle X8-2 hardware documentation for information on how to configure the terminal application to connect to the console, and how to establish communications with the Oracle X8-2.

3. For installation procedures, insert the USB stick in the USB port.
4. Lead the cables neatly away from the rear panel.

5. Plug in the cables to their respective destination components.

Connect ILOM to the Oracle X8-2

Use the following procedure to make a connection to the Oracle X8-2 Oracle Integrated Lights Out Manager (ILOM) port. For a remote permanent connection to the Service Processor over the ILOM connection, use the rear panel NET MGT port.

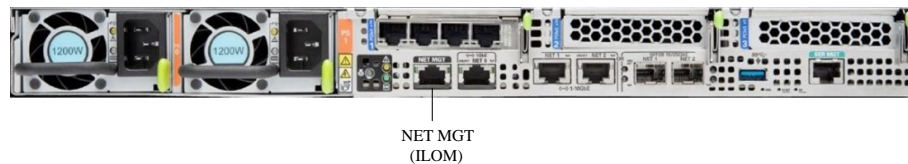
Note:

Keep Ethernet cables separated from power cables by at least 60mm where possible and never run them in the same channel of the rack without segregation.

- Category 6 (or better) Ethernet

1. Locate the cable to connect to the Oracle X8-2 for Communications.
2. Plug the RJ-45 connector into the ILOM port.

Figure 1-20 Connecting to ILOM over the Network



3. Lead the cable neatly away from the rear panel.
 4. Connect the other end of the cable to the LAN.
- Refer to the [Oracle X8-2 hardware documentation](#) for information about how to configure the Web browser application to connect to the console, and how to establish communications with the Oracle X8-2.

Install the Software on the X8-2

The Oracle Communications Session Router (OCSR) requires software installation when deployed on the Oracle X8-2.

Software installation to Oracle X8-2 includes the following high-level steps:

1. Insert your installation media into the USB slot or connect the ISO image by way of Oracle Integrated Lights Out Manager (ILOM) virtual media.
2. Power on the Oracle X8-2.
3. Observe the startup process, and press F8 to enter the boot menu when it becomes available.
4. Select the bootable USB or ISO setting.

Note:

You may need to scroll through the list to reach the ISO setting.

5. Save and exit the boot menu.
The Oracle X8-2 starts the OCSR installation.
6. Change the Console Device boot parameter to COM1 during installation. If you miss this change during the installation, power on and off the device or catch the boot parameter interrupt and change as soon as possible.
7. Remove the USB media when prompted by the Oracle X8-2.
8. Allow the Oracle X8-2 complete the installation process and boot to the newly installed OCSR software.

Next Steps After the Software Installation

Oracle recommends the following steps after installation on the Oracle X8-2 platform on the OCSR.

1. Execute the OCSR **format hard-disk** command, per your requirements. See the "Formatting the Disk Volume" for reference and instructions. .
2. Turn off the OCSR using the **Halt** command. This provides a graceful software shutdown, after which the hardware is still powered on.
3. Power cycle the hardware using the power switch, a power controller, or by physically disconnecting and reconnecting the power cable.

To configure the OCSR, refer to the *ACLI Configuration Guide*.

Boot parameter changes to consider prior to service configuration include:

- Set the **Target Name** to your preferred OCSR name.
- Set the **Console Device** to COM1 (serial).
- Set the **IP Address** to your preferred management port IP address.
- Set the **Netmask** for your management port IP address.
- Set the **Gateway** address for your management port IP address.

Note:

The boot parameters default Boot File is "/boot/bzImage". Be aware that upgrading code includes obtaining images with, for example, an SCz prefix and the .bz file extension.

Cable a Netra Server X8-2 for Acme Packet HA Node

The following procedure explains how to cable a Netra Server X8-2 for Acme Packet High Availability (HA) node using single rear interface support.

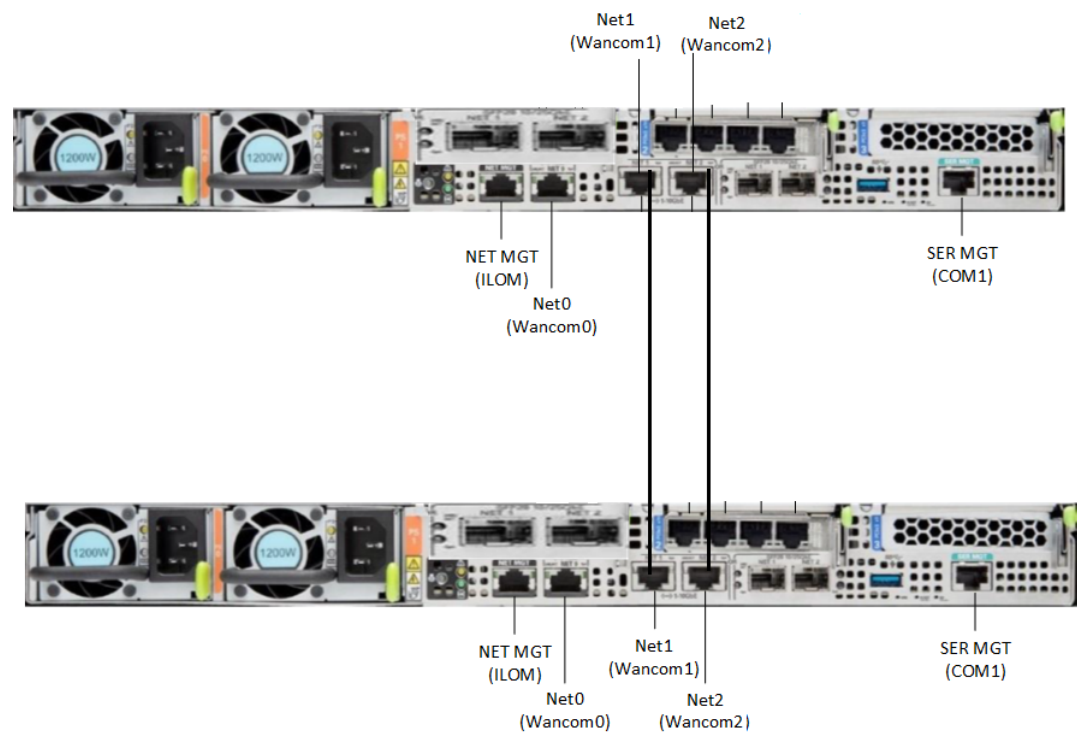
1. Insert one end of an Ethernet cable into **wancom1 or wancom2** on the rear panel of Netra Server X8-2 for Acme Packet #1. The release tab on the RJ-45 jack clicks into place when you insert it properly.

NOT_SUPPORTED:

The Communications Broker 4.0.0 Release enables support for High Availability using the Wancom1 or Wancom2 ports. With this enhancement you can use Communications Broker as HA with either wancom1 or wancom2 or both wancom1 and wancom2 ports (wancom2 as redundant interface). Please note, support for this feature is available on X8-2 hardware. For Wancom2 support, connect the cable between the Wancom2 ports on both hardware (similar to Wancom1).

- Review the Cabling diagram:

Figure 1-21 Cable the Rear Interface for HA



- Insert the other end of the Ethernet cable into the corresponding management interface on the rear panel of the Netra Server X8-2 for Acme Packet #2 as shown in the following illustration.

For example, If you use **wancom1** or **wancom2** on Netra Server X8-2 for Acme Packet #1, then you connect it to **wancom1** or **wancom2** on Netra Server X8-2 for Acme Packet #2.

HA Cabling

Category 6 Ethernet cables are required for cabling two HA nodes together.

Rear Panel Cabling for HA

You can use one connection for High Availability (HA) redundancy support between the two members of an HA node. Oracle recommends reserving **wancom0** as the boot and maintenance interface. You can use **wancom1** or **wancom2** for sharing HA information.

Figure 1-22 Oracle X8-2 Configuration A (4x10 GigE NIC)

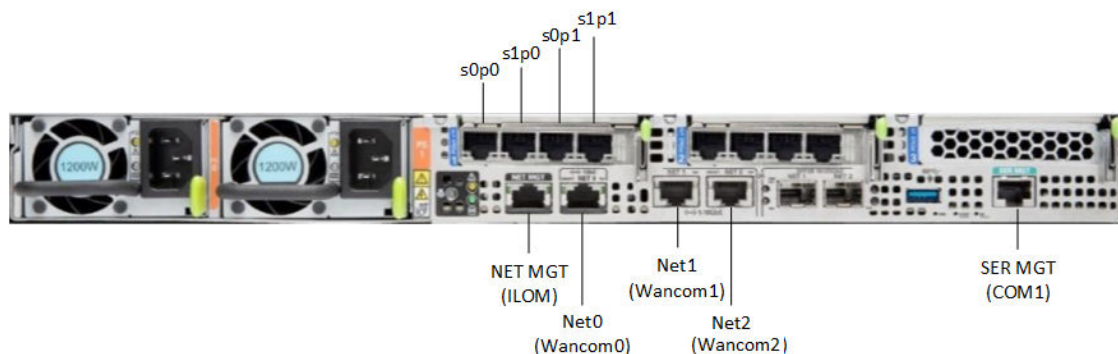


Figure 1-23 Oracle X8-2 Configuration B (Two 4x10 GigE NICs)

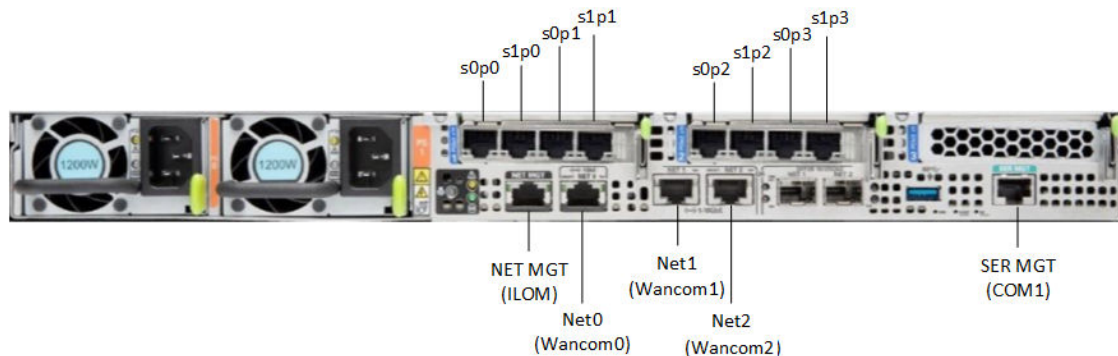
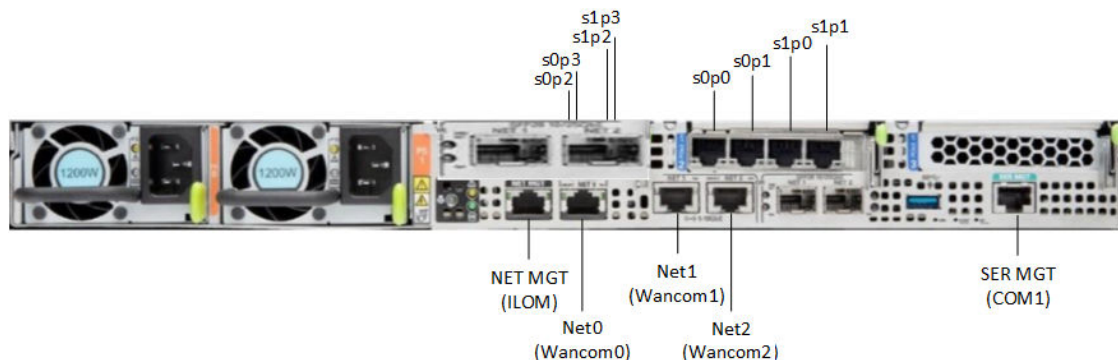


Figure 1-24 Oracle X8-2 Configuration B (One QSFP and One 4x10 GigE NICs)



Cable a Netra Server X8-2 for Acme Packet HA Node

The following procedure explains how to cable a Netra Server X8-2 for Acme Packet High Availability (HA) node using single rear interface support.

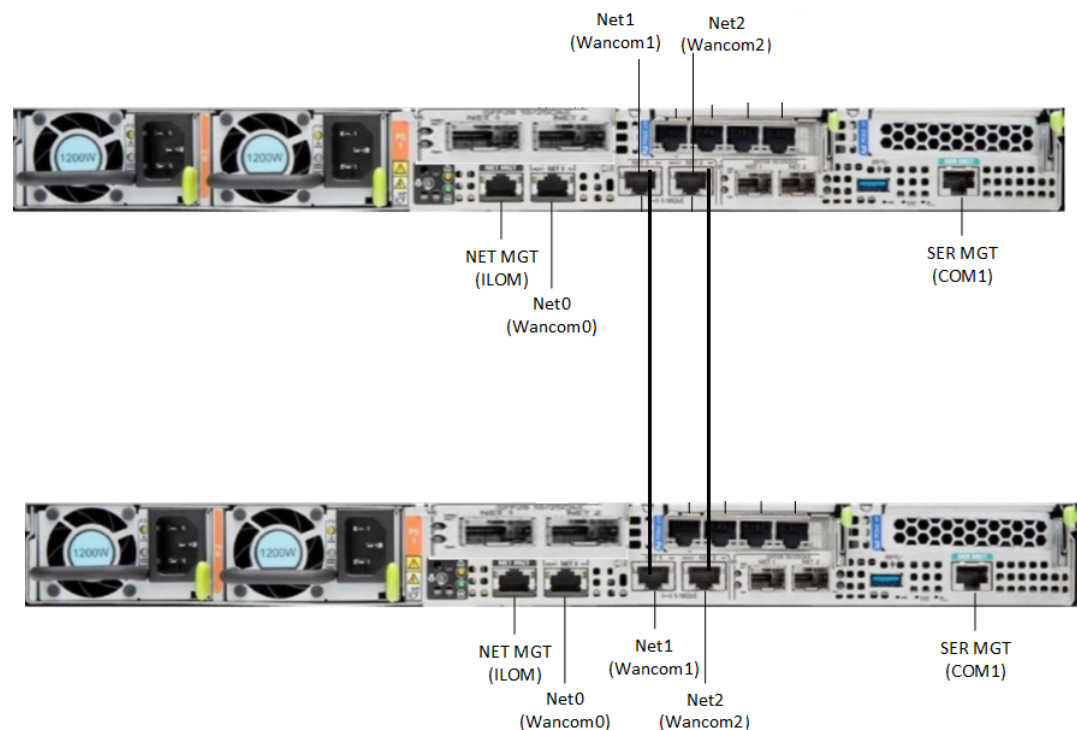
1. Insert one end of an Ethernet cable into **wancom1** or **wancom2** on the rear panel of Netra Server X8-2 for Acme Packet #1. The release tab on the RJ-45 jack clicks into place when you insert it properly.

NOT_SUPPORTED:

The Communications Broker 4.0.0 Release enables support for High Availability using the Wancom1 or Wancom2 ports. With this enhancement you can use Communications Broker as HA with either wancom1 or wancom2 or both wancom1 and wancom2 ports (wancom2 as redundant interface). Please note, support for this feature is available on X8-2 hardware. For Wancom2 support, connect the cable between the Wancom2 ports on both hardware (similar to Wancom1).

2. Review the Cabling diagram:

Figure 1-25 Cable the Rear Interface for HA



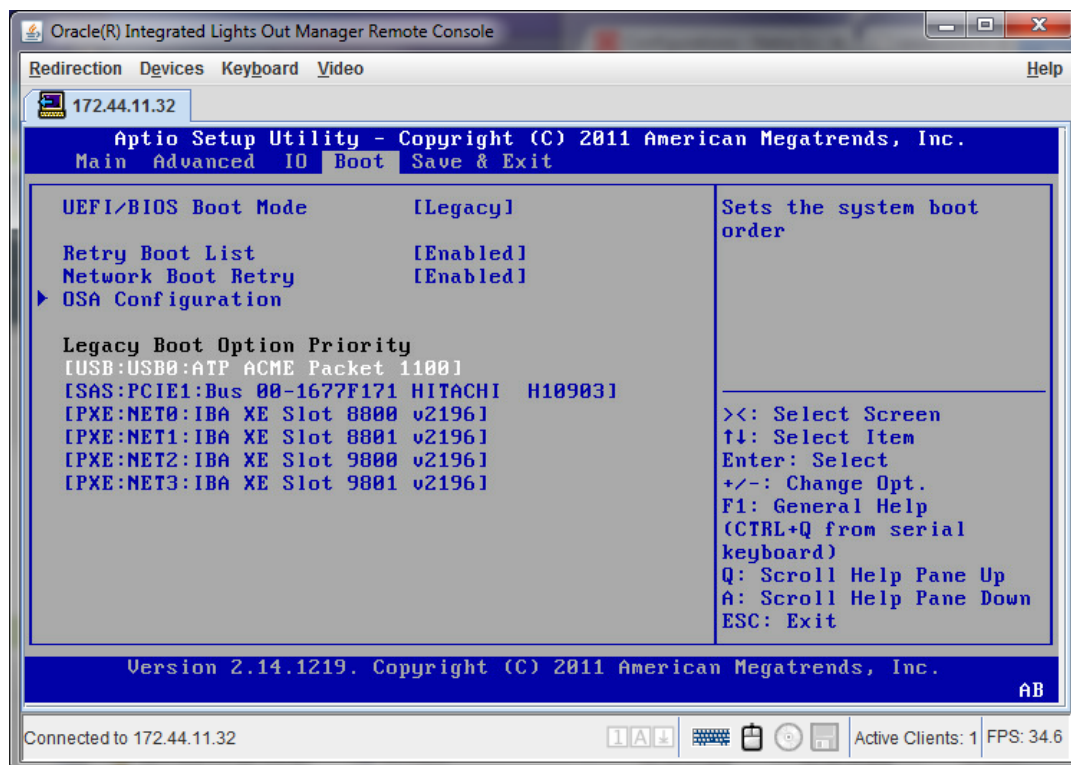
3. Insert the other end of the Ethernet cable into the corresponding management interface on the rear panel of the Netra Server X8-2 for Acme Packet #2 as shown in the following illustration.

For example, If you use **wancom1** or **wancom2** on Netra Server X8-2 for Acme Packet #1, then you connect it to **wancom1** or **wancom2** on Netra Server X8-2 for Acme Packet #2.

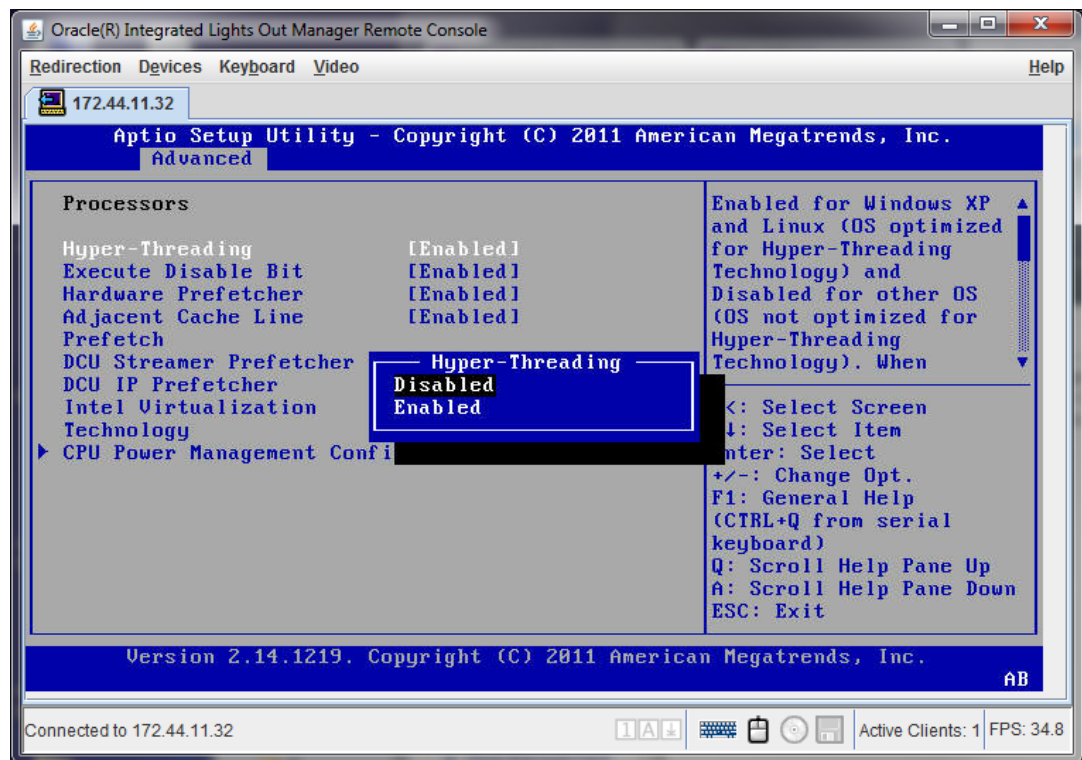
Configure the BIOS Setting

The Netra Server X3-2 requires the following changes to run Oracle Enterprise Communications Broker. This procedure shows where to make changes in the BIOS setup utility.

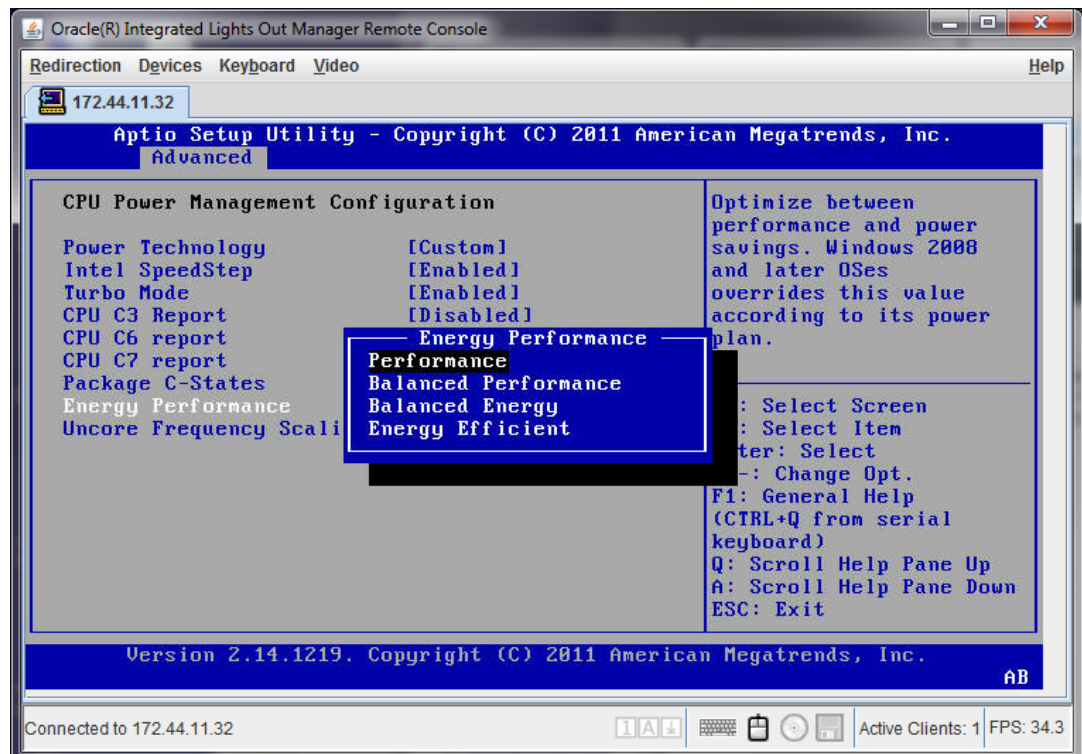
1. Set the USB slot as the first boot device, making the disk controller the second boot device.



2. Set Hyper-Threading to **Disabled**.



3. Change Energy Performance to **Performance**.



4. After setting Performance, press **Escape** to return to the main menu.
 5. Select **Save and Exit** to apply the changes.
- The system reboots using the newly configured settings.

Virtual Systems

The Communications Broker Software Only distribution is designed to operate on virtual machines running on generic, off-the-shelf servers. Refer to your version's Release Notes to see what hypervisors can support the .Communications Broker.

You can install the virtual machine software on the hardware of your choice. The number of VMs supported by a server is constrained only by the resources on your system.

Minimum VM Resources

Each VM instance requires the following minimum allocation or network resources.

- CPU cores: 5
- Memory: 8GB RAM
- Hard drive storage: 40GB
- Interfaces: 4 recommended



Note:

These resources support up to 120,000 users in the database.

If your deployment requires supporting more than 120,000 entries in the user database (up to one million), use the following resources.

- CPU cores: 8
- Memory: 16GB RAM
- Hard drive storage: 40GB
- Interfaces: 4 recommended

Format Hard Drive

Run the command **format-hard-drive**, as described in the *Oracle® Enterprise Session Director Configuration Guide* immediately after successful installation.

Log On to the System

Communications Broker requires you to set passwords for the Admin and User accounts the first time you power up a new or factory reset system by way of local access. You cannot access the Admin and User accounts until you set the corresponding passwords. Use either an SSH connection or console connection when setting passwords. You can log on to the system after setting passwords.

Before you begin, plan your passwords to meet the following requirements:

- 8-64 characters
- Include three of the following:
 - Lower case letters
 - Uppercase letters

- Numerals
- Punctuation

The system leads you through the process for setting the Admin and User passwords, as follows:

1. Power up Communications Broker.

The system prompts you to set the User account password.

2. At the prompt, type **acme**, and press ENTER. The system prompts you to enter the password that you want for the User account.

3. Set your User account password, and press ENTER.

4. Type **enable**, and press ENTER.

The system prompts you to set the Admin account password.

5. Type **packet**, and press ENTER.

The system prompts you to enter the password that you want for the Admin account.

6. Set your Admin account password, and press ENTER.

The system logs you in as Admin.

7. Type **setup prod**, and press ENTER.

8. Type 1, and press ENTER.

9. Type 5, press ENTER, and save.

10. Type **setup entitlements**, and press ENTER.

11. Type 1, and press ENTER.

12. Set the Session Capacity from 0-32000, press ENTER, and save.

13. Type **run setup**, and press ENTER. This initializes the product and configures, for example, the HTTP Server object for using the GUI.

See [Setting Up System Basics](#) for a complete description of the **run setup** command and its equivalent Set Initial Configuration system operation.

14. Access the GUI using the configured management address to proceed with further configuration.

2

Hardware Installation Summary

Installing the Oracle Enterprise Communications Broker (Communications Broker) in your rack requires the steps summarized below. This checklist is only an overview. It is not designed to substitute for following the detailed procedures in the hardware installation guides.

If running the Communications Broker as a virtual application, refer to the hardware vendor's installation instructions for hardware to learn how to access the software while it boots. From a console connection, there is little difference to the way successful startup appears as an appliance versus a virtual machine.

1. Unpack the Communications Broker.
2. Install the Communications Broker into your rack.
3. Install power supplies.
4. Install fan modules.
5. Install physical interface cards.
6. Cable the Communications Broker.

Make sure you complete installation procedures fully and note the safety warnings to prevent physical harm to yourself and/or damage to your Communications Broker.

After you complete the hardware installation procedures, establish a connection to the Communications Broker. Then load the Communications Broker software image you want to use and establish basic operating parameters.

Connecting to The Oracle Enterprise Communications Broker

By default, Oracle delivers the Oracle Enterprise Communications Broker (Communications Broker) with no management IP address. You must set this address the first time you start the system. See the System Boot section.

You can connect to the Communications Broker through a direct console connection or by creating a remote SSH session. Both methods provide a wide range of configuration, monitoring, and management options. IP-based management access, including SSH and the web GUI, requires an IP address for your management port. This address is specified in the **IP Address** boot parameter.



Note:

The system displays the **IP Address** parameter with different names, depending on the context.

- The boot parameters field name is **IP Address**.
- The initial configuration wancom0 address field name is **Management Interface IP Address**.
- The ACLI **Show Interfaces** command field name is **wancom0**.

By default, Oracle enables SSH, SFTP, and web GUI connections to the Communications Broker. The connections are only accessible by way of the **IP Address**. You cannot use SSH, SFTP, or the web GUI until you set the IP address.

Depending on the platform, you may need to install the software installation upon first startup. Use the console connection to perform and monitor software installation. The Communications Broker requires most configuration by way of the web GUI. Procedures requiring the ACLI include the following:

- Change the default management interface IP address.
- Format the hard drive.
- Set and change the password.
- Set and change the SIP Monitor and Trace filters.

Local Connections and Time-outs

The ACLI is available through serial and SSH connections. Prior to software installation, you reach the ACLI through a local, serial connection.

When deploying the Oracle Enterprise Communications Broker (Communications Broker) on a virtual machine, the virtual machine manager provides console access through a virtual serial connection. See the documentation for your virtual machine to learn how to access the console. Working with the virtual machine console is the same as working on dedicated hardware.

When deploying on dedicated hardware, refer to "Applicable Platforms" in the hardware documentation for instructions on connecting to the Communications Broker console.

Plug one end of the cable into your terminal and the other end into the RJ-45 port, located on the back of your server.

To set up a console connection to the Communications Broker:

1. Set the connection parameters for your terminal to the default boot settings:
 - Baud rate: 115,200 bits/second
 - Data bits: 8
 - Parity: No
 - Stop bit: 1
 - Flow control: None
2. Connect your PC to the Communications Broker with a serial cable. Refer to your hardware documentation for the location of your server's serial port.
3. Power on the Communications Broker.

The system boots. Upon successful boot, the system prompts you to log on.

Password:

4. Enter **acme** when prompted to log into User mode of the ACLI.

The system displays the ACLI's user mode prompt :

ORACLE>

5. If necessary, enter Superuser mode by typing **enable** at the ACLI prompt, and press Enter. The system ACLI prompts you for the superuser password:

```
ORACLE>enable
Password:
```

6. Enter **packet** to log into Superuser mode of the ACLI. The system changes the ACLI prompt to:

```
ORACLE#
```

7. Proceed with system configuration or setup.

You can control the amount of time it takes for your console connection to time out by setting the **Console Timeout** parameter in the system configuration. Default: 0, which means no time out enforcement. When your connection times out, the Communications Broker displays the login sequence again and prompts you for the passwords.

SSH Connections and Time-outs

You can use SSH to connect to the Oracle Enterprise Communications Broker (Communications Broker) and provision the Communications Broker remotely through the management interface over IP. You configure the management interface IP during system setup, or by way of the Communications Broker boot parameters.

The Oracle Enterprise Communications Broker can support up to five concurrent SSH and SFTP sessions. Note that only one user can carry out configuration tasks at a time.

To connect to the Communications Broker, you need to know the IP address of its administrative interface (wancom0). You can find the Communications Broker wancom0 IP address by using the ACLI to display the boot parameter value named **IP Address**.

You can manage the SSH connections to the Communications Broker by setting certain ACLI parameters and by using certain commands:

- To view the users who are currently logged into the system, use the **show users** command. You can see the ID, timestamp, connection source, and privilege level for active connections.
- From Superuser mode in the ACLI, you can terminate the connections of other users to free up connections. Use the **kill user** command, with the corresponding connection ID.
- When you reboot the Communications Broker from an SSH session, you lose IP access and the connection.

Initiate SSH without Username and Password

Many SSH clients allow you to initiate an SSH connection without specifying a username. To initiate an SSH connection to the Oracle Enterprise Communications Broker (Communications Broker) without specifying usernames and SSH user passwords:

1. Open your SSH client.
2. At the prompt in the SSH client, type the **ssh** command, a space, the IPv4 address of your Oracle Enterprise Communications Broker, and press Enter.

The SSH client prompts you for a password before connecting to the Communications Broker. Enter the Communications Broker User mode password. After authentication, an

SSH session is initiated and you can continue with tasks in User mode or enable Superuser mode.

Note that some clients interpret SSH session initiation without a Username as a means of logging in with your system login name. The preceding procedure does not work for such clients.

 **Note:**

You can also create connections to the Communications Broker using additional Username and password options.

SSH with Username and Password

To initiate an SSH connection to the Oracle Enterprise Communications Broker with an SSH username and password:

1. In the ACLI at the Superuser prompt, type the **ssh-password** and press Enter. Enter the name of the user you want to establish. Then enter a password for that user when prompted. Passwords do not appear on your screen.

```
SYSTEM# ssh-password
SSH username [saved]: MJones
Enter new password: 95X-SD
Enter new password again: 95X-SD
```

 **Note:**

After you configure ssh-password, the SSH login accepts the username and password you set, as well as the default SSH/SFTP usernames: `User` and `admin`.

2. Configure your SSH client to connect to your Oracle Enterprise Communications Broker's management IPv4 address using the username you just created. The standard version of this command would be:

```
ssh -l MJones 10.0.1.57
```

3. Enter the SSH password you set in the ACLI.

```
MJones@10.0.2.54 password: 95X-SD
```

4. Enter your User password to work in User mode on the Oracle Enterprise Communications Broker. Enable Superuser mode and enter your password to work in Superuser mode.
5. An SSH session window opens and you can enter your password to use the ACLI.

GUI Access

To access the Oracle Enterprise Communications Broker (Communications Broker) for ongoing configuration and management, you must use the GUI. The system allows only a few user and provisioning procedures by way of the ACLI, such as setting the initial management IP address and changing GUI access passwords. The system does not allow disabling the GUI.

You can configure GUI access by way of HTTP or HTTPS at the configured management address, which you must set prior to attempting to log on.

When a user accesses the GUI, the Communications Broker displays the log on screen. Upon successful log on, the system allows access to the System Administration and Service Provisioning controls.

Setting Your Login Banner

The Oracle Enterprise Communications Broker allows the user to create and edit the message displayed in the Login banner dialog, which appears upon successful login.

1. Click the **System** tab.

The Oracle Enterprise Communications Broker displays the system panel.

2. Open the **System Operations** Link.

The Oracle Enterprise Communications Broker lists the Systems Operations commands.

3. Click the **Set login banner** link.

The Oracle Enterprise Communications Broker displays the **Set login banner** dialog, which includes a text box allowing the user to write a login message.

4. Type your banner text and save.

The Oracle Enterprise Communications Broker sets the login banner.

System Boot

Whenever your Oracle Enterprise Communications Broker boots, the following information about the tasks and settings for the system appear in your terminal window.

- System boot parameters
- From what location the software image is being loaded: an external device or internal flash memory
- Requisite tasks that the system is starting
- Log information: established levels and where logs are being sent
- Any errors that might occur during the loading process

After the loading process is complete, the ACLI login prompt appears.

Note:

You can set boot parameters using the ACLI or the GUI. Boot parameter definitions, which help you understand what you should set them to, are provided below.

Oracle Enterprise Communications Broker Boot Parameters

Boot parameters specify the information that your Oracle Enterprise Communications Broker uses at boot time when it prepares to run applications. The Oracle Enterprise Communications Broker's boot parameters:

- Allow you to set the IP address for the management interface (wancom0).

- Allow you to set a system prompt. The target name parameter also specifies the title name displayed in your web browser and SNMP device name parameters.
- Determine the software image to boot and from where the system boots that image.
- Sets up the username and password for network booting from an external FTP server.

In addition to providing details about the Oracle Enterprise Communications Broker's boot parameters, this section explains how to view, edit, and implement them.

When displaying the boot parameters, your screen shows a help menu and the first boot parameter (boot device). Press Enter to continue down the list of boot parameters.

Upload the Stage 3 Boot Loader and System Image

Whenever you upgrade the software image, upload the Stage 3 boot loader and the new system image file to the system.

The Stage 3 boot loader is generally backward compatible with previous releases, but Oracle recommends that you install the Stage3 boot loader from the same Major.Minor version as the system image. It is not normally necessary to update the boot loader when installing a maintenance or patch release when the Major.Minor release is the same.

System upgrades typically consist of transferring the new system image and Stage 3 boot loader to the system and setting boot parameters to the new system software. To ensure compatibility, copy the Stage 3 boot loader to `/boot/bootloader` before you update the boot parameters to use the new software image file. You must name the boot loader file `/boot/bootloader` on the target system with no file extension. When upgrading an HA pair, you must perform the upgrade procedure on each HA node.

Use the following procedure to upload the Stage 3 boot loader and system image.

1. Obtain the Stage 3 boot loader image file (*.boot).
2. Upload the Stage 3 boot loader image file (*.boot) as `/boot/bootloader` to your system using an SSH File Transfer Protocol (SFTP) client.
3. Upload the new system software image (*.bz) to `/boot/`.
4. Validate the boot loader by rebooting the Oracle Enterprise Communications Broker after renaming the boot loader.

```
[Downloads]$ ls -la
total 148820
drwxr-xr-x  2 bob src      4096 Jun 17 15:16 .
drwxr-xr-x 28 bob src      4096 May 21 14:17 ..
-rw-r--r--  1 bob src 10164527 Jun 17 15:15 nnPCZ300.64.boot
-rw-r--r--  1 bob src 73849839 Jun 17 15:15 nnPCZ300.64.bz
[Downloads]$ sftp admin@123.45.67.890
admin@123.45.67.890's password:
Connected to 123.45.67.890.
sftp> cd /boot
sftp> put nnPSCZ300.64.boot
Uploading nnPCZ300.64.boot to /boot/nnPCZ300.64.boot
nnPCZ300.64.boot                100% 9926KB   9.7MB/s   00:01
sftp> rm /boot/bootloader
sftp> rename nnPCZ300.64.boot /boot/bootloader
sftp> put nnPCZ300.64.bz
Uploading nnPCZ300.64.bz to /boot/nnPCZ300.64.bz
nnPCZ300.64.bz                  100%  70MB   14.1MB/s   00:05
```

```
sftp> bye
Received disconnect from 123.45.67.890: 11: Logged out.
[Downloads]$
```

 **Note:**

The Stage 3 boot loader is ready for operation after upload and filename change, but validating it before booting the new system software is good practice.

Boot Parameter Changes

You can access and edit boot parameters by using either the ACLI or by interrupting the system boot process.

 **Note:**

Changes to boot parameters do not go into effect until you reboot the Oracle Enterprise Communications Broker.

Oracle recommends that you use management port 0 (wancom0) as the boot interface, and that your management network is either:

- directly a part of your LAN for management port 0
- accessible through management port 0

Otherwise, your management messages may use an incorrect source address.

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 Boot Parameters from the ACLI

To access and change boot parameters from the ACLI:

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

```
ORACLE# configure terminal
```

2. Type `bootparam`, and press Enter. The boot device parameters display.

```
ORACLE(configure)# bootparam
'.' = clear field; '-' = go to previous field; ^D = quit
Boot File      : /boot/nnPCz100.gz
```

To navigate through the boot parameters, press Enter and the next parameter appears on the following line.

You can navigate through the entire list this way. To go back to a previous line, type a hyphen (-) and press Enter. Any value that you enter entirely overwrites the existing value and does not append to it.

3. To change a boot parameter, type the new value that you want to use next to the old value. For example, if you want to change the image you are using, type the new filename next to

the old one. You can clear the contents of a parameter by typing a period and then pressing Enter.

```
ORACLE(configure)# bootparam
'.' = clear field; '-' = go to previous field; ^D = quit
Boot File      : /boot/nnPCz100.gz /boot/nnPCz200.gz
```

After you scroll through all of the boot parameters, the system prompt for the configure terminal branch displays.

```
ORACLE(configure)#
```

4. Exit the configure terminal branch.
5. Reboot the Oracle Enterprise Communications Broker for the changes to take effect.

The ACLI **reboot** and **reboot force** commands initiate a reboot. With the **reboot** command, you must confirm that you want to reboot. With the **reboot force** command, you do not have make this confirmation.

```
ORACLE# reboot force
```

The Oracle Enterprise Communications Broker completes the full booting sequence. If necessary, you can stop the auto-boot at countdown to fix any boot parameters.

If you configured boot parameters correctly, the system prompt displays and you can go ahead with configuration, management, or monitoring tasks.

 **Note:**

If you configured the boot parameters incorrectly, the Oracle Enterprise Communications Broker goes into a booting loop and displays an error message.

```
Error loading file: errno = 0x226.
Can't load boot file!!
```

Press the space bar to stop the loop. Correct the error in the boot parameter, and reboot the system.

Change Boot Parameters by Interrupting a Boot in Progress

To access and change boot parameters by interrupting a boot in progress:

1. When the Oracle Enterprise Communications Broker is in the process of booting, you can press the space bar on your keyboard to interrupt when you see the following message:

```
Press the space bar to stop auto-boot...
```

2. After you stop the booting process, enter the letter `p` to display the current parameters, the letter `c` to change the boot parameters, or the `@` (at-sign) to continue booting.

```
[Acme Packet Boot]: c
'.' = clear field; '-' = go to previous field; ^D = quit
Boot File      : /boot/bzImage-bones64
```

To navigate through the boot parameters, press `Enter` and the next parameter displays on the following line.

You can navigate through the entire list this way. To go back to a previous line, type a hyphen (`-`) and press `Enter`. Any value that you enter entirely overwrites the existing value and does not append to it.

3. To change a boot parameter, type the new value that you want to use next to the old value. For example, if you want to change the image you are using, type the new filename next to the old one.

```
[Acme Packet Boot]: c
'.' = clear field; '-' = go to previous field; ^D = quit
Boot File      : /boot/bzImage-bones64 /boot/bzImage.gz
```

4. After you scroll through the complete list of boot parameters, you return to the boot prompt. To reboot with your changes taking effect, type `@` (the at-sign), and press `Enter`.

```
[Acme Packet Boot]: @
```

The Oracle Enterprise Communications Broker completes the full booting sequence, unless there is an error in the boot parameters.

If you have configured boot parameters correctly, the system prompt displays and you can go ahead with configuration, management, or monitoring tasks.

Note:

If you have configured the boot parameters incorrectly, the Oracle Enterprise Communications Broker goes into a booting loop and displays an error message.

```
Error loading file: errno = 0x226.
Can't load boot file!!
```

Press the space bar to stop the loop. Correct the error, and reboot the system.

Set Management IP Address

You must manually set your management IP address within the Oracle Enterprise Communications Broker's boot parameters.

To set your management interface IP, access the boot parameters using a serial console connection within the context of one of the methods described above.

1. Type the letter `c` (change) to start boot parameter editing.
2. Press `Enter` until you reach the parameter named **IP Address**.

3. Type in the desired IP address.
4. Press Enter until you reach the end of the boot parameter list.
5. Reboot your Oracle Enterprise Communications Broker.

After being set, the management interface IP address provides access to your system via ssh and the web GUI. You can verify the status of this interface using the following command to display the address and status of wancom0.

```
Oracle ECB# show interfaces brief
Slt Prt Vlan Interface IP Gateway Adm Oper
Num Num ID Name Address Address Stat Stat
-----
- - - lo 127.0.0.1 - up up
- - - wancom0 122.30.204.127/16 - up up
0 0 0 M00 122.170.1.200/16 0.0.0.0 up up
-----
Oracle ECB#
```

Format Hard Drive

Manual software installation, performed on virtual and COTs machines, does not include formatting the hard drive automatically. After manual software installation and boot parameter configuration, the user must format the hard drive from the ACLI.

Generic installation documentation may not include the requirement to format the hard-disk. Run the command **format hard-disk** from the Oracle Enterprise Communications Broker ACLI to create a persistent partition for your /opt directory, within which you can store data needed after a reboot. Perform this procedure the FIRST time you start your Oracle Enterprise Communications Broker.

Partial output is presented below. Be sure to accept all defaults presented during the format by typing the letter **y** when prompted.

```
ORACLE# format hard-disk
WARNING: Please ensure device is not currently in use by any applications
before proceeding
Continue [y/n]?: y
The following system partitions will now be created:
1: /opt 8000000 bytes
2: /crash 16218284032 bytes
Create the system partitions and filesystems as configured above [y/n]?: y
```

After the drive(s) are formatted, the system mounts the newly created partitions.

System Image Filename

The system image filename is a name you set for the image. This is also the filename the bootloader uses whenever booting your system. This filename must match the filename specified in the boot parameters. When your image is located on your Oracle Enterprise Communications Broker, the parameter should start with /boot/ to indicate that the Oracle Enterprise Communications Broker is booting from it's local /boot directory.

If the filename set in the boot parameters does not point to the image you want sent to the Oracle Enterprise Communications Broker via SFTP, then you could not only fail to load the

appropriate image, but you could also load an image from a different directory or one that is obsolete for your purposes. This results in a boot loop condition that you can fix by stopping the countdown, entering the appropriate filename, and rebooting the Oracle Enterprise Communications Broker.

Setting Up System Basics

Before configuring and deploying your Oracle Enterprise Communications Broker, you must establish some basic attributes such as new User and Superuser passwords, system prompt and enabling the Web GUI.

New User and Superuser Passwords

Acme Command Line (ACLI) passwords provide access for SSH, SFTP, and GUI sessions. Common security practices include changing these passwords from their defaults, and at intervals defined by your organization. Refer to the ACLI `secret` command, documented in the *Oracle Communications Session Border Controller ACLI Reference Guide*, for information about changing user and superuser passwords. Refer to the "Password Policy" section in the *Administrative Security Essentials Guide* for information about password requirements and policy configuration.

New System Prompt

You can set the ACLI system prompt using **Configure system** or the **Set boot parameters** System Operation. Change the **target name** value to make it meaningful within your network. The target name may be up to 38 characters. A value that identifies the system in some way is often helpful.

Setting the Initial Configuration

You can initialize the Oracle Enterprise Communications Broker (Communications Broker) from both the ACLI and the GUI, but you must use the ACLI first. You initialize the Communications Broker from the ACLI using the **run setup** command.

Immediately after installation, setting passwords, running **setup product** and running **setup entitlements** from the ACLI, perform the **run setup** procedure from the ACLI. This sets basic system elements, including the GUI, by answering a series of questions. Questions provide a default value that you can accept to enable reliable and consistent system connectivity. After completing **run setup**, use the Web GUI to configure the Oracle Enterprise Communications Broker (Communications Broker).

 **Note:**

ACLI output sometimes refers to System Operations, including the **setup product** operation using the common term, wizard.

The **run setup** process initializes or configures the following:

- Management IP address and Gateway IP address
- Web GUI
- High Availability settings

- Communications Broker access setting

You can perform this task after installation from either the ACLI or the GUI.

Initializing with Run Setup

Use the following procedure to initialize the Communications Broker so you can configure using the Communications Broker Web GUI.

1. In Superuser mode, type **run setup**, and press Enter.

```
ORACLE# run setup
```

Note:

The **run setup quiet** command, which enables a less verbose presentation, also initializes the Communications Broker.

The following displays.

```
=====
-----
Thank you for purchasing the Oracle ECB. The following
short wizard will guide you through the initial set-up.
-----
'? ' = Help; '.' = Clear; 'q' = Exit

CONFIGURATION

WARNING: Proceeding with wizard will result in existing configuration
being erased.
  Erase config and proceed (yes/no) [no]           :
=====
```

You can use:

- ‘?’ key to obtain query-specific help
- ‘.’ key to clear the field of the current setting.
- ‘q’ key to exit the process. Initiating the **q** displays the following prompt:

```
Discarding changes and quitting wizard. Are you sure? [y/n]?:
```

Enter **y** to discard any changes and quit the installation . The root prompt displays.

A “Warning” displays stating that using **run setup** overwrites the existing running configuration. If you want to back out of the setup process and not overwrite the current running configuration, you can enter **q** or enter **No** at the Erase config and proceed prompt. The root prompt displays.

2. The system offers you the opportunity to establish an HA deployment. Proceed with your setup.

3. The system offers you the opportunity to create a unique target name, which the system requires. Proceed with your setup.
4. The system offers you the opportunity to establish an management address. Proceed by setting the management address parameters. If you previously set these parameters, you can simply accept the existing settings.

```
Management interface IP address [address]      :
Management interface subnet mask [mask]       :
Management interface gateway IP address [address] :
SIP interface VLAN id (0 - 4095) [0]          :
```

5. Depending on your command syntax, **run setup** or **run setup quiet**, the system offers you the opportunity to set a SIP interface, timezone, specify the number of licensed sessions, and so forth. Follow the prompts to continue the setup process, and press Enter.
6. Your setup sequence provides you with the opportunity to save these settings.

```
Enter 1 - 27 to modify, 'd' to display summary, 's' to save, 'q' to exit.
[s]:
```

When you proceed with saving, the following displays.

```
Saving changes and quitting wizard. System will reboot. Are you sure? [y/
n]?:
```

'-' key to navigate to the previous step in the setup process, if required, and change your response.

Set Up High Availability Mode

Use the following procedure to configure High Availability (HA) on a primary and secondary Oracle Enterprise Communications Broker (Communications Broker).

Confirm that you own an HA license.

In the following procedure, enter **y** to discard any changes and quit the initialization. The system displays the root prompt.

The **run setup** command warns you that it overwrites (erases) the existing running configuration. If you want to back out of the setup process and not overwrite the current running configuration, you can enter **q** or enter **No** at the Erase config and proceed prompt. The root prompt displays.

Note:

For HA environments, running setup on the secondary system is also required to set the wancom0 address and secondary targetname. The targetname must match the same secondary targetname specified on the primary system.

To configure the primary Communications Broker for HA :

1. In Superuser mode, type **run setup**, and press Enter.

```
ORACLE# run setup
```

2. Type **yes** to continue the setup process, and press Enter.
3. Type **2** to configure an HA device, and press Enter.

```
Enter choice [1 - standalone] :2
```

The system displays the following:

```
=====
This ECB may be a standalone or part of a highly available redundant pair.
ECB role
  1. primary
  2. secondary
Enter choice [1-primary] :
=====
```

4. Enter **2**, and press Enter.

```
Enter choice [1-primary] :2
```

The system displays the following:

```
=====
SBC SETTINGS
Unique target name of this ECB [ECB name>] :
=====
```

5. Type **a unique target name for the ECB (or keep the default in [])**, and press Enter.

```
Unique target name of this ECB [ECB1] :NNESD2
```

The system displays the following:

```
=====
SBC SETTINGS
IP address on management interface [<ECB IP address>] :
=====
```

 **Note:**

The setup process provides default IP addresses for the HA primary and secondary peers (Redundancy interface address and Peer IP address). These default addresses are link-local addresses as specified in RFC 3927, *Dynamic Configuration of IPv4 Link-local addresses*.

6. Type the IP address on the management interface of the secondary Web Server (or keep the default in []), , and press Enter.

```
IP address on management interface [<ECB IP address>]      : 164.30.85.52
```

The system displays the following:

```
Subnet mask [255.255.0.0] :  
=====
```

7. Type the subnet mask of the secondary Web Server (or keep the default in []), , and press Enter.

```
Subnet mask [255.255.0.0]      : 255.255.0.0
```

The system displays the following:

```
=====
```

```
Gateway IP address [164.30.0.1]      :  
=====
```

8. Type the gateway IP address of the secondary Web Server (or keep the default in []), and press Enter.

```
Gateway IP address [<SBC gateway address>]      : 164.30.0.1
```

The system displays the following:

```
=====
```

```
AUTOMATIC CONFIGURATION  
Acquire config from the Primary (yes/no) [yes]      :  
=====
```

9. Type **y** for the secondary HTTP Server to acquire the configuration from the primary Web Server during switchover, and press Enter.
10. Type **s** to save the configuration, and press Enter. Or, Select an item number from the summary view to modify the value for that item. Or, type **q** to exit.

```
Enter 1 - 16 to modify, 'd' to display summary, 's' to save, 'q' to exit.  
[s]      :s
```

The system displays the following:

```
=====
```

```
Saving changes and quitting wizard. Are you sure? [y/n]?      :  
=====
```

Type **y** to verify you want to save the configuration, and press Enter.

he system displays the following:

```

=====
Running configuration is backed up as
'bkup_setup_wizard_Mar__7_13_58_26_545.gz'
*****
Deleting configuration
Erase-Cache received, processing.
waiting 1200 for request to finish
Request to 'ERASE-CACHE' has Finished,
Erase-Cache: Completed
Request to 'RESTORE-CONFIG' has Finished,
Restore Backup Completed Successfully
checking configuration
Save-Config received, processing.
waiting for request to finish
Request to 'SAVE-CONFIG' has Finished,
Save complete
Currently active and saved configurations do not match!
To sync & activate, run 'activate-config' or 'reboot activate'.
-----
Verification successful! No errors nor warnings in the configuration
Activate-Config received, processing.
waiting for request to finish
Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete
You may access the GUI via http://164.30.85.52:80/ after reboot.
=====

```

You have completed the **run setup** .

Run **run setup** on the secondary peer of the HA pair.

Initialize the System from the GUI

The Oracle Enterprise Communications Broker (Communications Broker) requires initialization upon the first startup. You must do this from the ACLI upon the first startup. Later, you can re-initialize the Communications Broker from the GUI.

Caution:

If you ever need to initialize the system again, be aware that all configuration is lost during initialization and that the system reboots when you click **Complete**.

If you plan to configure High Availability (HA), note that you can use the Set Initial Configuration procedure to configure the primary Communications Broker first. Upon successful configuration of the primary, HA operations begin as soon as you complete the Set Initial Configuration on the secondary Communications Broker.

1. To initialize the system from the GUI, navigate to the System tab and select the **Set initial configuration** Operation from the System Operations link.

The system displays the Configure system dialog.

2. On the Configure System dialog, do the following:

High Availability mode	Select one of the following modes: <ul style="list-style-type: none"> standalone—You want to deploy a single Communications Broker. high availability—You want to deploy Communications Brokers in pairs, connecting them together and configuring one as primary and the other as a secondary.
Unique target name of this Communications Broker	Type the name of this system. This setting has an operational impact on your high availability configuration.
Management interface IP address	Type the IP address to use for accessing the Web GUI, and press Enter.
Management interface subnet mask	Type subnet mask to use for accessing the Web GUI, and press Enter.
Management interface gateway IP address	Type the IP address to use for reaching this network's gateway, and press Enter.
SIP interface VLAN ID	Type the VLAN ID, if any, required for operation on the network of your SIP interface. Range: 0-4095
SIP interface IP address	Type the IP address to use for accessing the SIP interface, and press Enter. This step is required.
SIP interface subnet mask	Type subnet mask to use for accessing the SIP interface, and press Enter.
Setup system time zone	Select one of the following settings: <ul style="list-style-type: none"> Yes—to set the system time zone. No—to skip setting the system time zone.
System time zone	Select your time zone from the drop-down list.
Session capacity	Type the number of sessions you purchased for this Communications Broker.

- Click **Complete** to proceed with deleting the existing configuration, setting the values, and rebooting your Communications Broker.

Add a License with the Set License Function

TLS is the only software feature for which you need a license on the Oracle Enterprise Communications Broker (Communications Broker). You must obtain a TLS license before you can add it. To obtain a TLS license, you must present the correct system serial number to Oracle.

- Go to the **System** tab, **System Operations**, and select **Set License**.
The Communications Broker displays the **Set License** dialog.
- Copy the serial number for your Communications Broker and contact Customer Support by logging into My Oracle Support or calling Oracle Customer support to make the request. Oracle replies with your license.
- After you receive your license from Oracle, enter your license in the Add License field in **A License**.

The system checks the license and, if correct, installs it. If the license is incorrect, the system displays a error message.

3

System Administration

The initial configuration of the Oracle Enterprise Communications Broker (Communications Broker) establishes system operations, which you configure before configuring SIP operations with the Service Provisioning controls. The Communications Broker GUI displays controls for establishing the system operations under System Administration on the Configuration page. Use the controls to specify how to manage the system.

The following information provides high-level descriptions of the System Administration controls used to configure system operation.

- Accounting—Configure connections to RADIUS servers to collect Call Detail Records (CDR) generated by the system.
- General—Specify standard system management information parameters, such as system identification information, system management information interfaces (SNMP and Syslog), and global service configurations including Denial of Service and High Availability settings.
- LDAP—Define servers and server access rules for using an external LDAP database as a source for user authentication and routing procedures.
- Network—Specify your network and High Availability settings, and add host routes.
- Security—Configure login authentication, certificate records, TLS profiles, TL Global settings. Generate certificate requests and import certificates, add a public key, enable audit logging.
- SIP Interface—Specify the SIP interface and add SIP service ports. Configure SIP monitoring and SIP monitoring filters.
- SIP Manipulation—Create SIP header and element manipulation rules that change session service messages for interoperability, policy, and other deployment purposes.
- SIP Registrar—Create and manage a SIP registrar object on the Communications Broker to offload Agent of Record registration processes from other network elements.
- SNMP—Specify SNMP community for allowing access to READ functions and trap receivers.
- Sync—Specify Sync configuration settings and add Sync agents. Provides control over multiple Communications Broker synchronization processes, including defining applicable Communications Brokers and initiating the synchronization.
- Web Server Settings—Specify web server functionality, including HTTP and HTTPS operation. Specify the applicable TLS profile and inactivity timeout.

Save and Activate

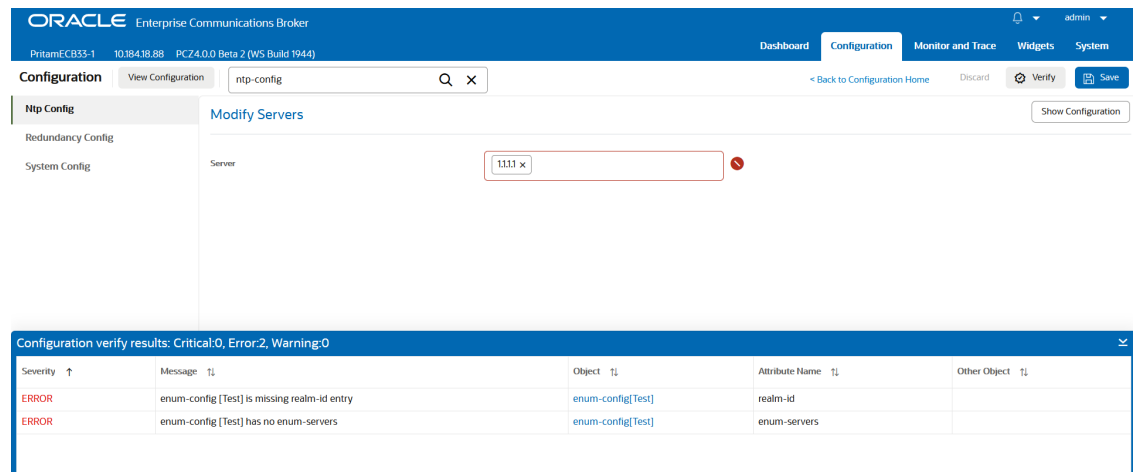
The GUI retains configuration changes until you send them to your device or discard them from the GUI.

You must also Save, then Activate your changes before your device can apply your changes. The Save button on the Configuration page initiates configuration Save and Activate procedures to the system.

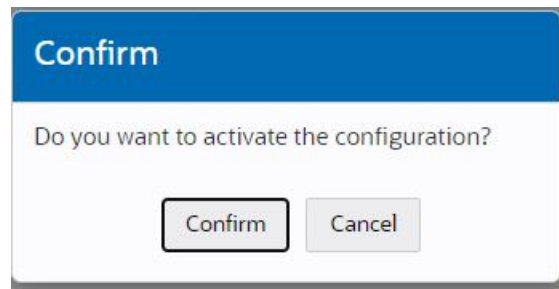
When you click Save, the GUI either saves the configuration to your device or prevents you from saving invalid data. The system highlights any fields containing invalid data, allowing you to find and correct the mistake. The system displays any configuration errors in a list at the bottom of the GUI. The following screen capture shows an example of the errors list.

The GUI also checks your configuration for errors every time you click the Save button, the system displays the following dialog if any errors occur.

Figure 3-1 Configuration Errors Screen



After the save is successful, the GUI displays a dialog asking you if you want to activate this configuration. Note that you can save without activating, for example, when you want to wait for a preferred maintenance window to apply the changes to avoid any service disruption.



The confirmation dialog defaults to “Cancel,” which leaves your changes saved to your system but not activated. Select “Cancel” if you want to activate your configuration at a later time. Select “Confirm” to activate the changes now. The GUI provides a final confirmation message indicating success when activation finishes.

General Settings and System Config Settings

Use the General Settings link on the Configuration tab to reach the General Settings and System Config pages, where you can set the following system-wide parameters.

General

Use **General Settings** to specify the following:

- Network Time Protocol (NTP) servers—Add one or more NTP servers.

- Redundancy Config—Enable and disable HA, identify the primary and secondary devices, and specify synchronization.
- System Config - Set system-wide parameters.

System Config

Use **System Config** to specify the following:

- System Settings—Set the hostname, location, and default gateway, console timeout, and restart.
- SNMP—Enable and disable SNMP, specify the MIB system, and set SNMP traps and notifications.
- Syslog Servers—Add one or more Syslog servers, specify the system log level, and specify the process log level.
- Communications Monitoring Probe—Enable and disable the Communications Monitor, set the group ID, set the TLS profile, enable and disable QoS, and add one or more Monitor collectors.
- Alarm Threshold—Set the thresholds for one or more types of alarms.

Configure an NTP Server

You can specify one or more Network Time Protocol (NTP) servers for the Communications Broker on the General Settings page by adding their Fully Qualified Domain Names or addresses in a comma-separated list.

Note:

The Communications Broker media interface does not support management traffic for NTP. When configuring connectivity to these resources, do not configure these resources within a media interface subnet range.

1. Access the NTP Configuration object.
Configuration tab, **General settings**, **Ntp Config**.
2. On the Add Servers page, enter the address or FQDN of the NTP server that you want to add.
3. (Optional) Add another NTP server to the list. (comma-separated in the NTP Servers field)
4. Click **OK**.
5. Save the configuration.

Redundancy Config / High Availability Settings

You can deploy the Communications Broker in pairs to deliver High Availability (HA) or Redundancy Config.

Two Communications Brokers operating in this way are called an HA node. In the HA node, one Communications Broker operates in the Active mode and the other one operates in Standby mode. In the event of the Active member of the node losing functionality, the system switches over to the Standby member. In the HA model, the two Communications Brokers

share the call state and communicate with each other, which keeps sessions and calls from dropping in the event of a call flow disruption.

Oracle recommends configuring High Availability with the **run setup** command from the CLI. The command populates the HA fields, which you can also find within the GUI under General Settings.

- The Active Communications Broker checks itself for internal process and IP connectivity issues. If it detects that it is experiencing certain faults, it hands over its role as the Active system to the Standby Communications Broker in the node.
- The Standby Communications Broker is the backup system, fully synchronized with the session status of the Active Communications Broker. The Standby Communications Broker monitors the status of the Active system so that, if needed, it can assume the Active role without the Active system instructing it to do so. If the Standby system takes over the Active role, it notifies network management using an SNMP trap.

Refer to the *Oracle Enterprise Session Border Controller CLI Configuration Guide* for more detail about High Availability operations, including:

- Synchronization
- Checkpointing

Overview

To produce seamless switch overs from one Communications Broker to the other, the HA node uses shared virtual MAC and virtual IP addresses for the media interfaces in a way that is similar to VRRP (virtual router redundancy protocol). Sharing addresses eliminates the possibility that the MAC and IPv4 address set on one Communications Broker in an HA node will be a single point of failure. The standby Communications Broker sends ARP requests using a utility IPv4 address and its hard-coded MAC addresses to obtain Layer 2 bindings.

When there is a switchover, the standby Communications Broker issues gratuitous ARP messages using the virtual MAC address, establishing that MAC on another physical port within the Ethernet switch. To the upstream router, the MAC and IP are still alive, meaning that existing sessions continue uninterrupted.

Within the HA node, the Communications Brokers advertise their current state and health to one another in checkpointing messages; each system is apprised of the other's status. Using Communications Broker's HA protocol, the Communications Brokers communicate with UDP messages sent out and received on the interfaces carrying "heartbeat" traffic between the active and standby devices.

The standby Communications Broker assumes the active role when:

- It has not received a checkpoint message from the active Communications Broker for a certain period of time.
- It determines that the active Communications Broker's health score has decreased to an unacceptable level.
- The active Communications Broker relinquishes the active role.

Establishing Active and Standby Roles

Communications Brokers establish active and standby roles in the following ways.

- If a Communications Broker boots up and is alone in the network, it is automatically the active system. If you then pair a second Communications Broker with the first to form an

HA node, then the second system to boot up will establish itself as the standby automatically.

- If both Communications Brokers in the HA node boot up at the same time, they negotiate with each other for the active role. If both systems have perfect health, then the Communications Broker with the lowest HA interface IPv4 address will become the active Communications Broker. The Communications Broker with the higher HA interface IPv4 address will become the standby Communications Broker.

If the physical link between the two Communications Brokers fails during boot up or operation, both will attempt to become the active Communications Broker. In this case, processing will not work properly.

Configure Redundancy Config

The Communications Broker supports configuring an pair for High Availability (HA) operations.

1. Access **Configuration, General-settings, Redundancy Config**.
2. On the Redundancy Config page, do the following:
 - State—Select to enable redundancy. Default: enable. Valid values: enable | disable.
 - Advertisement Time—Set the time, in milliseconds, between transmitting redundancy protocol updates. Default: 500ms. Valid values: 50-2147483647ms.
 - Percent Drift—Set the percentage of the advertisement time after which the peer is considered unresponsive. Default: 210. Valid values: 100-65535.
 - Becoming Standby Time—Set the time, in milliseconds, to wait for complete synchronization. Default:180000. Valid values: 5-50-2147483647ms.
 - Cfg Max Trans—Set the maximum number of redundancy config sync transactions to keep on active. Default: 10000. Valid values: 0-4294967295.
 - Cfg Sync Comp Time—Set the timeout for subsequent config sync requests after complete redundancy configuration occurs. Default: 1000. Valid values: 0-4294967295.
3. Peers—Click **Add** and do the following to add the redundancy peer.
 - Name—Enter the name of the peer.
 - Type—Click **Add** and set the type of peer. Default: Primary. Valid values: Primary | Secondary | Unknown.
 - Expand **Destinations** and do the following.
 - Address—Set the destination address for the peer. Default: 0.0.0.0.
 - Network Interface—Set the name for originating messages. Default: eth1:0.
4. Click **OK**.
5. Click **Save** to activate the configuration.

Force an HA Switchover

The Communications Broker allows you to cause a High Availability (HA) switchover manually. Executing the procedure forces the two Communications Brokers in your HA node to trade roles. The Active system becomes Standby, and the Standby becomes Active.

A successful manual switchover requires the following conditions:

- The Communications Broker from which you trigger the switchover must be in one of the following states: Active, Standby, or becoming Standby.

- A manual switchover to the Active state is allowed only on a Communications Broker in the Standby or becoming Standby state when it achieves full media, signaling, and configuration synchronization.
- A manual switchover to the Active state is allowed only on a Communications Broker in the Standby or becoming Standby state when its health score is above the value you configure for the threshold.

1. Click the **System** tab.
2. Click **Force HA Switchover**.

On clicking on Force HA Switchover, Communications Broker displays a dialog box about the current state of Communications Broker and the new state. You are prompted for a confirmation to perform the switchover.


The Communications Broker displays the **Force HA Switchover** dialog, which includes **Switch to Standby**. The Communications Broker executes the HA role change.

Configure System Config

The Communications Broker allows you to specify system identification and global settings by way of the parameters that you specify on the System Config page.

Set the following parameters to configure global system identification information.

1. Access the System Config configuration object.
Configuration tab, **General-settings**, **System Config**.
2. On the Modify System Config page, do the following.

Hostname	Enter the hostname used to identify the Communications Broker by the software. For example, the IP address for Fully Qualified Domain Name.
Description	Enter a textual description of the Communications Broker for informational purposes.
Location	Enter the location of the Communications Broker for informational purposes. For example, you might include the site name and physical address of the Communications Broker.
Default Gateway	Set the default gateway for this Communications Broker for egress traffic with no explicit destination. Default: 0.0.0.0.
	<div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #0070c0;"> <p> Note:</p> <p>Changing this parameter can cause you to lose connectivity with the Communications Broker GUI. Be prepared to access the Communications Broker console, if you lose connectivity. See the <i>Oracle Communications Session Border Controller CLI Configuration Guide</i> for instructions on setting the default gateway using the CLI.</p> </div>
Restart	Select to cause the system to restart after a service disruption. Default: enabled.

SSH Timeout	Set the length of time, in seconds, that the system waits for the next command before disconnecting. Default: 0. Range: 0-65535.
Console Timeout	Set the length of time, in seconds, that the system waits to terminate an ACLI administrative session due to inactivity. Use 0 to disable console session timeout. Default: 0. Range: 0-65535.
PKO Rate Pkt	Set the rate per second at which the system sends packet data. Default: 0. Range: 0-32768.
PKO Rate Burst	Set the maximum burst rate of packets. Default: 0. Range: 0-0124.
SNMP Agent Mode	SNMP Agent Mode. You can select the SNMP Agent mode. Allowed values <ul style="list-style-type: none"> v1v2: All SNMP V1 and V2 features are enabled. This is the default value. v3: SNMP v3 features are enabled.

3. Save the configuration.

SNMP Configuration

Use SNMP to support monitoring of devices attached to the network for conditions that warrant administrative attention on the Communications Broker

Use the MIB settings for informational purposes. The remainder of the parameters enable SNMP and the specific Communications Broker events that you want reported to the SNMP system.

Note that you configure the SNMP community and the trap receiver settings by way of the SNMP icon.

Configure SNMP Settings

Use System Config to enable SNMP on the Communications Broker and to set global SNMP settings.

Note that neither the MIB system name nor the MIB system location that you enter in the following procedure correlate to the name and location fields in System Configuration.

1. Access the System Config configuration object.
Configuration tab, **System Administration** section, **General Settings**, **System Config**.
2. On the Modify System Config page, do the following.

MIB System Contact	Set the contact information displayed in the Communications Broker MIB transactions. You can enter a textual identification of your company's contact person for the Communications Broker and information about how to contact that person.
MIB System Name	Set the identification of this Communications Broker to display in MIB transactions. Use the FQDN.
MIB System Location	Set the physical location of this Communications Broker to report in MIB transactions.

SNMP Enabled	Select to enable SNMP. Note that you must also enable SNMP, and set a snmp-syslog-level. Default: enabled.
Enable SNMP Auth Traps	Select to enable sending an SNMP trap in response to an unsuccessful authentication attempt. Default: disabled.
Enable SNMP Syslog Notify	Select to enable sending SNMP traps when the system generates an alarm. Default: disabled.
Enable SNMP Monitor Traps	<ul style="list-style-type: none"> Select to generate traps with unique trap-IDs for each syslog event. Deselect to generate a single trap-ID for all events, with different values in the description string. Default: disabled.
Enable Env Monitor Traps	Select to enable environment monitor traps for main board PROM temperature, CPU voltage, power supplies, and fan speeds. Default: disabled.

3. Save the configuration.

Logging (Syslog)

Logging events is a critical part of diagnosing mis-configurations and optimizing operations. Communications Brokers can send both syslog and process log data to appropriate hosts for storage and analysis.

Overview

The Oracle Enterprise Communications Broker generates two types of logs, syslogs and process logs. Syslogs conform to the standard used for logging servers and processes as defined in RFC 3164.

Process logs are Oracle proprietary logs. Process logs are generated on a per-task basis and are used mainly for debugging purposes. Because process logs are more data inclusive than syslogs, their contents encompass syslog log data when they are sent off box. A special application must be run on a remote server to receive process logs. Please contact your Oracle sales representative directly or calling Oracle Customer support for more information about the process log application.

Syslog and process log servers are both identified by an IPv4 address and port pair.

Process Log Messages

Process log messages are sent as UDP packets in the following format:

```
<file-name>:<log-message>
```

In this format, <file-name> indicates the log filename and <log-message> indicates the full text of the log message as it would appear if it were written to the normal log file.

Add a Syslog Server

The Oracle Enterprise Communications Broker requires a connection to at least one Syslog Server to process the log events that the system can generate for diagnosing mis-configurations and for optimizing operations. The Communications Broker supports adding up to eight Syslog servers.

1. Access the System Config configuration object.
Configuration tab, **System Administration** section, **General Settings, System Config**.
2. On the Modify System config page, under Syslog Servers, click **Add**.
3. In the Add Syslog server dialog, do the following:

Address	Set the IP address or FQDN of the server to which you want to send Syslog messages from the Communications Broker. Default: 0.0.0.0.
Port	Enter the port number on the Syslog server to which the Communications Broker sends log messages. Range: 0-65535. Default: 514.
Facility	Enter the user-defined facility value sent in every syslog message from the Communications Broker to the syslog server. This value must conform to IETF RFC 3164. Range: 0-99999999. Default: 4.

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

Configure Syslog Settings

Set the following parameters to configure system-wide Syslog and Process log functionality. Oracle recommends that you configure Debug and Trace levels temporarily and only when required because both log levels are verbose and can adversely impact system performance.

1. Access the System Config configuration object.
Configuration tab, **System Administration** section, **General Settings, System Config**.
2. On the Modify System Config page, do the following.

System Log Level	Select the severity level from the drop-down list that you want to cause the system to send a syslog trap to the Network Management System. Default: Warning.
Process Log Level	Select the severity level from the drop-down list that you want to cause the system to send a process trap to the Network Management System. Default: Notice.

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

Enterprise Operations Monitor

As a proactive call monitoring solution, the Enterprise Operations Monitor captures and analyzes all required signaling messages and media from the network, providing full correlation and quality metrics in real time.

The Enterprise Operations Monitor enables you to drill down into the captured data for troubleshooting and root-cause analysis of any reported problem related to a user, user group, trunk, network device, or Internet Protocol (IP) address. The Enterprise Operations Monitor Mediation Engine is the application that collects SIP, DNS, ENUM and protocol message traffic received from one or more EOM probes.

You can configure the Communications Broker to act as an EOM probe, or as an exporter, that can:

- Establish an authenticated, persistent, reliable TCP connection between itself and one or more Oracle Enterprise Operations Monitor Mediation Engines.
- Send UTC-timestamped, unencrypted copy of a protocol messages to the Oracle Enterprise Operations Monitor Mediation Engine.
- Accompany the copied message with related data to include the port or vlan on which the message was sent and received, the local and remote IP:port information, and the transport layer protocol.

Add a Monitor Collector

You can configure the probes embedded in the Oracle Enterprise Communications Broker to establish an IPFIX connection with one or more Oracle Enterprise Operations Monitor Mediation Engines (ME) to collect SIP, DNS, ENUM and protocol message traffic for the Enterprise Operations Monitor (EOM) to analyze.

You might want to connect the Communications Broker to multiple MEs, for example, to support monitoring continuity in the event of a service disruption.

- Configure at least one network interface.
- Obtain the IP address and port number of each target Oracle Enterprise Operations Monitor Mediation Engine that you want to connect.

In the following procedure, the Monitor Collector is the ME.

1. Access the System Config configuration object.
Configuration tab, **System Administration** section, **General Settings**, **System Config**.
2. On the Modify System Settings page, under Monitor Collector, click **Add**, and do the following:

Address	Set the IP address of the target ME. Default: 0.0.0.0.
Port	Set the port number on which the ME listens. Range: 1025-65535. Default: 4739
Network Interface	Select the local network interface from which to export traffic to the ME from the drop-down list. Default: wancom0:0.

3. Click **OK**.
4. (Optional) Repeat steps 2-3 for each additional monitor collector you want to connect to the Communications Broker.
5. Click **OK**.
6. Save the configuration.

Configure Communications Monitoring Probe Settings

Configuring Communications Monitoring Probe settings allows you to make the Oracle Enterprise Communications Broker act as a probe, sending network traffic information to an Oracle Communications Session Monitor Mediation Engine.

The Communications Session Monitor is Oracle's Communication Experience Manager. The manager is powered by the Oracle Communications Session Monitor Mediation Engine, a platform that collects SIP, DNS, ENUM, and protocol message traffic received from Oracle Communications Session Monitor Probes. The mediation engine stores the traffic in an internal database, and analyzes aggregated data to provide comprehensive multi-level monitoring, troubleshooting, and interoperability information.

Acting as a Probe, or as an exporter, the Communications Broker can:

- Establish an authenticated, persistent, reliable TCP connection between itself and the Oracle Communications Session Monitor Mediation Engines.
- Send UTC time-stamped, unencrypted copy of a protocol messages to the Mediation Engine.
- Accompany the copied message with related data to include: the port and VLAN on which the message was sent or received, local and remote IP:port information, and the transport layer protocol.

1. Access the System Config configuration object.

Configuration tab, **System Administration** section, **General Settings**, **System Config**.

2. Expand **Comm Monitor**.

State	Select to enable the probe.
SBC Grp ID	Set the <code>SBC group id</code> parameter to assign an integer value to the Communications Broker in its role as an information exporter. Default: 0.
Monitor Collector	<p>Click Add, and do the following:</p> <ol style="list-style-type: none"> a. Address—Enter the collector IP address to specify the IP address of the target Oracle Communications Session Monitor Mediation Engine. b. Port—Enter the collector port number of the target Oracle Communications Session Monitor Mediation Engine. Default: 4739. Range: 1025-65535. c. Network Interface—Select the network interface from which to export traffic to the Oracle Communications Session Monitor Mediation Engine. Most systems use M00:0. d. Click OK. e. Optional—Repeat to add another monitor collector.

3. Do one of the following:
 - Configure other settings on the Modify System Config page, and click **OK**.
 - Click **Back**.
4. Save the configuration.

Support for Multiple VLANs

Communications Broker allows you to configure up to four separate Virtual Local Area Networks (VLAN) to help manage your deployment. For example, you might want separate networks for certain departments or locations. Each VLAN connects to its own uniquely defined network interface and SIP interface, which allows you to create separate networks.

When you perform the initial configuration, the system creates a default network called "ecb" and one SIP interface with the Realm ID set to "ecb." The system also populates the Realm ID parameter in the Session Agent, LDAP, and ENUM configurations with "ecb."

 **Note:**

Only the default "ecb" network can act as the registrar, and LDAP support is also limited to the default "ecb" network. Additional networks cannot act as the registrar or support LDAP.

Add Multiple VLANs

To add additional network interfaces, go to the Networks configuration page and use the Service multi-instance configuration object. When you click **Add**, the system displays the Network settings page with the parameters necessary to configure a network. See "Configure a Network Interface" for instructions.

Note the following guidelines for adding VLANs:

- All VLANs use the S0P0 physical interface.
- Each VLAN must correspond to one, unique SIP interface
- The network Realm identifier, VLAN ID, and network IP address cannot repeat across networks

Accounting Settings

The Oracle Enterprise Communications Broker offers support for RADIUS, an accounting, authentication, and authorization (AAA) system. In general, RADIUS servers are responsible for receiving user connection requests, authenticating users, and returning all configuration information necessary for the client to deliver service to the user.

You can configure your Oracle Enterprise Communications Broker to send call accounting information to one or more RADIUS servers. This information can help you to see usage and QoS metrics, monitor traffic, and even troubleshoot your system.

Configure an Accounting Server

Use the following procedure to configure an accounting server to receive accounting detail from the Oracle Enterprise Communications Broker (Communications Broker). You can also edit and delete existing accounting servers with this procedure,.

The remote server to which the accounting configuration sends messages uses at least one of two pieces of information for purposes of identification. The Communications Broker

accounting messages always include the NAS IP address, while some may include the NAS ID:

- Network Access Server (NAS) IP address (the IP address of the Communications Broker SIP proxy).
- NAS ID. If you enter a value, the Communications Broker sends the NAS ID to the remote server.

If you have more than one Communications Broker pointing to the same accounting server, you can use the NAS ID to identify which Communications Broker generated the record.

1. Access the Accounting Configuration object.

Configuration, System Administration, Accounting, Accounting.

2. On the Accounting Configuration page, go to **Account Servers**, click **Add**, and do the following:

Hostname	Enter the name of the host associated with the account server in hostname format (FQDN) or as an IP address.
Port	Enter the number of the UDP port associated with the account server to which messages are sent. Default: 1813. Range: 1025-65535.
Secret	Enter the secret to pass from the account server to the client in text format.
NAS ID	(Optional) Enter the NAS ID in text format (FQDN allowed). The account server uses this value to identify the Communications Broker for the transmittal of accounting messages.

3. Click **OK**.
4. (Optional) Repeat steps 2-3 to add more accounting servers.
5. Save the configuration.

Configure Accounting

You can configure the Oracle Enterprise Communications Broker (Communications Broker) to perform accounting tasks and send the information to multiple FTP push servers and accounting servers. Accounting information can help you to see usage and QoS metrics, monitor traffic, and troubleshoot the system.

Set the accounting configuration parameters to indicate where and when you want the system to produce accounting messages. Specify one or more FTP push servers and accounting servers.

1. Access the Accounting Configuration object.

Configuration, System Administration, Accounting, Accounting.

2. On the Accounting Configuration page, do the following:

State	Select to enable the accounting configuration. Default: Disabled.
Generate Start	Specify how you want the Communications Broker to handle generating a CDR Start message. Default: OK. Valid values: <ul style="list-style-type: none"> • OK—Generate the Start message when the system receives an OK message in response to an INVITE.

	<ul style="list-style-type: none"> • None—Do not generate a Start message. • Invite—Generate a Start message when the system receives a SIP session INVITE.
Generate Interim	<p>Specify an interim message to indicate to the accounting server that the SIP session parameters changed. Default: Re-invite Response. Valid values:</p> <ul style="list-style-type: none"> • OK—Generate a Start message when the Communications Broker receives an OK message in response to an INVITE. • Re-invite—Cause the Communications Broker to transmit an interim message. • Reinvite—Generate an interim message when the Communications Broker receives a SIP session reINVITE message. • Re Invite Cancel—Generate a Start message when the Communications Broker receives a SIP session ReINVITE, and the ReINVITE is cancelled before the system responds. • Unsuccessful-Attempt—Generate an interim message when a SIP session set-up attempt from a preference-ordered list of next-hop destinations is unsuccessful. The interim message contains: the destination IP address, the disconnect reason, a timestamp for the failure, and the number that was called.
Intermediate Period	<p>Set the periodic recording interval in seconds. Default: 0. Range: 0-21447483647.</p>
File Output	<p>Select to enable the Communications Broker to generate local files containing accounting records. Default: Disabled.</p>
File Path	<p>Specify where you want the system to store accounting record files on the Communications Broker. (Do not use the /boot or /code file systems.) Default: /opt/logs/.</p>
File Rotate Time	<p>Set how often, in minutes, that you want to push the stored files. Default: 1440 (24 hours). Range: 0-21447483647. 0--means do not rotate the files. Note that the Communications Broker overwrites the oldest file first.</p>
FTP Push	<p>Select to enable the Communications Broker to push files to the FTP server. Default: Disabled.</p>
Push Receiver	<p>Click Add, and do the following:</p> <ol style="list-style-type: none"> 1. <ul style="list-style-type: none"> • Server—Enter the IP address of the FTP or SFTP server that you want to receiver the pushed files. Default: 0.0.0.0. • Port—Set the port for the server. Default: 21 Valid values: For FTP-21. For SFTP-22. • Admin state—Select to enable this server. Default: Enabled. • Remote path—Enter the path name for sending files to the push receiver. Default: Empty. Valid values: <string> prefix for filenames.

	<ul style="list-style-type: none"> • File name prefix—Enter the file name prefix to prepend to the files the Communications Broker sends to the push receiver. Default: Empty. Valid values: <string> prefix for filenames. • Priority—Set the priority for this push receiver. Default: 0. Range: 0-4. (0 is the highest priority.) • Protocol—Set the protocol you want the Communications Broker to use. Default: FTP. Valid values: FTP SFTP. • Username—Enter the user name you want the Communications Broker to use to connect to the push receiver. • Do one of the following: Password—Click Set to enter and confirm the password you want to use to access the push receiver. Public key—Select the public key profile that you want from the drop-down list. • Temp remote file—Select to use a temporary remote file name for the file transfer. <p>2. Click OK.</p> <p>3. (Optional) Repeat this procedure to add another push receiver.</p>
Accounting Servers	<p>To add one or more accounting servers, click Add and do the following:</p> <ol style="list-style-type: none"> 1. <ul style="list-style-type: none"> • Hostname—Enter the host name of the accounting server. • Port—Enter the port for the accounting server. Default: 1813. Range: 1025-65535. • Secret—Enter the secret for authentication. • NAS path—Enter the ID for the remote network accounting receiver. 2. Click OK. <p>The system adds the server to the list in the Accounting server dialog.</p> <p>3. (Optional) Repeat to add another accounting server.</p>

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

FTP Push

In addition to local and RADIUS server storage, the Oracle Enterprise Communications Broker (Communications Broker) can send accounting files to an FTP server. The information sent to the FTP server is the same as that stored locally.

Use FTP push to copy local CDR files to a remote FTP server on a periodic basis. You configure FTP push by defining push receivers with the login and FTP server credentials of the remote server. At the specified time interval (file rotate time), the Communications Broker closes the current file and pushes the files that are complete and have not yet been pushed, including the just-closed file to the FTP server.

Push receiver configurations must include:

- Enabling the FTP push server

- The server's IP address and port
- Remote path to the upload destination
- File name prefix
- Account login credentials

Multiple Push Receivers

Communications Broker (Communications Broker) supports up to five CDR push receivers for use with the local file storage and FTP push feature. For each receiver you configure, you can set the file transfer protocol that you want to use. (FTP or SFTP). The system uses the push receivers according to the priorities you assign by setting a 0 through 4 priority number to the server. 0 is the highest priority, and 4 (default) is the lowest.

Based on the priority level you set, the Communications Broker uses the strategy that you set to select a CDR push receiver. If the highest priority push receiver selected using the strategy becomes unavailable, the Communications Broker uses the strategy (hunt, round robin) to select another.

This feature is dynamically configurable. When you change the configuration, the Communications Broker updates the list of push receivers if it has changed.

Secure FTP Push Configuration

You can configure the Oracle Enterprise Communications Broker (Communications Broker) to securely log on to a push receiver using one of the following methods that creates a secure connection.

Password authentication—

1. Set the **protocol** parameter on the push receiver to SFTP.
2. Configure a username and password.
3. Leave the **public-key** parameter blank.
4. Import the host key from the SFTP server to the Communications Broker as a known-host key.
See "SSH Key Management" in the *Configuration Guide*.

Public key authentication—

1. Set the **protocol** parameter on the push receiver to SFTP.
2. Configure the username.
3. Leave the **public-key** parameter blank, regardless of authentication type.
4. Export the Communications Broker's public key with the `show security public-host-key rsa` command.
5. Append the Communications Broker's public-key to the SFTP server's `authorized_keys` file.
6. Import the host key from the SFTP server to the Communications Broker as a known-host key.
See "SSH Key Management" in the *Configuration Guide*.

It is often difficult to determine whether the SFTP server uses its RSA key or its DSA key for its server application. For this reason, Oracle recommends that you import both the RSA key and the DSA key to the Communications Broker to ensure a successful FTP Push.

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 rsa sftp.example.com
```

Add an FTP Push Receiver

The Oracle Enterprise Communications Broker (Communications Broker) supports configuring up to five FTP push servers to receive accounting files. Use the **Push Receiver Add** dialog located on the **account-config** page to access the parameters for creating a list of FTP push receivers.

- If you plan to use a public key for authentication to the push receiver, create public key profile. See "Configure Secure FTP Push with Public Key Authentication."
- Configure accounting. See "Configure Accounting."

To add an FTP push server to the accounting configuration,

1. Access the Accounting Configuration object.
Configuration, System Administration, Accounting, Accounting.
2. On the **Accounting config** page, do the following:

Table 3-1 Accounting Config

Field	Description
FTP push	Select to enable FTP push. Default: Disabled.

Table 3-1 (Cont.) Accounting Config

Field	Description
Push receiver	<p>Click Add, and do the following:</p> <ol style="list-style-type: none"> <ul style="list-style-type: none"> Server—Enter the IP address of the FTP or SFTP server that you want to receiver the pushed files. Default: 0.0.0.0. Port—Set the port for the server. Default: 21 Valid values: For FTP-21. For SFTP-22. Admin State—Select to enable this server. Default: Enabled. Remote Path—Enter the path name for sending files to the push receiver. Default: Empty. Valid values: <string> prefix for filenames. File Name Prefix—Enter the file name prefix to prepend to the files the Communications Broker sends to the push receiver. Default: Empty. Valid values: <string> prefix for filenames. Priority—Set the priority you want the Communications Broker to use when hunting for this push receiver versus your other ones. Default: 4. Range: 0-4. (0 is the highest priority.) Protocol—Set the protocol you want the Communications Broker to use. Default: FTP. Valid values: FTP SFTP. Username—Enter the user name you want the Communications Broker to use to connect to the push receiver. Do one of the following: <ul style="list-style-type: none"> Password—Click Set to enter and confirm the password you want to use to access the push receiver. Public Key—Select the public key profile that you want from the drop-down list. Temp Remote File—Select to use a temporary remote file name for the file transfer. Click OK. (Optional) Repeat this procedure to add another push receiver.

- Click **OK**.
- Save the configuration.

Network Interface Configuration

The network interface configuration specifies a logical network interface. The Communications Broker supports up to four Virtual Local Area Networks (VLAN). You configure a SIP interface and one or more application (SIP) ports over each network interface.

Configure a Network Interface

Set the following parameters to configure a network interface. The network Realm identifier, VLAN ID, and network IP address cannot repeat across networks. They must be unique for each network.

1. Access the Network configuration object. **Configuration, System Administration, Network, Network Settings.**
2. On the **Service** page, click **Add**, and do the following on the Network Settings page:

Name	Add a name for the Network Interface. This is to enable support of multiple Network Interfaces by Communications Broker.
Sub Port ID	Enter the identification of a specific virtual interface in a physical interface (For example, VLAN tag). A value of 0 indicates that this element is not using a virtual interface. The sub-port-id field value is only required if the operation type is Media. You can add a value from 0 to 4095.
Realm Identifier	Enter the name of this realm for access control.
VLAN ID	Enter the identification of a specific virtual interface in a physical interface, for example, a VLAN tab. If this network interface is not channelized, leave this field blank, and the value will correctly default to 0. The sub-port-id is only required if the operation type is Media. Default: 0. Range: 0-4095.
Hostname	Enter the fully qualified domain name.
Network IP Address	Enter the IPv4 address of this network interface.
Network IP Subnet Mask	Enter the net mask of this network interface in dotted decimal notation.
Network IP Gateway Address	Enter the gateway that this network interface uses to communicate with the next hop. You can set an additional, secondary gateway with the sec-gateway parameter.
Preferred DNS Server IP Address	Enter the IP address of the targeted DNS server.
Alternate DNS Server IP Address	Enter an alternate IP address for the targeted DNS server.
Alternate DNS Server IP Address	Enter an alternate IP address for the targeted DNS server.
DNS Domain	Enter the default domain name.
Enable REFER Termination	Select to terminate and process SIP REFER messages. Default: Disabled.
Send NOTIFY for REFER Provisional Responses	Select which NOTIFY messages for provisional responses you want the system to act on. Default: None. Valid values: None Initial All.
Enable ToS Marking	Select to ToS mark egress packets. Default: Disabled.

ToS Value	Set the ToS value to apply to egress packets. Default: 0x00.
HIP IP List	Enter all possible local IP address(es) to which a remote system can send administrative traffic. This parameter specifies addresses that can reply to requests. You must also configure them in a service list, such as ICMP Address, to specify the service to which they can reply. This parameter can accept multiple IP addresses.
ICMP Address	Enter all possible local IP address(es) to which a remote system can ping the Communications Broker and expect replies. You must also configure these addresses to the HIP IP List parameter.
Enable Gateway Heartbeat	For High Availability, check this checkbox to allow the network interface to continually confirm that its gateway is reachable.
High Availability Settings	Use the arrow control to display the HA parameters.
Primary Utility IP Address	Enter the utility IP address for the primary peer to use.
Secondary Utility IP Address	Enter the utility IP address for the secondary peer to use.
Interface Virtual MAC	Enter the virtual MAC address of the interface. (This address moves to which ever peer is active.)

3. Click **OK**.
4. (Optional) Repeat steps 2 and 3 to add another network interface (up to 4 total).
5. Save the configuration.

Enable ICMP

To configure Internet Control Message Protocol (ICMP) functionality on a media interface, define the IPv4 address on your Oracle Enterprise Communications Broker (Communications Broker) network interface and enable ICMP in Network Settings. Enabling ICMP entries automatically opens the well-known port associated with a service. For security ICMP is disabled buy default, so the Communications Broker discards ICMP requests or responses for the address. Oracle recommends that you enable ICMP only temporarily on a network interface.

Do the following to enable ICMP functionality on a network interface.

1. Access the Network configuration object. **Configuration, System Administration, Network, Network Settings**.
2. Select **Enable ICMP** to enable ICMP traffic on this network interface, so that the Communications Broker can respond to ICMP pings. Default: Disabled.
3. Save the configuration.

Configure the Network Interface for High Availability Operations

The Modify Network Settings dialog also includes the High Availability (HA) setting fields, which allow you to manually specify the addressing to be used by this interface for HA operation. Oracle recommends, that you use `run setup` to configure HA. These fields provide you with another way to configure this addressing.

1. Scroll through the The Modify Network Settings dialog to locate the addressing fields shown below.

Pri Utility Addr

Sec Utility Addr

2. Primary utility IP address—Enter the utility IPv4 address for the primary HA peer. This address can be any unused IPv4 address within the subnet defined for the network interface. For example, given a network interface with the IPv4 address 168.0.4.15/24 (identifying the host associated with the network interface), the possible range of unused IPv4 addresses is 168.0.4.1 to 168.0.4.254. Ask your network administrator which IPv4 addresses are available for use.
3. Secondary utility IP address—Enter the utility IPv4 address for the secondary Oracle Enterprise Communications Broker peer. Usually, this IPv4 address is the next in the sequence up from the primary utility address. It is also generated from the range of unused IPv4 addresses within the subnet defined for the network interface.

Virtual MAC Addresses

To create an HA node, you create virtual MAC addresses for the media interfaces. You enter these addresses in virtual MAC address parameters for physical interface configurations.

This field is automatically populated with a valid virtual MAC address during `run setup`. It is recommended that you retain this configuration.

The HA node uses shared virtual MAC (media access control) and virtual IP addresses for the interfaces. When there is a switchover, the standby Oracle Enterprise Communications Broker sends out an ARP message using the virtual MAC address, establishing that MAC on another physical port within the Ethernet switch.

A MAC address is a hardware address that uniquely identifies Oracle Enterprise Communications Broker components. Given that, the virtual MAC address you configure allows the HA node to appear as a single system from the perspective of other network devices. To the upstream router, the MAC and IP are still alive, meaning that existing sessions continue uninterrupted through the standby Oracle Enterprise Communications Broker.

To configure a virtual MAC, enter the virtual MAC address in the **Interface virtual MAC** field.

Configure a Realm

You can enable and configure a variety of constraints that the Oracle Enterprise Communications Broker (Communications Broker) applies to regulate session activity with the realm.

1. Access the Realm Config.
Click **Configuration, Network, Realm config**
2. On the **Realm Config** page, click **Add**, and do the following:

Table 3-2 Realm Config

Fields	Description
Identifier	Enter the name of the realm. This parameter uniquely identifies the realm. You use this parameter in other configurations when asked for a realm identifier value.
Network Interfaces	Enter the physical and network interface(s) that you want this realm to reference. These are the network interfaces through which this realm can be reached by ingress traffic, and through which this traffic exits the system as egress traffic. Enter the name and port in the correct format where the name of the interface comes first and is separated by a colon (:) from the port (VLAN) number. For example, s0p0:0. The parameters you set for the network interfaces must be unique.
Media Realm List	Not Used
Teams FQDN in URI	Not Used
SDP Inactive Only	Not Used
Access Control Trust Level	Indicate the trust level for the host with the realm. The default value is none . The valid values are: <ul style="list-style-type: none">• 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.

Table 3-2 (Cont.) Realm Config

Fields	Description
Invalid Signal Threshold	<p>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:</p> <ul style="list-style-type: none"> • Minimum—Zero (0) is disabled. • Maximum—999999999 <p>If the number of invalid messages exceeds this value based on the tolerance window parameter, configured in the media manager, the host is demoted.</p> <p>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.</p> <p>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.</p>
Maximum Signal Threshold	<ul style="list-style-type: none"> • Minimum—Zero (0) is disabled. • Maximum—999999999 <p>If the number of messages received exceeds this value within the tolerance window, the host is demoted.</p>
Untrusted Signal Threshold	<p>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:</p> <ul style="list-style-type: none"> • Minimum—Zero (0) is disabled. • Maximum—999999999
Deny Period	<p>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:</p> <ul style="list-style-type: none"> • Minimum—Zero (0) is disabled. • Maximum—999999999

Table 3-2 (Cont.) Realm Config

Fields	Description
Session Max Life Limit	Enter the maximum amount of time in seconds of a session. An additional, special case value of "Unlimited", is the highest possible value. <ul style="list-style-type: none"> • Minimum—0 (default) • Maximum—2073600 • Unlimited
Refer Call Transfer	Specifies whether and how to execute a REFER <ul style="list-style-type: none"> • 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	Provisional mode for sending Notify message. <ul style="list-style-type: none"> • none: No intermediate Notify messages are to be sent. • initial: Immediate 100 Trying Notify needs to be sent • all: Immediate 100 Trying NOTIFY and plus a NOTIFY for each non-100 provisional received by the SD are to be sent. • Default value: None.
Dyn Refer Term	Specify the behavior when a REFER is sent to this realm. <ul style="list-style-type: none"> • Disabled—Proxy the REFER (Default) • Enabled—Terminate and issue Re-INVITE
User Site	Not Used
SRVCC Trfo	Not Used

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

Configurable TCP Timers

You can configure your Oracle Enterprise Communications Broker to detect failed TCP connections more quickly so that data can be transmitted via an alternate connection before timers expire. Across all protocols, you can now control the following for TCP:

- Connection establishment
- Data retransmission
- Timer for idle connections

These capabilities all involve configuring an **options** parameter that appears in the network parameters configuration.

Configuring TCP Data Retransmission

TCP is considered reliable in part because it requires that entities receiving data must acknowledge transmitted segments. If data segments go unacknowledged, then they are retransmitted until they are finally acknowledged or until the maximum number of retries has been reached. You can control both the number of times the Oracle Enterprise

Communications Broker tries to retransmit unacknowledged segments and the periodic interval (how often) at which retransmissions occur.

You set two new options in the network parameters configuration to specify how many retransmissions are allowed and for how long: **atcp-rxmt-interval** and **atcp-rxmt-count**.

To configure TCP data retransmission:

1. Access the Network configuration object. **Configuration, System Administration, Network, Network Parameters.**
2. **Options**—Set the options parameter by typing the option name **atcp-rxmt-interval=x** (where x is a value in seconds between 2 and 60) into the Options field. This value will be used as the interval between retransmission of TCP data segments that have not been acknowledged.
3. **Options**—Now enter a second option to set the number of times the Oracle Enterprise Communications Broker will retransmit a data segment before it declares the connection failed. Set the options parameter by typing the option name **atcp-rxmt-count=x** (where x is a value between 4 and 12 representing how many retransmissions you want to enable) into the Options field.

```
atcp-rxmt-interval=30
atcp-rxmt-count=6
```

4. Save and activate your configuration.

Timer for Idle Connections

When enabled to do so, the Oracle Enterprise Communications Broker monitors inbound TCP connections for inactivity. These are inbound connections that the remote peer initiated, meaning that the remote peer sent the first SYN message. You can configure a timer that sets the maximum amount of idle time for a connection before the system consider the connection inactive. Once the timer expires and the connection is deemed inactive, the system sends a TCP RST message to the remote peer.

To configure the timer for TCP idle connections:

1. Access the Network configuration object. **Configuration, System Administration, Network, Network Parameters.**
2. **Options**—Set the options parameter by typing option name **atcp-idle-timer=x** (where x is a value in seconds between 120 and 7200) into the options field. This value will be used to measure the activity of TCP connections; when the inactivity on a TCP connection reaches this value in seconds, the system declares it inactive and drops the session.

```
atcp-idle-timer=120
```

3. Save and activate your configuration.

Configure the Network Parameters

Set the following parameters to configure network parameters that are common to all network interfaces.

1. Access the Network configuration object. **Configuration, System Administration, Network, Network Parameters.**
2. On the Service page, and do the following on the Network Parameters page:

TCP Keepinit Timer	<p>Enter the TCP connection timeout period if a TCP connection cannot be established. If you have upgraded the release you are running and a value outside of the acceptable range was configured in an earlier release, the default value is used and a log message is generated.</p> <ul style="list-style-type: none"> • Default: 75 • Values: 0-999999999
TCP Keepalive Count	<p>Enter the number of packets the system sends to the remote peer before it terminates the TCP connection.</p> <ul style="list-style-type: none"> • Default: 4 • Values: 0-223-1
TCP Keepalive Idle Timer	<p>Enter the idle time in seconds before triggering keepalive processing. If you have upgraded the release you are running and a value outside of the acceptable range was configured in an earlier release, the default value is used and a log message is generated.</p> <ul style="list-style-type: none"> • Default: 400 • Values: 30-7200
TCP Keepalive Interval Timer	<p>Enter the TCP retransmission time if a TCP connection probe has been idle for some amount of time.</p> <ul style="list-style-type: none"> • Default: 75 • Values: 15-75
TCP Keepalive Mode	<p>Enter the TCP keepalive mode.</p> <ul style="list-style-type: none"> • Default: 0 • Values: <ul style="list-style-type: none"> – 0—The sequence number is sent un-incremented. – 1—The sequence number is sent incremented. – 2—No packets are sent. – 3—Send RST (normal TCP operation).
Options	Enter any optional features or parameters.

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

DNS on the Communications Broker

DNS service is best known for providing resolution of internet domain names to IP addresses. Domain names are easy to remember, but connections require IP addresses. DNS deployments can also provide more comprehensive services, if required. For example, the a DNS client may need the resolution of multiple IP addresses to a single domain name, or the types of service provided by a given server. The Communications Broker uses DNS predominantly for resolving FQDNs to IP addresses so that it can support sessions.

When configured, the Communications Broker performs DNS client functions per RFC1034 and RFC1035. The user can define one primary DNS server and two backup DNS servers for the Communications Broker to query a domain for NAPTR (service/port), SRV (FQDN), AAAA (IPv6), and A (IP address) information. A common example of the Communications Broker using DNS is to locate a SIP server via server location discovery, as described in RFC 3263. An applicable context is identifying a callee so the Communications Broker can place a call.

There are multiple reasons for the Communications Broker to query a DNS server. In each case, the Communications Broker follows this high level procedure:

1. The system determines the egress realm.
2. The system identifies the egress network interface.
3. From the egress network interface, the system refers to the configured DNS server(s).
4. The system issues the DNS query to the primary server, then any configured backup servers, based on the function and the initial information it has.
5. The system performs recursive lookups or subsequent queries based on, for example, information provided in NAPTR resource responses, until it has one or more resolutions for the FQDN.
6. The system continues processing using the resolved FQDN(s) or indicates it cannot reach that FQDN.

**Note:**

DNS queries may require host routes.

DNS Functions on the Communications Broker

The Communications Broker Resolves session agent address when configured with FQDN. Routing to SIP Agents as static IP addresses is not scalable. Leveraging DNS-SRV allows dynamic routing and avoids routing to out of service agents. When configured, the Communications Broker can use RFC 3262 mechanism for balancing traffic to SIP servers. When the response to the DNS query for that hostname yields one or more IP addresses, the Communications Broker stores them in a cache to perform the role of the session agent. By obtaining multiple resolutions, the Communications Broker can then load balance signaling traffic.

The Communications Broker allows you to configure a session agent with an FQDN in the hostname field. When combined with an empty IP address field, the Communications Broker can initiate DNS queries that can result in multiple SRV or A records when triggered by SIP requests. For SRV requests, the Communications Broker uses the priority and weight in the DNS response to load balance. For A records, you configure the Communications Broker to use either the Hunt or Round-Robin strategy for load balancing at the applicable session agent. If there is a failure at the server, and you have enabled DNS load balancing, the Communications Broker queries the secondary server from the pool instead of recursing the request back to the Communications Broker.

The Communications Broker uses the target port number to determine what kind of query to perform, sending SRV requests if the value is 0, and A requests if it is nonzero. The Communications Broker can get this port value from either your agent configuration or from SRV responses from the DNS server.

The Communications Broker can report aggregate statistics of all IP targets that correspond to the session agent object and individual statistics per IP address (including per-method statistics) of each member that the FQDN query returns.

The Communications Broker refreshes its DNS cache when:

- The DNS result TTL expires and the new results are different or changed.
- The cached target associated with the UE is out of service.

Resolving A and SRV Records

The Communications Broker resolves **session-agent** FQDNs to either A or SRV records. The type of request is dependent on the port number used in the request. Port zero generates an SRV query and any other port number generates an A query.

You configure a **session-agent** with port number based on the Communications Broker

Load Balancing A and SRV Records

The Communications Broker achieves A record load balancing through hunt/round-robin strategy and SRV load balancing based on priority and weights received from DNS server response.

The Communications Broker monitors the availability of the dynamically resolved IP addresses obtained from DNS using OPTIONS pings (ping-per-DNS entry). The **ping-method** and **ping-interval** for each resolved IP address is copied from the original session-agent. This can be achieved by enabling **ping-all-addresses** in **session-agent** configuration. The default of **ping-all-addresses** is disabled, in which case the Communications Broker only pings the first available resolved IP addresses.

Load Balancing A Records

The Communications Broker performs server resolution for a registering UE starting with a DNS SRV query for a session agent with its port set to 0 and transport method given a setting other than **any** or *****. When it receives a UE'S initial registration message, the Communications Broker performs a DNS SRV query for the next hop target that handles the UE'S signaling. If the SRV target is an FQDN host that then resolves to multiple A RRs, the system then:

- If you have configured the **load-balance-dns-query** parameter to **hunt**, the Communications Broker selects a single A RR. The Communications Broker always selects the first of several A RRs unless that system goes out of service.
- If you have configured the **load-balance-dns-query** parameter to **round-robin**, the Communications Broker uses round-robin to select from multiple A RRs for a given SRV target.

By default, the system hunts for and selects a single A RR. If you have configured the **dns-load-balance** option, then **round-robin** does not work. This option distributes the requests to IP addresses that are resolved from SRV responses with the same weight and priority. Also, if you enable the session agent ping and set the **load-balance-dns-query** to **round-robin**, the Communications Broker pings all IP addresses resolved through the DNS query.

Note that when no session agent exists, the Communications Broker uses standard SRV ordering and A record hunting.

Load Balancing SRV Records

A Service record (SRV record) is a specification of data in the Domain Name System defining the location of servers for specified services as defined in RFC 2782. The SRV record includes contact information in addition to priority and weight fields.

- **Priority**—The priority field determines the precedence of use of the record's data. Client/Customer should use the SRV records with the lowest-numbered priority value first, and fall back to records of higher priority value if the connection fails.
- **Weight**—If a service has multiple SRV records with the same priority value, the Communications Broker load balances them in proportion to the values of their weight fields.

In the following example, both the priority and weight fields are used to provide a combination of load balancing and backup service.

```
_sip._tcp.provider.com. 86400 IN SRV 5 40 5060 SERVER.provider.com.  
_sip._tcp.provider.com. 86400 IN SRV 5 30 5060 server1.provider.com.  
_sip._tcp.provider.com. 86400 IN SRV 5 30 5060 server2.provider.com.  
_sip._tcp.provider.com. 86400 IN SRV 20 0 5060 backupserver.provider.com.
```

The first three records share a priority of 5, so the Communications Broker uses the weight field's value to determine which server (host and port combination) to contact. The sum of all three values is 100, so server.example.com will be used 40% of the time.

The Communications Broker uses server1 and server2 for 60% of requests, with half of them sent to server1, and the other half to server2. If SERVER becomes unavailable, the Communications Broker shares the load equally between server1 and server2.

If all three servers with priority 5 are unavailable, the Communications Broker chooses the record with the next lowest priority value, which in this case is backupserver.provider.com.

Load balancing provided by SRV records is inherently limited because the information is static. A server's current load is not taken into account unless TTL values are low enough (around a minute or lower) to allow fast updates to priority or weight values.

Retransmission Logic

The retransmission of DNS queries is controlled by three timers. These timers are derived from the configured DNS timeout value and from underlying logic that the minimum allowed retransmission interval should be 250 milliseconds; and that the Oracle Enterprise Communications Broker should retransmit 3 times before timing out to give the server a chance to respond.

- **Init-timer** is the initial retransmission interval. If a response to a query is not received within this interval, the query is retransmitted. To safeguard from performance degradation, the minimum value allowed for this timer is 250 milliseconds.
- **Max-timer** is the maximum retransmission interval. The interval is doubled after every retransmission. If the resulting retransmission interval is greater than the value of max-timer, it is set to the max-timer value.
- **Expire-timer**: is the query expiration timer. If a response is not received for a query and its retransmissions within this interval, the server will be considered non-responsive and the next server in the list will be tried.

The following examples show different timeout values and the corresponding timers derived from them.

```

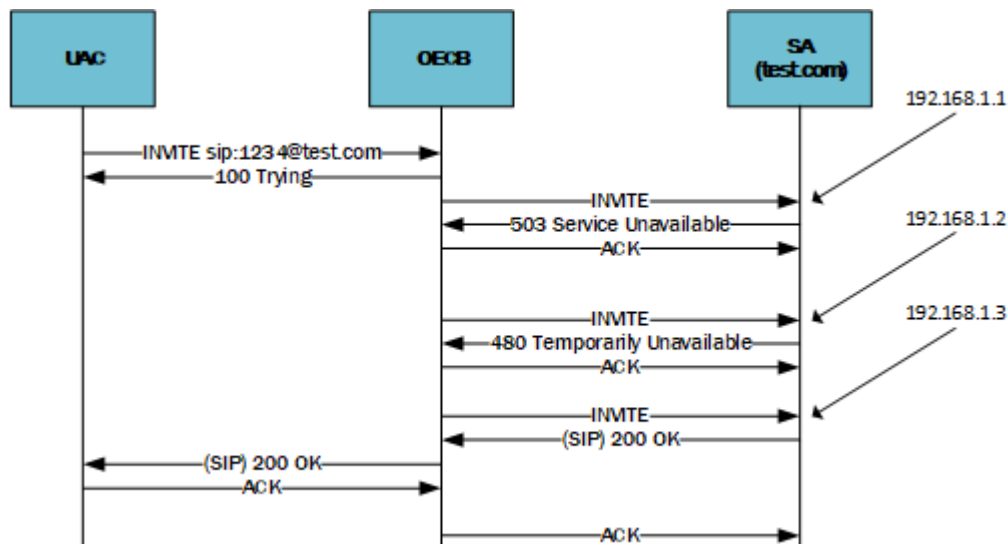
timeout >= 3 seconds
Init-timer = Timeout/11
Max-Timer = 4 * Init-timer
Expire-Timer = Timeout
timeout = 1 second
Init-Timer = 250 ms
Max-Timer = 250 ms
Expire-Timer = 1 sec
timeout = 2 seconds
Init-Timer = 250 ms
Max-Timer = 650 ms
Expire-Timer = 2sec
    
```

DNS-SRV Session Agent Recursion Error Handling

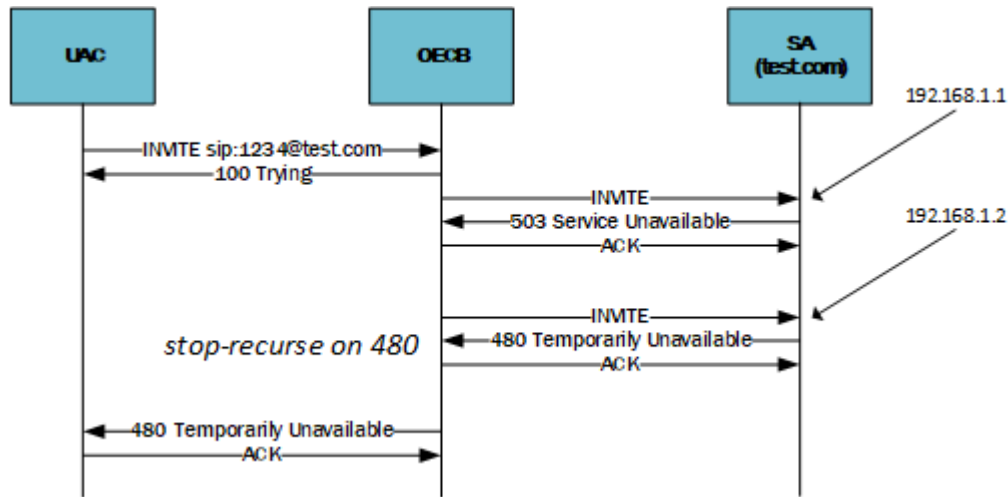
When a session request is sent from the Communications Broker to a session agent, and an error response is received (or a transport failure occurs), the Communications Broker attempts to reroute the message through the list of dynamically resolved IP addresses. The Communications Broker can be configured to resend session requests through the list of IP addresses under more failure conditions.

This feature concerns the case when a session agent is configured with an FQDN in the hostname parameter and the **dns-load-balance** or **ping-all-addresses** option is configured. This configuration sets up the load balancing / redundancy behavior for the SBC to use all addresses returned in the SRV/A-record for that session agent. In previous versions of the SBC software, only when a 503 failure from the SA was received would the SBC resend the session request to the next dynamically resolved IP address (on the SRV/A record list).

By adding the **recurse-on-all-failures** option to a session agent, the Communications Broker will resend a session request to the next address on the list after a 4xx or 5xx failure response has been received from a session agent.



If the Communications Broker receives a failure response from the session agent, and the number of that failure is configured in the **stop-recurse** parameter, no further session requests will be forwarded to additional addresses from the SRV/A record list. The error message will be forwarded back to the UA.



DNS Transaction Timeout

You can configure a DNS transaction timeout interval on a per **network-interface** basis using the **dns-timeout** parameter. As described, you configure the Communications Broker with a primary and two optional backup DNS servers. The Communications Broker queries the primary DNS server. If it does not receive a response within the configured number of seconds, it queries the backup1 DNS server. If that also times out, the Communications Broker contacts the backup2 DNS server.

DNS Server Status via SNMP

The Oracle Enterprise Communications Broker monitors the status of all configured DNS servers used by a SIP daemon. If a DNS server goes down, a major alarm is sent. If all DNS servers used by a SIP daemon are down, a critical alarm is sent. The `apAppsDnsServerStatusChangeTrap` is sent for both events.

You can poll the status of a DNS server using the `apAppsDNSServerStatusTable` in the `ap-apps.mib`.

Once the `apAppsDnsServerStatusChangeTrap` has been sent, a 30 second window elapses until the server status is checked again. At the 30 second timer expiration, if the server is still down, another trap and alarm are sent. If the server has been restored to service, the `apAppsDnsServerStatusChangeClearTrap` is sent.

Configure DNS on the Network Interface

DNS configuration includes procedures to the **network-interface** and **session-agent** elements.

To make DNS operational, configure addressing that is version compatible to the **network-interface** address on the network interface itself.

1. Access the Network configuration object. **Configuration, System Administration, Network, Network Settings.**
2. On the Service page, click **Add**, and do the following on the Network Settings page:

DNS IP Primary	Enter the IP address of the targeted DNS server.
DNS IP Backup1	Enter an alternate IP address for the targeted DNS server.

DNS IP Backup2	Enter an alternate IP address for the targeted DNS server.
DNS Domain	Enter the default domain name.
DNS Timeout	Enter the number of seconds to wait for a DNS response.
DNS Max TTL	Enter the maximum number of hops a query can take.

3. Save the configuration.

The system performs DNS query procedures with these servers every time processing encounters an FQDN for which the system needs resolution.

Review the ensuing sections and configure DNS components to refine DNS operation to your environment, including interface, realm, session agent, and ENUM operation refinement.

Security Settings

Security configuration from the GUI consists of creating the building blocks you can use to establish TLS-secured paths for your signaling traffic. The overall process includes generating certificate requests and certificate import.

The TLS configuration procedures that you can perform from the GUI includes:

- Configure Certificate Records.
- Generate Certificate Request for your CA.
- Import Certificates.
- Upload certificate files.
- Download certificate files.
- Configure TLS Profiles, which utilize your certificate records.
- Apply TLS Profiles to SIP Interfaces, agents and the web-server-config.

The dialogs available from the Security icon allow you to perform all procedures with the exception of applying a TLS profile to a configuration element. You apply TLS profiles to configuration elements using controls within their respective dialogs.

SHA 2 Support

The Oracle Enterprise Communications Broker (Communications Broker) supports Secure Hash Algorithm (SHA) 2 for improved security.

The Communications Broker supports SHA 2 for:

- Generating certificate requests, signing certificates, and verifying certificates.
- Configuring SHA-2 digital certificates on all interfaces through the dashboard, for example, the LDAP, SIP, and web/HTTPS: interfaces.
- Using the 2048 key size as the default for the signing algorithm.
- TLS 1.2 using the SHA-2 algorithm for certificates.

Add a Certificate Record

Use the certificate-record element to add certificate records to the Oracle Enterprise Communications Broker (Communications Broker).

A certificate record represents either the end-entity or the Certificate Authority (CA) certificate on the Communications Broker. When you configure a certificate for the Communications Broker, the name that you enter must be the same as the name that you use to generate a certificate request. If configuring for an end stations CA certificate for mutual authentication, the certificate name must be the same name used during the import procedure.

- If this certificate record is used to present an end-entity certificate, associate a private key with this certificate record by using a certificate request.
- If this certificate record is created to hold a CA certificate or certificate in pkcs12 format, a private key is not required.

1. Access the Certificates configuration object.

Configuration, Security, Certificates.

2. On the **Certificates** page, click **Add**.
3. On the **Add Certificates** page, do the following:

Table 3-3 Add Certificate

Field	Description
Name	Enter the name of the certificate record.
Country	Enter a two character country name abbreviation. For example, US for the United States.
State	Enter a two character state or province name abbreviation. For example, NE for Nebraska.
Locality	Enter the name of the locality in the state or province. For example, a city, a township, or a parish. Range: 1-128 characters.
Organization	Enter the name of the organization holding the certificate. For example, a company name. Range: 1-64 characters.
Unit	Name of the unit within the organization holding the certificate. For example, a business unit or a department. Range: 1-64 characters.
Common Name	Common name for the certificate record. For example, your name. Range: 1-64 characters.
Key Size	Size of the key for the certificate. Supported values: 512 1024 2048. Default: 2048.
Alternate Name	Alternate name of the certificate holder.
Trusted	Select to trust this certificate record.
Key Usage List	Select a key that you want to use with this certificate record from the drop-down list. This parameter defaults to the combination of digitalSignature and keyEncipherment. For a list of other valid values and their descriptions, see the section "Key Usage Control" in the <i>ACLI Configuration Guide</i> .

Table 3-3 (Cont.) Add Certificate

Field	Description
Extended Key Usage List	Add one or more extended keys that you want to use with this certificate record. This parameter defaults to serverAuth. For a list of other valid values and their descriptions, see the section “Key Usage Control” in the <i>ACLI Configuration Guide</i> .
Options	Optional Features or Parameters

4. Click **OK**.
 5. Save the configuration.
- Create TLS profiles, using the certificate records to further define the encryption behavior and to provide an entity that you can apply to a SIP interface.

TLS Profile Configuration

Certificate records must exist prior to this configuration.

Configure a TLS profile to further define the encryption behavior you want between these systems and to establish an entity that you can apply to SIP Interfaces.

1. Access the TLS Profile configuration object.
Click **Configuration, Security, TLS Profiles**.
2. On the **TLS profiles** page, click **Add**, and do the following:
3. Name—Enter the name of the TLS profile. This parameter is required.
4. End entity certificate—Enter the name of the Certificate Record for the applicable entity.
5. Trusted CA certificates—Enter the names of the trusted CA certificate records.
6. Cipher list—The following cipher-lists are supported for the GUI only:
 - AES256-SHA (TLS_RSA_WITH_AES_256_CBC_SHA) - Firefox (version 12) and Chrome (version 19.0.1084.46m)
 - AES128-SHA (TLS_RSA_WITH_AES_128_CBC_SHA) - Firefox (version 12) and Chrome (version 19.0.1084.46m)
 - DES-CBC-SHA (SSL_RSA_WITH_DES_CBC_SHA or TLS_RSA_WITH_DES_CBC_SHA) - Internet Explorer (Version 9)
7. Verify depth—Specify the maximum depth of the certificate chain that will be verified. Default: 10. Range: 0-10.
8. Mutual authenticate—Define whether or not you want the Oracle Enterprise Communications Broker to mutually authenticate the client. Default: disabled. Valid values: enabled | disabled.
9. TLS version—Enter the TLS version you want to use with this TLS profile. Default: compatibility. Valid values: TLSv1, SSLv3, and compatibility.
10. Ignore dead responder—Enables your device to establish a client connection when the OCSP responder is unavailable, assuming the associated certificate was signed by a trusted certificate authority. Default: disabled. Valid values: enabled | disabled.

11. Allow self signed cert—Enables your device to establish client connections to clients that present self-signed certificates. Default: disabled. Valid values: enabled | disabled.

Apply the TLS profile to a SIP Interface by selecting if from the SIP Interface TLS Profile drop-down list.

TLS Global Configuration

Add tls-global configuration details to configure global TLS parameters.

1. Access the TLS Global object by navigating to **Configuration, Security, TLS Global**.
2. On the **Add TLS Global** page, add details as described here:

Table 3-4 Fields in the Add TLS Global page

Field	Description
Session Caching	Enable or disable the Communications Broker's session caching capability. The default value is disabled.
Session Cache Timeout	Enter the session cache timeout in hours The default value is 12. Possible values are: Min: 0 (disabled)/ Max: 24.
Diffie Hellman Key Size	Enter the size of the Diffie-Hellman key offered by the Communications Broker when negotiating TLS on a SIP interface. <ul style="list-style-type: none"> • Default value: DH_KeySize_1024 • Possible values: DH_KeySize_1024 or DH_KeySize_2048 • Setting the key size to 2048 bits significantly decreases performance.

Generate a Certificate Request

Use the Certificate Record configuration object to select a certificate record and generate a certificate request.

- Confirm that the certificate record exists.

To get a certificate authorized by a Certificate Authority (CA), you must generate a certificate request from the certificate record on the device and send it to the CA.

1. Access the Certificates configuration object.

Configuration, Security, Certificates.

2. Click on the three dots next to the object.

3. Click **Generate**.

The system creates the request and displays it in a dialog.

4. Copy the information from the dialog and send it to your CA as a text file.
- When the CA replies with the certificate, import the certificate to the device with the corresponding certificate record.

Import a Certificate

Use the Certificate Record configuration object to import a certificate into the Oracle Enterprise Communications Broker (Communications Broker).

Use this procedure to import either a device certificate or an end-station CA certificate for a mutual authentication deployment. You must import the certificate to the corresponding certificate record for the Communications Broker. End-station CA certificates may or may not need to be imported against a pre-configured certificate record.

1. Access the Certificates configuration object.

Configuration, Security, Certificates.

2. Click on the three dots next to the object.
3. Click **Import**.

The system displays a dialog from which you can import the certificate.

4. Select one of the following format types from the **Format** drop down list:
 - pkcs7
 - x509
 - Try-all. The system tries all possible formats until it can import the certificate.
5. Select either **File** or **Paste** for the Import method.
6. Browse to the certificate file, and select the certificate to import, or paste it in the **Paste** field.
7. Click **Import**.

The Communications Broker imports the certificate.
8. Reboot the system.
 - Apply the corresponding certificate record to the intended SIP interface.

RADIUS Authentication

The User Authentication and Access control feature supports authentication using one or more RADIUS servers. In addition, you can set two levels of privilege, one for all privileges and more limited set that is read-only.

User authentication configuration also allows you to use local authentication, localizing security to the Oracle Enterprise Communications Broker (Communications Broker) log-in modes. These modes are User and Superuser, each requiring a separate password.

The components involved in the RADIUS-based user authentication architecture are the Communications Broker and your RADIUS servers. In these roles:

- The Communications Broker restricts access and requires authentication through the RADIUS server. The Communications Broker communicates with the RADIUS server using either port 1812 or 1645, but does not know whether or not the RADIUS server listens on these ports
- Your RADIUS server provides an alternative method for defining Communications Broker users and authenticating them through RADIUS. The RADIUS server supports the VSA called ACME_USER_CLASS, which specifies what kind of user is requesting authentication and what privileges to grant.

The Communications Broker also supports the use of the Cisco Systems Inc.™ Cisco-AVPair vendor specific attribute (VSA). This attribute allows for successful administrator login to servers that do not support the Oracle authorization VSA. While using RADIUS-based authentication, the Communications Broker authorizes you to enter Superuser mode locally even when your RADIUS server does not return the ACME_USER_CLASS VSA or the Cisco-AVPair VSA. For this VSA, the Vendor-ID is 1 and the Vendor-Type is 9. The following below shows the values this attribute can return, and the result of each:

- shell:priv-lvl=15—User automatically logged in as an administrator
- shell:priv-lvl=1—User logged in at the user level, and not allowed to become an administrator
- Any other value—User rejected

When RADIUS user authentication is enabled, the Communications Broker communicates with one or more configured RADIUS servers that validates the user and specifies privileges. On the Communications Broker, you configure:

- What type of authentication you want to use on the Communications Broker
- If you are using RADIUS authentication, you set the port from which you want the Communications Broker to send messages
- If you are using RADIUS authentication, you also set the protocol type you want the Communications Broker and RADIUS server to use for secure communication

Although most common deployments use two RADIUS servers to support this feature, you may configure up to six. Among other settings for the server, there is a class parameter that specifies whether the Communications Broker should consider a specific server as primary or secondary. As implied by these designations, the primary servers are used first for authentication, and the secondary servers are used as backups. If you configure more than one primary and one secondary server, the Communications Broker chooses servers to which it sends traffic in a round-robin strategy. For example, if you specify three servers are primary, the Communications Broker will round-robin to select a server until it finds an appropriate one. The system does the same for secondary servers.

The VSA attribute assists with enforcement of access levels by containing one of the following classes:

- None—All access denied
- User—Monitoring privileges are granted; your user prompt will resemble ORACLE>
- Admin—All privileges are granted (monitoring, configuration, etc.); your user prompt will resemble ORACLE#

After the system selects a RADIUS server, the Communications Broker initiates communication and proceeds with the authentication process. The authentication process between the Communications Broker and the RADIUS server takes place uses one the following methods, all of which are defined by RFCs:

Protocol	RFC
PAP (Password Authentication Protocol)	B. Lloyd and W. Simpson, PPP Authentication Protocols, RFC 1334, October 1992
CHAP (Challenge Handshake Authentication Protocol)	B. Lloyd and W. Simpson, PPP Authentication Protocols, RFC 1334, October 1992 W. Simpson, PPP Challenge Handshake Authentication Protocol (CHAP), RFC 1994, August 1996
MS-CHAP-V2	G. Zorn, Microsoft PPP CHAP Extensions, Version 2, RFC 2759, January 2000

**Note:**

MS-CHAP-V2 support includes authentication, only. The Communications Broker does not support or allow password exchange.

Management Protocol Behavior

When you use local authentication, management protocols behave the same way that they do when you are not using RADIUS servers. When you use RADIUS servers for authentication, management protocols behave as follows:

- SSH in pass-through mode—The User and Admin accounts are authenticated locally, not through the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication. When authentication is successful, the user is granted privileges depending on the ACME_USER_CLASS VSA attribute.
- SSH in non-pass-through mode—When you create an SSH account on the Oracle Enterprise Communications Broker (Communications Broker), you are asked to supply a user name and password. When local authentication succeeds, you are prompted for the ACLI user name and password. If your user ACLI name is user, then you are authenticated locally. Otherwise, you are authenticated using the RADIUS server. If RADIUS authentication is successful, the privileges you are granted depend on the ACME_USER_CLASS VSA attribute.
- SFTP in pass-through mode—When you do not configure an SSH account on the Oracle Enterprise Communications Broker, the RADIUS server is contacted for authentication for any user that does not have the user name user. The Oracle Enterprise Communications Broker uses local authentication if the user name is user.
- SFTP in non-pass-through mode—The User and Admin accounts are authenticated locally, not through the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication.

RADIUS Authentication Configuration

To enable RADIUS authentication and user access on your Oracle Enterprise Communications Broker, you need to configure global parameters for the feature and then configure the RADIUS servers that you want to use.

Global Authentication Settings

To configure the global authentication settings on the Oracle Enterprise Communications Broker (Communications Broker).

1. Access the Authentication Configuration object.
Configuration, Security, Authentication.
2. On the Modify Authentication page, do the following:

Source Port	Set the number of the port you want to use from message sent from the Communications Broker to the RADIUS server. Default: 1812. Valid values: 1645 1812
Type	Set the type of user authentication you want to use on this Communications Broker. Default: Local. Valid values: local radius.

Protocol	If you are using RADIUS user authentication, set the protocol to use with your RADIUS server(s) from the Protocol drop-down list. Default: pap. Valid values: ascii pap chap mschapv2.
Allow Local Authorization	Enable to authorize users to enter Superuser mode (administration) locally when the RADIUS server does not return the ACME_USER_CLASS VSA or the Cisco-AVPair VSA. Default: Disabled.
Login as Admin	Select if you want the Communications Broker to log users in automatically in Superuser (administrative) mode. Default: Disabled.

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

RADIUS Server Settings

The parameters you set for individual RADIUS servers identify the RADIUS server, establish a password common to the Oracle Enterprise Communications Broker (Communications Broker) and the server, and establish trying times.

The **Authentication Method** parameter has a specific relationship to the global protocol parameter for the authentication configuration. Exercise care when setting it. If the authentication method that you set for the RADIUS server does not match the global authentication protocol, then the RADIUS server is not used. The Communications Broker overlooks it and does not send authentication requests to it. You can enable use of the server by changing the global authentication protocol so that it matches.

To configure a RADIUS server to use for authentication:

1. Access the Authentication Configuration object.
Configuration, Security, Authentication.
2. In the Radius servers section, click **Add** and do the following:

Address	Set the remote IP address for the RADIUS server. Required. No default.
Port	Set the port at the remote IP address for the RADIUS server. Default: 1812. Valid values: 1645 1812.
State	Set the state of the RADIUS server in the State field. Enable this parameter to use this RADIUS server to authenticate users. Default: Enabled.
Secret	Set the password that the RADIUS server and the Communications Broker share to communicate when authentication is initialed. <ol style="list-style-type: none"> a. Click Set. b. Enter the secret. c. Confirm the secret. d. Click OK.
NAS ID	Set the NAS ID for the RADIUS server.
Realm ID	Set the realm to associate with this configuration.

Authentication Methods	Set the authentication method you want the Communications Broker to use with this RADIUS server. Default: pap. Valid values: all pap chap mschapv2.
------------------------	---

3. Save the configuration.

TACACS+ Overview

Like Diameter and Remote Authentication Dial-In User Service (RADIUS), Communications Broker uses a client-server model in which a Network Access Server (NAS) acts in the client role and a TACACS+ equipped device (a daemon in TACACS+ nomenclature) assumes the server role. For purposes of the current implementation, the Communications Broker functions as the TACACS+ client. Unlike RADIUS, which combines authentication and authorization, TACACS+ provides three distinct applications to provide finer grade access control.

Authentication is the process that confirms a user's purported identity. Authentication is most often based on a simple username/password association, but other, and more secure methods, are becoming more common. The following authentication methods are support by the current implementation: simple password, PAP (Protocol Authentication Protocol), and CHAP (Challenge Handshake Authentication Protocol).

Authorization is the process that confirms user privileges. TACACS+ can provide extremely precise control over access to system resources. In the current implementation, TACACS+ controls access to system administrative functions.

TACACS+ provides secure communication between the client and daemon by encrypting all packets. Encryption is based on a shared-secret, a string value known only to the client and daemon. Packets are encrypted in their entirety, save for a common TACACS+ header.

The cleartext header contains, among other fields, a version number, a sequence number, and a session ID. Using a methodology described in Section 5 of the TACACS+ draft RFC, the sender encrypts outbound cleartext messages by repetitively running the MD5 hash algorithm over the concatenation of the session ID, shared-secret, version number, and sequence number values, eventually deriving a virtual one-time-pad of the same length as the message body. The sender encrypts the cleartext message with an XOR (Exclusive OR) operation, using the cleartext message and virtual one-time-pad as inputs.

The message recipient, who possesses the shared-secret, can readily obtain the version number, sequence number, session ID, and message length from the cleartext header. Consequently, the recipient employs the same methodology to derive a virtual one-time-pad identical to that derived by the sender. The recipient decrypts the encrypted message with an XOR operation, using the encrypted message and virtual one-time-pad as inputs.

Details on the TACACS+ functions and configuration can be found in the *Oracle Communications Session Border Controller CLI Configuration Guide*.

The TACACS+ implementation is based upon the following internet draft.

draft-grant-tacacs-02.txt, *The TACACS+ Protocol Version 1.78*

Other relevant documents include

RFC 1321, *The MD-5 Message Digest Algorithm*

RFC 1334, *PPP Authentication Protocols* .

RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

**Note:**

TACACS documentation in this guide excludes per-message definitions that duplicate IETF standards documentation.

TACACS+ Authentication

The Oracle Enterprise Communications Broker (Communications Broker) uses Terminal Access Controller Access-Control System Plus (TACACS+) authentication services solely for the authentication of user accounts. Administrative users must be authenticated locally by the Communications Broker.

The current TACACS+ implementation supports three types of user authentication: simple password (referred to as ASCII by TACACS+), PAP, and CHAP.

ASCII Log In

ASCII login is analogous to logging into a standard PC. The initiating peer is prompted for a username, and, after responding, is then prompted for a password.

PAP Log In

Password Authentication Protocol (PAP) is defined in RFC 1334, *PPP Authentication Protocols*. PAP offers minimal security because passwords are transmitted as unprotected clear text. PAP log in differs from ASCII log in because the username and password are transmitted to the authenticating peer in a single authentication packet, as opposed to the two-step prompting process used in ASCII log in.

CHAP Log In

Challenge Handshake Authentication Protocol (CHAP) is defined in RFC 1994, *PPP Challenge Handshake Authentication Protocol*. CHAP is a more secure than Password Authentication Protocol (PAP) because it is based on a shared-secret (known only to the communicating peers), and therefore avoids the transmission of clear text authentication credentials. CHAP operations occur as follows.

1. After a login attempt, the authenticator tests the initiator by responding with a packet containing a challenge value — an octet stream with a recommended length of 16 octets or more.
2. Receiving the challenge, the initiator concatenates an 8-bit identifier (carried within the challenge packet header), the shared-secret, and the challenge value, and uses the shared-secret to compute an MD-5 hash over the concatenated string.
3. The initiator returns the hash value to the authenticator, who performs the same hash calculation, and compares results. If the hash values match, authentication succeeds. If hash values differ, authentication fails.

Authentication Message Exchange

All TACACS+ authentication packets consist of a common header and a message body. Authentication packets are of three types: START, CONTINUE, and REPLY.

START and CONTINUE packets are always sent by the Oracle Enterprise Communications Broker, the TACACS+ client. START packets initiate an authentication session, while CONTINUE packets provide authentication data requested by the TACACS+ daemon. In

response to every client-originated START or CONTINUE, the daemon must respond with a REPLY packet. The REPLY packet contains either a decision (pass or fail), which terminates the authentication session, or a request for additional information needed by the authenticator.

TACACS+ Header

The Terminal Access Controller Access-Control System Plus (TACACS+) header format is as follows.

```
+-----+-----+-----+-----+
|maj |min | type  | seq_no | flags |
|ver |ver |       |       |       |
+-----+-----+-----+-----+
| session_id                                     |
+-----+-----+-----+-----+
| length                                         |
+-----+-----+-----+-----+
```

maj ver

This 4-bit field identifies the TACACS+ major protocol version, and must contain a value of 0xC .

min ver

This 4-bit field identifies the TACACS+ minor protocol version, and must contain either a value of 0x0 (identifying TACACS+ minor version 0) or a value of 0x1 . (identifying TACACS+ minor version 1). Minor versions 0 and 1 differ only in the processing of PAP and CHAP logins.

type

This 8-bit field identifies the TACACS+ AAA service as follows:

0x1 — TACACS+ Authentication

0x2 — TACACS+ Authorization

0x3 — TACACS+ Accounting

sequence-no

This 8-bit field contains the packet sequence for the current session.

The first packet of a TACACS+ session must contain the value 1; each following packet increments the sequence count by 1. As TACACS+ sessions are always initiated by the client, all client-originated packets carry an odd sequence number, and all daemon-originated packets carry an even sequence number. TACACS+ protocol strictures do not allow the sequence_no field to wrap. If the sequence count reaches 255, the session must be stopped and restarted with a new sequence number of 1.

flags

This 8-bit field contains flags as described in Section 3 of the draft RFC; flags are not under user control.

session_id

This 32-bit field contains a random number that identifies the current TACACS+ session — it is used by clients and daemons to correlate TACACS+ requests and responses.

length

This 32-bit field contains the total length of the TACACS+ message, excluding the 12-octet header — in other words, the length of the message body.

Authentication START Packet

The Oracle Enterprise Communications Broker, acting as a TACACS+ client, sends an authentication START packet to the TACACS+ daemon to initiate an authentication session. The daemon must respond with a REPLY packet.

The authentication START packet format is as follows.

```

+-----+
|          Common Header          |
|          type contains 0x1      |
+-----+-----+-----+-----+
|action |priv_lvl|authen_ |service |
|        |        |type    |        |
+-----+-----+-----+-----+
|user_len|port_len|rem_addr|data_len|
|        |        |_len   |        |
+-----+-----+-----+-----+
|          user ...              |
+-----+-----+-----+-----+
|          port ...              |
+-----+-----+-----+-----+
|          rem-addr ...          |
+-----+-----+-----+-----+
|          data ...              |
+-----+-----+-----+-----+

```

action

This 8-bit field contains an enumerated value that identifies the requested authentication action. For the current TACACS+ implementation, this field always contains a value of 0x01 , indicating user login authentication.

priv_lvl

This 8-bit field contains an enumerated value that identifies the privilege level requested by an authenticating user. For the current TACACS+ authentication implementation, this field always contains a value of 0x01 , indicating the user level.

authen-type

This 8-bit field contains an enumerated value that identifies the authentication methodology. Supported values are as follows:

0x01 ASCII — simple login, Oracle Enterprise Communications Broker prompts for username and password

0x02 PAP — as specified in RFC 1334

0x03 CHAP — as specified in RFC 1994

service

This 8-bit field contains an enumerated value that identifies the service requesting the authentication. For the current TACACS+ implementation, this field always contains a value of 0x01 , indicating user login authentication.

user_len

This 8-bit field contains the length of the user field in octets.

port_len

This 8-bit field contains the length of the port field in octets. As the port field is not used in the current TACACS+ authentication implementation, the port_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

rem_addr_len

This 8-bit field contains the length of the rem_addr field in octets. As the rem_addr field is not used in the current TACACS+ authentication implementation, the rem_addr_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

data_len

This 8-bit field contains the length of the data field in octets.

user

This variable length field contains the login name of the user to be authenticated.

port

This variable length field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .

rem_addr

This variable length field contains the location of the user to be authenticated. This field contains the localhost address.

data

This optional variable length field contains miscellaneous data.

Authentication REPLY Packet

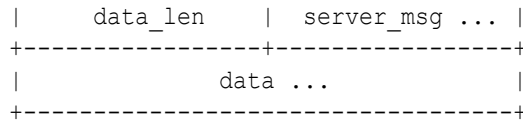
The TACACS+ daemon sends an authentication REPLY packet to the Oracle Enterprise Communications Broker in response to a authentication START or authentication CONTINUE packet. Depending on the contents of the status field, the authentication REPLY packet either ends the authentication transaction, or continues the transaction by requesting addition information needed by the authenticator.

The authentication REPLY packet format is as follows.

```

+-----+
|           Common Header           |
|           type contains 0x1       |
+-----+-----+-----+-----+
| (type field contains 0x1)        |
+-----+-----+-----+-----+
| status | flags | server_msg_len |
+-----+-----+-----+-----+

```

**status**

This 16-bit field contains an enumerated value that specifies the current state of the authentication process. Supported values are as follows:

0x01 PASS — the user is authenticated, thus ending the session

0x02 FAIL — the user is rejected, thus ending the session

0x04 GETUSER — daemon request for the user name

0x05 GETPASS — daemon request for the user password

0x06 RESTART — restarts the transaction, possibly because the sequence number has wrapped, or possibly because the requested authentication type is not supported by the daemon

0x07 ERROR — reports an unrecoverable error

flags

This 8-bit field contains various flags that are not under user control.

server_msg_len

This 16-bit field contains the length of the server_msg field in octets. As the server_msg field is not used in REPLY packets sent by the current TACACS+ authentication implementation, the server_msg_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

data_len

This 16-bit field contains the length of the data field in octets. As the data field is not used in REPLY packets sent by the current TACACS+ authentication implementation, the data_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

server_msg

This optional variable length field contains a server message intended for display to the user. The current TACACS+ authentication implementation does not use this field.

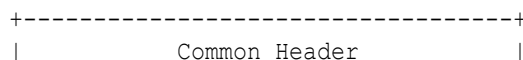
data

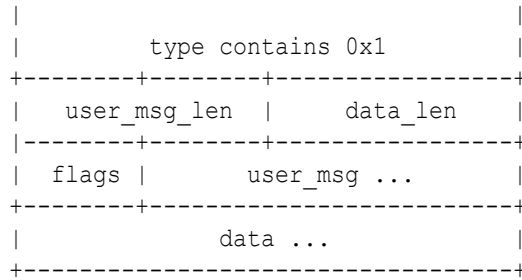
This optional variable length field contains data pertinent to the authentication process. The current TACACS+ authentication implementation does not use this field.

Authentication CONTINUE Packet

The Oracle Enterprise Communications Broker, acting as a TACACS+ client, sends an authentication CONTINUE packet to the TACACS+ daemon in response to a REPLY message which requested additional data required by the authenticator.

The authentication CONTINUE packet format is as follows.





user_msg_len

This 16-bit field contains the length of the user_msg field in octets.

data_len

This 16-bit field contains the length of the data field in octets. As the data field is not used in the current TACACS+ authentication implementation, the data field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

flags

This 8-bit field contains various flags that are not under user control.

user_msg

This variable length field contains a string that responds to an information request contained in a REPLY message.

data

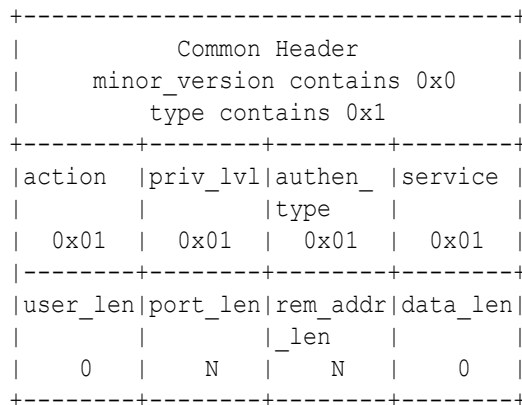
This optional variable length field contains miscellaneous data, often in response to a daemon request. The current TACACS+ authentication implementation does not use the data field in Authentication CONTINUE packets.

Authentication Scenarios

Each of the supported user authentication scenarios is described in terms of packet flow in the following sections.

ASCII Authentication

The Oracle Enterprise Communications Broker initiates the authentication with an authentication START packet.



```

|           port           |
|           tty10         |
+-----+
|           rem_addr      |
|           localhost address |
+-----+

```

- The action field specifies the requested authentication action — 0x01 for TAC_PLUSAUTHEN_LOGIN (authentication of a user login).
- The priv_lvl field specifies the privilege level requested by the user — 0x01 for TAC_PLUS_PRIV_LVL_USER.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len and data_len fields contain a value of 0 , as required by the TACACS+ protocol.
- The port_len and rem_addr_len fields contain the length, in octets, of the port and rem_addr fields.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.

The TACACS+ daemon returns an authentication REPLY requesting the username.

```

+-----+
|           Common Header           |
|           minor_version contains 0x0 |
|           type contains 0x1         |
+-----+
| status | flags | server_msg_len |
| 0x04  |      | 0              |
+-----+
|           data_len           |
|           0                  |
+-----+

```

- The status field specifies a daemon request — 0x04 for TAC_PLUS_AUTH_STATUS_GETUSER (get username).
- The server_msg_len data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

The Oracle Enterprise Communications Broker responds with an authentication CONTINUE packet.

```

+-----+
|           Common Header           |
|           minor_version contains 0x0 |
|           type contains 0x1         |
+-----+

```

```

|   user_msg_len   |   data_len   |
|                  |              |
|-----+-----+-----+
| flags |   user_msg ... |
|-----+-----+-----+

```

- The `user_msg_len` field contains the length, in octets, of the `user_msg` field.
- The `data_len` field contains a value of 0 , as required by the TACACS+ protocol.
- The `user_msg` field contains the username to be authenticated.

The TACACS+ daemon returns a second authentication REPLY requesting the user password.

```

+-----+
|           Common Header           |
|   minor_version contains 0x0       |
|           type contains 0x1       |
|-----+-----+-----+
| status | flags | server_msg_len |
| 0x05  |     |             0 |
|-----+-----+-----+
|   data_len   |
|             0 |
|-----+

```

- The `status` field specifies a daemon request — 0x05 for `TAC_PLUS_AUTH_STATUS_GETPASS` (get user password).
- The `server_msg_len` and `data_len` fields both contain a value of 0 , as required by the TACACS+ protocol.

The Oracle Enterprise Communications Broker responds with a second authentication CONTINUE packet.

```

+-----+
|           Common Header           |
|   minor_version contains 0x0       |
|           type contains 0x1       |
|-----+-----+-----+
| user_msg_len |   data_len   |
|              |             0 |
|-----+-----+-----+
| flags |   user_msg ... |
|-----+-----+-----+

```

- The `user_msg_len` field contains the length, in octets, of the `user_msg` field.
- The `data_len` field contains a value of 0 , as required by the TACACS+ protocol.
- The `user_msg` field contains the user password to be authenticated.
- Other, optional fields are not used.

The TACACS+ daemon returns a third authentication REPLY reporting the authentication result, and terminating the authentication session.

```

+-----+
|           Common Header           |

```

```

|   minor_version contains 0x0   |
|         type contains 0x1     |
+-----+-----+-----+
| status | flags | server_msg_len |
| 0x01  |      | 0              |
+-----+-----+-----+
|   data_len   |
|         0   |
+-----+

```

- The status field specifies the authentication result — 0x01 for TAC_PLUS_AUTH_STATUS_PASS (authorization succeeds), or 0x02 for TAC_PLUS_AUTH_STATUS_FAIL (authorization fails).
- The server_msg_len , and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

PAP Authentication

The Oracle Enterprise Communications Broker initiates the Password Authentication Protocol (PAP) authentication with an authentication START packet.

```

+-----+-----+-----+-----+
|           Common Header           |
|   minor_version contains 0x1     |
|         type contains 0x1       |
+-----+-----+-----+-----+
| action |priv_lvl|authen_ |service |
|        |        |type    |        |
| 0x01  | 0x01  | 0x02   | 0x01   |
+-----+-----+-----+-----+
| user_len|port_len|rem_addr|data_len|
|         |        |_len   |        |
|  N     |  N     |  N     |  N     |
+-----+-----+-----+-----+
|           user           |
+-----+-----+-----+-----+
|           port           |
|          tty10          |
+-----+-----+-----+-----+
|           rem_addr           |
|        localhost address   |
+-----+-----+-----+-----+
|           data ...           |
+-----+-----+-----+-----+

```

- Action—specifies the requested authentication action — 0x01 for TAC_PLUSAUTHEN_LOGIN (authentication of a user login).
- Priv_lvl—specifies the privilege level requested by the user — 0x01 for TAC_PLUS_PRIV_LVL_USER.
- Authen_type—specifies the authentication methodology — 0x02 for TAC_PLUS_AUTHEN_TYPE_PAP (PAP login).
- Service—specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).

- User_len—contains the length, in octets, of the user field.
- Port_len—contains the length, in octets, of the port field.
- Rem_addr_len—contains the length, in octets, of the rem_addr field.
- Data_len—contains the length, in octets, of the date field.
- User—contains the username to be authenticated.
- Port—contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- Rem_addr—specifies the location of the user to be authenticated. This field contains the localhost address.
- Data—Contains the password to be authenticated.

The TACACS+ daemon returns an authentication REPLY reporting the authentication result.

```

+-----+
|          Common Header          |
|  minor_version contains 0x1     |
|          type contains 0x1     |
+-----+-----+-----+-----+
| status | flags | server_msg_len |
| 0x01  |      |           0   |
+-----+-----+-----+-----+
|          data_len              |
|           0                    |
+-----+

```

- Status—specifies the authentication result — 0x01 for TAC_PLUS_AUTH_STATUS_PASS (authorization succeeds), or 0x02 for TAC_PLUS_AUTH_STATUS_FAIL (authorization fails).
- The server_msg_len and data_len—both contain a value of 0 , as required by the TACACS+ protocol.
- Other, optional fields are not used.

CHAP Authentication

The Oracle Enterprise Communications Broker initiates the Challenge Handshake Authentication Protocol (CHAP) with an authentication START packet.

```

+-----+
|          Common Header          |
|  minor_version contains 0x1     |
|          type contains 0x1     |
+-----+-----+-----+-----+
| action | priv_lvl | authen_ | service |
|        |         | type   |         |
| 0x01  | 0x01   | 0x03  | 0x01   |
+-----+-----+-----+-----+
| user_len | port_len | rem_addr | data_len |
|          |         | _len   |         |
|   N    |   N    |   N    |   N    |
+-----+-----+-----+-----+

```

```

|          user          |
+-----+
|          port         |
|          tty10        |
+-----+
|          rem_addr     |
|          localhost address |
+-----+
|          data ...     |
+-----+

```

- Action—specifies the requested authentication action — 0x01 for TAC_PLUSAUTHEN_LOGIN (authentication of a user login).
- Priv_lvl—specifies the privilege level requested by the user — 0x01 for TAC_PLUS_PRIV_LVL_USER.
- Authen_type—specifies the authentication methodology — 0x03 for TAC_PLUS_AUTHEN_TYPE_CHAP (CHAP login).
- Service—specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- User_len—contains the length, in octets, of the user field.
- Port_len—contains the length, in octets, of the port field.
- Rem_addr_len—contains the length, in octets, of the rem_addr field.
- Data_len—contains the length, in octets, of the date field.
- User—contains the username to be authenticated.
- Port—contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- Rem_addr—specifies the location of the user to be authenticated. This field contains the localhost address.
- Data—contains the password to be authenticated.

The TCACS+ daemon returns an authentication REPLY reporting the authentication result.

```

+-----+
|          Common Header          |
|          minor_version contains 0x1          |
|          type contains 0x1          |
+-----+
| status | flags | server_msg_len |
| 0x01  |      |          0      |
+-----+
|          data_len          |
|          0          |
+-----+

```

- Status—specifies the authentication result — 0x01 for TAC_PLUS_AUTH_STATUS_PASS (authorization succeeds), or 0x02 for TAC_PLUS_AUTH_STATUS_FAIL (authorization fails).

- Server_msg_len and data_len—both contain a value of 0 , as required by the TACACS+ protocol.
- Other, optional fields are not used.

TACACS+ Authorization

The Oracle Enterprise Communications Broker uses Terminal Access Controller Access-Control System Plus (TACACS+) services to provide administrative authorization. With TACACS+ authorization enabled, each individual CLI command issued by an admin user is authorized by the TACACS+ authorization service. The Oracle Enterprise Communications Broker replicates each CLI command in its entirety, sends the command string to the authorization service, and suspends command execution until it receives an authorization response. If TACACS+ grants authorization, the pending command is executed; if authorization is not granted, the Oracle Enterprise Communications Broker does not execute the CLI command, and displays an appropriate error message.

The daemon's authorization decisions are based on a database lookup. Data base records use regular expressions to associate specific command string with specific users. The construction of such records is beyond the scope of this document.

Authorization Message Exchange

All Terminal Access Controller Access-Control System Plus (TACACS+) authorization packets consist of a common header and a message body. Authorization packets are of two types: REQUEST and RESPONSE.

The REQUEST packet, which initiates an authorization session, is always sent by the Oracle Enterprise Communications Broker. Upon receipt of every REQUEST, the daemon must answer with a RESPONSE packet. In the current TACACS+ implementation, the RESPONSE packet must contain an authorization decision (pass or fail). The exchange of a single REQUEST and the corresponding RESPONSE completes the authorization session.

Authorization REQUEST Packet

The Oracle Enterprise Communications Broker, acting as a Terminal Access Controller Access-Control System Plus (TACACS+) client, sends an authorization REQUEST packet to the TACACS+ daemon to initiate an authorization session.

The authorization REQUEST packet format is as follows.

```

+-----+
|           Common Header           |
|                                   |
|           type contains 0x2       |
+-----+-----+-----+-----+
|authen_ |priv_lvl|authen_ |authen- |
|method  |         |type     |service |
+-----+-----+-----+-----+
|user_len|port_len|rem_addr|arg_cnt |
|         |         |_len    |         |
+-----+-----+-----+-----+
|arg1_len|arg2_len| ...   |argN_len|
|         |         |         |         |
+-----+-----+-----+-----+
|           user ...                 |
+-----+

```

```

|          port ...          |
+-----+
|          rem-addr ...     |
+-----+
|          arg1 ...         |
+-----+
|          arg2 ...         |
+-----+
|          argN ...         |
+-----+

```

authen_method

This 8-bit field contains an enumerated value that identifies the method used to authenticate the authorization subject — that is, an admin user. Because the admin user was authenticated locally by the Oracle Enterprise Communications Broker, this field always contains a value of 0x05 , indicating authentication by the requesting client.

priv_lvl

This 8-bit field contains an enumerated value that identifies the privilege level associated with the authorization subject. For the current TACACS+ authorization implementation, this field always contains a value of 0x00 .

authen-type

This 8-bit field contains an enumerated value that identifies the methodology. used to authenticate the authorization subject. Because the admin user was authenticated with a simple username/password exchange, this field always contains a value of 0x01 , indicating ascii login.

authen_service

This 8-bit field contains an enumerated value that identifies the service that requested authentication. Because an admin user is authenticated with a simple username/password exchange, this field always contains a value of 0x01 , the login service.

user_len

This 8-bit field contains an integer that specifies the length, in octets, of the user field.

port_len

This 8-bit field contains an integer that specifies the length, in octets, of the port field.

rem_addr_len

This 8-bit field contains an integer that specifies the length, in octets, of the rem_addr field.

arg_cnt

This 8-bit field contains an integer that specifies the number of arguments contained with the REQUEST. Given the design of the current TACACS+ implementation, this field always contains a value of 0x02 .

arg1_len

This 8-bit field contains an integer that specifies the length, in octets, of the first argument.

Subsequent fields contain the length of each sequential argument.

user

This variable length field contains the login name of the user to be authorized.

port

This variable length field contains the name of the Oracle Enterprise Communications Broker port on which authorization is taking place. Following Cisco Systems convention, this field contains the string `tty10`.

rem_addr

This variable length contains the location of the user to be authorized. This field contains the localhost address.

arg...

This variable length field contains a TACACS+ attribute value pair (AVP); each arg field holds a single AVP.

A TACACS+ AVP is an ASCII string with a maximum length of 255 octets. The string consists of the attribute name and its assigned value separated by either an equal sign (=) or by an asterisk (*). The equal sign (=) identifies a mandatory argument, one that must be understood and processed by the TACACS+ daemon; the asterisk (*) identifies an optional argument that may be disregarded by either the client or daemon.

Administrative authorization requires the use of only two TACACS+ AVPs: `service` and `cmd`.

The `service` AVP identifies the function to be authorized. In the case of the current implementation, the attribute value is always `shell`. Consequently the attribute takes the follow format:

```
service=shell
```

The `cmd` AVP identifies the specific ACLI command to be authorized. The command is passed in its entirety, from the administrative configuration root, **configure terminal**, through the final command argument. For example,

```
cmd=configure terminal security authentication type tacacsplus
```

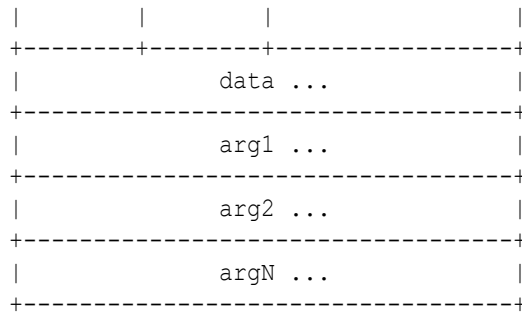
Note the equal sign (=) used in the attribute examples, indicating that both are mandatory arguments.

Authorization RESPONSE Packet

The Terminal Access Controller Access-Control System Plus (TACACS+) daemon sends an authorization RESPONSE packet to the Oracle Enterprise Communications Broker to report authorization results.

The authorization RESPONSE packet format is as follows.

```
+-----+
|           Common Header           |
|           type contains 0x2       |
+-----+-----+-----+-----+
|status |arg_cnt | server_msg len |
|       |      |      |      |
+-----+-----+-----+-----+
|   data_len   |arg1_len|arg2_len|
|              |      |      |
+-----+-----+-----+-----+
|   ... |argN_len|   server_msg   |
```



status

This 8-bit field contains an enumerated value that specifies the results of the authorization process. Supported values are 0x01 (Pass), 0x10 (Fail), and 0x11 (Error). Fail indicates that the authorization service rejected the proposed operation, while Error indicates the authorization service failed

If authorization succeeds (status=0x01), the ACLI command is executed; if authorization fails, for whatever the reason (status=0x10 or 0x11), the ACLI command is not executed, and an appropriate error message is generated.

arg_cnt

This 8-bit field contains an integer that specifies the number of arguments contained with the RESPONSE. Given the design of the current TACACS+ implementation, this field always contains a value of 0x02 .

server_msg_len

This 16-bit field contains an integer that specifies the length, in octets, of the server_msg field.

data_len

This 16-bit field contains an integer that specifies the length, in octets, of the data field.

arg1_len

This 8-bit field contains an integer that specifies the length, in octets, of the first argument.

Subsequent fields contain the length of each sequential argument.

server-msg

This optional variable length field contains a string that can be presented to the user.

data

This optional variable length field contains a string that can be presented to an administrative display, console, or log.

arg...

This optional variable length field contains a TACACS+ attribute value pair (AVP); each arg field holds a single AVP.

No arguments are generated in RESPONSE packets within the current TACACS+ implementation.

Authorization Pass

The Oracle Enterprise Communications Broker initiates the authorization with an authorization REQUEST packet.

```

+-----+
|          Common Header          |
|          |          |          |
|          type contains 0x2      |
+-----+-----+-----+-----+
|authen_ |priv_lvl|authen_ |authen_ |
|method  |        |type    |service |
| 0x05   | 0x00  | 0x01   | 0x01   |
+-----+-----+-----+-----+
|user_len|port_len|rem_addr|arg_cnt |
|         |        |_len   |        |
|  N     |  N    |  N    |  2    |
+-----+-----+-----+-----+
|arg1_len|arg2_len|      user ... |
|         |        |      login name |
+-----+-----+-----+-----+
|          port          |
|          tty10         |
+-----+-----+-----+-----+
|          rem_addr     |
|          localhost address |
+-----+-----+-----+-----+
|          arg1         |
|          AVP          |
|          service=shell |
+-----+-----+-----+-----+
|          arg2         |
|          AVP          |
| cmd=configure terminal security |
+-----+-----+-----+-----+

```

- The `authen_method` field specifies the method used to authenticate the subject — 0x05 for `TAC_PLUS_AUTHEN_METHOD_LOCAL` (authentication by the client).
- The `priv_lvl` field specifies the privilege level requested by the user — 0x00 for `TAC_PLUS_PRIV_LVL_MIN`.
- The `authen_type` field specifies the authentication methodology — 0x01 for `TAC_PLUS_AUTHEN_TYPE_ASCII` (simple login).
- The `authen_service` field specifies the requesting service — 0x01 for `TAC_PLUS_AUTHEN_SVC_LOGIN` (login service).
- The `user_len` field contains the length, in octets, of the user field.
- The `port_len` field contains the length, in octets, of the port field.
- The `rem_addr_len` field contains the length, in octets, of the `rem_addr` field.
- The `arg_cnt` field contains the number of arguments in the message body.
- The `arg1_len` field contains the length, in octets, of the service AVP.

- The `arg2_len` field contains the length, in octets, of the service AVP.
- The `user` field contains the login name of an admin user.
- The `port` field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string `tty10`.
- The `rem_addr` field specifies the location of the user to be authenticated. This field contains the localhost address.
- The `arg1` field contains the mandatory service AVP.
- The `arg2` field contains the mandatory cmd AVP.

The TACACS+ daemon returns a authorization RESPONSE reporting the status, and terminating the authorization session.

```

+-----+
|           Common Header           |
|                                   |
|           type contains 0x2       |
+-----+-----+-----+-----+
| status |arg_cnt | server_msg_len |
| 0x01  |  0     |         0     |
+-----+-----+-----+-----+
|           data_len               |
|           0                       |
+-----+

```

- The `status` field specifies the authorization status — `0x01` for `TAC_PLUS_AUTHOR_STATUS_PASS_ADD` (authorization approved).
- The `arg_cnt` field contains a value of `0` — the authorization RESPONSE returns no arguments.
- The `server_msg_len` and `data_len` fields both contain a value of `0`, as required by the TACACS+ protocol.

Authorization Fail

The Oracle Enterprise Communications Broker initiates the authorization with an authorization REQUEST packet.

```

+-----+
|           Common Header           |
|                                   |
|           type contains 0x2       |
+-----+-----+-----+-----+
| authen_ |priv_lvl| authen_ | authen_ |
| method  |        | type    | service |
| 0x05    | 0x00  | 0x01   | 0x01   |
+-----+-----+-----+-----+
| user_len|port_len|rem_addr|arg_cnt |
|         |        |_len   |        |
|  N     |  N    |  N    |  2    |
+-----+-----+-----+-----+
| arg1_len|arg2_len|         user ... |
|         |        |                 |
+-----+

```

```

|   N   |   N   |   login name   |
+-----+-----+-----+
|           | port   |                 |
|           | tty10  |                 |
+-----+-----+-----+
|           | rem_addr |                 |
|           | localhost address |                 |
+-----+-----+-----+
|           | arg1   |                 |
|           | AVP    |                 |
|           | service=shell |                 |
+-----+-----+-----+
|           | arg2   |                 |
|           | AVP    |                 |
|           | cmd=configure terminal security |                 |
+-----+-----+-----+

```

- The `authen_method` field specifies the method used to authenticate the administrative subject — 0x05 for `TAC_PLUS_AUTHEN_METHOD_LOCAL` (authentication by the client).
- The `priv_lvl` field specifies the privilege level requested by the user — 0x00 for `TAC_PLUS_PRIV_LVL_MIN`.
- The `authen_type` field specifies the authentication methodology — 0x01 for `TAC_PLUS_AUTHEN_TYPE_ASCII` (simple login).
- The `authen_service` field specifies the requesting service — 0x01 for `TAC_PLUS_AUTHEN_SVC_LOGIN` (login service).
- The `user_len` field contains the length, in octets, of the user field.
- The `port_len` field contains the length, in octets, of the port field.
- The `rem_addr_len` field contains the length, in octets, of the rem-addr field.
- The `arg_cnt` field contains the number of arguments in the message body.
- The `arg1_len` field contains the length, in octets, of the service AVP.
- The `arg2_len` field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string `tty10`.
- The `rem_addr` field specifies the location of the user to be authenticated. This field contains the localhost address.
- The `arg1` field contains the mandatory service AVP.
- The `arg2` field contains the mandatory cmd AVP.

The TACACS+ daemon returns an authorization RESPONSE reporting the status, and terminating the authorization session.

```

+-----+-----+-----+
|           | Common Header |                 |
|           |               |                 |
|           | type contains 0x2 |                 |
+-----+-----+-----+
| status | arg_cnt | server_msg_len |

```

```

| 0x10 | 0 | 0 |
|-----+-----+
| data_len |
| 0 |
|-----+

```

- The status field specifies the authorization status — 0x10 for TAC_PLUS_AUTHOR_STATUS_FAIL (authorization rejected).
- The arg_cnt field contains a value of 0 — the authorization RESPONSE returns no arguments.
- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

TACACS+ Accounting

The Oracle Enterprise Communications Broker uses Terminal Access Controller Access-Control System Plus (TACACS+) accounting to log administrative actions. With accounting enabled, each individual ACLI command executed by an admin user is logged by the accounting service.

Accounting Message Exchange

All Terminal Access Controller Access-Control System Plus (TACACS+) accounting packets consist of a common header and a message body. Accounting packets are of two types: REQUEST and REPLY.

The REQUEST packet has three variant forms. The START variant initiates an accounting session; the STOP variant terminates an accounting session; the WATCHDOG variant updates the current accounting session. REQUEST packets are always sent by the Oracle Enterprise Communications Broker (Communications Broker). Upon receipt of every REQUEST, the daemon must answer with a REPLY packet.

A TACACS+ accounting session proceeds as follows.

1. Immediately following successful authorization of an admin user, the Communications Broker sends an accounting REQUEST START packet.
2. The daemon responds with an accounting REPLY packet, indicating that accounting has started.
3. For each ACLI command executed by an admin user, the Communications Broker sends an accounting REQUEST WATCHDOG packet requesting accounting of the ACLI command. As the Communications Broker sends the WATCHDOG only after an admin user's access to the ACLI command is authorized, the accounting function records only those commands executed by the user, not those commands for which authorization was not granted.
4. The daemon responds with an accounting REPLY packet, indicating that the ACLI operation has been recorded by the accounting function.
5. Steps 3 and 4 are repeated for each authorized ACLI operation.
6. Immediately following logout (or timeout) of an admin user, the Communications Broker sends an accounting REQUEST STOP packet.
7. The daemon responds with an accounting REPLY packet, indicating that accounting stopped.

Accounting REQUEST Packet

The Oracle Enterprise Communications Broker (Communications Broker), acting as a Terminal Access Controller Access-Control System Plus (TACACS+) client, sends an accounting REQUEST START variant to the TACACS+ daemon following the successful authorization of an admin user. It sends an accounting REQUEST WATCHDOG variant to the daemon following the authorization of an admin user's access to an ACLI command. It sends an accounting REQUEST STOP variant to the daemon at the conclusion of the ACLI session.

The accounting REQUEST packet format is as follows.

```

+-----+
|           Common Header           |
|           type contains 0x3       |
+-----+-----+-----+-----+
| flags |authen_|priv_lvl|authen-|
|       |method |        |type   |
+-----+-----+-----+-----+
|authen_|user_len|port_len|rem_addr|
|service|        |        |_len   |
+-----+-----+-----+-----+
|arg_cnt|arg1_len|arg2_len|argN_len|
|        |        |        |        |
+-----+-----+-----+-----+
|argN_len|        user ... |
+-----+-----+-----+-----+
|        port ...   |
+-----+-----+-----+-----+
|        rem-addr ...|
+-----+-----+-----+-----+
|        arg1 ...   |
+-----+-----+-----+-----+
|        arg2 ...   |
+-----+-----+-----+-----+
|        argN ...   |
+-----+-----+-----+-----+

```

flags

This 8-bit field contains an enumerated value that identifies the accounting REQUEST variant.

0x2 — START

0x4 — STOP

0x8 — WATCHDOG

authen_method

This 8-bit field contains an enumerated value that identifies the method used to authenticate the accounting subject — that is, an admin user. Because an admin user is authenticated locally by the Oracle Enterprise Communications Broker, this field always contains a value of 0x05, indicating authentication by the requesting client.

priv_lvl

This 8-bit field contains an enumerated value that identifies the privilege level associated with the accounting subject. For the current TACACS+ accounting implementation, this field always contains a value of 0x00 .

authen-type

This 8-bit field contains an enumerated value that identifies the methodology. used to authenticate the accounting subject. Because an admin user is authenticated with a simple username/password exchange, this field always contains a value of 0x01 , indicating ascii login.

authen_service

This 8-bit field contains an enumerated value that identifies the service that requested authentication. Because an admin user is authenticated with a simple username/password exchange, this field always contains a value of 0x01 , the login service.

user_len

This 8-bit field contains an integer that specifies the length, in octets, of the user field.

port_len

This 8-bit field contains an integer that specifies the length, in octets, of the port field.

rem_addr_len

This 8-bit field contains an integer that specifies the length, in octets, of the rem_addr field.

arg_cnt

This 8-bit field contains an integer that specifies the number or arguments contained with the accounting REQUEST.

arg1_len

This 8-bit field contains an integer that specifies the length, in octets, of the first argument.

Subsequent fields contain the length of each sequential argument.

user

This variable length field contains the login name of the accounting subject.

port

This variable length field contains the name of the Oracle Enterprise Communications Broker port on accounting is taking place. Following Cisco System convention, this field always contains the string tty10 .

rem_addr

This variable length contains the location of the authorization subject. This field always contains the localhost address.

arg...

This variable length field contains a TACACS+ attribute value pair (AVP); each arg field holds a single AVP.

A TACACS+ AVP is an ASCII string with a maximum length of 255 octets. The string consists of the attribute name and its assigned value separated by either an equal sign (=) or by an asterisk (*). The equal sign (=) identifies a mandatory argument, one that must be understood

and processed by the TACACS+ daemon; the asterisk (*) identifies an optional argument that may be disregarded by either the client or daemon.

Administrative accounting requires the use of five TACACS+ AVPs: `service`, `task-id`, `start_time`, and `stop_time`.

The `task_id` AVP, included in accounting REQUEST START, STOP, and WATCHDOG variants, correlates session initiation, watchdog updates, and termination packets; each associated START, STOP, and WATCHDOG packet must contain matching task-id AVPs.

```
task_id=13578642
```

The `start_time` AVP, included in accounting REQUEST START and WATCHDOG variants, specifies the time at which a specific accounting request was initiated. The start time is expressed as the number of seconds elapsed since January 1, 1970 00:00:00 UTC.

```
start_time=1286790650
```

The `stop_time` AVP, included in accounting REQUEST STOP variants, specifies the time at which a specific accounting session was terminated. The stop time is expressed as the number of seconds elapsed since January 1, 1970 00:00:00 UTC.

```
stop_time=1286794250
```

The `service` AVP, included in accounting REQUEST START, STOP, and WATCHDOG variants, identifies the function subject to accounting. In the case of the current implementation, the attribute value is always `shell`. Consequently the attribute takes the follow format:

```
service=shell
```

The `cmd` AVP, included in accounting REQUEST WATCHDOG variants, identifies the specific ACLI command to be processed by the accounting service. The command is passed in its entirety, from the administrative configuration root, **configure terminal**, through the final command argument. For example,

```
cmd=configure terminal security authentication type tacacsplus
```

Note the equal sign (=) used in the attribute examples, indicating that all are mandatory arguments.

Accounting REPLY Packet

The Terminal Access Controller Access-Control System Plus (TACACS+) daemon sends an accounting REPLY packet to the Oracle Enterprise Communications Broker to report accounting results.

The accounting REPLY packet format is as follows.

```
+-----+
|           Common Header           |
|                                     |
|           type contains 0x3        |
+-----+-----+-----+
| server_msg_len | data_len |
+-----+-----+-----+
| status | server_msg ... |
+-----+-----+-----+
|           data ...           |
+-----+
```

server_msg_len

This 16-bit field contains the length, in octets, of the server_msg field.

data_len

This 16-bit field contains the length, in octets, of the data field.

status

This 8-bit field contains the status of the previous accounting request. Supported values are:

0x1 — Success

0x2 — Error/Failure

server_msg

This optional variable length field can contain a message intended for display to the user. This field is unused in the current TACACS+ implementation.

data

This optional variable length field can contain miscellaneous data. This field is unused in the current TACACS+ implementation.

Accounting Scenario

The Oracle Enterprise Communications Broker initiates the accounting session with an accounting REQUEST START.

```

+-----+
|           Common Header           |
|           type contains 0x3       |
+-----+-----+-----+-----+
| flags | authen_ | priv_lvl | authen- |
|       | method  |         | type    |
| 0x02 | 0x05   | 0x00   | 0x01   |
+-----+-----+-----+-----+
| authen_ | user_len | port_len | rem_addr |
| service |         |         | _len    |
| 0X01   | N      | N      | N      |
+-----+-----+-----+-----+
| arg_cnt | arg1_len | arg2_len | arg3_len |
| 3      | N      | N      | N      |
+-----+-----+-----+-----+
|           user                     |
| login name of an admin user       |
+-----+-----+-----+-----+
|           port                     |
|           tty10                    |
+-----+-----+-----+-----+
|           rem_addr                 |
|           localhost address        |
+-----+-----+-----+-----+
|           AVP                      |
|           task-id=13578642         |
+-----+-----+-----+-----+
|           AVP                      |

```

```

|          start_time=1286790650          |
+-----+
|                   AVP                   |
|          service=shell                   |
+-----+

```

- The flags field contains an enumerated value (0x02) that identifies an accounting REQUEST START.
- The authen_method field specifies the method used to authenticate the ACCOUNTING subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the task_id AVP.
- The arg2_len field contains the length, in octets, of the start_time AVP.
- The arg3_len field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory task_id AVP.
- The arg2 field contains the mandatory start_time AVP.
- The arg3 field contains the mandatory service AVP.

The TACACS+ daemon returns an accounting REPLY reporting the status, indicating that accounting has started.

```

+-----+
|          Common Header          |
|          |
|          type contains 0x3      |
+-----+
| server_msg_len | data_len |
|          0      |          0      |
+-----+
| status |
| 0x01  |
+-----+

```

- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- The status field specifies the authorization status — 0x01 for TAC_PLUS_ACCT_STATUS_SUCCESS (accounting processed).

The Oracle Enterprise Communications Broker reports ACLI command execution with an accounting REQUEST WATCHDOG.

```

+-----+
|           Common Header           |
|                                     |
|           type contains 0x3       |
+-----+-----+-----+-----+
| flags | authen_ | priv_lvl | authen- |
|       | method  |         | type    |
| 0x08 | 0x05   | 0x00   | 0x01   |
+-----+-----+-----+-----+
| authen_ | user_len | port_len | rem_addr |
| service |         |         | _len    |
| 0X01   | N       | N       | N       |
+-----+-----+-----+-----+
| arg_cnt | arg1_len | arg2_len | arg3_len |
| 4      | N       | N       | N       |
+-----+-----+-----+-----+
| arg4_len |         user         |
|         | login name of admin user |
+-----+-----+-----+-----+
|         port         |
|         tty10        |
+-----+-----+-----+-----+
|         rem_addr     |
|         localhost address |
+-----+-----+-----+-----+
|         AVP          |
|         task-id=13578642 |
+-----+-----+-----+-----+
|         AVP          |
|         start_time=1286790650 |
+-----+-----+-----+-----+
|         AVP          |
|         service=shell |
+-----+-----+-----+-----+
|         AVP          |
|         cmd=configure terminal security |
+-----+-----+-----+-----+

```

- The flags field contains an enumerated value (0x08) that identifies an accounting REQUEST WATCHDOG.
- The authen_method field specifies the method used to authenticate the ACCOUNTING subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).

- The `authen_service` field specifies the requesting service — 0x01 for `TAC_PLUS_AUTHEN_SVC_LOGIN` (login service).
- The `user_len` field contains the length, in octets, of the user field.
- The `port_len` field contains the length, in octets, of the port field.
- The `rem_addr_len` field contains the length, in octets, of the `rem_addr` field.
- The `arg_cnt` field contains the number of arguments in the message body.
- The `arg1_len` field contains the length, in octets, of the `task_id` AVP.
- The `arg2_len` field contains the length, in octets, of the `start_time` AVP.
- The `arg3_len` field contains the length, in octets, of the service AVP.
- The `arg4_len` field contains the length, in octets, of the `cmd` AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string `tty10`.
- The `rem_addr` field specifies the location of the user to be authenticated. This field contains the localhost address.
- The `arg1` field contains the mandatory `task_id` AVP.
- The `arg2` field contains the mandatory `start_time` AVP.
- The `arg3` field contains the mandatory service AVP.
- The `arg4` field contains the mandatory `cmd` AVP.

The TACACS+ daemon returns an accounting REPLY reporting the status, indicating that the ACLI operation has been processed.

```

+-----+
|           Common Header           |
|                                     |
|           type contains 0x3       |
+-----+-----+
| server_msg_len | data_len |
|           0   |         0   |
+-----+-----+
| status |
| 0x01  |
+-----+

```

- The `server_msg_len` and `data_len` fields both contain a value of 0, as required by the TACACS+ protocol.
- The status field specifies the authorization status — 0x01 for `TAC_PLUS_ACCT_STATUS_SUCCESS` (accounting processed).

The Oracle Enterprise Communications Broker reports an admin user logout or timeout with an accounting REQUEST STOP.

```

+-----+
|           Common Header           |
|                                     |
|           type contains 0x3       |

```

```

+-----+-----+-----+-----+
| flags  | authen_ | priv_lvl | authen- |
|        | method  |         | type    |
| 0x04  | 0x05   | 0x00    | 0x01   |
+-----+-----+-----+-----+
| authen_ | user_len | port_len | rem_addr |
| service |         |         | _len    |
| 0X01   | N      | N      | N      |
+-----+-----+-----+-----+
| arg_cnt | arg1_len | arg2_len | arg3_len |
| 3      | N      | N      | N      |
+-----+-----+-----+-----+
|                user                |
| login name of an admin user        |
+-----+-----+-----+-----+
|                port                |
|                tty10                |
+-----+-----+-----+-----+
|                rem_addr              |
|                localhost address     |
+-----+-----+-----+-----+
|                AVP                   |
|                task-id=13578642      |
+-----+-----+-----+-----+
|                AVP                   |
|                stop_time=1286790650  |
+-----+-----+-----+-----+
|                AVP                   |
|                service=shell         |
+-----+-----+-----+-----+

```

- The flags field contains an enumerated value (0x04) that identifies an accounting REQUEST STOP.
- The authen_method field specifies the method used to authenticate the ACCOUNTING subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the task_id AVP.
- The arg2_len field contains the length, in octets, of the start_time AVP.
- The arg3_len field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.

- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory task_id AVP.
- The arg2 field contains the mandatory start_time AVP.
- The arg3 field contains the mandatory service AVP.

The TACACS+ daemon returns an accounting REPLY reporting the status, indicating that accounting has terminated.

```

+-----+
|           Common Header           |
|                                   |
|           type contains 0x3       |
+-----+-----+
| server_msg_len | data_len |
|         0      |         0   |
+-----+-----+
| status |
| 0x01  |
+-----+

```

- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- The status field specifies the authorization status — 0x01 for TAC_PLUS_ACCT_STATUS_SUCCESS (accounting processed).

Managing TACACS+ Operations

Terminal Access Controller Access-Control System Plus (TACACS+) management is supported by the following utilities.

TACACS+ MIB

An Oracle proprietary MIB provides external access to Terminal Access Controller Access-Control System Plus (TACACS+) statistics.

MIB counters are contained in the apSecurityTacacsPlusStatsTable that is defined as follows.

```

SEQUENCE {
    apSecurityTacacsPlusCliCommands           Counter32
    apSecurityTacacsPlusSuccess Authentications Counter32
    apSecurityTacacsPlusFailureAuthentications Counter32
    apSecurityTacacsPlusSuccess Authorizations Counter32
    apSecurityTacacsPlusFailureAuthorizations Counter32
}

```

apSecurityTacacsPlusStats Table (1.3.6.1.4.1.9148.3.9.9.4)

Object Name	Object OID	Description
apSecurityTacacsCliCommands	1.3.6.1.4.1.9148.3.9.1.4.3	Global counter for ACLI commands sent to TACACS+ Accounting
apSecurityTacacsSuccess Authentications	1.3.6.1.4.1.9148.3.9.1.4.4	Global counter for the number of successful TACACS+ authentications
apSecurityTacacsFailure Authentications	1.3.6.1.4.1.9148.3.9.1.4.5	Global counter for the number of unsuccessful TACACS+ authentications
apSecurityTacacsSuccess Authorizations	1.3.6.1.4.1.9148.3.9.1.4.6	Global counter for the number of successful TACACS+ authorizations
apSecurityTacacsFailure Authorizations	1.3.6.1.4.1.9148.3.9.1.4.7	Global counter for the number of unsuccessful TACACS+ authorizations

SNMP Trap

SNMP traps are issued when

- a Terminal Access Controller Access-Control System Plus (TACACS+) daemon becomes unreachable
- an unreachable TACACS+ daemon becomes reachable
- an authentication error occurs
- an authorization error occurs

TACACS+ Faults

The Oracle Enterprise Communications Broker (Communications Broker) supports (TACACS+) traps to notify you of operational status. Traps from the apSysMgmt tree include:

- apSysMgmtTacacsDownTrap (1.3.6.1.4.1.9148.3.2.6.0.78) - Generated when a TACACS+ server becomes unreachable.
- apSysMgmtTacacsDownClearTrap (1.3.6.1.4.1.9148.3.2.6.0.79) - Generated when a TACACS+ server that was unreachable becomes reachable.

The Communications Broker searches for a TACACS+ server until it finds an available one and then stops searching. However, in the TACACS+ SNMP implementation, SNMP expects the Communications Broker to make connection attempts to all servers.

- When there is only one TACACS+ server and that server goes down, the Communications Broker behaves normally, sending a apSysMgmtTacacsDownTrap trap when the server goes down, and a apSysMgmtTacacsDownClearTrap trap when the server comes back up.
- When there is more than one TACACS+ server and the active server goes down, an apSysMgmtTacacsDownTrap trap is sent, indicating that some servers are down and the next server is tried.
 - If all servers fail, an apSysMgmtTacacsDownTrap is sent indicating that all servers are down.
 - If one of the servers comes back up while the rest are still down, an apSysMgmtTacacsDownTrap is sent indicating that some servers are still down.

Traps from the apSecurity tree include:

- apSecurityTacacsFailureNotification (1.3.6.1.4.1.9148.3.9.3.1.0.4) - Generated when the system detects TACACS daemon reachability changes as well as TACACS authentication and authorization errors.
- apSecurityTacacsDownLocalAuthUsedTrap (1.3.6.1.4.1.9148.3.9.3.9.0.1) - Generated when a user remotely logs into a system configured for TACACS+ authentication and is authenticated locally by the system because all of the configured and enabled TACACS+ servers have become unreachable or unresponsive
- apSecurityTacacsDownLocalAuthUsedClearTrap (1.3.6.1.4.1.9148.3.9.3.9.0.2) - Generated when a user remotely logs into a system configured for TACACS+ authentication and is successfully authenticated (i.e., access accepted or denied) remotely by a configured and enabled TACACS+ server.

TACACS+ Logging

All messages between the Oracle Enterprise Communications Broker and the Terminal Access Controller Access-Control System Plus (TACACS+) daemon are logged in a clear text format, allowing an admin user to view all data exchange, except for password information.

TACACS+ Configuration

Configuration of Terminal Access Controller Access-Control System Plus (TACACS+) consists of the following steps.

1. Enable TACACS+ client services
2. Specify one or more TACACS+ servers (daemons)

Add TACACS+ Authentication and Servers

To configure Terminal Access Controller Access-Control System Plus (TACACS+), you enable TACACS+ client services and specify one or more TACACS+ servers.

1. Access the Login Config configuration object: **Configuration, Security, Authentication.**
2. On the Authentication page, do the following:

Source Port	Default: 1812. Range: 1645-1812.
Type	Select TACACS from the drop-down list.
Protocol	Select the authentication protocol. PAP is the default.
Tacacs Authentication Only	<p>Select whether to limit authentication to TACACS only.</p> <ul style="list-style-type: none"> • Enable - If a TACACS+ server is configured and available, then authentication shall use ONLY TACACS+ and the system rejects any attempts to authenticate using other methods. <ul style="list-style-type: none"> – If a TACACS+ server is configured and available, then authentication uses ONLY TACACS+ and the Communications Broker rejects any attempts to authenticate from other methods. – If a TACACS+ server is configured but unavailable, the Communications Broker shall allow authentication using other methods.

	<ul style="list-style-type: none"> Disable (default) - The system supports authentication using all configured methods.
Tacacs Accounting	Select to enable accounting of admin operations. Default: Enabled.
Server Assigned Privilege	Select to allow only Admin users to use configuration commands. Default: Disabled.
Allow Local Authentication	Select to enable local authentication. Default: Disabled.
Login as Admin	Select to enable logging in as Admin.
Management Strategy	<p>Select an authentication management strategy from the drop-down list.</p> <ul style="list-style-type: none"> Use either Hunt or Round-Robin when using multiple TACACS+ servers. Use Hunt when using a single TACACS+ server. <p>Default: Hunt.</p>
Management Servers	Enter the IP address of a management server.
TACACS Servers	<p>Click Add, and do the following:</p> <ol style="list-style-type: none"> Address—Enter the IP address of this server. Port—Enter the port number of the server you want to receive TACACS+ client requests. Default: 49. Range: 1025-65535. State—Select to enable this server. Default: Enabled. Secret—Enter and confirm the 16-digit string for the shared secret used by the TACACS+ client and the server to encrypt and decrypt TACACS+ messages. Dead Time—Enter the time, in seconds, for the quarantine period imposed upon a TACACS+ server that becomes unreachable. Default: 10. Range: 10-10000 seconds. Authentication Methods—Add one or more authentication methods. Default: All.

- Click **OK**.
- Save the configuration.

SIP Interface Settings

A SIP Interface is an application layer interface logically residing "over" a network interface. The SIP interface defines the transport addresses (IP address and port) upon which the Oracle Enterprise Communications Broker receives and sends SIP messages. You can define a SIP interface for each network to which the Oracle Enterprise Communications Broker is connected. Note that these networks must be within the Oracle Enterprise Communications Broker's Network Interface subnet. SIP interfaces support UDP, TCP and TLS transport.

In addition to defining a SIP interface's network participation (**Port**), you can also define forking and other functionality (**Interface settings**).

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.

Global SIP Timers

This section explains how to configure SIP retransmission and expiration timers.



Note:

you can also set timers and counters per SIP interface.

Overview

SIP timers define the transaction expiration timers, retransmission intervals when UDP is used as a transport, and the lifetime of dynamic TCP connections. The retransmission and expiration timers correspond to the timers defined in RFC 3261.

- **init timer:** is the initial request retransmission interval. It corresponds to Timer T1 in RFC 3261.
This timer is used when sending requests over UDP. If the response is not received within this interval, the request is retransmitted. The retransmission interval is doubled after each retransmission.
- **max timer:** is the maximum retransmission interval for non-INVITE requests. It corresponds to Timer T2 in RFC 3261.
The retransmission interval is doubled after each retransmission. If the resulting retransmission interval exceeds the max timer, it is set to the max timer value.
- **trans expire:** is the transaction expiration timer. This value is used for timers B, D, F, H and J as defined in RFC 3261.
- **invite expire:** defines the transaction expiration time for an INVITE transaction after a provisional response has been received. This corresponds to timer C in RFC 3261.
If a final response is not received within this time, the INVITE is cancelled. In accordance with RFC 3261, the timer is reset to the invite expire value when any additional provisional responses are received.
- **Inactive dynamic conn timer** defines the idle time of a dynamic TCP connection before the connection is torn down. Idle is defined as not transporting any traffic. There is no timer in RFC 3261 corresponding to this function.

SIP Timers Discreet Configuration

Previous releases controlled various SIP timers with a single setting, **Trans Expire**, available in both SIP Config and SIP Interface modes. When executed in SIP Config mode, the

command essentially established a global default transaction expiration timer value. Executed at the SIP Interface level, the command established a local, interface-specific value that overrides the global default.

Specific timers controlled by **Trans Expire** are as follows:

- Timer B, the INVITE transaction timeout timer, defined in Section 17.1.1.2 and Appendix A of RFC 3261, *SIP: Session Initiation Protocol*.
- Timer D, the Wait-Time for response retransmits timer, defined in Section 17.1.1.2 and Appendix A of RFC 3261, *SIP: Session Initiation Protocol*.
- Timer F, the non-INVITE transaction timeout timer, defined in Section 17.1.2.2 and Appendix A of RFC 3261, *SIP: Session Initiation Protocol*.
- Timer H, the Wait-Time for ACK receipt timer, defined in Section 17.2.1 and Appendix A of RFC 3261, *SIP: Session Initiation Protocol*.
- Timer J, the Wait-Time for non-INVITE requests timer, defined in Section 17.2.2 and Appendix A of RFC 3261, *SIP: Session Initiation Protocol*.

The **Initial Inv Trans Expire** parameter enables user control over SIP Timer B for initial INVITE transactions. Other timers, namely B for non-initial INVITES, D, F, H, and J remain under the control of **Trans Expire**.

Use **Initial Inv Trans Expire** in the SIP Config to establish a global, default transaction timeout value (expressed in seconds) used exclusively for initial INVITE transactions.

Allowable values are integers within the range 0 (the default) through 999999999. The default value, 0, indicates that a dedicated INVITE Timer B is not enabled. Non-default integer values enable a dedicated Timer B and set the timer value.

The default value retains compatibility with previous operational behavior in that Timers B, D, F, H, and J all remain subject to the single timer value set by **Trans Expire**. However, when **Initial Inv Trans Expire** is set to a supported non-zero value, SIP Timer B as it applies to initial INVITES, assumes that value rather than the value assigned by **Trans Expire**. This functionality is available in both SIP Config and in SIP Interface objects.

If a dedicated Timer B is set in the SIP Config, you can use **initial-inv-trans-expire** in the SIP Interface to establish a local interface-specific Timer B timeout value that overrides the global default value.

Timer to Tear Down Long Duration Calls

In most call scenarios, long duration calls are terminated with the expiration of this timer, but there are some cases where a call can stay connected for a longer duration. For example, if a user connects to an IVR service and does not hang up the phone receiver properly, there is no way for the network provider to free up the IVR resources if the user devices send session updating requests. To prevent this situation, set the **Session Max Life Limit** timer in the SIP Config, which starts when the call or session is established and does not reset for any session update, keep-alive or system switchover. On expiry, the call is torn down if it's in established state.

The **Session Max Life Limit** parameter can be provisioned in the following configuration elements, in order of precedence from highest to lowest: **Session Agent**, **Realm Config**, **SIP Interface**, and **SIP Config**. Its range of values is {0-2073600} seconds with an additional special case value of "Unlimited", which is treated as the highest possible value. The default value is 0 (no timer).

Difference between 0 and Unlimited

No timer is created when **Session Max Life Limit** is configured to either the value 0 or “Unlimited”, so no timeout can occur. The difference between the two values is how they are handled when determining which value of **Session Max Life Limit** to use when there are several specified within the various configuration elements.

When a session is created the timer examines both the ingress side and the egress side and, in cases where both sides have a configured value for **Session Max Life Limit**, uses the side with the lower (stricter) value. On each side, the system reviews the configuration elements relevant to the session and uses the value of **Session Max Life Limit** from the configuration element with the highest precedence (**Session Agent**, then **Realm Config**, then **SIP Interface**, and lastly **SIP Config**).

- When the value is set to 0, the configuration element is ignored and the next configuration element in the precedence chain is looked at.
- A value between 1 and 2073600 (24 days) or the value “Unlimited” is treated as a valid configured value. In this case the system will not move onto the next element in the precedence chain and the value is used in the final comparison between the egress and ingress values. The value “Unlimited” is viewed as the highest possible value, and therefore is considered greater than any other value it is compared against.

For example, on the ingress side the value of **Session Max Life Limit** in **Realm Config** is set to 86400 and the value of **Session Max Life Limit** in **Session Agent** is set to “Unlimited”. The **Session Agent** value has a higher precedence than the **Session Agent** value so, therefore, the value “Unlimited” is used for the ingress side. On the egress side the value of **Session Max Life Limit** in **Realm Config** is set to 43200 and the value of **Session Max Life Limit** in **Session Agent** is set to 0 (no timer), so the value of **Session Max Life Limit** in **Realm Config** is used. When compared against the ingress side the value 43200 is less than “Unlimited”; therefore, the value set for the timer is 43200.

Add a SIP Interface

The SIP interface defines the signaling interface through which the Oracle Enterprise Communications Broker (Communications Broker) receives and sends SIP messages.

- Consider any SIP options that you want to add.
- Configure any inbound and outbound manipulation rules that you want to use with this interface.

In the configuration, you specify how the Communications Broker handles SIP messages and you can add SIP options.

1. Access the SIP Interface configuration object.

Configuration, System Administration, SIP Interface, Interfaces.

2. On the SIP Interface page, click **Add**.
3. On the Add Interface page, do the following:

Field	Description
State	Select to enable this configuration. Default: Enabled.

Field	Description
Enable Early Media Inhibit	Select to extract and store Session Description Protocol (SDP) messages from provisional responses before call setup.
Realm ID	Select the realm for this interface. Required to set SIP ports.
Description	Enter a description for this interface.
SIP Ports	Click Add , and do the following: <ul style="list-style-type: none"> a. Address—Enter the IP address of this interface. b. Port—Enter the port. Default: 5060. Range: 1-65535. c. Transport protocol—Select a protocol from the drop-down list. Default: UDP. Valid values: SCTP TCP TLS UDP d. Allow anonymous—Set how you want the system to handle requests from a SIP realm. Default: All. Valid values: All (allow all anonymous connections) Registered (session agents and registered endpoints, only) e. TLS profile—Select the TLS profile you want for this port. f. Click OK. g. (Optional) Add more SIP ports.
Initial Inv Trans Expire	Enter the default transaction timeout value (expressed in seconds) used exclusively for initial INVITE transactions. <ul style="list-style-type: none"> • Minimum—0 • Maximum—999999999
Session Max Life Limit	Enter the maximum amount of time in seconds of a session. An additional, special case value of “Unlimited”, is the highest possible value. <ul style="list-style-type: none"> • Minimum—0 (default) • Maximum—2073600 • Unlimited
Options	Add optional parameters or features by entering them in comma-separated format.
Stop Recurse	Enter one or more response codes that you want to cause this session agent to stop route recursion. You can enter individual response codes separated by a comma, such as 301,305 or a range such as 300-380. Default: 401,407. Valid values: 300-599.
Inbound Manipulation	Select an inbound manipulation rule from the drop-down list.
Outbound Manipulation	Select an outbound manipulation rule from the drop-down list.

Field	Description
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: <ul style="list-style-type: none"> • None (default) • Disabled—Do not support the interworking. • Enabled—Support the interworking.


4. Click **OK**.
5. (Optional) Add another SIP interface.
6. Save the configuration.

Configure SIP Config

Use SIP Config to set the parameters that apply to all SIP call traffic on the Oracle Enterprise Communications Broker (Communications Broker).

1. Access the SIP Interface configuration object.
Configuration tab, **System Administration** section, **SIP Interface**, **SIP Config**.
2. On the **SIP Config** page, do the following:

Home Realm ID	Enter the identifier of the home realm. This is the network to which the Communications Broker SIP proxy (B2BUA) is logically connected. If configured, this field must correspond to a valid identifier field entry in a realm-config.
Init Timer	Enter the initial timeout value in milliseconds for a response to an INVITE request, and it applies to any SIP request in UDP. In RFC 3261, this value is also referred to as <code>TIMER_T1</code> . The default is 500 . The valid range is: <ul style="list-style-type: none"> • Minimum—0 • Maximum—999999999
Max Timer	Enter the maximum transmission timeout (T2) for SIP in milliseconds. When sending SIP over UDP, a re-transmission timer is used. If the timer expires and the message is re-transmitted, the re-transmission timer is then set to twice the previous value (but will not exceed the maximum timer value). Using the default values of 500 milliseconds and 4000 milliseconds, the re-transmission timer is 0.5, then 1, 2, and finally 4. The incrementing continues until the transmission expire timer activates. The default is 4000 . The valid range is: <ul style="list-style-type: none"> • Minimum—0 • Maximum—999999999
Trans Expire	Enter the transaction expire timeout value (Timer B) in seconds to set the time for SIP transactions to live. The same value is used for Timers D, F, H and J. The default is 32 . The valid range is: <ul style="list-style-type: none"> • Minimum—0 • Maximum—999999999

Initial Inv Trans Expire	<p>Enter the default transaction timeout value (expressed in seconds) used exclusively for initial INVITE transactions.</p> <ul style="list-style-type: none"> • Minimum—0 • Maximum—999999999
Invite Expire	<p>Enter the invite expire timeout value (Timer C) in seconds to indicate the time for SIP client transaction will live after receiving a provisional response. The default is 180. The valid range is:</p> <ul style="list-style-type: none"> • Minimum—0 • Maximum—999999999
Session Max Life Limit	<p>Enter the maximum amount of time in seconds of a session. An additional, special case value of “Unlimited”, is the highest possible value.</p> <ul style="list-style-type: none"> • Minimum—0 (default) • Maximum—2073600 • Unlimited
Inactive Dynamic Conn	<p>Enter the inactive dynamic connection value in seconds to set the time limit for inactive dynamic connections.</p> <p>If the connection between the SIP proxy and a session agent is dynamic (for example, through dTCP), and the connection has been idle for the amount of time specified here, the SIP proxy breaks the connection. Idle is defined as not transporting any traffic. The default value is 32. The valid range is:</p> <ul style="list-style-type: none"> • Minimum—0 • Maximum—999999999 <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> Note: Setting this parameter to 0 disables this parameter.</p> </div>
Options	<p>Enter the option syntax for the option. Add more options, separated by a comma.</p>
ENUM SAG Match	<p>Select to enable ENUM session agent group matching.</p>
Refer Source Agent Routing	<p>The refer-source-agent-routing setting is applicable to REFER scenarios like Unattended Transfer, Attended Transfer with fresh INVITE and Attended Transfer with re-INVITE.</p> <p>Use this attribute to define the routing table look-up preferences based on source-agent, whether it is the call originator or the referring agent. This is specifically for terminating REFER scenario, and works only when the Refer Call Transfer parameter to the applicable Session Agent or Realm, and Dyn Refer Term in the applicable Realm are enabled.</p>

	<ul style="list-style-type: none"> Click Enable for Communications Broker to perform a routing table look-up based on the source agent of the referring party or referrer. The routing entry with Source Agent of REFERRER (and not of Call Originator) gets selected and is used to route the new INVITE to the REFERREE. Note: The INVITE message is not modified. Only the routing look-up is based on the REFERRING agent. If you do not select Enable, Communications Broker performs a routing table look-up based on the source agent of the calling party.
Default Context	Set the default source context for the system to use for a given call when unable to identify source context by any other method.
Parallel Forking	Select to cause the system to fork all sessions to all contacts of an Agent of Record.
Fork Group Timeout	Set the timeout value, in seconds, after which the Communications Broker tries the next fork group with the highest priority. Default: 0. Range: 0-32.
ASCII Based Routing	Select to enable routing with alphanumeric entries in the user database and routing table.
Proxy Registration	Select to allow the Communications Broker to accept a registration from an unauthorized domain, and proxy the registration to the intended registry.
Source Based Routing	Enable to allow the Communications Broker to perform source routing based on agent hostname.

3. Save and activate the configuration.

Restricting Session Initiation

The Oracle Enterprise Communications Broker (Communications Broker) can restrict the set of end stations that can initiate sessions to those originating through active session agents and previously registered users. By default, the Communications Broker does not restrict session initiation. You can enable the functionality in the **SIP Port** configuration.

The **SIP Port** configuration includes the **Allow Session Agents and Registered End-Points** control that you use to restrict session initiation. When selected, the Communications Broker responds to session initiation by endpoints that are not behind an agent or not already registered with a **403: Unauthorized** message.

Configure a SIP Interface Port

A SIP Interface port configuration defines the transport address and protocol that the Oracle Enterprise Communications Broker (Communications Broker) uses for sending and receiving messages through a SIP interface. You can apply a TLS profile to the configuration, and you can limit SIP requests from session agents and registered end points. You must configure at least one port per SIP interface. You can optionally configure multiple SIP ports per SIP interface. For example, suppose you configure the Communications Broker to receive calls by way of TCP and to send calls by way UDP, you must configure a SIP port for each protocol.

- Create the TLS profile that you want for this configuration.

In the following procedure, use step 4 to add more SIP interface ports.

1. Access the SIP Interface configuration object.
Configuration, System Administration, SIP Interface, Interfaces.
2. On the SIP Interface page, under SIP Port, click **Add**, and do the following:

Fields	Description
Address	Enter the IP address of the SIP interface.
Port	Enter port number for the SIP interface. Default: 5060. Range: 0-65535.
Transport Protocol	Select a protocol from the drop-down list. Default: UDP. Valid values: SCTP TCP TLS UDP
TLS Profile	Select the TLS profile you want for this port.
Allow Anonymous	Set how you want the system to handle requests from a SIP realm. Default: All. Valid values: All (allow all anonymous connections) Registered (session agents and registered endpoints, only)

3. Click **OK**.
The system displays the SIP Ports page with a list of SIP Interface ports you configured.
4. Optional—Click **Add** to add another SIP Interface port.
5. Click **Back**.
The system displays the SIP Interface page, where you can add another SIP Interface.
Optional—Configure SIP monitoring.

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 3-2 filter config dialog

The screenshot shows a web-based configuration interface. At the top, there is a 'Configuration' tab and a search bar containing 'sessions list'. A navigation pane on the left lists several categories: 'Interfaces', 'Monitoring', 'Monitoring Filters' (which is highlighted), and 'SIP Config'. The main content area is a modal dialog titled 'Add Filter Config'. It contains three text input fields: 'Name' (containing 'Filter1'), 'Address' (containing '0.0.0.0'), and 'User' (which is empty).

- 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 3-3 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 3-4 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.

SIP REFER

SIP REFER provides the Oracle Enterprise Communications Broker with the ability to terminate SIP REFER messages and perform attended or unattended call transfers. You can enable REFER termination at both the agent and Real Config, with agent configuration taking precedence. You can also configure the **Refer Notify Provisional** parameter to send NOTIFY messages for provisional responses.

SIP REFER Method Call Transfer for Communications Broker

The Communications Broker supports a handling mode for the REFER method that automatically converts a received REFER method into an INVITE method. This allows the Communications Broker to transfer a call without having to proxy the REFER back to the other User Agent (UA).

The Communications Broker provides the **Enable REFER Termination** parameter for provisioning the handling of REFER methods as call transfers. When you enable ISP REFER Method Call Transfer, the Communications Broker creates an INVITE message whenever it receives a REFER. The Communications Broker sends the INVITE message to the address in the Refer-To header. The INVITE message includes all of the unmodified information contained in the REFER message. The Communications Broker uses the previously negotiated SDP in the new INVITE message, and sends the NOTIFY and BYE messages to the UA upon call transfer completion. You configure this function at the SIP interface or agent with agent configuration taking precedence.

When a REFER method is received containing no Referred-By header, the Communications Broker adds one, allowing the Communications Broker to support all call agent screen applications.

The SIP REFER method call transfer feature supports the following:

- Both unattended and attended call transfers.
- Both successful and unsuccessful call transfers.
- Early media from the Referred-To party to the transferee.
- REFER method transfer from different sources.
- The REFER event package as defined in RFC 3515. This applies for situations where multiple REFER methods are used within a single dialog.
- Third party initiated REFER method signaling the transfer of a call by associating the REFER method to the dialogue through the REFER TargetDialog.

Unsuccessful Transfer Scenarios

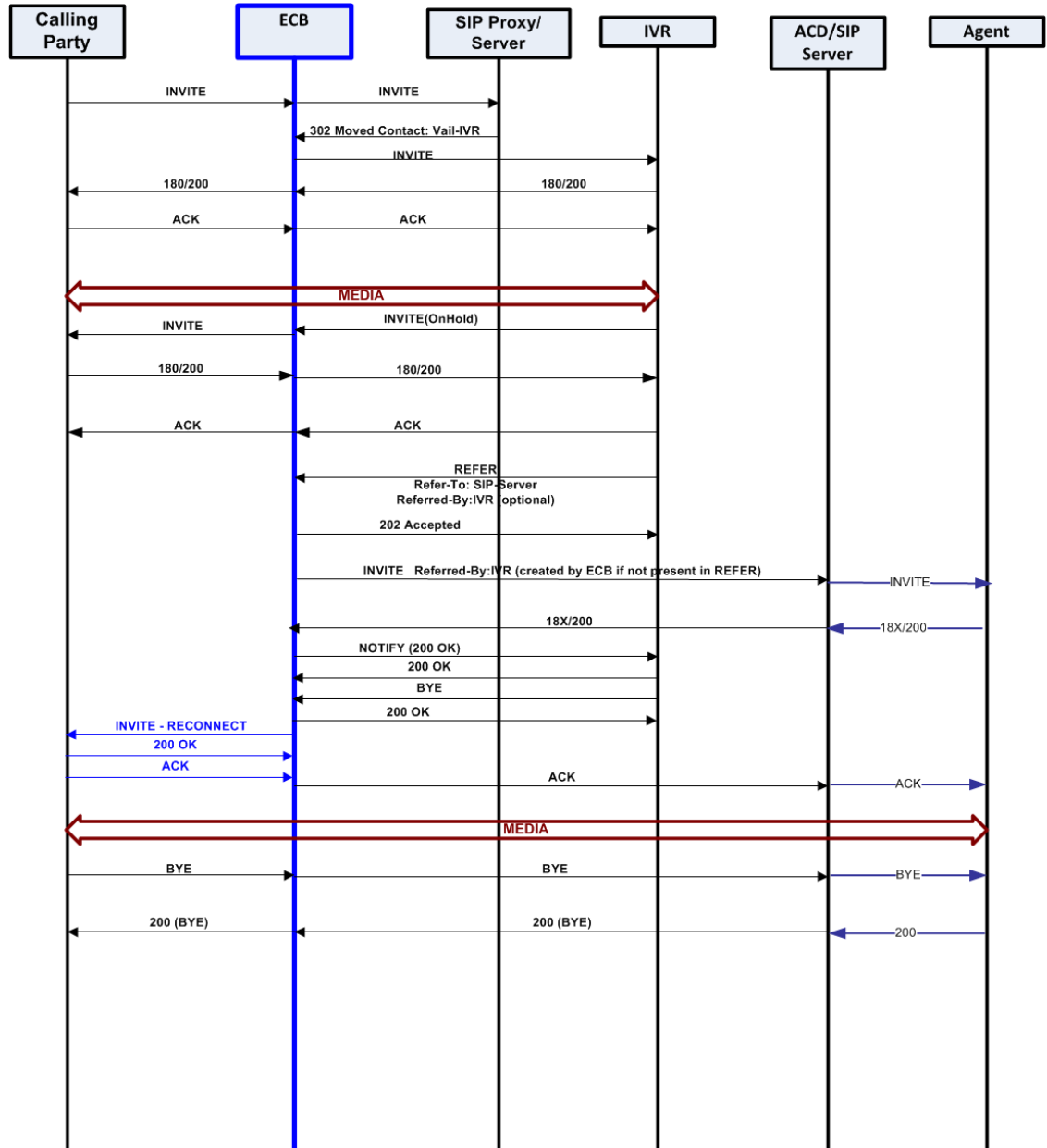
The Oracle Enterprise Communications Broker (Communications Broker) does not successfully handle the following unsuccessful, unusual, and unexpected transfer scenarios:

- The new INVITE to the Referred-To party gets challenged, the Communications Broker does not answer the challenge. It is treated with the 401/407 response just as any other unsuccessful final response.
- The header of the REFER message contains a method other than INVITE or contains URI-parameters or embedded headers not supported by the Communications Broker.
- The Communications Broker allows the Referred-To URI that happens to resolve to the same next-hop as the original INVITE went to, to do so.
- The Communications Broker ignores any MIME attachments within a REFER method.
- The Communications Broker recurses (when configured to do so) when the new INVITE sent to the Referred-To party receives a 3xx response.
- The transferee indicated support for 100rel, and the original two parties agreed on using it, yet the Referred-To party does not support it.
- The original parties negotiated SRTP keys.

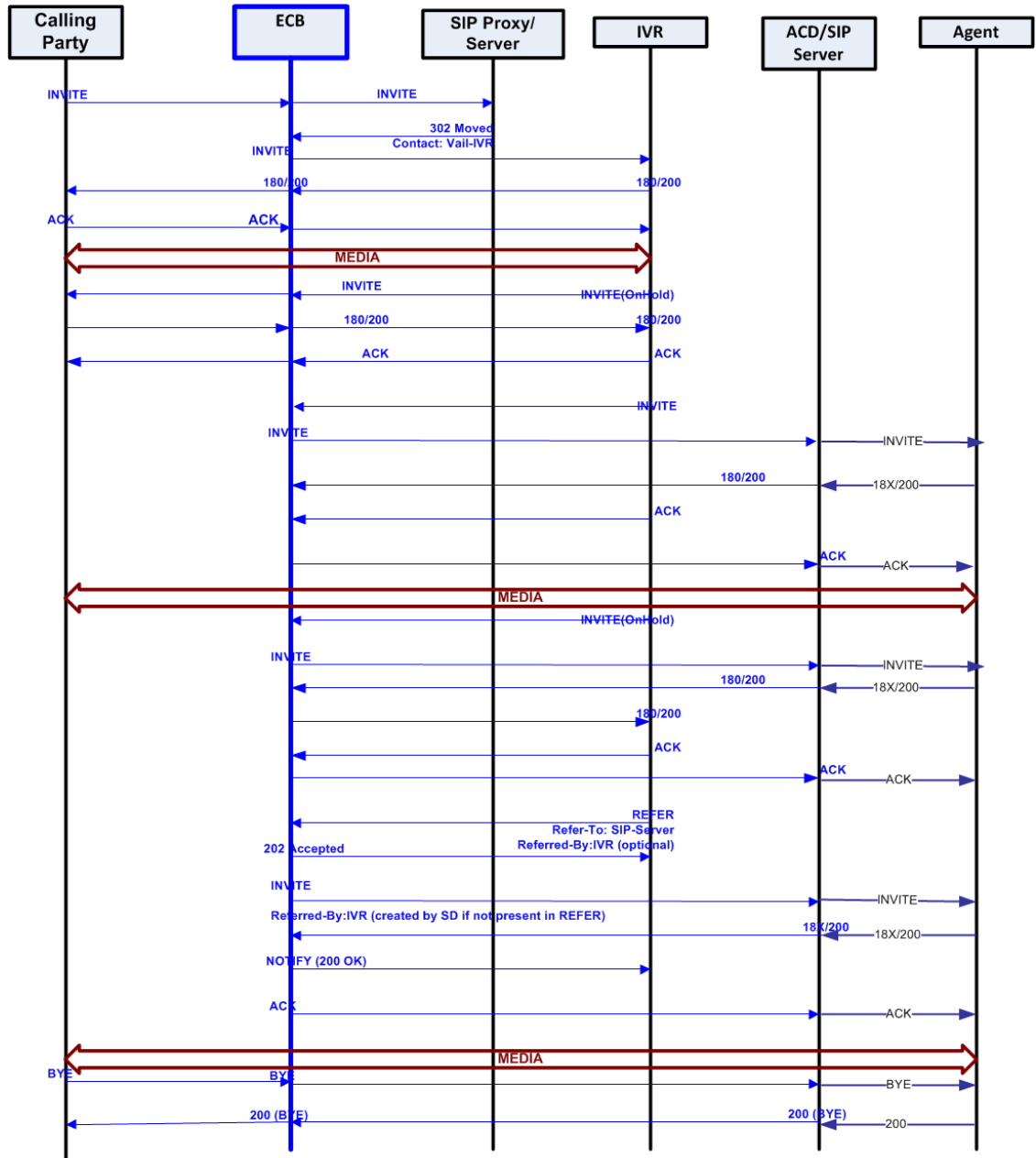
- The original parties agreed on a codec using a dynamic payload type, and the Referred-To party happens to use a different dynamic payload number for that codec.

Call Flows

The following ladder diagram shows an example of call flow for an unattended call transfer:



The following ladder diagram shows an example call flow of an attended call transfer:



Configure SIP REFER Method

The Oracle Enterprise Communications Broker (Communications Broker) allows you to set REFER termination on a per-agent and SIP interface basis. Agent configuration takes precedence over the SIP interface configuration.

Configure the SIP Interface.

Select **Enable REFER Termination** in the SIP Interface configuration to allow the specified agent to support SIP REFER method call transfers.

Use the following procedure to enable SIP REFER termination support.

1. Access the SIP Interface configuration object.
Configuration, System Administration, SIP Interface, Interfaces.
2. On the SIP Interface page, do one of the following:

- Open an existing interface.
 - Add a new interface.
3. In the interface configuration, select **Enable REFER termination**.
 4. Exit the SIP Interface configuration.
 5. Click **Service Provisioning, Agents, Session Agent**.
 6. Do one of the following:
 - Select an existing agent to edit.
 - Add a new agent.
 7. In the Agent configuration, select **Enable REFER termination**.
 8. Save and activate the configuration.

Dynamic REFER Support

In the simplest scenarios, the Communications Broker supports Dynamic REFER, also known as REFER-initiated call transfer, either by proxying the REFER to the other User Agent in the dialog, or by terminating the received REFER and issuing a new INVITE to the referred party. In addition, the Communications Broker provides dynamic refer support, which determines on a call-by-call basis whether to proxy the REFER to the next hop, or terminate the REFER and issue an INVITE. You configure dynamic REFER with the **Refer Call Transfer** parameter to the applicable Session Agent or Realm, and **Dyn Refer Term** in the applicable Realm.

By default, the Communications Broker proxies REFER sessions. You can configure the Communications Broker for both REFER Re-INVITE call flows and dynamic flows. You configure non-dynamic Re-INVITE scenarios using the ingress agent configuration only. You establish dynamic flows by also configuring the egress realm appropriately. The source agent determines whether to perform dynamic REFER, and the target **Realm Config** determines whether to dynamically proxy or issue a Re-INVITE.

Table 3-5 Refer-call-transfer

Refer-call Transfer	Description	Action
Enabled	To configure non-dynamic REFER	Enable the Refer Call Transfer parameter on the source agent for Re-INVITE
Dynamic	To configure Dynamic REFER	you set the Refer Call Transfer parameter in the applicable source agent to Dynamic
Disabled	To configure Proxy Mode	Disable the Refer Call Transfer parameter on the source agent for proxy mode

You also enable the **Dyn Refer Term** parameter in the target **Realm Config**.

The Communications Broker also includes a **Refer Call Transfer** setting in the **Realm Config**. If the REFER comes from a source that is not an agent, the Communications Broker uses the realm's **Refer Call Transfer** setting to determine how to handle the refer. Behavior is the same whether configured on session agent or realm.

The overall behavior proceeds as follows:

- When you set the source agent's **Refer Call Transfer** parameter to **Disabled** (the default), all received REFERs are simply proxied to the peer User Agent.

- When you set the source agent's **Refer Call Transfer** parameter to **Enabled**, the Communications Broker terminates all REFERs, generates a new INVITE, and sends the INVITE to the address in the Refer-To header.
- When you set the source agent's **Refer Call Transfer** parameter to **Dynamic**, the Communications Broker determines REFER handling on a call-by-call basis

This Communications Broker processing for non-dynamic and dynamic REFER proceeds as follows:

1. Determine whether the REFER comes from a session agent.
2. If the REFER comes from a session agent, check the **Refer Call Transfer** value for the session agent from which the REFER was received:
 - If the value is disabled, proxy the REFER to the peer User Agent, to complete REFER processing.
 - If the value is enabled, terminate the REFER and issue an new INVITE to the referred party, to complete REFER processing.
 - If the value is dynamic, identify the next hop egress realm.
3. If the REFER does not come from a session agent, the Communications Broker checks the ingress realm's **Refer Call Transfer** value. Behavior is the same whether configured on session agent or realm.
4. When the Communications Broker determines the next hop, it check the **Dyn Refer Term** value for that egress realm.
 - a. If **Dyn Refer Term** is **Disabled** (the default), proxy the REFER to the next hop to complete REFER processing.
 - b. If **Dyn Refer Term** is **Enabled**, terminate the REFER and issue an new INVITE to the referred party to complete REFER processing.

 **Note:**

The Communications Broker identifies the next hop realm based on either the Refer-To header or Based on the routing configuration, with the routing table taking higher priority.

Supported Scenarios

In the basic scenario for REFER initiated call transfer, a call is established between two User Agents (Alice and Bob). User Agent Bob then sends a REFER request to transfer the call to a third User Agent Eva. With dynamic call-transfer enabled, the Communications Broker prevents the REFER from being sent to Alice and generates the INVITE to Eva.

- If the INVITE to Eva succeeds, the Communications Broker sends a re-INVITE to Alice modifying the SIP session as described in Section 14 of RFC 3261, *SIP: Session Initiation Protocol*. At this point the Communications Broker cancels the original dialog between the Communications Broker and Bob.
- If the INVITE to Eva fails, call disposition depends on whether or not Bob issued a BYE after the REFER call transfer. If the Communications Broker did receive a BYE from Bob (for instance, a blind transfer), it proxies the BYE to A. Otherwise, the Communications Broker retains the original SIP session and media session, thus allowing Bob to re-establish the call with Alice by sending a re-INVITE. In this case, the Communications Broker sets a timer (32 seconds), and then sends a BYE.

- If a REFER method is received containing no Referred-By header, the Communications Broker adds one, allowing the Communications Broker to support all call agent screen applications.

In addition, the SIP REFER method call transfer feature supports the following:

- Both unattended and attended call transfers
- Both successful and unsuccessful call transfers
- Early media from the Referred-To party to the transferee
- REFER method transfer from different sources within the destination realm
- The REFER event package as defined in RFC 3515. This applies for situations where multiple REFER methods are used within a single dialog.
- Third party initiated REFER method signalling the transfer of a call by associating the REFER method to the dialogue via the REFER TargetDialog.
- The Referred-To party can be both in a different realm (and thus a different steering pool) from the referrer, and in the same realm
- The associated latching should not prohibit the Referred-To party from being latched to while the referee is still sending media.

The Communications Broker does not successfully handle the following anomalous transfer scenarios:

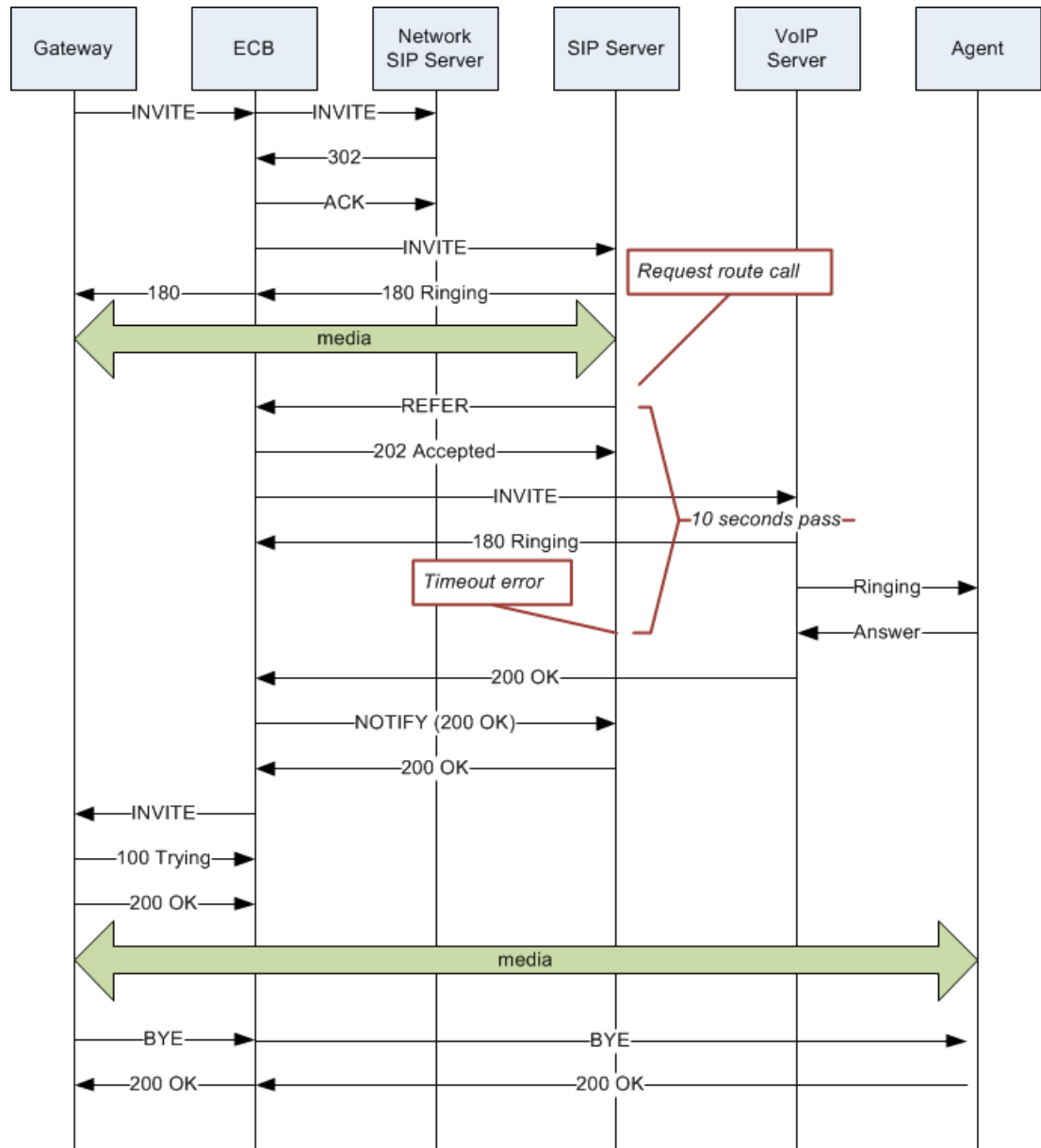
- The new INVITE to the Referred-To party gets challenged — the Communications Broker does not answer the challenge. It is treated with the 401/407 response just as any other unsuccessful final response.
- The header of the REFER message contains a method other than INVITE or contains URI-parameters or embedded headers not supported by the Communications Broker.
- The Communications Broker shall allow the Referred-To URI that happens to resolve to the same next-hop as the original INVITE went to, to do so.
- The Communications Broker ignores any MIME attachment(s) within a REFER method.
- The Communications Broker recurses (when configured to do so) when the new INVITE sent to the Referred-To party receives a 3xx response.
- The transferee indicated support for 100rel, and the original two parties agreed on using it, yet the Referred-To party does not support it.
- The original parties negotiated SRTP keys.

180 and 100 NOTIFY in REFER Call Transfers for the Communications Broker

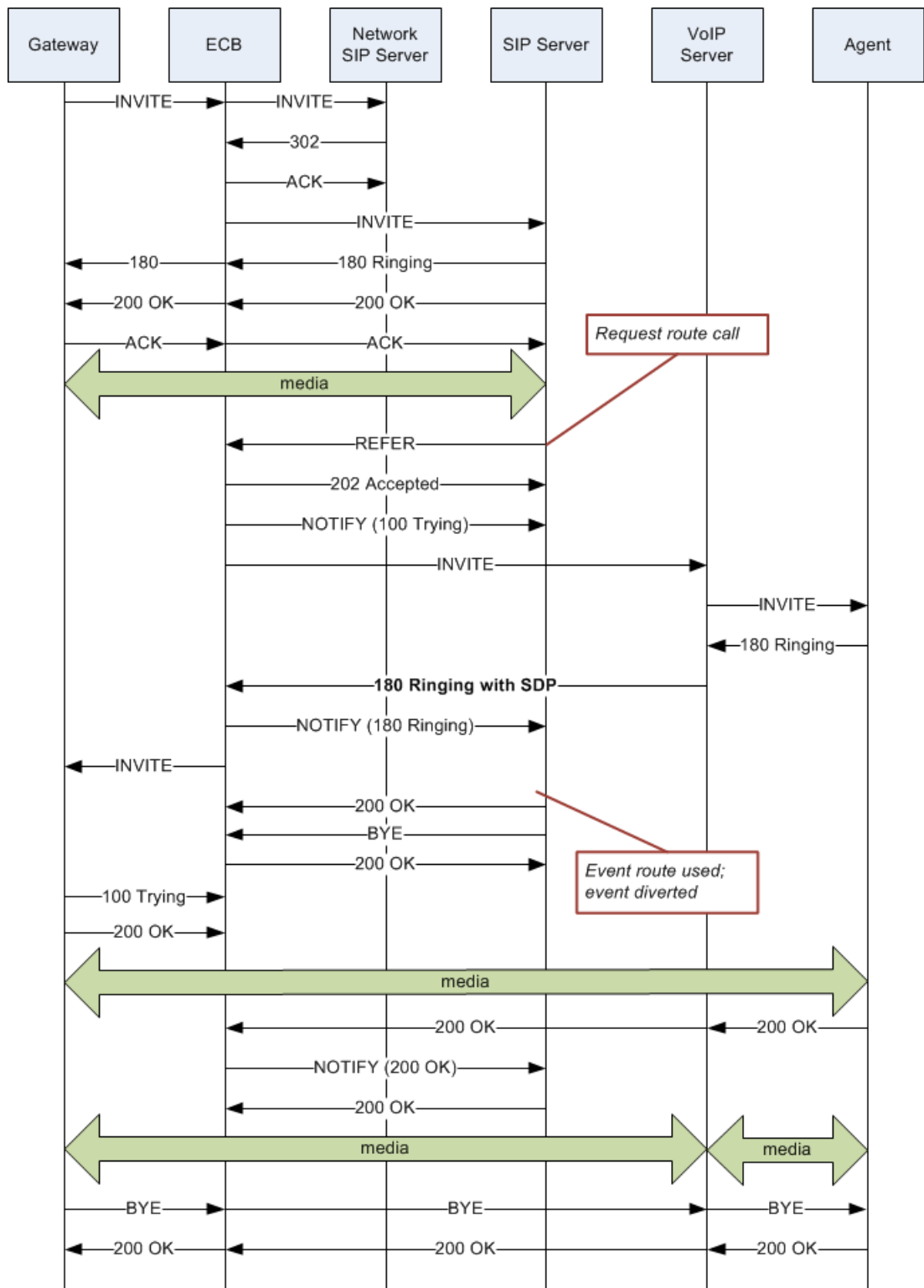
When you configure the Communications Broker to support REFER call transfers, you can enable it to send a NOTIFY message after it sends either a 202 Accepted or a 180 Ringing message. If your network contains elements that comply with RFC 5589, and therefore expect the NOTIFY message after the 202 Accepted and each provisional 180 Ringing, set the **Send NOTIFY messages for REFER Provisional Responses** to either **Initial** or **All**, according to your deployment needs.

Without this parameter changed from its default (**None**), the Communications Broker does not return send the NOTIFY until it receives the 200 OK response from the agent being called. If the time between the REFER and the NOTIFY exceeds time limits, this sequencing can cause the Communications Broker's NOTIFY to go undetected by devices compliant with RFC 5589. Failures during the routing process can result.

The following ladder diagram shows how a sample call flow times out when the **Send NOTIFY Messages for REFER Provisional Responses** parameter is not set.



When you compare the call flow above to the following one depicting the scenario when the Communications Broker has the **Send NOTIFY Messages for REFER Provisional Responses** changed from its default, the difference is that the Communications Broker now responds with a **NOTIFY** in response to the **202 Accepted** and it sends another one after the **180 Ringing**. This prevents the timeout and allows the event to be diverted successfully.



Sample Messages

In compliance with RFC 5589, the NOTIFY message with 100 Trying as the message body looks like the sample below. Note that the expires value in the subscription state header is

populated with a value that equals $2 * \text{TIMER C}$, where the default value of **TIMER C** is 180000 milliseconds.

```
NOTIFY sips:4889445d8kjdk3@atlanta.example.com;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjwt@biloxi.example.com;gr=3413kj2ha>;tag=a6c85cf
Call-ID: a84b4c76e66710
CSeq: 73 NOTIFY
Contact: <sips:3ld812adkjwt@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: active;expires=360
Content-Type: message/sipfrag
Content-Length: ...
SIP/2.0 100 Trying
```

Also in compliance with RFC 5589, the NOTIFY message with 180 Ringing as the message body looks like the sample below. Again, the expires value in the subscription state header is populated with a value that equals $2 * \text{TIMER C}$, where the default value of **TIMER C** is 180000 milliseconds.

```
NOTIFY sips:4889445d8kjdk3@atlanta.example.com;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjwt@biloxi.example.com;gr=3413kj2ha>;tag=a6c85cf
Call-ID: a84b4c76e66710
CSeq: 73 NOTIFY
Contact: <sips:3ld812adkjwt@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: active;expires=360
Content-Type: message/sipfrag
Content-Length: ...
SIP/2.0 180 Ringing
```

Also in compliance with RFC 5589, the NOTIFY message with 200 OK as the message body looks like the sample below.

```
NOTIFY sips:4889445d8kjdk3@atlanta.example.com;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjwt@biloxi.example.com;gr=3413kj2ha>;tag=a6c85cf
Call-ID: a84b4c76e66710
CSeq: 74 NOTIFY
Contact: <sips:3ld812adkjwt@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: terminated;reason=noresource
```

```
Content-Type: message/sipfrag
Content-Length: ...
SIP/2.0 200 OK
```

180 and 100 NOTIFY Configuration

You can specify the provisional mode for sending Notify messages using the **Refer Notify Provisional** setting in the **Realm Config** or the **Session Agent** configuration objects. By default, the Oracle Enterprise Communications Broker (Communications Broker) sends only the final result NOTIFY message.

Do the following to enable 100 and 180 NOTIFY messages in REFER call transfers.

1. Navigate to either the **Session Agent** or **Realm Config** configuration objects.
2. On the **Realm Config/Session Agent** page, in the **Add Realm Config/Add Agents** page, select a setting for the **Refer Notify Provisional** parameter.
3. Select any one of the following:
 - **None**—No intermediate Notify messages need to be sent.
 - **Initial**—Send an immediate 100 Trying NOTIFY, and the final result NOTIFY.
 - **All**—Send an immediate 100 Trying NOTIFY, plus a NOTIFY for each non-100 provisional messages the Communications Broker receives; and the final result NOTIFY.
4. Save and activate the configuration.

SNMP

This section explains how to configure Simple Network Management Protocol (SNMP) communities and trap receivers. These features are not essential for baseline Oracle Enterprise Communications Broker service, but they are necessary to use an element management system to manage Oracle Enterprise Communications Brokers. They provide important monitoring and system health information that contribute to a robust deployment of the Oracle Enterprise Communications Broker.

Overview

SNMP is used to support monitoring of network-attached devices for conditions that warrant administrative attention. SNMP is comprised of three groups of settings on a Oracle Enterprise Communications Broker. These settings are system-wide configurations including MIB contact information, SNMP community settings, and trap receivers.

Basic SNMP Parameters

The Oracle Enterprise Communications Broker includes several parameters that control basic SNMP functionality. The MIB-related elements are for informational purposes, and are helpful if set. The remainder of the parameters determines if certain Oracle Enterprise Communications Broker events are reported to the SNMP system.

SNMP Community

An SNMP community is a grouping of network devices and management stations used to define where information is sent and accepted. An SNMP device or agent might belong to

more than one SNMP community. SNMP communities provide a type of password protection for viewing and setting management information within a community. You can define multiple SNMP communities on a Oracle Enterprise Communications Broker to segregate access modes per community and NMS host.

Trap Receivers

A trap receiver is an application used to receive, log, and view SNMP traps for monitoring the Oracle Enterprise Communications Broker. An SNMP trap is the notification sent from a network device, the Oracle Enterprise Communications Broker in this case, that declares a change in service. Multiple trap receivers can be defined on a Oracle Enterprise Communications Broker either for redundancy or to segregate alarms with different severity levels to individual trap receivers.

Each server that an element management system is installed on should be configured as a trap receiver on all Oracle Enterprise Communications Broker's managed by that element management system.

SNMP Community Settings

Follow the steps below to configure an SNMP community on the Oracle Enterprise Communications Broker (Communications Broker).

1. Access the SNMP Communities configuration object.

Configuration, System Administration, SNMP, SNMP Communities.

2. On the SNMP community page, do the following:

Community	Enter an SNMP community name of an active community where this Communications Broker can send or receive SNMP information. A community name value can also be used as a password to provide authentication, thereby limiting the NMSs that have access to this Communications Broker. With this field, the SNMP agent provides trivial authentication based on the community name that is exchanged in plain text SNMP messages. For example, public. Default: blank Valid values: alpha-numeric characters.
IP Addresses	Enter an IPv4 address that is valid within this SNMP community. This IPv4 address corresponds with the IPv4 address of the NMS application that monitors or configures this Communications Broker. You can enter multiple addresses, separated by commas.

3. Save the configuration.

Set Trap Receiver Settings

Follow the steps below to configure trap receivers on your device.

1. From the Trap receiver list, click **Add**.

The system displays the Add SNMP Trap Settings dialog.

2. Community name—Enter the SNMP community name to which this trap receiver belongs. For example, **Public**. Valid values: Alpha-numeric characters. Default: Blank.
3. IP address—Enter the IPv4 address of an authorized NMS. This value is the IPv4 address of an NMS where traps are sent. Enter the IP address in dotted decimal format.

4. IP Port—Enter the port number of an authorized NMS. If you do not specify a port number, the default SNMP trap port of 162 is used.

HTTP Server Settings

Configure your preferences for the Oracle Enterprise Communications Broker HTTP server in the HTTP Server page. If you correctly performed the **run setup** procedure after installation, the HTTP Server object for wancom0 is already present.



Note:



If the HTTP state and HTTPS state parameters are set via **run setup**, you can edit those parameters as required.

1. Access the Web Server Settings configuration object.

Configuration, System Administration, HTTP Server.

2. On the **Add HTTP Server** page, do the following:

Name	This is a multi-instance element. You must provide a name. The run setup procedure creates an object named wancom0 .
State	Status of HTTP server configuration element. The default value is Enabled.
HTTP State	Specify whether or not to enable HTTP for accessing the Web server. Default is enabled. A check mark indicates enabled, and a blank box indicates disabled.
Realm	The name of the realm where this server resides. This parameter can be empty.
IP Address	Specifies the address of the network interface used within the specified realm. This parameter can be empty.
HTTP Port	Enter the port for the HTTP connection. Default: 80. Valid values: 1-65535.
HTTP Strict Transport Security Policy	<p>Enable or disable the HSTS (HTTP Strict Transport Security) policy. The HSTS policy redirects a insecure HTTP connection to a secure HTTPS connection.</p> <ul style="list-style-type: none"> • If HSTS is enabled the max-age in the Strict-Transport-Security will be set to 31536000 • If HSTS is disabled the max-age in the Strict-Transport-Security will be set to 0 always. The header is not removed completely.

	<p> Note:</p> <ul style="list-style-type: none"> • HSTS only works for default ports (HTTP-80 and HTTPS-443). • HSTS does not enforce HTTPS for GUI on localhost that are tunneled.
HTTPS State	Specify whether or not to enable HTTPS (secure connection) for accessing the Web server. Default is disabled. A check mark indicates enabled, and a blank box indicates disabled.
HTTPS Port	Enter the port for the HTTPS connection. Default: 443. Valid values: 1-65535.
HTTP Interface List	Specifies the application that uses this object. The run setup procedure creates sets this object as GUI . Additional settings include REST .
HTTP File Upload Size	Maximum size of the file to be uploaded in MB. The range is 0 to 999. Default is 0.
TLS Profile	Enter the Transport Layer Security (TLS) Protocol profile name to use with HTTPS. Valid values: Alpha-numeric characters. Default: Blank.
	<p> Note:</p> <p>If you specify a TLS profile, and HTTP is enabled, the Communications Broker checks against the TLS profile table for a match. If there is no match, the applicable errors display during the verification of the configuration.</p>
Auth Profile	Not Used

3. Click OK.
4. Save the configuration.

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.

Admin Security - Feature Set


This section describes the implications of adding and removing the Admin Security feature set on Communications Broker.

Enabling the Admin Security Feature

Use the **Set Entitlements** option to provision the Admin Security feature in Communications Broker.

1. Click the **System** option in the Communications Broker home page.
2. Click **System Operations** menu in the left-side pane.
3. Click **Set Entitlements**. In the **Set entitlements** page complete the following fields:

Table 3-6 Fields in the Set Entitlements page

Field	Description
Admin Security	<ul style="list-style-type: none"> • Select the Enable checkbox to activate Admin Security features. • With AdminSecurity enabled, Shell access is denied, Password-policy and login-config parameters are enabled. For more information, see Password Policy. <div style="border: 1px solid #0070C0; padding: 5px; margin-top: 10px;"> <p> Note:</p> <p>Enabling the Admin Security feature activates enhanced security functions. Once enabled, security cannot be reverted without resetting the system back to the factory default state.</p> </div> <ul style="list-style-type: none"> • With Admin Security disabled, Shell access is denied, and Password-policy features are disabled.
Admin Security With ACP/NNC	ACP/NNC license add more password security.

Login Policy

The Login Policy controls concurrent system access to a specified number of users, sets the maximum number of unsuccessful login attempts, and specifies the response to login failure.

The single instance **login-config** configuration element defines the login policy.

1. Click the **Configuration** tab, and then click **Security**.
2. Click **Login Config** in the left-side pane.
3. In the **Add Login Config** page, provide the following details:

Table 3-7 Fields in the Add Login Config page

Field	Description
Enable Login Banner	Check the enable check box to enable the display of the Login Banner. The default option is Enabled. For more information on the Login Banner, see Login Banner .
Concurrent Session Limit	Specify the maximum number of simultaneous connections allowed per username. Valid values are integers within the range of 1 through 10, with a default value of 2 (simultaneous connections). Retain the default value, or specify a new connection limit.
Max Login Attempts	Specify the number of consecutive unsuccessful login attempts that trigger disconnection of a console, SSH, or SFTP session. Valid values are integers within the range of 2 through 100, with a default value of 3 (sessions). Retain the default value, or specify a new threshold value.
Login Attempt Interval	Enter the idle interval in seconds imposed after an unsuccessful login attempt. Valid values are integers within the range of 4 through 60, with a default value of 4 seconds. Retain the default value, or specify a new login interval.
Lockout Interval	Add the number of seconds that logins from an interface are not allowed after the max-login-attempts threshold has been reached. Valid values are integers within the range of 15 through 300. The default value is 60 seconds.
Send Alarm	enables the generation and transmission of alarms in the event of an interface lockout. Allowable values are enabled (the default) or disabled. Retain the default value, or select disabled to squelch alarm generation.

4. Click **OK**.

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 complete 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.

Password Policy

The Admin Security feature supports the creation of password policies that enables the authentication process by imposing requirements for:

- password length
- password strength
- password history and re-use
- password expiry and grace period

The Admin Security feature set needs the following password length/strength requirements:

- user class passwords must contain at least 9 characters (Admin Security only)
- admin class passwords must contain at least 15 characters
- passwords must contain at least 2 lower case alphabetic characters
- passwords must contain at least 2 upper case alphabetic characters
- passwords must contain at least 2 numeric characters
- passwords must contain at least 2 special characters (such as !, ", #, \$, %, &, ', (,), *, +, ,, -, ., /, :, ;, <, =, >, ?, @, [, \,], ^, _ , ` , {, |, }, ~)
- passwords must differ from the prior password by at least 4 characters
- characters in password must differ from the prior password in at least 8 positions
- passwords cannot contain, repeat, or reverse the entire username
- passwords cannot contain three consecutive identical characters

When you enable the **password-policy**, and **password-policy-strength** as part of the Admin Security ACP feature, you impose the following requirements in addition to those enforced with the Admin Security feature:

- Passwords cannot contain two or more characters from the user ID
- Passwords cannot contain a sequence of three or more characters from any password contained in the password history cache
- Passwords cannot contain a sequence of two or more characters more than once
- Passwords cannot contain either sequential numbers or characters, or repeated characters more than once.

In the absence of the Admin Security ACP feature, you may safely ignore the password-policy-strength and retain the default value (disabled).

For more information, see [Configuring the Admin Security with ACP Password Rules](#).

Configuring Password Policy Properties

To enforce the stronger password rules and restrictions that the Administrative Security ACP license it provides, you must enable the password-policy-strength parameter.

1. Click the **Configuration** tab, click **Security**.
2. Click **Password Policy**.
3. In the **Add Password Policy** page, add values to the fields as described in the table below

Table 3-8 Fields in the Add Password Policy page

Field	Description
Min Secure Pwd Len	Ignored when the Admin Security with the ACP feature is installed and the password-policy-strength is set to enabled. The default value is 8 (characters). The allowable values 8 through 64.
Expiry Interval	Specify the password lifetime in days. Password lifetime tracking begins when a password is changed. The default value is 90 (days). The allowable values are integers within the range 0 through 65535.
Expiry Notify Period	Specify the number of days before expiration that users will begin to receive password expiration notifications. The default value is 30 (days). The allowable values are integers within the range 1 through 90. During the notification period, users are reminded of the impending password expiry at login and logout.
Grace Period	Works in conjunction with grace-logins field. After the password expires, you are granted some number of logins (as specified in the grace-logins field) for some number of days (as specified in the graceperiod field). Once the number of grace-logins is exceeded, or graceperiod has expired, you are forced to change your password. The default value for grace-period is 30 (days). The allowable values for grace-period are integers within the range 1 through 90.
Grace Logins	Works in conjunction with grace-period field. See description for the grace-period field for more information. The default value for the grace-logins field is 3 (logins). The allowable values for the grace-logins field are integers within the range 1 through 10
Password History Count	Specify the number of previously used passwords retained in an encrypted format in the password history cache. The default value is 8. (retained passwords). The allowable values are integers within the range 1 through 24. By default, a user's eight most recently expired passwords are retained in the password history. As the user's current password is changed, the password is added to the history, replacing the oldest password entry. New, proposed passwords are evaluated against the contents of the password cache, to prevent password re-use, and guard against minimal password changes.

4. Click **OK**.

Configure the Administrative Security with ACP Password Rules

To enforce stronger password rules and restrictions that the Administrative Security ACP license provides, enable the **password-policy-strength**.

Confirm that the Admin Security With ACP/NNC has been enabled. For more information, see [Enabling the Admin Security Feature](#).

1. Click the **Configuration** tab, click **Security**.
2. Click **Password Policy**.
3. In the **Add Password Policy** page, add values to the fields as described in the table below
4. Click the **enable** check box next to the **Password Policy Strength** field.

When you set the **password-policy-strength** to enable as part of the Admin Security ACP feature, you impose requirements in addition to those enforced with the Admin Security feature.

Changing a Password

The Login Banner provides prior notice of an impending password expiration.

For more information on changing the password, see the section [Changing a Password](#) in the Oracle Communications Session Border Controller Admin Security Guide Release 9.2.0.

4

Maintenance and Debugging

Oracle Enterprise Communications Broker (Communications Broker) software closely aligns with Oracle Session Border Controller (SBC) software. The vast majority of reference and debugging processes, procedures, and information is common across Oracle SBC products.

Common Maintenance and Debugging Documentation

The following table directs you to other Oracle documentation that provides monitoring and debugging information.

Log File Definition and Descriptions Fault Information Management Manual Configuration Management Process and Procedures	Oracle SBC Maintenance and Troubleshooting Guide
MIB Descriptions MIB Definition and Identification (OID Reference) SNMP GETs SNMP Trap Definition and Descriptions	Oracle SBC MIB Reference Guide
Manual HDR Management HDR Group Definition and Descriptions	Oracle SBC Historical Data Recording (HDR) Resource Guide

Your Oracle Enterprise Communications Broker Image

Your Oracle Enterprise Communications Broker arrives with the most recent, manufacturing-approved run-time image installed on the flash memory. If you want to use this image, you can install your Oracle Enterprise Communications Broker, establish a connection to the Oracle Enterprise Communications Broker, and then begin to configure it. On boot up, your system displays information about certain configurations not being present. You can dismiss these displays and begin configuring your Oracle Enterprise Communications Broker.

If you want to use an image other than the one installed on your Oracle Enterprise Communications Broker when it arrives, you can use the information in this section to obtain and install it.

Obtain a New Image

You can download software images onto the platform of your Oracle Enterprise Communications Broker (Communications Broker) from various sources. You can take any one of the following actions:

- Obtain an image from the Oracle Software Delivery Cloud.

- Obtain an image from your Oracle customer support representative, who will transfer it to your system.

Regardless of how you obtain the image, you need to use Secure File Transfer Protocol (SFTP) to copy it from its source to your Communications Broker.

Upgrade Software - Web GUI System Tab

You can upgrade the system software from the System tab on the Web GUI. The system requires a reboot after the upgrade.

1. From the GUI, click the **System** tab.
2. Click **Software Operations**.
3. Click **Upgrade Software**.
4. Click **Verification**.
5. Verify that system health, synchronization health, current configuration version, and disk usage are appropriate and adequate for the upgrade.
6. From the drop-down list, select **Upload Method**, and select one of the following methods.
 - Local—Use to select a file from your system for transfer.
 - Flash—Use to select a file already on the device.
 - Network—Use to specify parameters for network boot by way of file transfer.

The system displays the Upgrade Software dialog with the fields required for your upgrade.

7. Complete the required fields.
 - Software file to upload. (If the Upload Method is Local) Use **Browse** to locate the file on your local system.
 - Boot file to upload. (If the Upload Method is Local) Use **Browse** to locate the boot file on your local system. The existing bootloader is backed up in the folder `/code/images` and the new uploaded bootloader file is placed in `/boot` directory and is renamed to *bootloader*.
 - Software file. (If the Upload Method is Flash) The location and name of the file on the device.
 - Boot file. (If the Upload Method is Network) The complete name of the boot file.
 - Host IP. (If the Upload Method is Network) The IP address of the FTP server.
 - FTP username. (If the Upload Method is Network) The user name to log onto the FTP server.
 - FTP password. (If the Upload Method is Network) The password to log onto the FTP server.
8. Optional. Select **Reboot After Upload**.
9. 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.
10. Click **OK**.

If you selected **Reboot After Upload**, the system reboots.

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

The Oracle Enterprise Communications Broker (Communications Broker) allows you obtain a pre-defined file containing information that support personnel normally request.

1. Access the **Support Information** page. Click **System**, **Support Information**.
2. On the **Support Information** page, click **Support Information**.
3. Click **Support Information**.

The Communications Broker displays a **Progress** message box, which indicates the system is generating support information output. When complete, your browser displays a dialog where you to decide what to do with the support-info.log file.

4. Do one of the following:
 - Follow the dialog's instructions to select the application you want to use to display your support-info.log file.
 - Save the file locally.

5

Procedure to Avoid the activate_config Error

Use the information in this topic to avoid the error that you may see when you try to load the baseconfig on the Public cloud VM.

Scenario: You may see the error when you try to save the configuration and activate the configuration using the activate_config ACLI command.

1. Create a Communications Broker Release 4.2.0 VM on Azure or AWS.
2. Load the baseconfig using the command:

```
spl load acli baseconfig
```

3. Save the configuration using save-config
4. Activate the configuration using activate-config. You may observe the error message:

```
ERROR: sip-config [] has reference to home-realm-id [ecb] which does not exist
```

Procedure to Avoid the Error

Perform the following tasks to avoid encountering the error:

1. Create the phy-interface.
2. Create network-interface with the above created phy-interface.
3. Create realm-config with the above created network_interface
4. Create sip-interface along with its sip-port with above created network_interface and realm-config
5. Perform spl load acli baseconfig
6. Save and activate the configuration

6

Deploying Communications Broker on Oracle Cloud Infrastructure (OCI)

You can deploy Communications Broker on the Oracle Cloud Infrastructure (OCI). Before installing Communications Broker, SSH keys must be generated to access the Communications BrokerVM instances.



Note:

The Communications Broker Administrator Guide Release P-Cz4.1.0 documents information that is specific to deploying Communications Broker in an OCI environment. For generic instructions and more information, deploying in OCI, see the section [Create and Deploy on OCI](#) in the Platform Preparation and Installation Guide Release S-Cz9.2.0 - for Service Provider and Enterprise.

Deploying Communications Broker on OCI - Important Points to Note

Here are some important points you should note before you deploy Communications Broker on OCI.

- The s0p0 interface is sufficient to run traffic on Communications Broker.
- The OCI Resource Manager is not supported by Communications Broker.
- The run setup operation must not be done on the OCI platform. To load the default configurations (dialing-context, ldap-config, policy, http-server), execute the command: `sp1 load acli baseconfig` in the ACLI session and perform the Save and Activate operation.

Standard Shapes Supported for Communications Broker Deployment

The section lists the supported standard shapes that you can use to deploy Communications Broker on OCI.

Table 6-1 Standard Shapes Supported

Shape	OCPUs	VCPUs	Memory (GB)	Max Forwarding Cores	Max VNICs	Max Rx/Tx Queues
VM.Optimize d3.Flex (Small)	4	8	16	8	4	8

Table 6-1 (Cont.) Standard Shapes Supported

Shape	OCPUs	VCPUs	Memory (GB)	Max Forwarding Cores	Max VNICs	Max Rx/Tx Queues
VM.Optimize d3.Flex (Medium)	8	16	32	9	8	9
VM.Optimize d3.Flex (Large)	16	32	64	15	16	15
VM.Standard 3.Flex (Small)	4	8	16	8	4	8
VM.Standard 3.Flex (Medium)	8	16	32	15	8	15
VM.Standard 3.Flex (Large)	16	32	64	15	16	15

7

Deploying Communications Broker on AWS

You can deploy Communications Broker on AWS. Before installing Communications Broker, SSH keys must be generated to access the Communications Broker VM instances.

 **Note:**

Ensure that you read the contents of this section that contains information specific to deploying Communications Broker in an AWS environment. For generic instructions and more information, see the Session Border Controller 9.3.0 Platform Preparation and Installation Guide - for Service Provider and Enterprise.

Standard Shapes Supported for AWS Deployment

The section lists the supported standard shapes that you can use to deploy Communications Broker on AWS.

Table 7-1 Standard Shapes Supported

Shape	Instance Type	CPUs	Memory (GB)
Small	c5.2xlarge	8	16 GB
Medium	c5.4xlarge	16	32 GB
Large	c5a.8xlarge	32	64 GB

8

Deploying Communications Broker on Azure

You can deploy Communications Broker on the Azure. Before installing Communications Broker, SSH keys must be generated to access the Communications Broker VM instances.

 **Note:**

Ensure that you read the contents of this section that contains information specific to deploying Communications Broker in an Azure environment. For generic instructions and more information, see the Session Border Controller 9.3.0 Platform Preparation and Installation Guide - for Service Provider and Enterprise.

Standard Shapes Supported for Azure Deployment

The section lists the supported standard shapes that you can use to deploy Communications Broker on Azure.

The following table lists the Azure instance sizes that you can use for the Communications Broker.

Table 8-1 Standard Shapes Supported

Shape	Size(Fs Series)	CPUs	Memory (GB)
Small	Standard_F8s	8	16 GB
Medium	Standard_F16s	16	32 GB
Large	Standard F32s v2	32	64 GB