



CHAPTER 15

Configuring the Cisco Application Control Engine

The following sections describe how to configure the Cisco Application Control Engine:

- [About the Cisco Application Control Engine, page 15-1](#)
- [Configuring the Cisco Application Control Engine, page 15-4](#)

About the Cisco Application Control Engine

This section describes the Cisco Application Control Engine (ACE) and includes the following topics:

- [ACE Overview, page 15-1](#)
- [ACE Topology, page 15-1](#)
- [Configuration Overview, page 15-2](#)

ACE Overview

The ACE provides access control, load balancing, and high availability functionality for the Cisco TelePresence Exchange System server cluster.

Clients gain access to the server cluster through the ACE. The ACE provides a virtual IP address (VIP) that acts as a proxy for the servers. The ACE distributes client requests to the servers based on the service requested, the load-balancing algorithm, the health of the servers, and session persistence requirements.

The ACE distributes the following types of incoming Cisco TelePresence Exchange System traffic:

- SIP traffic to the call engines
- HTTP traffic to the IVR application on the call engines
- HTTP traffic to the administration servers

ACE Topology

You can configure up to four interfaces on the ACE appliance.

- You must configure one interface to serve as the outside interface.

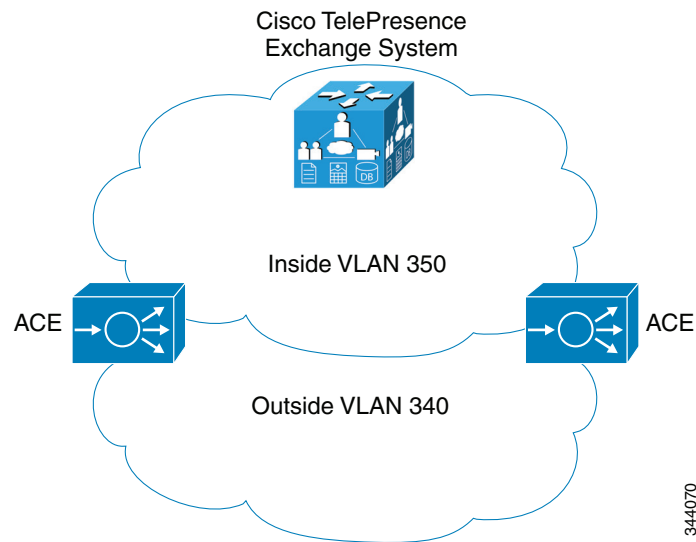
The outside interface connects to the users of the Cisco TelePresence Exchange System cluster.

If you have a redundant ACE in the application, you must configure the outside interface as a trunk to support both a native VLAN for untagged traffic, and a fault tolerant (FT) VLAN to provide a communication path between the two ACE appliances. The two ACE appliances are in an active/standby configuration. The ACE in standby is known as the peer.

- You must configure one interface to serve as the inside interface to provide access to the Cisco TelePresence Exchange System.

**Note**

The inside and outside interfaces must belong to different VLANs.



Configuration Overview

The ACE appliance provides server load balancing for three types of message traffic:

- SIP call control
- HTTP messages for the IVR service
- HTTP messages for the administration console

To configure the ACE for the Cisco TelePresence Exchange System, complete the following procedures:

**Note**

For links to the ACE configuration procedures listed below, see the [“Configuring the Cisco Application Control Engine”](#) section on page 15-4.

1. Configure the hostname.
2. Configure the physical interfaces.
Assign VLANs to the interfaces.
3. Configure the real servers.
Create a real server for each server in the Cisco TelePresence Exchange System cluster.
4. Configure access control lists.

Create access control lists (ACLs) to filter incoming or outgoing traffic on an interface based on configurable criteria (such as protocol type or IP address ranges).

5. Configure health probes.

Create a health probe for each traffic type supported by Cisco TelePresence Exchange System. A health probe defines the type of message that the ACE will periodically send to the servers, and the expected responses.

6. Configure the server farms.

Create a server farm for each Cisco TelePresence Exchange System traffic type. A server farm is a virtual server that provides a specific service. The ACE load-balances the incoming requests among the real servers that are associated with the server farm. The ACE also monitors server health (by sending periodic probes) and distributes work only to the operational real servers.

7. Configure session persistence.

Create a sticky group for each server farm. A sticky group defines how to identify the session that is associated with each incoming message.

8. Configure a management class map and a policy map.

Create these policies to allow remote management access to the Cisco TelePresence Exchange System cluster.

9. Configure Layer 7 load balancing policy maps and class maps.

Define Layer 7 policy maps and class maps for each of the three traffic types. Layer 7 class maps and policy maps define the classification and policy for traffic based on upper-layer message parameters such as HTTP header fields and SIP header fields.

10. Configure Layer 3 and Layer 4 policy maps and class maps.

Define Layer 3 and Layer 4 policy maps and class maps for each of the three traffic types. These class maps and policy maps define the classification and policy for traffic based on Layer 3 and Layer 4 message parameters such as source IP address, port, and protocol.

Each Layer 7 policy must be included in a Layer 3 and Layer 4 policy.

11. Configure VLAN interfaces.

Activate the management and load-balancing policies by associating the policy maps with the VLAN interfaces.

12. Configure miscellaneous ACE parameters and logging options.

Configure various parameters and settings that are important for correct operation of the Cisco TelePresence Exchange System.

An overview of the ACE appliance is available in the *Cisco ACE 4700 Series Application Control Engine Appliance Quick Start Guide*, at

http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/ace_appliances/vA3_2_2/configuration/quick/guide/ace_appliance_qsg.html.

Additional information about ACE appliance configuration for server load balancing is available in the *Cisco ACE 4700 Series Appliance Server Load-Balancing Configuration Guide*, at

http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/ace_appliances/vA4_2_0/configuration/slb/guide/slbgd.html.

Additional information about configuring redundant ACE appliances is available in the *Cisco ACE 4700 Series Appliance Administration Guide*, at

http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/ace_appliances/vA4_2_0/configuration/administration/guide/redundcy.html.

Other documents related to the ACE appliance are available at http://www.cisco.com/en/US/products/ps7027/tsd_products_support_series_home.html.

Configuring the Cisco Application Control Engine

This section describes how to configure the ACE and includes the following topics:

- [Configuring the Hostname, page 15-4](#)
- [Configuring Interfaces, page 15-5](#)
- [Configuring Real Servers, page 15-7](#)
- [Configuring Access Control Lists, page 15-8](#)
- [Configuring Health Probes, page 15-8](#)
- [Creating Server Farms, page 15-10](#)
- [Configuring Session Persistence, page 15-12](#)
- [Configuring Class Maps, page 15-14](#)
- [Configuring Policy Maps, page 15-16](#)
- [Configuring VLAN Interfaces, page 15-19](#)
- [Configuring Miscellaneous Parameters, page 15-23](#)
- [Configuring ACE Logging Options, page 15-24](#)



Note All IP addresses shown in the configurations are for example purposes only.

Configuring the Hostname

By default the hostname of the ACE is switch. You can assign a specific name to the ACE. For configurations in which a redundant pair of ACEs is in use, you need to define both a hostname for the primary ACE (active system) and a peer hostname for the standby system.

All configuration for the ACE is done on the primary ACE. All configuration and changes in status are regularly communicated to the standby ACE through the fault-tolerant VLAN.

To configure the hostname for the ACE, do the following task:

	Command	Purpose
Step 1	switch/Admin# configure terminal	Enters configuration mode.
Step 2	switch/Admin(config)# peer hostname name	Configures the hostname for the peer (standby) ACE. The active ACE regularly communicates its configuration to the peer ACE. (Required only for redundant ACE configuration). The hostname is a case-sensitive text string from 1 to 32 alphanumeric characters in length. The default value of hostname is switch .

	Command	Purpose
Step 3	switch/Admin(config)# hostname <i>name</i>	Configures the hostname for the active ACE.
Step 4	hostname/Admin(config)# exit	Exits configuration mode.

The following example shows how to set the hostname for an ACE in a non-redundant configuration to ACE_1:

```
switch/Admin# configure terminal
switch/Admin(config)# hostname ACE_1
ACE_1/Admin(config)# exit
```

The following example shows how to set hostnames for two ACEs in a redundant configuration where ACE_1 is the active ACE and ACE_2 is the peer ACE that is in standby:

```
switch/Admin# configure terminal
switch/Admin(config)# peer hostname ACE_2
switch/Admin(config)# hostname ACE_1
ACE_1/Admin(config)# exit
```

Configuring Interfaces

You can configure up to four interfaces on the ACE. You must configure at least one outside interface and one inside interface. The outside interface connects to the users of the Cisco TelePresence Exchange System server cluster and the inside interface connects to the server cluster.

The inside and outside interfaces must belong to different VLANs.

To configure an interface on the ACE, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin# configure terminal	Enters configuration mode.
Step 2	ACE_1/Admin(config)# interface gigabitEthernet <i>slot_number / port_number</i>	Enters interface configuration mode to define the first interface.
Step 3	ACE_1/Admin(config-if)# switchport access vlan <i>vlan_ID</i>	Assigns an access VLAN to the interface. When this is a new VLAN, the VLAN interface is automatically created.
Step 4	ACE_1/Admin(config-if)# no shutdown	Enables the first interface.
Step 5	ACE_1/Admin(config)# interface gigabitEthernet <i>slot_number / port_number</i>	Defines a second interface.
Step 6	ACE_1/Admin(config-if)# speed 1000	Assigns a speed of 1000Mbps to the interface. (Required only for the redundant ACE configuration).
Step 7	ACE_1/Admin(config-if)# duplex full	Assigns full-duplex mode to the interface. (Required only for the redundant ACE configuration).

	Command	Purpose
Step 8	ACE_1/Admin(config-if) # carrier-delay {down milliseconds [up milliseconds] up milliseconds [down milliseconds]}	Delays the processing of hardware link down and link up notifications. Delay values are in ms. (Required only for the redundant ACE configuration).
Step 9	ACE_1/Admin(config-if) # qos trust cos	Sets the trusted state of an interface by defining which packet classifications the interface can carry. Definable classifications are CoS, ToS, and DSCP. (Required only for the redundant ACE configuration).
Step 10	ACE_1/Admin(config-if) # switchport trunk native vlan vlan_ID	Assigns a native trunk VLAN to the interface for untagged traffic. (Required for redundant ACE configurations.)
Step 11	ACE_1/Admin(config-if) # switchport trunk allowed vlan vlan_ID	Assigns a VLAN to the interface that can receive and transmit traffic on the trunk. You can define multiple VLANs on this trunk. In redundant ACE configurations, you define a fault-tolerant VLAN to provide a communication path for the heartbeat between the redundant ACE pair, in addition to a native VLAN. (Required for redundant ACE configurations.)
Step 12	ACE_1/Admin(config-if) # no shutdown	Enables the interface.
Step 13	ACE_1/Admin(config) # interface gigabitEthernet slot_number / port_number	Enters interface configuration mode to define the third interface.
Step 14	ACE_1/Admin(config-if) # switchport access vlan vlan_ID	Assigns an ACE access VLAN to the interface.
Step 15	ACE_1/Admin(config-if) # no shutdown	Enables the interface. Note Repeat steps 13 through 15 to define the fourth interface.
Step 16	ACE_1/Admin(config-if) # exit	Exits interface configuration mode.

Non-Redundant Configuration

The following example shows how to configure and enable port 1 as the inside interface and port 2 as the outside interface for a non-redundant ACE configuration:

Interfaces 3 and 4 are not configured or enabled in this configuration and instead are shut down.

```
ACE_1/Admin# config
ACE_1/Admin(config) # interface gigabitEthernet 1/1
ACE_1/Admin(config-if) # switchport access vlan 350
ACE_1/Admin(config-if) # no shutdown

ACE_1/Admin(config) # interface gigabitEthernet 1/2
ACE_1/Admin(config-if) # switchport access vlan 340
ACE_1/Admin(config-if) # no shutdown

ACE_1/Admin(config) # interface gigabitEthernet 1/3
ACE_1/Admin(config-if) # shutdown

ACE_1/Admin(config) # interface gigabitEthernet 1/4
ACE_1/Admin(config-if) # shutdown
ACE_1/Admin(config-if) # exit
```

Redundant Configuration

The following example shows how to configure port 1 as the inside interface and port 2 as the outside trunk interface and ports 3 and 4 as access interfaces in a redundant ACE configuration:

```
ACE_1/Admin# config
ACE_1/Admin(config)# interface gigabitEthernet 1/1
ACE_1/Admin(config-if)# switchport access vlan 350
ACE_1/Admin(config-if)# no shutdown

ACE_1/Admin(config)# interface gigabitEthernet 1/2
ACE_1/Admin(config-if)# speed 1000
ACE_1/Admin(config-if)# duplex full
ACE_1/Admin(config-if)# carrier-delay down 30 up 30
ACE_1/Admin(config-if)# switchport trunk native vlan 340
ACE_1/Admin(config-if)# switchport trunk allowed vlan 340, 999
ACE_1/Admin(config-if)# no shutdown

ACE_1/Admin(config)# interface gigabitEthernet 1/3
ACE_1/Admin(config-if)# switchport access vlan 390
ACE_1/Admin(config-if)# no shutdown

ACE_1/Admin(config)# interface gigabitEthernet 1/4
ACE_1/Admin(config-if)# switchport access vlan 410
ACE_1/Admin(config-if)# no shutdown
ACE_1/Admin(config-if)# exit
```

Configuring Real Servers

Configure a real server for each physical administration and call engine server in the cluster.

To configure a real server, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# rserver <i>name</i>	Enters real server configuration mode for the specified real server.
Step 2	ACE_1/Admin(config-rserver-host)# ip address <i>ip_address</i>	Configures the IP address for the real server.
Step 3	ACE_1/Admin(config-rserver-host)# inservice	Places the real server in-service.
Step 4	ACE_1/Admin(config-rserver-host)# exit	Exits real server configuration mode.

The following example shows how to configure the administration real servers:

```
ACE_1/Admin(config)# rserver CTX-ADMIN-1
ACE_1/Admin(config-rserver-host)# ip address 10.22.139.123
ACE_1/Admin(config-rserver-host)# inservice
ACE_1/Admin(config-rserver-host)# exit
ACE_1/Admin(config)# rserver CTX-ADMIN-2
ACE_1/Admin(config-rserver-host)# ip address 10.22.139.124
ACE_1/Admin(config-rserver-host)# inservice
ACE_1/Admin(config-rserver-host)# exit
```

The following example shows how to configure the call engine real servers:

```
ACE_1/Admin(config)# rserver SIPE-1
ACE_1/Admin(config-rserver-host)# ip address 10.22.139.125
ACE_1/Admin(config-rserver-host)# inservice
ACE_1/Admin(config-rserver-host)# exit
ACE_1/Admin(config)# rserver SIPE-2
ACE_1/Admin(config-rserver-host)# ip address 10.22.139.126
ACE_1/Admin(config-rserver-host)# inservice
ACE_1/Admin(config-rserver-host)# exit
ACE_1/Admin(config)#
```

Configuring Access Control Lists

Access control lists (ACLs) allow you to filter incoming or outgoing traffic on an interface based on configurable criteria (such as protocol type or IP address ranges).

For the Cisco TelePresence Exchange System, configure an ACL to permit all IP traffic from any source address to any destination address. To create the ACL, enter the following command in configuration mode:

```
ACE_1/Admin(config)# access-list ALL line 8 extended permit ip any any
```

Configuring Health Probes

You can configure health probes to monitor the health of the Cisco TelePresence Exchange System server cluster. The ACE appliance periodically sends a probe message to each server and evaluates the response to determine the state of the server.

The following sections describe the health probes that you can configure for the server cluster:

- [Configuring an HTTP Health Probe, page 15-8](#)
- [Configuring a SIP Health Probe, page 15-9](#)

Configuring an HTTP Health Probe

You can configure HTTP health probes to monitor the IVR application on the call engines and the administration console on the administration servers.

To configure an HTTP health probe, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# probe http <i>probe_name</i>	Creates an HTTP probe with the specified name and enters HTTP probe configuration mode.
Step 2	ACE_1/Admin(config-probe-http)# port <i>port-number</i>	Configures the destination port number to use for the probe.
Step 3	ACE_1/Admin(config-probe-http)# interval <i>seconds</i>	Configures the time interval between probes (in seconds). The default value is 15 seconds.
Step 4	ACE_1/Admin(config-probe-http)# faildetect <i>retry-count</i>	Configures the number of consecutive failed probes before the server state is marked as failed. The default value is 2.
Step 5	ACE_1/Admin(config-probe-http)# passdetect interval <i>seconds</i>	Configures the time interval (in seconds) between sending probes to a failed server.

	Command	Purpose
Step 6	ACE_1/Admin(config-probe-http) # request method get [<i>url</i> <i>url_string</i>]	Configures the probe to use the HTTP GET method to get the page for the specified universal resource locator (URL). The default value for the URL is forward slash. Note Use Ctrl+V to escape special characters in the CLI, such as question mark (?). For example, type HttpPing[Ctrl+V]?healthCheck to enter HttpPing?healthCheck in the configuration.
Step 7	ACE_1/Admin(config-probe-http) # expect status <i>min_number</i> <i>max_number</i>	Configures the range (minimum and maximum values) of HTTP status codes that an ACE expects in the probe response. To configure a single status code, enter the same number for <i>min_value</i> and <i>max_value</i> .
Step 8	ACE_1/Admin(config-probe-http) # open <i>timeout</i>	Configures the time interval (in seconds) to wait for a TCP connection to be established. By default, the ACE waits 10 seconds to open and establish the connection with the server.

The following example shows how to configure the HTTP health probe for the administration server:

```
ACE_1/Admin(config) # probe http ctx-admin
ACE_1/Admin(config-probe-http) # port 8080
ACE_1/Admin(config-probe-http) # interval 2
ACE_1/Admin(config-probe-http) # faildetect 2
ACE_1/Admin(config-probe-http) # passdetect interval 4
ACE_1/Admin(config-probe-http) # request method get url /ctxadmin/ping
ACE_1/Admin(config-probe-http) # expect status 200 200
ACE_1/Admin(config-probe-http) # open 1
```

The following example shows how to configure the HTTP health probe for the IVR application on the call engines:

```
ACE_1/Admin(config) # probe http IVR
ACE_1/Admin(config-probe-http) # port 8080
ACE_1/Admin(config-probe-http) # interval 5
ACE_1/Admin(config-probe-http) # faildetect 2
ACE_1/Admin(config-probe-http) # passdetect interval 4
ACE_1/Admin(config-probe-http) # request method get url /MeetMePing/HttpPing?healthCheck=true
ACE_1/Admin(config-probe-http) # expect status 200 200
ACE_1/Admin(config-probe-http) # open 1
```

Configuring a SIP Health Probe

You can define SIP (UDP and TCP) probes to monitor the health of the call processing service.

To configure a SIP health probe, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config) # probe sip { <i>udp</i> <i>tcp</i> } <i>name</i>	Enter the type of SIP probe (UDP or TCP) and the name of the probe.
Step 2	ACE_1/Admin(config-probe-sip) # interval <i>seconds</i>	Configures the time interval between probes (in seconds). The default value is 15 seconds.
Step 3	ACE_1/Admin(config-probe-sip) # faildetect <i>retry-count</i>	Configures the number of consecutive failed probes before the server state is marked as failed. The default value is 2.

	Command	Purpose
Step 4	ACE_1/Admin(config-probe-sip)# passdetect interval <i>seconds</i>	Configures the time interval (in seconds) between sending probes to a failed server, or the number of consecutive successful probe responses before marking the server state as active.
Step 5	ACE_1/Admin(config-probe-sip)# passdetect count <i>number</i>	Configures the number of consecutive successful probe responses before marking the server state as active.
Step 6	ACE_1/Admin(config-probe-sip)# expect status <i>min_number</i> <i>max_number</i>	Configures the range (minimum and maximum values) of status codes that an ACE expects in the probe response. To configure a single status code, enter the same number for min_value and max_value.
Step 7	ACE_1/Admin(config-probe-sip)# open <i>timeout</i>	Configures the time interval (in seconds) to wait for a TCP connection to be established. By default, the ACE waits 10 seconds to open and establish the connection with the server.

The following example shows how to configure a SIP UDP probe:

```
ACE_1/Admin(config)# probe sip udp SIP-OPTION
ACE_1/Admin(config-probe-sip)# interval 2
ACE_1/Admin(config-probe-sip)# faildetect 1
ACE_1/Admin(config-probe-sip)# passdetect interval 4
ACE_1/Admin(config-probe-sip)# passdetect count 2
ACE_1/Admin(config-probe-sip)# expect status 200 200
ACE_1/Admin(config-probe-sip)# open 1
```

The following example shows how to configure a SIP TCP probe:

```
ACE_1/Admin(config)# probe sip tcp SIP-TCP-OPTION
ACE_1/Admin(config-probe-sip)# interval 2
ACE_1/Admin(config-probe-sip)# faildetect 1
ACE_1/Admin(config-probe-sip)# passdetect interval 4
ACE_1/Admin(config-probe-sip)# passdetect count 2
ACE_1/Admin(config-probe-http)# expect status 200 200
ACE_1/Admin(config-probe-http)# open 1
```

Creating Server Farms

A server farm is a connected group of real servers that perform the same function. You must define at least two real servers to include in a server farm.

To create a server farm and define real server membership for those server farms, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# serverfarm host <i>name</i>	Creates the server farm and enters the server farm configuration mode for the specified server farm.
Step 2	ACE_1/Admin(config-sfarm-host) # failaction <i>purge</i>	Configures the action that is taken if a real server in the server farm goes down. Purge indicates that ACE removes the connection to the real server and sends a reset (RST) to the server.
Step 3	ACE_1/Admin(config-sfarm-host) # probe <i>name</i>	Specifies the probe to use for monitoring the health of real servers in this server farm.
Step 4	ACE_1/Admin(config-sfarm-host) # rserver <i>name</i>	Associates the specified real server as a member of this server farm.

	Command	Purpose
Step 5	ACE_1/Admin(config-sfarm-host-rs)# inservice	Places the real server in service.
Step 6	ACE_1/Admin(config-sfarm-host-rs)# exit	Exits server farm real-server configuration mode
Step 7	ACE_1/Admin(config-sfarm-host)# exit	Exits server farm configuration mode.

For the Cisco TelePresence Exchange System:

- Create a server farm for the administration console service and associate at least two administration servers (on which the administration console runs) to the server farm.
- Create a server farm for the IVR application and associate at least two call engine servers (on which the IVR application runs) to the server farm.
- Create a server farm for the SIP (call processing) service and associate at least two call engine servers (on which the SIP service runs) to the server farm.

Real servers can belong to multiple server farms. Although the SIP service and IVR application both run on the call engine (real server), you define a separate server farm for each service because the health probes and the session persistence criteria are different for the two services.

The following example shows how to configure a server farm for the administration console on the administration servers:

```
ACE_1/Admin(config)# serverfarm host CTX-ADMIN
ACE_1/Admin(config-sfarm-host)# failaction purge
ACE_1/Admin(config-sfarm-host)# probe ctx-admin
ACE_1/Admin(config-sfarm-host)# rserver CTX-ADMIN-1
ACE_1/Admin(config-sfarm-host-rs)# inservice
ACE_1/Admin(config-sfarm-host-rs)# exit
ACE_1/Admin(config-sfarm-host)# rserver CTX-ADMIN-2
ACE_1/Admin(config-sfarm-host-rs)# inservice
ACE_1/Admin(config-sfarm-host-rs)# exit
```

The following example shows how to configure a server farm for the IVR application on the call engine servers:

```
ACE_1/Admin(config)# serverfarm host IVR_SERVERS
ACE_1/Admin(config-sfarm-host)# failaction purge
ACE_1/Admin(config-sfarm-host)# probe IVR
ACE_1/Admin(config-sfarm-host)# rserver SIPE-1
ACE_1/Admin(config-sfarm-host-rs)# inservice
ACE_1/Admin(config-sfarm-host-rs)# exit
ACE_1/Admin(config-sfarm-host)# rserver SIPE-2
ACE_1/Admin(config-sfarm-host-rs)# inservice
ACE_1/Admin(config-sfarm-host-rs)# exit
```

The following example shows how to create a server farm for the SIP service on the call engine servers:

```
ACE_1/Admin(config)# serverfarm host SIP_FARM
ACE_1/Admin(config-sfarm-host)# failaction reassign
ACE_1/Admin(config-sfarm-host)# probe SIP_UDP-OPTION
ACE_1/Admin(config-sfarm-host)# rserver SIPE-1
ACE_1/Admin(config-sfarm-host-rs)# inservice
ACE_1/Admin(config-sfarm-host-rs)# exit
ACE_1/Admin(config-sfarm-host)# rserver SIPE-2
ACE_1/Admin(config-sfarm-host-rs)# inservice
ACE_1/Admin(config-sfarm-host-rs)# exit
```

Configuring Session Persistence

Session persistence ensures that the system directs all messages for a session to the same real server. Session persistence is also known as stickiness.

On the ACE, you configure session persistence by defining sticky groups. The sticky group defines how to identify sessions based on the value of specific fields within the incoming messages.

For the Cisco TelePresence Exchange System, configure a sticky group for each of the server farms.

This section addresses sticky group configuration and includes the following topics:

- [Creating SIP Header Sticky Groups, page 15-12](#)
- [Creating HTTP Cookie Sticky Groups, page 15-12](#)
- [Creating HTTP Header Sticky Groups, page 15-13](#)

Creating SIP Header Sticky Groups

The SIP header sticky group identifies sessions based on fields in the SIP message header.

For the call processing service, create a sticky group based on the SIP Call ID field. All messages with the same call ID will be directed to the same real server.

To create a SIP header sticky group, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# sticky sip-header Call-ID <i>name2</i>	Creates a SIP header sticky group, which recognizes sessions based on the Call ID field in the header.
Step 2	ACE_1/Admin(config-sticky-header)# timeout <i>minutes</i>	Configures a timeout value for the sticky group. The value is the number of minutes that the ACE retains the sticky information for each client session. The default value is 1440 minutes.
Step 3	ACE_1/Admin(config-sticky-header)# serverfarm <i>name1</i>	Associates a server farm with this sticky group.

The following example shows how to create a sticky group that uses the SIP call ID field to identify sessions:

```
ACE_1/Admin(config)# sticky sip-header Call-ID SIP_FARM
ACE_1/Admin(config-sticky-header)# timeout 5
ACE_1/Admin(config-sticky-cookie)# serverfarm SIP_FARM
```

Creating HTTP Cookie Sticky Groups

The HTTP cookie sticky group identifies sessions based on the cookie value in the HTTP header. The system directs all messages with the same cookie value to the same server. The ACE can insert a cookie into the server response for the first client message. The ACE uses this cookie value to identify the session, and then forwards this same cookie value in all subsequent client messages.

To create the HTTP cookie sticky group, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# sticky http-cookie name1 name2	Creates an HTTP cookie sticky group, which recognizes sessions based on the cookie value (name1) in the HTTP header. Name2 is the name of the sticky group.
Step 2	ACE_1/Admin(config-sticky-cookie)# cookie insert browser-expire name	Enables cookie insertion. The ACE inserts a session cookie in the server response to the client, to ensure stickiness to the same server. Browser-expire allows the client browser to expire the cookie after the session ends.
Step 3	ACE_1/Admin(config-sticky-cookie)# serverfarm name	Associates the sticky group with the specified SIP server farm.

The following example shows how to configure an HTTP cookie sticky group for the administration console:

```
ACE_1/Admin(config)# sticky http-cookie ctx_1 WEB_STICKY
ACE_1/Admin(config-sticky-cookie)# cookie insert browser-expire
ACE_1/Admin(config-sticky-cookie)# serverfarm CTX-ADMIN
```

Creating HTTP Header Sticky Groups

The HTTP header sticky group identifies sessions based on the value of fields in the HTTP header. You can configure the sticky group to use a specific portion of the header.

To create an HTTP header sticky group, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# sticky http-header name1 name2	Creates an HTTP header sticky group. Name1 is the HTTP header name. Name2 is the name of the sticky group.
Step 2	ACE_1/Admin(config-sticky-header)# header offset number1 [length number2]	The header offset specifies how many bytes to ignore (starting from the first byte of the header). Length specifies the number of bytes of header that the ACE uses to identify the session.
Step 3	ACE_1/Admin(config-sticky-header)# serverfarm name	Associates the sticky group with the specified SIP server farm.

The following example shows how to define an HTTP header sticky group for the IVR application:

```
ACE_1/Admin(config)# sticky http-header Host IVR_STICKY
ACE_1/Admin(config-sticky-header)# header offset 0 length 0
ACE_1/Admin(config-sticky-header)# serverfarm IVR_SERVERS
```

Configuring Class Maps

A Layer 3 and Layer 4 class map classifies traffic based on the Layer 3 and Layer 4 information (such as IP address, IP protocol, or port number). A Layer 7 class map classifies traffic based on fields in the upper-layer protocols (such as HTTP or SIP). A management class map classifies traffic based on management protocols (such as ICMP, SNMP, SSH, or Telnet).

This section addresses configuration for class maps and includes the following topics:

- [Configuring Layer 7 HTTP Class Maps, page 15-14](#)
- [Configuring Layer 7 SIP Class Maps, page 15-15](#)
- [Configuring Layer 3 and Layer 4 Class Maps, page 15-15](#)
- [Configuring Management Class Maps, page 15-16](#)

Configuring Layer 7 HTTP Class Maps

To create a Layer 7 class map for server load balancing based on the URL value in the HTTP header, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# class-map type http loadbalance match-any <i>map_name</i>	Creates a Layer 7 class map for HTTP server load balancing. The match-any keyword indicates that a message matches this class map if any of the configured match statements are true. The name has a maximum of 64 alphanumeric characters and must not contain spaces.
Step 2	ACE_1/Admin(config-cmap-http- lb)# [<i>line_number</i>] match http url expression [method <i>name</i>]	Configures a URL (or portion of a URL) to match when making the load-balancing decision. The optional method keyword specifies the HTTP 1.1 method name to include in the match.
Step 3	ACE_1/Admin(config-cmap-http- lb) exit	Exits the class map HTTP load balancing configuration mode.

The following example shows how to create a class map for Layer 7 load balancing of HTTP traffic to the IVR application:

```
ACE_1/Admin(config)# class-map type http loadbalance match-any IVR
ACE_1/Admin(config-cmap-http-lb)# match protocol http url /MeetMeIVR/.*
ACE_1/Admin(config-cmap-http-lb)# exit
```

Configuring Layer 7 SIP Class Maps

To create a Layer 7 SIP class map for load balancing, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# class-map type sip loadbalance match-any map_name	Creates a Layer 7 class map for load balancing SIP traffic. The match-any keyword indicates that a message matches this class map if any of the configured match statements are true.
Step 2	ACE_1/Admin(config-cmap-sip-lb)# match source-address ip_address [mask]	Specifies the source IP address (with optional mask) to match for this class map.
Step 3	ACE_1/Admin(config-cmap-sip-lb)# match sip header header_name header-value expression	Configures a value (or set of values) in the specified SIP header to match for this class map. Expression uses regular expression syntax.
Step 4	ACE_1/Admin(config-cmap-sip-lb)# exit	Exits SIP class map load balancing configuration mode.

The following example shows how to configure a SIP load-balancing class map to match traffic with any value of Call-ID:

```
ACE_1/Admin(config)# class-map type sip loadbalance match-any SIP-L7
ACE_1/Admin(config-cmap-sip-lb)# match sip header Call-ID header-value ".*"
```

Configuring Layer 3 and Layer 4 Class Maps

To create a Layer 3 and Layer 4 class map, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# class-map match-any map_name	Creates a Layer 3 and Layer 4 class map.
Step 2	ACE_1/Admin(config-cmap)# match virtual-address vip_address { tcp udp } eq port_number	Configures the ACE virtual IP address, protocol, and port number to match for this class map.
Step 3	ACE_1/Admin(config-cmap)# match port { tcp udp } eq sip	Configures the TCP or UDP port number to match for this class map. SIP has the value 5060.

The following example shows how to create a Layer 3 and 4 class map that matches the IVR traffic arriving at the virtual IP address:

```
ACE_1/Admin(config)# class-map match-any IVR-VIP
ACE_1/Admin(config-cmap)# match virtual-address 10.22.139.103 tcp eq 8080
```

The following example shows how to create a Layer 3 and 4 class map for all SIP UDP traffic:

```
ACE_1/Admin(config)# class-map match-any SIP_UDP_CLASS
ACE_1/Admin(config-cmap)# match port udp eq sip
ACE_1/Admin(config-cmap)# exit
```

The following example creates a Layer 3 and 4 class map for all SIP traffic:

```
ACE_1/Admin(config)# class-map match-any SIP_TRAFFIC
ACE_1/Admin(config-cmap)# match port udp eq sip
ACE_1/Admin(config-cmap)# match port tcp eq sip
ACE_1/Admin(config-cmap)# exit
```

The following example shows how to create a Layer 3 and 4 class map to match all SIP traffic (UDP and TCP) arriving at the specified virtual IP address:

```
ACE_1/Admin(config-if)# class-map match-any SIP_VIP_CLASS
ACE_1/Admin(config-cmap-mgmt)# match virtual-address 10.22.139.103 udp eq sip
ACE_1/Admin(config-cmap-mgmt)# match virtual-address 10.22.139.103 tcp eq sip
```

Configuring Management Class Maps

To allow remote network traffic to pass through the ACE, you must create a management traffic policy, which requires a management class map.

To configure a management class map, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# class-map type management match-any <i>map_name</i>	Creates a Layer 3 and Layer 4 class map for management traffic. The match-any keyword indicates that a message matches this class map when any of the configured match statements are true.
Step 2	ACE_1/Admin(config-cmap-mgmt) # match protocol <i>protocol_name</i> [any source_address <i>ip_address</i> <i>mask</i>]	Configures a management protocol to match for this class map. You can configure the match statement to match any source address or configure a specific source IP address and mask.
Step 3	ACE_1/Admin(config-cmap-mgmt) exit	Exits class map HTTP load balancing configuration mode.

The following example shows how to create a management-type class map that matches traffic from any source that matches any of the specified protocols:

```
ACE_1/Admin(config-if)# class-map type management match-any REMOTE_ACCESS
ACE_1/Admin(config-cmap-mgmt)# match protocol xml-https any
ACE_1/Admin(config-cmap-mgmt)# match protocol icmp any
ACE_1/Admin(config-cmap-mgmt)# match protocol telnet any
ACE_1/Admin(config-cmap-mgmt)# match protocol ssh any
ACE_1/Admin(config-cmap-mgmt)# match protocol http any
ACE_1/Admin(config-cmap-mgmt)# match protocol https any
ACE_1/Admin(config-cmap-mgmt)# match protocol snmp any
ACE_1/Admin(config-cmap-mgmt)# exit
```

Configuring Policy Maps

A policy map defines a series of actions that you want to apply to traffic that matches one or more of the associated class maps.

This section addresses policy map configuration and includes the following topics:

- [Configuring Management Policy Maps, page 15-17](#)
- [Configuring Layer 7 Load Balancing Policy Maps, page 15-17](#)

- [Configuring Layer 4 Policy Maps, page 15-18](#)

Configuring Management Policy Maps

A management policy map specifies policy for network management traffic that is received by the ACE.

To create a management policy map, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# policy-map type management first-match match-any map_name	Creates a policy map for management traffic.
Step 2	ACE_1/Admin(config-pmap-mgmt) # class name1	Associates a class map with this policy map. You can associate multiple class maps with a policy map.
Step 3	ACE_1/Admin(config-pmap-mgmt-c) # permit deny	Specifies whether to permit or deny the traffic that matches the class map.
Step 4	ACE_1/Admin(config-pmap-mgmt-c) exit	Exits management policy map configuration mode.

The following example shows how to create a policy map to allow remote management access to the Cisco TelePresence Exchange System:

```
ACE_1/Admin(config)# policy-map type management first-match REMOTE_MGMT_ALLOW_POLICY
ACE_1/Admin(config-pmap-mgmt)# class REMOTE_ACCESS
ACE_1/Admin(config-pmap-mgmt-c)# permit
```

Configuring Layer 7 Load Balancing Policy Maps

A Layer 7 load balancing policy map specifies the traffic (based on a class map) to send to each server farm for load balancing. The order of classes in the policy map is significant, as traffic is sent to the server farm that is associated with the first matching traffic class in the policy.

To create a Layer 7 load balancing policy map, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# policy-map type loadbalance first-match match-any map_name	Creates a Layer 7 load-balancing policy map for HTTP traffic.
Step 2	ACE_1/Admin(config-pmap-lb)# class name1	Associates a class map with this policy map. You can associate multiple class maps with a policy map.
Step 3	ACE_1/Admin(config-pmap-lb-c) # sticky-serverfarm name	Specifies that the traffic that matches this class is load-balanced to the specified sticky server farm.
Step 4	ACE_1/Admin(config-pmap-lb-c) exit	Exits class map HTTP load balancing configuration mode.

The following example shows how to create a Layer 7 policy map to load balance IVR traffic by using the IVR_STICKY server farm. The system load balances all other traffic by using the WEB_STICKY server farm:

```
ACE_1/Admin(config)# policy-map type loadbalance first-match VXML-LB
ACE_1/Admin(config-pmap-lb)# class IVR
ACE_1/Admin(config-pmap-lb-c)# sticky-serverfarm IVR_STICKY
ACE_1/Admin(config-pmap-lb-c)# class class-default
ACE_1/Admin(config-pmap-lb-c)# sticky-serverfarm WEB-STICKY
```

**Note**

Class-default is a pre-configured class map that matches all traffic.

The following example shows how to create a policy map to load balance SIP traffic across the SIP_FARM server farm:

```
ACE_1/Admin(config)# policy-map type loadbalance sip first-match L7-POLICY
ACE_1/Admin(config-pmap-lb)# class SIP-L7
ACE_1/Admin(config-pmap-lb-c)# sticky-serverfarm SIP_FARM
```

Configuring Layer 4 Policy Maps

To create a Layer 4 policy map, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# policy-map multi-match <i>map_name</i>	Creates a Layer 4 load balancing policy map. Multi-match allows the inclusion of multiple network traffic-related actions in the same policy map.
Step 2	ACE_1/Admin(config-pmap)# class <i>name1</i>	Associates a class map with this policy map.
Step 3	ACE_1/Admin(config-pmap-c)# loadbalance vip inservice	Enables the VIP for server load-balancing.
Step 4	ACE_1/Admin(config-pmap-c) loadbalance policy <i>name</i>	Specifies a Layer 7 load-balancing policy map to associate with this Layer 4 policy map.
Step 5	ACE_1/Admin(config-pmap-c)# appl-parameter sip advanced-options syslog	Associates a SIP parameter map with this policy.
Step 6	ACE_1/Admin(config-pmap-c)# loadbalance vip icmp-reply	Enables the VIP to respond to ICMP ECHO requests.
Step 7	ACE_1/Admin(config-pmap-c)# connection advanced-options	Associates a connection parameter map with this policy.
Step 8	ACE_1/Admin(config-pmap-c)# inspect sip	Enables packet inspection of the SIP packets.
Step 9	ACE_1/Admin(config-pmap-c) exit	Exits policy map configuration mode.

The following example shows how to create a Layer 4 policy map for incoming HTTP traffic on a VIP (specified in the class) and apply a Layer 7 load balancing policy:

```
ACE_1/Admin(config)# policy-map multi-match IVR_LB
ACE_1/Admin(config-pmap)# class IVR-VIP
ACE_1/Admin(config-pmap-c)# loadbalance vip inservice
ACE_1/Admin(config-pmap-c)# loadbalance policy VXML-LB
ACE_1/Admin(config-pmap-c)# loadbalance vip icmp-reply active
```

The following example shows how to create a Layer 4 policy map for incoming SIP traffic on a VIP (specified in the class) and apply a Layer 7 load balancing policy:

```
ACE_1/Admin(config)# policy-map multi-match L4-POLICY
ACE_1/Admin(config-pmap)# class SIP_VIP_CLASS
ACE_1/Admin(config-pmap-c)# loadbalance vip inservice
ACE_1/Admin(config-pmap-c)# loadbalance policy L7-POLICY
ACE_1/Admin(config-pmap-c)# loadbalance vip icmp-reply active
ACE_1/Admin(config-pmap-c)# appl-parameter sip advanced-options syslog
ACE_1/Admin(config-pmap-c)# inspect sip
```

The following example shows how to create a Layer 4 policy map to enable traffic inspection for all SIP traffic:

```
ACE_1/Admin(config)# policy-map multi-match SIP_INSPECT
ACE_1/Admin(config-pmap)# class SIP_TRAFFIC
ACE_1/Admin(config-pmap-c)# inspect sip
```

The following example shows how to apply UDP connection timeout settings for all SIP UDP traffic:

```
ACE_1/Admin(config)# policy-map multi-match UDP_TIMEOUT
ACE_1/Admin(config-pmap)# class SIP_UDP_CLASS
ACE_1/Admin(config-pmap-c)# connection advanced-options UDP-Timeout
```

Configuring VLAN Interfaces

Each Gigabit Ethernet port must be associated with a VLAN. For redundant configurations of the Cisco TelePresence Exchange System using the ACE, you must also define a fault-tolerant (FT) VLAN. The redundant ACE pair constantly communicate over the dedicated FT VLAN to determine the operating status of each appliance. The standby member uses the heartbeat packet to monitor the health of the active member. The active member uses the heartbeat packet to monitor the health of the standby member. Each ACE peer can also contain one or more FT groups. Each FT group consists of two members: one active context and one standby context. An FT group has a unique group ID that you assign.

You also must configure a different IP address within the same subnet on each appliance for the FT VLAN.



Note

Do not use this dedicated VLAN for any other network traffic, including HSRP and data.

For multiple contexts, the FT VLAN resides in the system configuration file. Each FT VLAN on the ACE has one unique MAC address that is associated with it. The ACE uses these device MAC addresses as the source or destination MACs for sending or receiving redundancy protocol state and configuration replication packets.



Note

An ACE appliance and an ACE module operating as peers cannot operate as redundant pairs for the Cisco TelePresence Exchange System. System redundancy must employ the same ACE device type and software release.

To configure a VLAN interface, do the following task:

	Command	Purpose
Step 1	ACE_1/Admin(config)# interface vlan <i>vlan_number</i>	Enters configuration mode for the specified VLAN interface.
Step 2	ACE_1/Admin(config-if)# ip address <i>ip-address mask</i>	Configures the IP address and mask for the VLAN interface.
Step 3	ACE_1/Admin(config-if)# alias ip address <i>ip-address mask</i>	Defines the default route when a redundant ACE configuration exists. (Required only for redundant ACE configurations).
Step 4	ACE_1/Admin(config-if)# peer ip address <i>ip-address mask</i>	Defines the IP address and mask for the fault tolerant VLAN interface. (Required only for redundant ACE configurations).
Step 5	ACE_1/Admin(config-if)# normalization send-reset	Enables sending a RST to the peer so it can reset its TCP connections for any non-SYN packets that are a connection miss.
Step 6	ACE_1/Admin(config-if)# access-group { input output } <i>name</i>	Associates the specified access group list (ACL) with the VLAN. The ACL will be applied to all incoming traffic (input) or outgoing traffic (output).
Step 7	ACE_1/Admin(config)# service-policy { input output } <i>name</i>	Associates the specified service policy with the VLAN. The service policy will be applied to all incoming traffic (input) or outgoing traffic (output). (Not configured on fault tolerant VLANs).
Step 8	ACE_1/Admin(config)# ft interface <i>interface_name</i>	Creates a fault tolerant VLAN to provide a communication path for updates from the active ACE to its peer (standby). (Required only for redundant ACE configurations).
Step 9	ACE_1/Admin(config-ft-intf)# ip address <i>ip-address mask</i>	Configures the IP address and mask for the VLAN interface. (Required only for redundant ACE configurations).
Step 10	ACE_1/Admin(config-ft-intf)# peer ip address <i>ip-address</i> <i>mask</i>	Specifies the IP address and mask of the ACE peer. (Required only for redundant ACE configurations).
Step 11	ACE_1/Admin(config-ft-intf)# no shutdown	Enables the VLAN interface.
Step 12	ACE_1/Admin(config-ft-intf)# exit	Exits fault tolerant interface configuration mode.
Step 13	ACE_1/Admin(config)# ft peer <i>peer_id</i>	Configures an ACE local redundancy peer.
Step 14	ACE_1/Admin(config-ft-peer)# ft-interface vlan <i>vlan_id</i>	Associates the fault-tolerant (FT) VLAN with the peer. Note This VLAN ID must also be configured on the switch. Only a layer 2 definition is required.
Step 15	ACE_1/Admin(config-ft-peer)# heartbeat interval <i>frequency</i> heartbeat count <i>number</i>	Configures the heartbeat interval and count for the fault-tolerant peer. Values are in milliseconds (ms).
Step 16	ACE_1/Admin(config-ft-peer)# query-interface vlan <i>vlan_id</i>	Defines the actual (routable) VLAN and interface that the fault-tolerant peer uses to send health-check and replication messages. A query interface allows the standby ACE to determine whether the active ACE is down or if there is a connectivity problem with the FT VLAN. A query interface helps prevent two redundant contexts from becoming active at the same time for the same FT group.

	Command	Purpose
Step 17	ACE_1/Admin(config-ft-peer) # no shutdown	Enables the query interface.
Step 18	ACE_1/Admin(config-ft-peer) # exit	Exits fault-tolerant peer configuration mode.
Step 19	ACE_1/Admin(config) # ft group <i>group_id</i>	Creates a fault-tolerant group for redundancy.
Step 20	ACE_1/Admin(config-ft-group) # peer <i>peer_id</i>	Associates the peer with the fault-tolerant group.
Step 21	ACE_1/Admin(config-ft-group) # no preempt	Disables preemption on the fault-tolerant group. Preemption ensures that the group member with the higher priority always asserts itself and becomes the active member.
Step 22	ACE_1/Admin(config-ft-group) # priority <i>number</i>	Configures the priority of the active group member. Values are 1 to 255. Configure a higher priority for the group on the module on which you want the active member to initially reside.
Step 23	ACE_1/Admin(config-ft-group) # associate-context <i>name</i>	Associates a context with each fault-tolerant group. You must associate the local ACE with the fault-tolerant group. You can assign multiple contexts.
Step 24	ACE_1/Admin(config-ft-group) # inservice	Places a fault-tolerant group in service.

Non-Redundant Configuration

The following example shows how to configure VLAN 340 as the outside interface. The **service-policy** commands activate the Layer 3 and Layer 4 policies on this VLAN. The Layer 7 load-balancing policies become active because they are encapsulated in the Layer 3 and Layer 4 policies:

```
ACE_1/Admin(config) # interface vlan 340
ACE_1/Admin(config-if) # description OUTSIDE network
ACE_1/Admin(config-if) # ip address 10.22.139.102 255.255.255.240
ACE_1/Admin(config-if) # normalization send-reset
ACE_1/Admin(config-if) # access-group input ALL
ACE_1/Admin(config-if) # service-policy input remote_mgmt_allow_policy
ACE_1/Admin(config-if) # service-policy input L4-POLICY
ACE_1/Admin(config-if) # service-policy input UDP_TIMEOUT
ACE_1/Admin(config-if) # service-policy input IVR_LB
ACE_1/Admin(config-if) # no shutdown
ACE_1/Admin(config-if) # exit
```

The following example shows how to configure the VLAN 350 interface as the inside interface:

```
ACE_1/Admin(config) # interface vlan 350
ACE_1/Admin(config-if) # description INSIDE network
ACE_1/Admin(config-if) # ip address 10.22.139.113 255.255.255.240
ACE_1/Admin(config-if) # normalization send-reset
ACE_1/Admin(config-if) # access-group input ALL
ACE_1/Admin(config-if) # service-policy input remote_mgmt_allow_policy
ACE_1/Admin(config-if) # service-policy input UDP_TIMEOUT
ACE_1/Admin(config-if) # service-policy input SIP_INSPECT
ACE_1/Admin(config-if) # no shutdown
ACE_1/Admin(config-if) # exit
```

Redundant Configuration

The following example shows how to configure VLAN 340 as the outside interface to support redundancy. The **service-policy** commands activate the Layer 3 and Layer 4 policies on this VLAN. The Layer 7 load-balancing policies become activated because they are encapsulated in the Layer 3 and Layer 4 policies:

```
ACE_1/Admin(config)# interface vlan 340
ACE_1/Admin(config-if)# description OUTSIDE network
ACE_1/Admin(config-if)# ip address 10.22.139.102 255.255.255.240
ACE_1/Admin(config-if)# alias 10.22.139.108 255.255.255.240
ACE_1/Admin(config-if)# peer ip address 10.22.139.104 255.255.255.240
ACE_1/Admin(config-if)# normalization send-reset
ACE_1/Admin(config-if)# access-group input ALL
ACE_1/Admin(config-if)# service-policy input remote_mgmt_allow_policy
ACE_1/Admin(config-if)# service-policy input UDP_TIMEOUT
ACE_1/Admin(config-if)# service-policy input L4-POLICY
ACE_1/Admin(config-if)# service-policy input IVR_LB
ACE_1/Admin(config-if)# no shutdown
ACE_1/Admin(config-if)# exit
```

The following example shows how to configure the VLAN 350 interface as the inside interface:

```
ACE_1/Admin(config)# interface vlan 350
ACE_1/Admin(config-if)# description INSIDE network
ACE_1/Admin(config-if)# ip address 10.22.139.114 255.255.255.240
ACE_1/Admin(config-if)# alias 10.22.139.113 255.255.255.240
ACE_1/Admin(config-if)# peer ip address 10.22.139.117 255.255.255.240
ACE_1/Admin(config-if)# no icmp-guard
ACE_1/Admin(config-if)# normalization send-reset
ACE_1/Admin(config-if)# access-group input ALL
ACE_1/Admin(config-if)# service-policy input remote_mgmt_allow_policy
ACE_1/Admin(config-if)# service-policy input UDP_TIMEOUT
ACE_1/Admin(config-if)# no shutdown
ACE_1/Admin(config-if)# exit
```

The following example shows how to configure the fault tolerant VLAN 999 interface on the trunk outside interface:



Note The fault-tolerant VLAN does not need to be routable; however, you must define the fault-tolerant VLAN on the switch that connects to the ACE to ensure layer 2 connectivity.

```
ACE_1/Admin(config)# ft interface vlan 999
ACE_1/Admin(config-ft-intf)# ip address 10.1.1.1 255.255.255.0
ACE_1/Admin(config-ft-intf)# peer ip address 10.1.1.2 255.255.255.0
ACE_1/Admin(config-ft-intf)# no shutdown
ACE_1/Admin(config-ft-intf)# exit
ACE_1/Admin(config)# ft peer 1
ACE_1/Admin(config-ft-peer)# heartbeat interval 200
ACE_1/Admin(config-ft-peer)# heartbeat count 10
ACE_1/Admin(config-if)# ft-interface vlan 999
ACE_1/Admin(config-if)# query-interface vlan 340
ACE_1/Admin(config)# ft group 1
ACE_1/Admin(config-ft-group)# peer 1
ACE_1/Admin(config-ft-group)# no preempt
ACE_1/Admin(config-ft-group)# priority 110
ACE_1/Admin(config-ft-group)# associate-context Admin
ACE_1/Admin(config-ft-group)# inservice
```

Configuring Miscellaneous Parameters

This section describes additional ACE configurations for the Cisco TelePresence Exchange System and includes the following topics:

- [Configuring the IP Default Route, page 15-23](#)
- [Configuring UDP Connection Timeout, page 15-23](#)
- [Enabling SysLog SIP Messages, page 15-23](#)
- [Configuring the Sticky Resource Class, page 15-23](#)
- [Assigning the Admin Context to the Sticky Resource Class, page 15-24](#)

Configuring the IP Default Route

Configure the default IP route for the inside VLAN to be the ACE inside interface. This configuration ensures that all traffic originating from the Cisco TelePresence Exchange System cluster transits through the ACE.

To define the default IP route (gateway), enter the following command:

```
ACE_1/Admin(config)# ip route 0.0.0.0 0.0.0.0 10.22.139.97
```

Configuring UDP Connection Timeout

Create a connection parameter map to define the UDP inactivity timeout value:

```
parameter-map type connection name  
set timeout inactivity seconds
```

The following example shows how to create a parameter map with a timeout value of one second:

```
ACE_1/Admin(config)# parameter-map type connection UDP-timeout  
ACE_1/Admin(config-parammap-conn)# set timeout inactivity 1
```

Enabling SysLog SIP Messages

Use the **parameter-map** command to set the logging value for SIP syslogs.

The following example shows how to create a parameter map to enable logging for SIP traffic:

```
ACE_1/Admin(config)# parameter-map type sip syslog  
ACE_1/Admin(config-parammap-conn)# logging all
```

Configuring the Sticky Resource Class

Sticky groups require system resources to store information about active sessions.

Create a sticky resource class to reserve the required system resources.

You define the resource requirement as a percentage of the total available resources.

For example, you can create a sticky resource class that allows access to the ACE for no less than 20 percent of the total number of stickiness connections that the ACE appliance supports. You must configure a minimum value for sticky to allocate resources for sticky entries, because the sticky software receives no resources under the unlimited (no limit) setting. The maximum value is either the same as the minimum value (equal-to-min) or has no limit.

To configure a sticky resource class and the number of sticky entries supported, do the following task:

Step 1 To define a resource class that allows call stickiness, enter the following command:

```
ACE_1/Admin#(config)# resource-class sticky
ACE_1/Admin#(config-resource)#
```

Step 2 To define the minimum and maximum entries allowed in the sticky resource class table, enter the following commands:

```
ACE_1/Admin#(config-resource)# limit-resource all minimum 0.00 maximum unlimited
ACE_1/Admin#(config-resource)# limit-resource sticky minimum 20.00 maximum
equal-to-min
```

Assigning the Admin Context to the Sticky Resource Class

You can operate the ACE in a single context or in multiple contexts. Multiple contexts use virtualization to partition the ACE into multiple virtual devices. Each context can contain its own set of policies, interfaces, resources, and administrators.

By default, the system enables a single virtual context known as the Admin context.

Use the **member** command to associate the sticky resource class to the Admin context.

The following example shows how to assign the sticky resource class to the default Admin context:

```
ACE_1/Admin(config)# context Admin
ACE_1/Admin(config-context)# member sticky
```

Configuring ACE Logging Options

You can configure the logging severity level, which specifies the severity system messages that the ACE logs. The ACE supports eight logging levels. Severity level values are 0 to 7; the lower the level number, the more severe the error.

The ACE logs messages of the specified level and those lower. For example, if the logging severity level is 3, the ACE logs messages with a severity level of 0, 1, 2, and 3.

Table 15-1 lists the log message severity levels.

Table 15-1 Log Message Severity Levels

Level Number	Level Keyword	Description
0	emergency	System unusable. For example, the ACE has shut down and cannot restart, or the system has experienced a hardware failure.
1	alert	Immediate action needed. For example, one of the ACE subsystems is not running.
2	critical	Critical condition. For example, the ACE has encountered a critical condition that requires immediate attention.

Table 15-1 Log Message Severity Levels (continued)

Level Number	Level Keyword	Description
3	error	Error condition. For example, error messages are conveyed about software or hardware malfunctions.
4	warning	Warning condition. For example, the ACE encountered an error condition that requires attention but is not interfering with the operation of the device.
5	notification	Normal but significant condition. For example, interface up/down transitions and system restart messages are conveyed.
6	informational	Informational message only. For example, reload requests and low-process stack messages are conveyed.
7	debugging	Appears during debugging only.

For more details on ACE SysLog Messages, see the *Cisco ACE 4700 Series Appliance System Message Guide*, at http://www.cisco.com/en/US/products/ps7027/products_installation_and_configuration_guides_list.html.

To enable logging of syslog messages on the ACE, do the following task:

Step 1 To enable logging to all output locations, enter the following commands:

```
ACE_1/Admin# configure
ACE_1/Admin#(config)# logging enable
```

To stop message logging to all output locations, enter the **no logging enable** command at the configuration mode.

Step 2 To enable logging of syslog messages and to assign a security level to specify which syslog messages the system logs, do this task:

- a. To enable logging of syslog messages during a console session by using the **logging console severity_level** configuration mode command, enter the following command:

```
ACE_1/Admin#(config)# logging console 2
```

By default, the ACE does not display syslog messages during console sessions. To disable logging on the ACE, enter the **no logging console** command at the configuration mode.

- b. To identify the date and time of a syslog message by using the **logging timestamp** configuration mode command, enter the following command:

```
ACE_1/Admin#(config)# logging timestamp
```

By default, the ACE does not generate a timestamp for syslog messages.

- c. To identify the severity level of messages that are sent to the syslog server by using the **logging trap severity_level** configuration mode command, enter the following command:

```
ACE_1/Admin#(config)# logging trap 3
```

To disable logging of traps, enter the **no logging trap** command at the configuration mode.

- d. To enable logging of Simple Network Management Protocol (SNMP) messages and to set the severity level for log messages that are sent to a network management system (NMS) by using the **logging history severity_level** configuration mode command, enter the following command:

```
ACE_1/Admin#(config)# logging history 7
```

To disable logging of SNMP messages, enter the **no logging history** command at the configuration mode.

- e. To enable system logging to a local buffer and to limit the messages sent to the buffer based on severity level by using the **logging buffered severity_level** configuration mode command, enter the following command:

```
ACE_1/Admin#(config)# logging buffered 7
```

- f. To change the logging facility to a value other than the default of 20 (LOCAL4) by using the **logging facility number** configuration mode command, enter the following command:

```
ACE_1/Admin#(config)# logging facility 23
```

The number can be a value from 16 (LOCAL0) to 23 (LOCAL7).

Most UNIX systems expect messages to use facility 20. The ACE allows you to change the syslog facility type to identify the behavior of the syslog daemon (syslogd) on the host.

To reset the logging facility to the default value of 20, enter the **no logging facility** command at the configuration mode.

- g. To specify that the ACE hostname serves as the device ID within the syslog message, enter the following command:

```
ACE_1/Admin#(config)# logging device-id hostname
```

To disable use of the hostname as the device ID in the syslog message, enter the **no logging device-id** command.

- h. To specify the syslog server (host) that receives the ACE syslog messages, enter the following command:

```
ACE_1/Admin#(config)# logging host ip_address
```

For the *ip_address* variable, enter the IP address of the host that serves as the syslog server.

You do not need to specify a port for the syslog server because by default it uses a UDP port of 514.

You can use multiple logging host commands to specify additional servers to receive the syslog messages.

To disable logging of ACE syslog messages to a syslog server, enter the **no logging host ip_address**.

- i. To control the display of a specific system logging message or to change the severity level that is associated with the specified system logging message by using the **logging message syslog_id [level severity_level]** configuration mode command, enter the following commands:

```
ACE_1/Admin#(config)# logging message 111088 level 3
ACE_1/Admin#(config)# logging message 607002 level 3
ACE_1/Admin#(config)# logging message 607004 level 3
ACE_1/Admin#(config)# logging message 607005 level 3
```

To disable logging of the specified syslog message, use the **no logging message** *syslog_id* command at the configuration mode.
