

# Configuring Remote Access to the ERX System

This chapter describes how to configure remote access to your ERX system.

<b>Topic</b>	<b>Page</b>
Overview	1-2
References	1-3
Before You Configure B-RAS	1-4
Configuration Tasks	1-4
Configuring a B-RAS License	1-5
Mapping a User Domain Name to a Virtual Router	1-5
Setting Up Domain Name and Realm Name Usage	1-9
Specifying a Single Name for Users from a Domain	1-12
Configuring Authentication and Accounting Servers	1-13
Configuring Name Server Addresses	1-25
Configuring Local Address Servers	1-27
Configuring DHCP Features	1-30
Creating an IP Interface	1-34
Configuring AAA Profiles	1-37
Using VSAs for Dynamic IP Interfaces	1-42
Configuring Timeout	1-43
Limiting Active Subscribers	1-44
Notifying RADIUS of AAA Failure	1-44
Configuring the SDX Client	1-45
Setting Baselines	1-46
Monitoring Remote Access	1-47

## Overview

---

Broadband Remote Access Server (B-RAS) is an application running on your system that:

- Aggregates the output from digital subscriber line access multiplexers (DSLAMs)
- Provides user PPP sessions or IP over ATM sessions
- Enforces quality of service (QoS) policies
- Routes traffic into an ISP's backbone network

A DSLAM collects data traffic from multiple subscribers into a centralized point so that it can be uploaded to your system over an ATM connection via a DS3, OC3, E3, or OC12 link.

The system provides the logical termination for PPP sessions, as well as the interface to authentication and accounting systems.

### *B-RAS Protocol Support*

The system supports the following protocols for B-RAS services:

- PPP
- PPP over Ethernet (PPPoE)
- Bridged Ethernet
- L2TP (LAC and LNS)
- L2F (NAS)

### *B-RAS Data Flow*

The system performs several tasks for a digital subscriber line (DSL) PPP user to establish a PPP connection. This is an example of the way B-RAS data may flow:

- 1 Authenticate the subscriber using RADIUS authentication.
- 2 Assign an IP address to the PPP/IP session via RADIUS, local address pools, or DHCP.
- 3 Terminate the PPP encapsulation or tunnel a PPP session.
- 4 Provide user accounting via RADIUS.



**Note:** For information about configuring RADIUS attributes see Chapter 2, *Configuring RADIUS Attributes*.

### *Configuring IP Addresses for Remote Clients*

A remote client can obtain an IP address from one of the following:

- RADIUS server
- Local address server
- DHCP proxy client and server
- DHCP relay agent (Bridged IP only)

Each method of configuring IP addresses for remote clients is discussed in this chapter.

Refer to your RADIUS server documentation for information on how to configure a RADIUS server.

### *AAA Overview*

Collectively, authentication, authorization, and accounting are referred to as AAA. Each has an important but separate function.

- Authentication – Determines who the user is, then determines whether that user should be granted access to the network. The primary purpose is to prevent intruders from our networks. It uses a database of users and passwords.
- Authorization – Determines what the user is allowed to do. Authorization gives the network manager the ability to limit network services to different users.
- Accounting – Tracks what the user did and when they did it. Accounting can be used for an audit trail or for billing for connection time or resources used.

Central management of AAA means the information is in a single, centralized, secure database, which is much easier to administer than information distributed across numerous devices. Both RADIUS and TACACS+ protocols are client-server systems which allow effective communication of AAA information.



**Note:** For information about TACACS+, see Chapter 3, *Configuring TACACS+*.

## References

---

For more information, see:

- RFC 2865 – Remote Authentication Dial In User Service (RADIUS) (June 2000)

## Before You Configure B-RAS

---

Before you begin to configure B-RAS, you need to collect some important information. You must determine the following for the RADIUS authentication and accounting servers:

- IP addresses
- UDP port numbers
- Secret keys

## Configuration Tasks

---

Each configuration task is presented in a separate section in this chapter. Note that most of the B-RAS configuration tasks are optional.

To configure B-RAS:

- Configure a B-RAS license.
- (Optional) Map a user domain name to a virtual router. By default, all requests go through a default router.
- (Optional) Set up domain name and realm name usage.
- (Optional) Specify a single name for users from a domain.
- Configure an authentication server on the system.
- (Optional) Configure an accounting server on the system.
- (Optional) Configure DNS and WINS name server addresses.
- (Optional) Configure a local address pool for remote clients.
- (Optional) Configure one or more DHCP servers.
- Create a PPP interface on which the system can dynamically create an IP interface.
- (Optional) Use VSAs for Dynamic Interfaces.
- (Optional) Configure UDP checksums.
- (Optional) Set idle or session timeout.
- (Optional) Limit the number of active subscribers on a VR or port.
- (Optional) Set up the system to notify RADIUS if a user fails AAA.
- (Optional) Configure the Service Deployment System client.
- (Optional) Set baselines for AAA statistics or RADIUS authentication and accounting statistics.

## Configuring a B-RAS License

---

From Global Configuration mode, configure a B-RAS license:

```
host1(config)#license b-ras k3n91s6gvtj
```

You can configure up to a total of 32,000 PPP and SDX interfaces for the system when you configure it for B-RAS. However, depending on the B-RAS license you purchased, no more than 2,000, 4,000, 8,000, 16,000, or 32,000 authenticated PPP sessions can be active at any one time.



**Note:** To use a B-RAS license for 16,000 or more interfaces, your SRP module(s) must have 512MB of memory.

The license key limits only the number of active subscribers; it does not limit the command set available on the CLI. When the limit of subscribers specified by the license is exceeded, the system issues warning log messages:

```
Subscriber limit has been exceeded - please contact Juniper  
Networks to upgrade your Subscriber Management Feature  
Pack license to support additional users.
```

If the limit is further exceeded, subscribers will be denied authentication.

### **license b-ras**

- Use to specify the B-RAS license.
- The license is a unique string of up to 15 alphanumeric characters.



**Note:** Acquire the license from Juniper Networks Customer Service or your Juniper Networks sales representative.

- You can purchase licenses that allow up to 2,000, 4,000, 8,000, 16,000, or 32,000 authenticated PPP sessions.
- Example

```
host1(config)#license b-ras jwmR4k8D
```

- Use the **no** version to disable the license.

## Mapping a User Domain Name to a Virtual Router

---

You can configure RADIUS authentication, accounting, and local address pools for a specific virtual router and then map a user domain to that virtual router.

The system keeps track of the domain-name-to-virtual-router mapping. Use the **aaa domain-map** command to map a user domain to a virtual router.



**Note:** This domain name is not the NT domain sometimes found on the Dialup Networking dialog box.

When the system is configured to require authentication of a PPP user, the system checks for the appropriate user domain-name-to-virtual-router mapping. If it finds a match, the system sends a RADIUS authentication request to the RADIUS server configured for the specific virtual router.

#### *Mapping User Requests Without a Valid Domain Name*

You can create a mapping between a domain name called **default** and a specific virtual router so that the system can map user names that contain a domain name that does not have an explicit map.

If a user request is submitted with a domain name for which the system cannot find a match, the system looks for a mapping between the domain name **default** and a virtual router. If a match is found, the user's request is processed according to the RADIUS server configured for the named virtual router. If no entry is found that maps **default** to a specific virtual router, the system sends the request to the server configured on the default virtual router.

#### *Mapping User Requests Without a Configured Domain Name*

You can map a domain name called **none** to a specific virtual router so that the system can map user names that do not contain a domain name.

If a user request is submitted without a domain name, the system looks for a mapping between the domain name **none** and a virtual router. If a match is found, the user's request is processed according to the RADIUS server configured for the named virtual router. If the system does not find the domain name **none**, it checks for the domain name **default**. If no matching entries are found, the system sends the request to the server configured on the default virtual router.

#### *Preauthenticating Users*

If users have a called number associated with them, the system searches the domain map for the called number. If it finds a match, the system uses the matching domain map entry information to authenticate the user. If the system does not find a match, it searches the domain map using normal processing.



**Note:** For this feature to work, the ERX system must be acting as the L2TP Network Server (LNS).

## Redirected Authentication

Redirected authentication provides a way to offload AAA activity on the system, by providing the domain-mapping-like feature remotely on the RADIUS server. Redirected authentication works as follows:

- 1 The system sends an authentication request (in the form of a RADIUS access-request message) to the RADIUS server that is configured in the default VR.
- 2 The RADIUS server determines the user's AAA VR context and returns this information in a RADIUS response message to the system.
- 3 The system then behaves in similar fashion as if it had received the VR context from the local system domain map.



**Note:** If the default VR does not exist, authentication fails.

To maintain local control, the only VR allowed to redirect authentication is the default VR. Also, to prevent loopbacks, the redirection may occur only once to a non-default VR.

To maintain flexibility, the redirection response may include idle time or session attributes that are considered as default unless the redirected authentication server overrides them. For example, if the RADIUS server returns the VR context along with an idle timeout attribute with the value set to 20 minutes, the system uses this idle timeout value unless the RADIUS server configured in the VR context returns a different value.

Since the system supports the RADIUS User-Name attribute [1] in the RADIUS response message, the default VR RADIUS server may override the user's name (this can be a stripped name or an entirely different name). Overriding is useful for the case when the user enters a login name containing a domain name that is significant only to the RADIUS server in the default VR.

## IP Hinting

You can allocate an address before authentication of PPP sessions. This address is included in the Access-Request sent to the authentication server as an IP address hint.

### **aaa domain-map**

- Use to map a user domain name to a virtual router or a loopback interface.
- When you specify only the domain name, the command sets the mode to Domain Map Configuration.

- Example

```
host1(config)#aaa domain-map juniper.net vrouter_1
host1(config)#aaa domain-map none vrouter_all_purpose
host1(config)#aaa domain-map default vrouter_all_purpose
host1(config)#aaa domain-map 8005558934 vrouter_78
host1(config)#aaa domain-map westford.com
host1(config-domain-map)#
```

- Use the **no** version to delete the map entry.

### ***ip-hint***

- Use to preallocate an IP address for the remote B-RAS user before authenticating the remote user.
- The address is passed as a *hint* in the authentication request.
- Example

```
host1(config-domain-map)#ip-hint enable
```

- The **no** version disables the feature.

### ***loopback***

- Use to map a user domain name to a loopback interface in Domain Map Configuration mode.
- The loopback identifies the interface information to use on the local (ERX) side of the subscribers interface.
- Example

```
host1(config)#aaa domain-map westford.com
host1(config-domain-map)#loopback 2
```

- Use the **no** version to delete the entry.

### ***virtual-router***

- Use to map a user domain name to a virtual router in Domain Map Configuration mode.
- Example

```
host1(config)#aaa domain-map westford.com
host1(config-domain-map)#virtual-router vrouter
```

- Use the **no** version to delete the entry.

## Setting Up Domain Name and Realm Name Usage

---

To provide flexibility in how the system handles different types of usernames, the software lets you specify the part of a username to use as the domain name, how the domain name is designated, and how the system parses names. It also allows you to set whether or not the system strips the domain name from the username before it sends the username to the RADIUS server.

By default, the system parses usernames as follows:

```
realmName/personalName@domainName
```

The string to the left of the forward slash (/) is the realm name, and the string to the right of the @ symbol is the domain name. For example, if you have a username of juniper/jill@abc.com, juniper is the realm name and abc.com is the domain name.

The system allows you to:

- Use the realm name as the domain name.
- Use delimiters other than / to designate the realm name.
- Use delimiters other than @ to designate the domain name.
- Use either the domain or the realm as the domain name when the username contains both a realm and domain name.

To provide these features, the system allows you to specify delimiters for the domain name and realm name. You can use up to eight one-character delimiters each for domain and realm names. The system also lets you specify how it parses usernames to determine which part of a username to use as the domain name.

### *Using the Realm Name as the Domain Name*

Typically, a realm appears before the user field and is separated with the / character; for example, usEast/jill@abc.com. To use the realm name usEast rather than abc.com as the domain name, set the realm name delimiter to /. For example:

```
host1(config)#aaa delimiter realmName /
```

This command causes the system to use the string to the left of the / as the domain name.

### *Using Delimiters Other Than @*

You can set up the system to recognize delimiters other than @ to designate the domain name. Suppose there are two users: bob@abc.com and pete!xyz.com, and you want to use both of their domain names. In this case you would set the domain name delimiter to @ and !. For example:

```
host1(config)#aaa delimiter domainName @!
```

### *Using Either the Domain or the Realm as the Domain Name*

If the username contains both a realm name and a domain name delimiter, you can use either the domain name or the realm name as the domain name. As previously mentioned, the system treats usernames with multiple delimiters as though the realm name is to the left of the first delimiter and the domain name is to the right of the last delimiter.

The system searches for a domain name by starting from the right-most character and searches until it reaches the first domain delimiter. The system searches for a realm name by starting from the left-most character and searches until it reaches the first realm delimiter. You can use the **aaa parse-order** command to change the order in which the system searches.

If you set the parse order to:

- **domain-first** – The system searches for a domain name starting with the right-most character. For username usEast/lori@abc.com, the domain name would be abc.com.
- **realm-first** – The system searches for a realm name starting with the left-most character and uses the realm name as the user's domain name. For username usEast/lori@abc.com, the domain would be usEast.

For example, if you set the delimiter for the realm name to / and set the delimiter for the domain name to @, the system would parse the realm first. The username usEast/lori@abc.com would result in a domain name of usEast. To cause the parsing to return abc.com as the domain, enter the **aaa parse-order domain-first** command.

### *Stripping the Domain Name*

The system provides another feature that strips the domain name from the username before it sends the name to the RADIUS server in an access-request message. You can enable or disable this feature using the **strip-domain** command.

By default, the domain name is the text after the last @ character. However, if you changed the domain name parsing using the **aaa delimiter** and **aaa parse-order** commands, the system strips the domain name and delimiter that results from the parsing.

### ***aaa delimiter***

- Use to configure delimiters for the domain and realm names. Specify one of the following keywords:
  - › **domainName** – configures domain name delimiters. The default domain name delimiter is @.
  - › **realmName** – configures realm name delimiters. The default realm name delimiter is NULL (no character). In this case, realm parsing is disabled (having no delimiter disables realm parsing).
- You can specify up to eight delimiters each for domain name and realm name.
- Example

```
host1(config)#aaa delimiter domainName @*/
```
- Use the **no** version to return to the default.

### ***aaa parse-order***

- Use to specify which part of a username the system uses as the domain name. If a user's name contains both a realm name and a domain name, you can configure the system to use either name as the domain name.
  - › **domain-first** – system searches for a domain name starting with the right-most character and searching until it reaches the first domain delimiter. For example, if the username is usEast/lori@abc.com, abc.com is the domain name. If no domain name is found, the system then searches for a realm name if realm delimiter is specified.
  - › **realm-first** – system searches for a realm name starting with the left-most character and searching until it reaches the first realm delimiter. For example, if the username is usEast/lori@abc.com, usEast is the domain name. If no realm name is found, the system searches for a domain name.
- Example

```
host1(config)#aaa parse-order domain-first
```
- Use the **no** version to return to the default, realm first.

### ***strip-domain***

- Use to strip the domain name from the username before sending an access-request message to the RADIUS server.
- By default, the domain name is the text after the last @ character. However, if you changed the domain name parsing using the **aaa delimiter** and **aaa parse-order** commands, the system strips the domain name and delimiter that results from the parsing.
- To stop stripping the username, use the **disable** keyword.

- Example

```
host1(config)#aaa domain-map xyz.com
host1(config-domain-map)#strip-domain enable
```

- Use the **no** version to return to the default, disabled.

## Specifying a Single Name for Users from a Domain

---

Assigning a single username and a single password for all users associated with a domain provides better compatibility with some RADIUS servers. You can use this feature for domains that require the system to tunnel, but not terminate, PPP sessions.

When users request a PPP session, they specify usernames and passwords. During the negotiations for the PPP session, the system authenticates legitimate users.



**Note:** This feature works only for users authenticated by PAP and not by CHAP.

If you configure this feature, the system substitutes the specified username and password for all authenticated usernames and passwords associated with that domain.

There are two options for this feature. The system can:

- Substitute the domain name for each username and one new password for each existing password.

For example, if the domain name is `xyz.com` and you specify the password `xyz_domain`, the system will associate the username `xyz.com` and the password `xyz_domain` with all users from `xyz.com`.

- Substitute one new username for each username and one new password for each existing password.

For example, if the domain name is `xyz.com` and you specify the username `xyz_group` and the password `xyz_domain`, the system will associate these identifiers with all users from `xyz.com`.

To use a single username and a single password for all users from a domain:

- 1 Access Domain Map Configuration mode using the **aaa domain-map** command.
- 2 Specify the new username and password using the **override-user** command.

### ***aaa domain-map***

- Use to map a domain name to a virtual router or to access Domain Map Configuration mode.

- Example

```
host1(config)#aaa domain-map xyz.com
host1(config-domain-map)#
```

- Use the **no** version to delete the map entry.

### ***override-user***

- Use to specify a single username and single password for all users from a domain.
- Use only for domains that require the system to tunnel and not terminate PPP sessions.
- If you specify a password only, the system substitutes the domain name for the username and associates the new password with the user.
- If you specify a name and password, the system associates both the new name and password with the user.
- If you specify only a password with this command and you have configured the domain name *none* with the **aaa domain-map** command, the system rejects any users without domain names.

- Example

```
host1(config-domain-map)override-user name boston password abc
```

- Use the **no** version to revert to the original username.

## Configuring Authentication and Accounting Servers

---

The number of RADIUS servers you can configure depends on available memory.

The order in which you configure servers determines the order in which your ERX system contacts those servers on behalf of clients.

Initially, a RADIUS client sends a request to a RADIUS authentication or accounting server. The RADIUS server uses the configured IP address, the UDP port number, and the secret key to make the connection. The RADIUS client waits for a response for a configurable timeout period and then retransmits the request. The RADIUS client retransmits the request for a user-configurable retry limit.

- If there is no response from the primary RADIUS server, the RADIUS client submits the request to the secondary RADIUS server using the timeout period and retry limit configured for the secondary RADIUS server.

- If the connection attempt fails for the secondary RADIUS server, the system submits the request to the tertiary server and so on until it either is granted access on behalf of the client or there are no more configured servers.
- If another authentication server is not configured, the ERX system attempts the next method in the method list; for accounting server requests, the information is dropped.

For example, suppose that you have configured the following authentication servers: Auth1, Auth2, Auth3, Auth4, and Auth5. Your system attempts to send an authentication request to Auth1. If Auth1 is unavailable, the system submits the request to Auth2, then Auth3, and so on until an available server is found. If Auth5, the last configured authentication server, is not available, the ERX system attempts the next method in the methods list. If the only method configured is RADIUS, then the system notifies the client that the request has been denied.

### *Server Access*

The system offers two options by which servers are accessed:

- **Direct** – The first authentication or accounting server that you configure is treated as the primary authentication or accounting server, the next server configured is the secondary, and so on.
- **Round-robin** – The first configured server is treated as a primary for the first request, the second server configured as primary for the second request, and so on. When the system reaches the end of the list of servers, it starts again at the top of the list until it comes full cycle through the list.

Use the **radius algorithm** command to specify the server access method.

When you configure the first RADIUS accounting server, a RADIUS Acct-On message is sent. When you delete the last accounting server, a RADIUS Acct-Off message is sent.

### *Server Request Processing Limit*

Each authentication and accounting server is capable of handling a maximum of 4,000 outstanding requests. Once that number is reached, the system submits the request to the next configured server.

### *SNMP Traps and System Log Messages*

The ERX system can alert network managers when RADIUS servers fail to respond to a request. You can set up the system to send SNMP traps

when a RADIUS server fails to respond to a request or when all RADIUS servers within a VR context fail to respond to a request. The system also generates syslog messages when RADIUS servers fail to respond; no configuration is required for syslog messages.

### SNMP Traps

The ERX system generates SNMP traps and syslog messages as follows:

- If the first RADIUS server fails to respond to the RADIUS request, the ERX RADIUS client issues a syslog and, if configured, an SNMP trap indicating that the RADIUS server timed out. The ERX RADIUS client will not issue another syslog or SNMP trap regarding this RADIUS server until the deadline expires, if configured, or for 3 minutes if deadline is not configured.
- The ERX RADIUS client then sends the RADIUS request to the second configured RADIUS server. If the second RADIUS server fails to respond to the RADIUS request, the ERX RADIUS client again issues a syslog and, if configured, an SNMP trap indicating that the RADIUS server timed out.
- This process continues until either the ERX RADIUS client receives a valid response from a RADIUS server or the list of configured RADIUS servers is exhausted. If the list of RADIUS servers is exhausted, the ERX RADIUS client issues a syslog and, if configured, an SNMP trap indicating that all RADIUS servers have timed out.

Note that if the ERX RADIUS client receives a RADIUS response from a “dead” RADIUS server during the deadline period, the RADIUS server is restored to active status.

### System Log Messages

You do not need to configure syslog messages. The system automatically sends them when individual servers do not respond to RADIUS requests and when all servers on a VR fail to respond to requests. The following are the formats of the warning level syslog messages:

```
RADIUS [ authentication | accounting ] server <server  
address> unavailable in VR <virtual router name>[; trying  
<next server address>]
```

```
RADIUS no [ authentication | accounting ] servers responding  
in VR <virtual router name>
```

## UDP Checksums

Each virtual router on which you configure B-RAS is enabled to perform UDP checksums by default. You can disable and reenable UDP checksums.

## Configuring AA Servers

The number of RADIUS servers you can configure depends on available memory. The system has an embedded RADIUS client for authentication and accounting.



**Note:** You can configure B-RAS with RADIUS accounting, but without RADIUS authentication. In this configuration, the username and password on the remote end are not authenticated and can be set to any value.

You must assign an IP address to a RADIUS authentication or accounting server to configure it.

If you do not configure a primary authentication or accounting server, all authentication and accounting requests will fail. You can configure other servers as backup in the event that the primary server cannot be reached. Configure each server individually.

To configure an authentication or accounting RADIUS server:

- 1 Specify a server address.

```
host1(config)#radius authentication server 10.10.10.1
```

- 2 (Optional) Specify a UDP port for RADIUS authentication or accounting server requests.

```
host1(config-radius)#udp-port 1645
```

- 3 Specify an authentication or accounting server secret.

```
host1(config-radius)#key gismo
```

- 4 (Optional) Specify the number of retries the system makes to an authentication or accounting server before it attempts to contact another server.

```
host1(config-radius)#retransmit 2
```

- 5 (Optional) Specify whether the ERX system should move on to the next RADIUS server when the system receives an access-reject message for the user it is authenticating.

```
host1(config)#radius rollover-on-reject enable
```

- 6 (Optional) Specify the interval (in seconds) between retries.

```
host1(config-radius)#timeout 5
```

- 7 (Optional) Specify the maximum number of outstanding requests.  

```
host1(config-radius)#max-sessions 100
```
- 8 (Optional) Enable duplicate address checking.  

```
host1(config)aaa duplicate-address-check enable
```
- 9 (Optional) Specify the amount of time to remove a server from the available list when a timeout occurs.  

```
host1(config-radius)#deadtime 10
```
- 10 Specify that duplicate accounting records be sent to the accounting server for a virtual router.  

```
host1(config)#aaa accounting duplication routerBoston
```
- 11 (Optional) Specify that tunnel accounting be enabled or disabled.  

```
host1(config)#radius tunnel-accounting enable
```
- 12 (Optional) Specify the default authentication protocol for PPP and DHCP clients.  

```
host1(config)#aaa authentication ppp default radius
```
- 13 (Optional) Disable UDP checksums on virtual routers you configure for B-RAS.  

```
host1:(config)#virtual router boston  
host1:boston(config)#radius udp-checksum disable
```

### Configuring SNMP Traps

This section shows how to configure the system to send traps to SNMP when RADIUS servers fail to respond to messages, and how to configure SNMP to receive the traps.

To set up the system to send traps:

- 1 (Optional) Enable SNMP traps when a particular RADIUS authentication server fails to respond to Access-Request messages.  

```
host1(config)#radius trap auth-server-not-responding enable
```
- 2 (Optional) Enable SNMP traps when all of the configured RADIUS authentication servers on a VR fail to respond to Access-Request messages.  

```
host1(config)#radius trap no-auth-server-responding enable
```

- 3 (Optional) Enable SNMP traps when a particular RADIUS accounting server fails to respond to a RADIUS accounting request.
 

```
host1(config)#radius trap acct-server-not-responding enable
```
- 4 (Optional) Enable SNMP traps when all of the RADIUS accounting servers on a VR fail to respond to a RADIUS accounting request.
 

```
host1(config)#radius trap no-acct-server-responding enable
```

To set up the SNMP to receive RADIUS traps:

- 1 Set up the appropriate SNMP community strings.
 

```
host1(config)#snmp-server community admin view everything rw
host1(config)#snmp-server community private view user rw
host1(config)#snmp-server community public view everything
ro
```
- 2 Specify the interface whose IP address is the source address for SNMP traps.
 

```
host1(config)#snmp-server trap-source fastEthernet 0/0
```
- 3 Configure the host that should receive the SNMP traps.
 

```
host1(config)#snmp-server host 10.10.132.93 version 2c 3
udp-port 162 trapFilters radius
```
- 4 Enable the SNMP router agent to receive and forward RADIUS traps.
 

```
host1(config)#snmp-server enable traps radius
```
- 5 Enable the SNMP on the router.
 

```
host1(config)#snmp-server
```



**Note:** For more information on these SNMP commands, see *Configuring Traps in ERX System Basics Configuration Guide, Chapter 4, Configuring SNMP*.

### **aaa accounting duplication**

- Use to specify that duplicate accounting records be sent to the accounting server on another virtual router.
- Example
 

```
host1(config)#aaa accounting duplication routerBoston
```
- Use the **no** version to disable the feature.

### ***aaa accounting interval***

- Use to specify the accounting interval between updates. Note that interim accounting is not supported for I-DAS.
- Select an interval in minutes from 10–1080. The default is 0, which means that the feature is disabled.
- Example

```
host1(config)#aaa accounting interval 60
```
- Use the **no** version to turn off interim accounting.

### ***aaa accounting ppp default***

- Use to specify the default accounting protocol for PPP. Currently, **radius** and **none** are the only values supported.
- Specify **radius** to select it as the accounting protocol.
- Example

```
host1(config)#aaa accounting ppp default radius
```
- Use the **no** version to set the accounting protocol to the default, **radius**.

### ***aaa authentication ppp default***

- Use to specify the default authentication protocol for PPP and DHCP clients. Currently, **radius** and **none** are the only values supported.
- Specify **radius** to select it as the authentication protocol.
- Specify **none** to grant all users access without authentication.
- If you use the **aaa authentication ppp default radius none** command, you are granted access when RADIUS servers are not available.
- Example

```
host1(config)#aaa authentication ppp default radius
```
- Use the **no** version to set the authentication protocol to the default, **radius**.

### ***aaa duplicate-address-check***

- Use this command to enable or disable routing table address lookup or duplicate address check.
- The system checks the routing table for returned addresses for PPP users. If the address existed, then the user was denied access.
- You can disable this routing table address lookup or duplicate address check with the **aaa duplicate-address-check** command.
- Example

```
host1(config)#aaa duplicate-address-check enable
```

**deadtime**

- Use to configure the amount of time (0–30 minutes) that a server is marked as unavailable if a request times out for the configured retry count. The default is 0.
- If a server fails to answer a request, it is marked *unavailable* by the system. The system does not send requests to the server until the system receives a response from the server or until the configured time is reached, whichever occurs first.
- If all servers fail to answer a request, then instead of marking all servers as unavailable, all servers are marked as available.
- To turn off the deadtime mechanism, specify a value of 0.
- Example

```
host1(config)#radius authentication server 10.10.0.1
host1(config-radius)#deadtime 10
```

- Use the **no** version to set the time to the default value.

**key**

- Use to configure secrets on the primary, secondary, and tertiary authentication servers.
- The authentication or accounting server secret is a text string used by RADIUS to encrypt the client and server *authenticator* field during exchanges between the system and a RADIUS authentication server. The system encrypts PPP PAP passwords using this text string.
- The default is no server secret.
- Example

```
host1(config)#radius authentication server 10.10.8.1
host1(config-radius)#key gismo
```

- Use the **no** version to remove the secret.



**Note:** Authentication fails if no key is specified for the authentication server.

**logout subscribers**

- Use to issue an administrative reset to the user's connection to disconnect the user.
- From Privilege Exec mode, you can choose to logout **all** subscribers, or logout subscribers by **username**, **domain**, **virtual-router**, or **port**.
- This command only applies to PPP users.
- Example

```
host1#logout subscribers username bmurphy
```

**max-sessions**

- Use to configure the number of outstanding requests to a server.
- Your system supports a maximum of 4,000 concurrent RADIUS requests. If the maximum number of outstanding requests is reached, the system sends the request to the next server.

- The default is 255.
- For each multiple of 255 (the RADIUS protocol limit), the ERX system opens a new UDP port to send and receive RADIUS requests and responses.
- Example
 

```
host1(config)#radius authentication server 10.10.0.1
host1(config-radius)#max-sessions 100
```



- Use the **no** version to restore the default value.
- **Note:** *The system can autosense up to 16,000 PPPoE or PPPoA sessions via dynamic interfaces.*

### ***no radius client***

- Use to remove all RADIUS servers for the virtual router context and to delete the ERX RADIUS client for the virtual router context.
- Example
 

```
host1:boston(config)#no radius client
```

### ***radius algorithm***

- Use to specify the algorithm—either **direct** or **round-robin**—that the ERX RADIUS client uses to contact the RADIUS server.
- Example
 

```
host1(config)#radius algorithm round-robin
```
- Use the **no** version to set the algorithm to the default, **direct**.

### ***radius rollover-on-reject***

- Use to specify whether the ERX system rolls over to the next RADIUS server when the system receives an access-reject message for the user it is authenticating.
- Example
 

```
host1(config)#radius rollover-on-reject enable
```
- Use the **no** version to set the default of disable.

### ***radius server***

- Use to specify the IP address of **authentication** and **accounting** servers.
- Example
 

```
host1(config)#radius authentication server 10.10.10.1
host1(config-radius)exit
host1(config)#radius authentication server 10.10.10.2
host1(config-radius)exit
host1(config)#radius authentication server 10.10.10.3
host1(config-radius)exit
host1(config)#radius authentication server 10.10.10.4
host1(config-radius)exit
```

```
host1(config)#radius authentication server 10.10.10.5
host1(config-radius)#exit
host1(config)#radius accounting server 10.10.10.20
host1(config-radius)#exit
host1(config)#radius accounting server 10.10.10.30
```

- Use the **no** version to delete the instance of the RADIUS server.

### ***radius trap acct-server-not-responding***

- Use to enable or disable traps when a particular RADIUS accounting server fails to respond to a RADIUS accounting request.

- Example

```
host1(config)#radius trap acct-server-not-responding enable
```

- Use the **no** version to return to the default setting, disabled.

### ***radius trap auth-server-not-responding***

- Use to enable or disable traps when a particular RADIUS authentication server fails to respond to a RADIUS Access-Request message.

- Example

```
host1(config)#radius trap auth-server-not-responding enable
```

- Use the **no** version to return to the default setting, disabled.

### ***radius trap no-acct-server-responding***

- Use to enable or disable traps when all of the configured RADIUS accounting servers per VR fail to respond to a RADIUS accounting request.

- Example

```
host1(config)#radius trap no-acct-server-responding enable
```

- Use the **no** version to return to the default setting, disabled.

### ***radius trap no-auth-server-responding***

- Use to enable or disable traps when all of the configured RADIUS authentication servers per VR fail to respond to a RADIUS Access-Request message.

- Example

```
host1(config)#radius trap no-auth-server-responding enable
```

- Use the **no** version to return to the default setting, disabled.

### ***radius tunnel-accounting***

- Use to specify that tunnel accounting be enabled or disabled.
- This command turns on accounting messages: Tunnel-Start, Tunnel-Stop, Tunnel-Reject, Tunnel-Link-Start, Tunnel-Link-Stop, and Tunnel-Link-Reject, as described in RFC 2867.

- Your system supports tunnel accounting for the L2TP LAC and LNS and for the L2F NAS.
- Example

```
host1(config)#radius tunnel-accounting enable
```
- Use the **no** version to set the default, *disable*.

### ***radius udp-checksum***

- Use to disable UDP checksums on virtual routers you configure for B-RAS.
- Issue this command in the context of the appropriate virtual router.
- Example

```
host1(config)#virtual router boston
host1:boston(config)#radius udp-checksum disable
```
- Use the **no** version to reenables UDP checksums on virtual routers you configure for B-RAS.

### ***radius update-source-addr***

- Use to specify an alternate source IP address for the system to use rather than the default router ID.
- Example

```
host1(config)#radius update-source-addr 192.168.40.23
```
- Use the **no** version to delete the parameter so that the system uses the router ID.

### ***retransmit***

- Use to configure the maximum number of times that the system retransmits a RADIUS packet to an authentication or accounting server.
- If there is no response from the primary RADIUS authentication or accounting server in the specified number of retries, the client sends the request to the secondary server. If there is no response from the secondary server, the system sends the request to the tertiary server, and so on.
- The default is three retry attempts.
- Example

```
host1(config)#radius authentication server 10.10.8.1
host1(config-radius)#retransmit 2
```
- Use the **no** version to set the value to the default.

### ***test aaa***

- Use to verify RADIUS authentication and accounting and IP address assignment setup.
- You must specify either a PPP or MLPPP user. PPP indicates a straight PPP user. MLPPP simulates Multilink PPP so that if multiple test commands are issued, all test users are bound by the same address.

- The command uses a username and password and attempts to authenticate a user, get an address assignment, and issue a start accounting request.
- Optionally, you can specify the virtual router context in which to authenticate the user.
- The command pauses for several seconds, then terminates the session by issuing a stop accounting request and an address release.
- Example

```
host1#test aaa ppp jsmith mypassword virtualroutercharlie2
```



**Note:** Specifying the password to associate with the username is optional. Specifying a virtual router is optional.

### **timeout**

- Use to configure the interval (in seconds) before the system retransmits a RADIUS packet to an authentication or accounting server.
- If the specified interval is reached and there is no response from the primary RADIUS authentication or accounting server, the system attempts another retry. When the retry limit is reached, the client sends the request to the secondary server. When the retry limit for the secondary server is reached, the system attempts to reach the tertiary server, and so on.



**Note:** After the fourth retransmission, the configured timeout value is ignored, and the system uses a “backoff” algorithm, which increases the timeout between each succeeding transmission.

The backoff algorithm is:

$$\text{timeout} = 2^{\text{retry-count}} + (\text{random}() \text{ modulo } 2^{\text{retry-count}})$$

- The default is 3 seconds.
- Example

```
host1(config)#radius authentication server 10.10.0.1
host1(config-radius)#timeout 5
```

- Use the **no** version to restore the default value.



**Note:** When a RADIUS server times out or when it has no available RADIUS identifier values, the system removes the RADIUS server from the list of available servers for a period of time. Your system restores all configured servers to the list if it is about to remove the last server. Restoring the servers avoids having an empty server list.

### **udp-port**

- Use to configure the UDP port on the system where the RADIUS authentication servers reside. The system uses this port to communicate with the RADIUS authentication servers.
- For an authentication server, you must specify a port number in the range 0–65536. The default is 1812.
- For an accounting server, you must specify a port number in the range 0–65536. The default is 1813.

- Example

```
host1(config)#radius authentication server 10.10.9.1
host1(config-radius)#udp-port 1645
```
- Use the **no** version to set the port number to the default value.

## Configuring Name Server Addresses

---

You can optionally assign IP addresses for Domain Name System (DNS) and Windows Internet Name Service (WINS) name servers.

During setup negotiations between the system and remote PC clients using PPP (Internet Protocol Control Protocol [IPCP] specifically), the remote client may request the DNS and WINS server IP addresses. If the IP addresses passed to the system by the remote PC client are different from the ones configured on your system, the system returns the values that you configured as the correct values to the remote PC client.

This behavior is controlled by the **ppp peer dns** and **ppp peer wins** interface commands.

If a PPP client request contains address values of 0.0.0.0 for the name servers, the system considers that the remote PC client is not configured and returns the configured values as the correct values to the remote PC client.

The DNS and WINS addresses are considered as part of the PPP user information. These addresses are provided to the PPP client as part of the IPCP negotiations between PPP peers (see *RFC 1877, PPP Internet Protocol Control Protocol Extensions for Name Server Addresses* for details).



**Note:** All name server address parameters are defined in the context of a virtual router.

### Configuration Tasks

Configure the DNS and WINS primary and secondary name server addresses.

- 1 Specify the IP address of the DNS primary name server.

```
host1(config)#aaa dns primary 10.10.10.5
```

- 2 Specify the IP address of the DNS secondary name server.

```
host1(config)#aaa dns secondary 10.10.10.6
```

- 3 Specify the IP address of the WINS primary name server.

```
host1(config)#aaa wins primary 192.40.10.05
```

- 4 Specify the IP address of the WINS secondary name server.

```
host1(config)#aaa wins secondary 192.40.10.40
```



**Note:** The name server address values are used exclusively for PPP clients and are not used by the system for domain name server resolution.

### ***aaa dns primary***

- Use to specify the IP address of the DNS primary name server.
- Example

```
host1(config)#aaa dns primary 10.10.10.5
```
- Use the **no** version to set the corresponding address to 0.0.0.0.

### ***aaa dns secondary***

- Use to specify the IP address of the DNS secondary name server.
- Example

```
host1(config)#aaa dns secondary 10.10.10.6
```
- Use the **no** version to set the corresponding address to 0.0.0.0.

### ***aaa wins primary***

- Use to specify the IP address of the WINS primary name server.
- Example

```
host1(config)#aaa wins primary 192.40.10.05
```
- Use the **no** version to set the corresponding address to 0.0.0.0.

### ***aaa wins secondary***

- Use to specify the IP address of the WINS secondary name server.
- Example

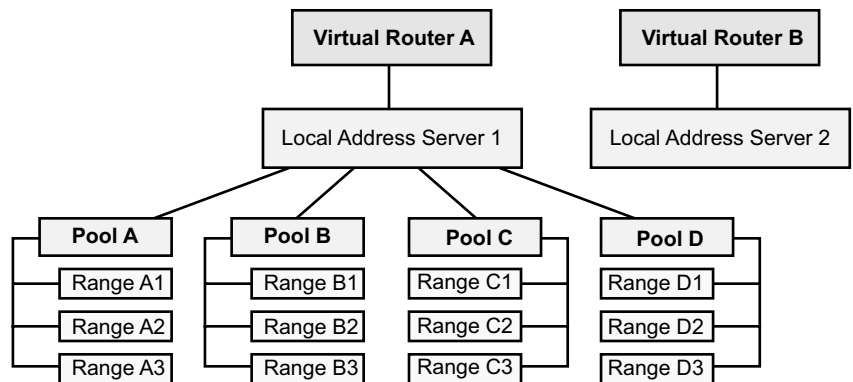
```
host1(config)#aaa wins secondary 192.40.10.40
```
- Use the **no** version to set the corresponding address to 0.0.0.0.

## Configuring Local Address Servers

The local address server allocates IP addresses from a local pool of addresses stored locally on your ERX system. Addresses are provided automatically to client sessions requiring an IP address from a virtual router that is configured to use a local address pool.

A local address server is defined in the context of a virtual router. You create a local address server when you configure the first local pool. Local address servers exist as long as the virtual router exists or until you remove them by deleting all configured pools.

Figure 1-1 illustrates the local address pool hierarchy. There may be multiple local address server instances, one per virtual router. Within each local address server, there can be one or more local address pools. Each pool can contain a number of IP addresses that are available for allocation and used by clients, such as PPP sessions.



**Figure 1-1** Local address pool hierarchy

### *Local Address Pool Ranges*

As shown in Figure 1-1, each local address pool is named and contains ranges of sequentially ordered IP addresses. These addresses are allocated when the AAA server makes a request for an IP address.

If a local address pool range is exhausted, the next range of addresses is used. If all pool ranges are exhausted, a new range can be configured to extend or supplement the existing range of addresses, or a new pool can be created. The newly created pool range is then used for future address allocation. If addresses allocated from the first pool range are released, then subsequent requests for addresses are taken from the first pool range.

Addresses are assigned sequentially from a range within a pool. If a range has no addresses available, the next range within that pool is used. If a

pool has no addresses available, the next configured pool is used, unless a specific pool is indicated.

### *SNMP Thresholds*

An address pool has SNMP thresholds associated with it that allow the local address server to signal SNMP traps when certain conditions exist. These thresholds include high utilization threshold and abated utilization threshold. If a pool's outstanding addresses exceed the high utilization threshold and the SNMP trap signaling is enabled, SNMP is notified. Likewise, when a pool's utilization drops below the abated threshold utilization threshold, SNMP is notified.

### *Configuring a Local Address Server*

You can create, modify, and delete address pools. You can display address pool information or status with the **show ip local pool** command described in the section *Monitoring Remote Access*.

The following are examples of tasks you can configure:

- Specify an addressing scheme.

```
host1(config)#ip address-pool local
```

- Map an address pool name to a range of local addresses. You can also use this command to add additional ranges to a pool.

```
host1(config)#ip local pool addrpool_10 192.34.56.10  
192.34.56.15
```

- Map an address pool name to a domain name.

```
host1(config)#aaa domain-map westford.com  
host1(config-domain-map)#address-pool-name poolA
```

- Delete an address pool.

```
host1(config)#no ip local pool addrpool_10
```



**Note:** If a pool or range is deleted and there are outstanding addresses, the AAA server logs out the client(s) using the address(es).

- Set SNMP variables by specifying an existing pool name and values.

```
host1(config)#ip local pool addrpool_10 warning 90 80
```

### ***address-pool-name***

- Use to specify the name of the local address pool from which the system allocates addresses for the domain that you are configuring.
- If the authentication server does not return an address, the system allocates an address from this pool. The authentication server may override this pool name using RADIUS attributes such as framed-pool.
- Example

```
host1(config)#aaa domain-map westford.com
host1(config-domain-map)#address-pool-name poolA
```
- Use the **no** version to remove the address pool name.

### ***ip address-pool***

- Use to specify the addressing scheme: **dhcp**, **local**, or **none**.
- The addressing scheme **none** returns a special indicator to AAA that allows the remote PPP client to assign its own address.
- Example

```
host1(config)#ip address-pool dhcp
```
- Use the **no** version to specify the default, **local**.

### ***ip local pool***

- Use to map an address pool name to a range of local addresses.
- You can create a pool with no address ranges configured for it.
- A name may contain up to 16 characters.
- Example

```
host1(config)#ip local pool addrpool_10 192.34.56.10
192.34.56.15
```
- Use the **no** version to remove the local pool (all ranges), or the specified range.

### ***ip local pool snmpTrap***

- Use to enable SNMP pool utilization traps.
- Example

```
host 1(config)#ip local pool addr_test snmpTrap
```
- Use the **no** version to disable SNMP pool utilization traps.

### ***ip local pool warning***

- Use to set SNMP utilization warning threshold values.
- Example

```
host1(config)#ip local pool addr_test warning 70 80
```
- Use the **no** version to reset the attributes to their default values; high threshold 85, abated threshold 75.

## Configuring DHCP Features

---

You can configure the following DHCP features:

- DHCP proxy client
- DHCP relay agent
- DHCP local server

### *DHCP Overview*

The Dynamic Host Configuration Protocol (DHCP) provides a mechanism through which computers using TCP/IP can obtain protocol configuration parameters automatically from a DHCP server on the network.

The most important configuration parameter carried by DHCP is the IP address. A computer must be initially assigned a specific IP address that is appropriate to the network to which the computer is attached, and that is not assigned to any other computer on that network. If you move a computer to a new network, it must be assigned a new IP address for that new network. You can use DHCP to manage these assignments automatically.

An IP client contacts a DHCP server for configuration parameters. The DHCP server is typically centrally located and operated by the network administrator. Because the server is run by a network administrator, DHCP clients can be reliably and dynamically configured with parameters appropriate to the current network architecture.

For additional information about DHCP, see *Chapter 6, Configuring DHCP Local Server*.

### *Integrated DHCP Access Server*

RADIUS accounting is supported for an Integrated DHCP Access Server (I-DAS). This feature allows you to use RADIUS start and stop attributes to track user events such as the lifetime of an IP address.

For information on supported accounting attributes, see *Chapter 2, Configuring RADIUS Attributes* and *Appendix A, ERX-Supported RADIUS Attribute Descriptions*.

For more information about DHCP, consult RFC 2131 – Dynamic Host Configuration Protocol (March 1997).

For more information about DHCP Relay Agent, see RFC 3046 – DHCP Relay Agent Information Option (January 2001).

### Configuring a DHCP Proxy Client

DHCP proxy client support enables the system to obtain an IP address from a DHCP server for a remote PPP client. Each system virtual router (acting as a DHCP proxy client) can query up to five DHCP servers.

For PPP users, the system acts as a DHCP client to obtain an address for the PPP user. This is referred to as DHCP proxy.

The process for PPP users is as follows:

- 1 The remote user dials in and the client requests RADIUS authentication.
- 2 The AAA server on the system sends a request to the DHCP proxy client on the system for an IP address to be assigned to the remote user's host.
- 3 The proxy client assumes the role of DHCP client and sends a discovery message to the DHCP server(s).
- 4 One or more of the DHCP servers responds with an offer message containing an IP address.
- 5 The proxy client determines which offer to accept and sends a message to that DHCP server requesting that IP address.
- 6 The DHCP server responds to the proxy client with an acknowledgment message.
- 7 The proxy client passes the IP address to the AAA server on the system, and the AAA returns the address to PPP. PPP then assigns the address to the remote host. The new IP address is included when the system next updates its routing table.

Dynamic IP addresses are *leased* to the remote host for a specific period of time that can range from minutes to days. At the halfway point in the lease period, the proxy client requests an extension from the DHCP server on behalf of the remote host. The lease is extended for a period specified in the ACK message returned by the DHCP server—typically equal to the original lease. If the DHCP server returns a NAK (negative acknowledgment) message to the proxy client, the proxy client notifies the server on the system that the extension has been denied. The AAA server logs out the remote host and frees the IP address for reuse.

When a remote host disconnects, the AAA server notifies the proxy client that the IP address is available for reuse. The proxy client informs the DHCP server, which can now reassign that IP address.

To configure a proxy client from Global Configuration mode:

- 1 Specify one or more DHCP servers.

```
host1(config)#ip dhcp-server 10.6.128.10
```

- 2 Direct the system to request IP addresses for remote users from the DHCP server(s).

```
host1(config)#ip address-pool dhcp
```

### ***ip address-pool***

- Use to specify to the system where to get an IP address for the remote user.
- In this case, select the **dhcp** option to get the address from a DHCP server.
- Example

```
host1(config)#ip address-pool dhcp
```

- Use the **no** version to set the default, **local**.

### ***ip dhcp-server***

- Use to specify the address of a DHCP server that will provide IP addresses for remote hosts.
- When you issue this command, the system adds the IP address to the list of DHCP servers (up to five).

- Example

```
host1(config)#ip dhcp-server 10.6.128.10
```

- Use the **no** version to remove the specified DHCP server or all DHCP servers.

## *Configuring DHCP Relay and BOOTP Relay*

The DHCP relay feature relays a request from a remote client to a DHCP server for an IP address. When the system receives a DHCP request from an IP client, it forwards the request to the DHCP server and passes the response back to the IP client.

Configuring DHCP relay also enables BOOTstrap Protocol (BOOTP) relay. The system relays any BOOTP requests it receives to the same set of servers that you configured for DHCP relay. A DHCP server can respond to the BOOTP request only if it is also a BOOTP server. The system relays any BOOTP responses it receives to the originator of the BOOTP request. If you do not configure DHCP relay, then BOOTP relay is disabled.

The system must wait for an acknowledgment from the DHCP server that the assigned address has been accepted. The IP client must accept an IP address from one of the servers. When the DHCP server sends an

acknowledgment message back to the DHCP client via the system, the system updates its routing table with the IP address of the client.

If a DHCP relay request is received on an unnumbered interface, the system determines the loopback address for that interface and passes that IP address to the server.

DHCP carries other important configuration parameters such as the subnet mask, default router, and Domain Name System (DNS) server. Using DHCP, a network administrator can avoid *having to* configure individual computers using a complex and confusing manual process. Instead, those computers can obtain all required configuration parameters automatically from a centrally managed DHCP server.

### **set dhcp relay**

- Use to enable DHCP relay and BOOTP relay and to specify an IP address for the DHCP server.
- The system relays any BOOTP requests it receives to the same set of servers that you configured for DHCP relay.
- When you issue this command, the system adds the IP address to the list of DHCP servers (up to five) and forwards all request packets to all configured servers.
- Issuing this command also enables relay of BOOTP requests to the configured DHCP servers. If one of the DHCP servers is also a BOOTP server and responds, the system relays the response to the request originator.
- Optionally, you can use the `discard-access-routes` parameter to remove existing access routes for an interface from routing tables and NVS.
- Example

```
host1(config)#set dhcp relay 192.168.29.10
```
- Use the **no** version to remove the specified DHCP server or remove all servers if no IP address is specified.



**Note:** Once configured, the client communicates directly with the DHCP server to request address renewal or to release the address. As such, the DHCP relay component does not know when or if it should remove the installed host route.

### **set dhcp relay agent**

- Use to enable the DHCP Relay Agent Information, which includes the *circuit-ID* and *remote-ID* suboptions.
- When you issue the **set dhcp relay agent** command and certain conditions are met, the system adds the DHCP relay agent information suboptions to every packet it relays from a DHCP client to a DHCP server.
- The suboptions include information known to the relay agent that the server can use to implement parameter assignment policies. The server echoes the suboptions when it replies to the client, but the system strips the suboptions before relaying the packets to the client.
- You can specify either suboption separately.

- The *circuit-ID* suboption contains *slot/port* information and either VPI/VCI information (ATM interfaces) or VLAN tag information (VLAN interfaces).
- The *remote-ID* suboption contains a value only when (1) the interface is a dynamic ATM interface and (2) the **subscriber** command is used to configure a user name and domain name for the interface. If both conditions are met, the suboption contains a string with the user name and domain name in the format: `username@domainname`.
- Example

```
host1(config)#set dhcp relay agent
```
- Use the **no** version to disable the addition of the DHCP relay agent information.

## Creating an IP Interface

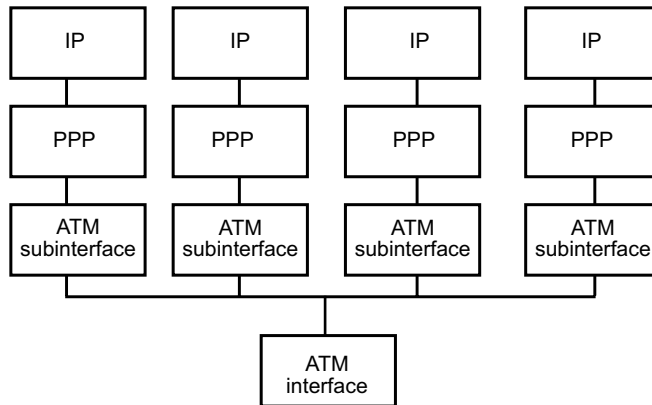
---

You can configure IP interfaces that support

- A single PPP client per ATM or Frame Relay subinterface
- Multiple PPP clients per ATM subinterface

### *Single Clients per ATM Subinterface*

Figure 1-2 shows a conceptual view of the configuration of a single PPP client per ATM subinterface.



**Figure 1-2** Single PPP clients per ATM subinterface

Configure an ATM interface by entering Configuration mode and performing the following tasks. See *ERX Physical and Link Layers Configuration Guide, Chapter 10, Configuring ATM*.

- 1 Configure a physical interface.  

```
host1(config)#interface atm 0/1
```
- 2 Configure the subinterface.  

```
host1(config-if)#interface atm 0/1.20
```
- 3 Configure a PVC by specifying the *vcd* (virtual circuit descriptor), the *vci* (virtual channel identifier), the *vpi* (virtual path identifier), and the encapsulation type.  

```
host1(config-if)#atm pvc 10 22 100 aal5snap
```
- 4 Configure PPP encapsulation.  

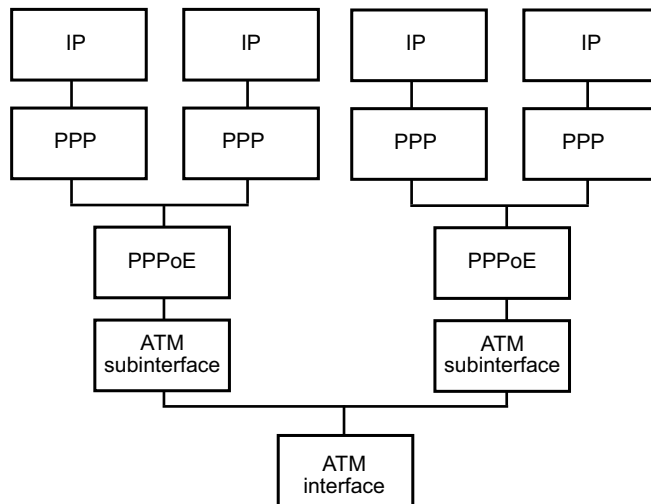
```
host1(config-if)#encapsulation ppp
```
- 5 Configure PAP or CHAP authentication.  

```
host1((config-if))#ppp authentication chap
```
- 6 Assign a profile to the PPP interface.  

```
host1(config-subif)#profile foo
```

*Multiple Clients per ATM Subinterface*

Figure 1-3 shows how PPPoE supports multiplexing of multiple PPP sessions per ATM subinterface.



**Figure 1-3** Multiple PPP clients per ATM subinterface

Configure an ATM interface by entering Configuration mode and performing the following tasks. See *ERX Physical and Link Layers Configuration Guide, Chapter 10, Configuring ATM*.

- 1 Configure a physical interface.

```
host1(config)#interface atm 0/1
```

- 2 Configure the subinterface.

```
host1(config-if)#interface atm 0/1.20
```

- 3 Configure a PVC by specifying the *vcd* (virtual circuit descriptor), the *vci* (virtual channel identifier), the *vpi* (virtual path identifier), and the encapsulation type.

```
host1(config-if)#atm pvc 10 22 100 aa15snap
```

- 4 Configure PPPoE encapsulation.

```
host1(config-if)#encapsulation pppoe
```

- 5 Configure the subinterface for one PPP client.

```
host1(config-if)#interface atm 0/1.20.1
```

- 6 Configure PPP encapsulation.

```
host1(config-if)#encapsulation ppp
```

- 7 Configure PAP or CHAP authentication.

```
host1((config-if))#ppp authentication chap
```

- 8 Apply the profile to the PPP interface.

```
host1(config-subif)#profile foo2
```

- 9 Configure the subinterface for a second PPP client.

```
host1(config-if)#interface atm 0/1.20.2
```

- 10 Configure PPP encapsulation.

```
host1(config-if)#encapsulation ppp
```

- 11 Configure PAP or CHAP authentication.

```
host1((config-if))#ppp authentication chap
```

- 12 Apply the profile to the PPP interface.

```
host1(config-subif)#profile foo2
```

## Configuring AAA Profiles

---

An AAA profile is a set of characteristics that act as a pattern which you can assign to domain names. Once you create an AAA profile, you can map it between a PPP client's domain name and certain AAA services on given interfaces. Using AAA profiles, you can:

- Allow a domain name access to AAA authentication
- Deny a domain name access to AAA authentication
- Map the original domain name to the mapped domain name for domain name lookup
- Use domain name aliases
- Force tunneling whenever a domain map contains tunnel attributes

An AAA profile contains a set of commands to control access for the incoming PPP subscriber. If no AAA profile is used, AAA continues as normal. The user's name and domain name are not changed as a result of an AAA profile mapping.



**Note:** There are two domain names with special meaning. The domain name **none** indicates that there is no domain name present in the subscriber's name. For more information on **none**, see the section *Mapping User Requests Without a Valid Domain Name*. The domain name **default** indicates that no other match occurs. For more information on **default**, see the section *Mapping User Requests Without a Configured Domain Name*.

### Allowing or Denying Domain Names

You can control a PPP subscriber's access to certain domains on given interfaces. As the administrator, you can use the **deny** command to prevent PPP subscribers from using unauthorized domain names. Using the **allow** command, you can allow PPP subscribers to use authorized domain names.

#### Configuration Example

In this example, the administrator wants to restrict access of a PPP interface to the specific domain **abc.com**.

- 1 Create an AAA profile.

```
host1(config)#aaa profile restrictToABC
```

- 2 Specify the domain name you want to allow.

```
host1(config-aaa-profile)#allow abc.com
```

- 3 Specify the domain name you want to restrict.
 

```
host1(config-aaa-profile)#deny default
```
- 4 Associate the AAA profile with the designated PPP interface.
 

```
host1(config-if)#ppp aaa-profile restrictToABC
```

When configured as such, the following is a likely scenario:

- PPP passes the AAA profile **restrictToABC** to AAA in the authentication request.
- AAA:
  - > Receives the authentication request from PPP with the subscriber's name **will@xyz.com**
  - > Parses the domain name **xyz.com** and examines the specified AAA profile **restrictToABC**
  - > Determines that the AAA profile **restrictToABC** is valid
  - > Searches **restrictToABC** for a match on the PPP subscriber's domain name and finds no match
  - > Searches **restrictToABC** for a match on the domain name **default**
  - > Finds a match and denies the user access

### Using Domain Name Aliases

You can translate an original domain name to a new domain name via the **translate** command. The command allows you to create domain name aliases; that is, the grouping of multiple domain names into a single domain name. You can partition PPP subscribers with the same domain into separate domains, based on the PPP interface.



**Note:** Partitioning subscribers does not cause modification of a user's name or domain.

When you use aliases, you greatly simplify the configuration process. When there are a large number of domains and you use aliases, it reduces the configuration volume, thus requiring less NVS and memory usage.

#### Configuration Example 1

In this example, an administrator wants to associate all subscribers of a PPP interface with a specific domain name.

- 1 Create an AAA profile.
 

```
host1(config)#aaa profile forwardToXyz
```

- 2 Map the original domain name to the mapped domain name for domain map lookup.

```
host1(config-aaa-profile)#translate default xyz.com
```

- 3 Associate the AAA profile with the designated PPP interface.

```
host1(config-if)#ppp aaa-profile forwardToXyz
```

When configured as such, the following scenario is typical:

- PPP passes the AAA profile **forwardToXyz** to AAA in the authentication request.
- AAA:
  - > Receives the authentication request from PPP with the subscriber's name **morris@abc.com**
  - > Parses the domain name **abc.com** and examines the specified AAA profile **forwardToXyz**
  - > Determines that the AAA profile **forwardToXyz** is valid
  - > Searches **forwardToXyz** for a match on the PPP subscriber's domain name and finds no match
  - > Searches **forwardToXyz** for a match on the domain name **default**
  - > Finds a match and continues as normal using the domain name **xyz.com**

### Configuration Example 2

In this example, an administrator wants to use aliases; that is, to associate multiple domain names with a specific domain name and not allow other domain names.

- 1 Create an AAA profile.

```
host1(config)#aaa profile toAbc
```

- 2 Map the original domain name to the mapped domain name for domain map lookup.

```
host1(config-aaa-profile)#translate abc1.com abc.com
host1(config-aaa-profile)#translate abc2.com abc.com
host1(config-aaa-profile)#translate abc3.com abc.com
```

- 3 Specify the domain name you want to restrict.

```
host1(config-aaa-profile)#deny default
```

- 4 Associate the AAA profile with the designated PPP interface.

```
host1(config-if)#ppp aaa-profile toAbc
```

When configured as such, the following scenario is typical:

- PPP passes the AAA profile **toAbc** to AAA in the authentication request.
- AAA:
  - > Receives the authentication request from PPP with the subscriber's name **jane@abc1.com**
  - > Parses the domain name **abc1.com** and examines the specified AAA profile **toAbc**
  - > Determines that the AAA profile **toAbc** is valid
  - > Searches **toAbc** for a match on the PPP subscriber's domain name and finds a match
  - > Continues as normal using the domain name **abc.com**

### **aaa profile**

- Use to configure a new AAA profile.
- Example

```
host1(config)#profile boston123
```
- Use the **no** version to delete the AAA profile.

### **allow**

- Use to specify the domain name(s) that you want to be allowed access to AAA authentication.
- This command does not indicate that the user will be granted access; it is simply the first access point to AAA authentication.
- Using this command does not implicitly deny all other domains.

- Example  

```
host1(config-aaa-profile)#allow xyz.com
```
- Use the **no** version to negate the command.

### **deny**

- Use to specify the domain name(s) that you want to be denied access to AAA authentication.
- Example  

```
host1(config-aaa-profile)#deny xyz.com
```
- Use the **no** version to negate the command.

### **ppp aaa-profile**

- Use to assign an AAA profile to static and dynamic, multilink and nonmultilink PPP interfaces.
- The PPP application associates the AAA profile with the interface and passes the AAA profile to AAA for authentication.
- If an AAA profile is deleted after it has been assigned to an interface, AAA will deny the authentication and log a message.
- When you remove an AAA profile, it does not remove any corresponding bindings between PPP interfaces or interface profiles and the AAA profile. If an AAA profile with the same name is added, the interface cannot authenticate until the AAA profile is reassigned.



**Note:** Although an AAA profile and an interface profile have similar functionality, they are not related and should be treated differently.

- Example  

```
host1(config-if)#ppp aaa-profile westford24
```
- Use the **no** version to remove the AAA profile assignment.

### **translate**

- Use to map the original domain name to the mapped domain name for domain map lookup.
- This command allows you to group multiple domain names into a single domain name (aliases).
- You can use this command to partition PPP subscribers with the same domain into separate domains, based on the PPP interface. By doing this, you do not cause modification of the user's name or domain.
- Example  

```
host1(config-aaa-profile)translate abc.com xyz.com
```
- Use the **no** version to negate the command.

## Using VSAs for Dynamic IP Interfaces

Table 1-1 describes the vendor-specific attributes (VSAs) that apply to dynamic IP interfaces and are supported on a per-user basis from RADIUS. See *ERX Physical and Link Layers Configuration Guide, Chapter 21, Configuring Dynamic Interfaces*.

**Table 1-1** VSAs that apply to dynamic IP interfaces

VSA	Description	Type	Length	Subtype	Subtype Length	Value
Ingress-Policy-Name	Specifies the name of the input (ingress) policy	26	len	10	sublen	string: <i>input-policy-name</i>
Egress-Policy-Name	Specifies the name of the output (egress) policy	26	len	11	sublen	string: <i>output-policy-name</i>
Ingress-Policy-Statistics	Indicates whether statistics are collected on input	26	12	12	6	integer: 0 – disable, 1 – enable
Egress-Policy-Statistics	Indicates whether statistics are collected on output	26	12	13	6	integer: 0 – disable, 1 – enable

You must perform the following tasks in order to successfully use these new VSAs:

- Specify the policy VSA(s) in the desired RADIUS user entries.
- Create the ingress or egress policy. Policies minimally consist of one or more policy commands and may include classifier control lists and rate limit profiles. See *ERX Policy Management and QoS Configuration Guide, Chapter 1, Configuring Policy Management*, for more information on policies and policy routing.

When a dynamic interface is created according to a profile, the system checks with RADIUS to determine whether an input or output policy must be applied to the interface. The VSA, if present, provides the name-enabling policy lookup. If found, the policy is applied to the dynamic interface. The system also determines whether the profile specifies any policies to be applied to the interface. Policies specified by the RADIUS VSA supersede any specified by the profile, as described in the following example:

The RADIUS user entry includes an Ingress-Policy-Name VSA that specifies the policy input5. The profile specifies two policies, input7 and output1. In this case, the RADIUS-specified input policy—input5—and the profile-specified output policy—output1—are applied to the dynamic interface.

See *ERX Policy Management and QoS Configuration Guide, Chapter i*, for information on assigning policies via profiles. Only attributes assigned by RADIUS appear in RADIUS Acct-Start messages. RADIUS attributes specified by a profile for dynamic interfaces do not appear in RADIUS Acct-Start messages because the profile is not active when the Acct-Start message is generated. These attributes will appear in RADIUS Acct-Stop messages for a profile that is active when the session is terminated.

### Traffic Shaping for PPP over ATM Interfaces

The system supports the configuration of traffic shaping parameters for PPP over ATM (PPPoA) via domain-based profiles and RADIUS. In connection with this feature, Table 1-2 describes VSAs which apply to dynamic IP interfaces and are supported on a per-user basis from RADIUS.

**Table 1-2** Traffic shaping VSAs that apply to dynamic IP interfaces

VSA	Description	Type	Length	Subtype	Subtype Length	Value
Atm-Service-Category	Specifies the type of service	26	12	14	6	integer: 1 – UBR 2 – UBR PCR 3 – NRT VBR 4 – CBR
Atm-PCR	Specifies the value for the peak cell rate (PCR)	26	12	15	6	integer
Atm-SCR	Specifies the value for the sustained cell rate (SCR)	26	12	16	6	integer
Atm-MBS	Specifies the maximum burst size (MBS)	26	12	17	6	integer

## Configuring Timeout

You can configure an idle or session timeout. The values you set are the default values for PPP B-RAS users. Attributes returned by RADIUS override these default settings on a per-user basis.

### **aaa timeout**

- Use to set either an idle or session timeout.
- The range in seconds for an idle timeout is 300–7200.
- The range in seconds for a *session* timeout is a minimum of 1 minute (60 seconds) to a maximum of 31 days (2678400 seconds). You need to round the setting to the nearest minute within the range.

- The current rounding behavior for the session, inactivity, and accounting timeouts is as follows:
  - › If the timeout is less than the minimum, it is rounded up to the minimum.
  - › If the timeout is greater than the maximum, it is rounded down to the maximum.
  - › If the timeout is not a multiple of 1 minute, it is rounded to the nearest minute.
- Example 1

```
host1(config)#aaa timeout idle 1200
```
- Example 2

```
host1(config)#aaa timeout session 3600
```
- Use the **no** version to delete either the idle or session timeout.

## Limiting Active Subscribers

---

You can limit the number of active subscribers on a port or virtual router.

### ***aaa subscriber limit per-port***

- Use to limit the number of active subscribers permitted on a port.
- Example

```
host1(config)#aaa subscriber limit per-port 2/0 20
```
- Use the **no** version to return to the default value, 0 (zero).

### ***aaa subscriber limit per-vr***

- Use to limit the number of active subscribers permitted on a virtual router.
- Because profiles are applied to subscribers after the PPP authentication phase, subscribers that have their VR context specified by profiles are not denied access. Instead, when IP notifies AAA of the subscribers VR context, AAA checks limits. If the subscriber exceeds the VR limit, AAA revokes the subscriber's access and logs out the subscriber.
- Example

```
host1:vr17(config)#aaa subscriber limit per-vr 20
```
- Use the **no** version to return to the default value, 0 (zero).

## Notifying RADIUS of AAA Failure

---

If a user passes RADIUS authentication, but fails AAA authentication, the RADIUS server may still allocate an address for the user from its internal address pool. To indicate to the RADIUS server to free the address, you can set up the system to send an Acct-Stop message if a user fails AAA.

**aaa accounting acct-stop on-aaa-failure**

- Use to cause the system to send an Acct-Stop message if a user fails AAA, but RADIUS grants access.
- Example

```
host1:vr17(config)#aaa accounting acct-stop on-aaa-failure
disable
```
- Use the **no** version to return to the default value, enabled.

**aaa accounting acct-stop on-access-deny**

- Use to cause the system to issue an Acct-Stop message if RADIUS denies access.
- Example

```
host1:vr17(config)#aaa accounting acct-stop on-access-deny
enable
```
- Use the **no** version to return to the default value, disabled.

## Configuring the SDX Client

---

The system has an embedded client that interacts with the Juniper Networks Service Deployment System (SDX; formerly SSC). To configure the SDX client, you specify the IP addresses of a primary, secondary, and/or tertiary SDX server(s). You also specify the port on which each SDX server listens for activity. Configuring the SDX client creates a Common Open Policy Service (COPS) protocol layer. You can configure SDX clients on a per virtual router basis.

**sscc address**

- Use to configure the SDX client with the IP addresses of the SDX servers and the ports on which the servers listen for activity.
- You can specify primary, secondary, and tertiary servers, and the port numbers on which each listens for activity.
- Example

```
host1(config)#sscc primary address 192.168.128.10 port 3310
```
- Use the **no** version to remove a specific SDX server (primary, secondary, or tertiary) from the list of servers.

**sscc enable**

- Use to enable SDX client support in the system.
- Example

```
host1(config)#sscc enable
```
- Use the **no** version to disable the feature.

### ***sscc retryTimer***

- Use to specify the delay period (from 5 to 300 seconds) during which the SDX client waits for a response from the SDX server. When the timer expires, the client attempts to reach the secondary server, and if that fails, the tertiary server, before trying the primary server again. The client waits for the delay period with each attempt.
- If only a primary server is configured, the request is sent again to the primary server.
- The default delay period is 90 seconds.
- Example

```
host1(config)#sscc retryTimer 90
```

- Use the **no** version to restore the default value.
- Use to specify a fixed source address for the TCP/COPS connection created for an SDX client session. This is the local address.
- If you do not specify a source address, the TCP/COPS connection is not bound to a specific source (that is, local) address.
- Example

```
host1(config)#sscc sourceAddress 10.9.123.8
```

- Use the **no** version to remove the specified SDX client source address.

### ***sscc transportRouter***

- Use to specify on which router the TCP/COPS (Common Open Policy Service) connection is to be established.
- The router can be the same as or different from the router the SDX client session is created in and associated with.
- If you do not specify the transport router for an SDX client session, the transport router defaults to the router associated with the session.
- Example

```
host1(config)#sscc transportRouter chicago
```

- Use the **no** version to remove the specified SDX client transport router.

## Setting Baselines

---

You can set a statistics baseline for AAA statistics or RADIUS authentication and authorization statistics using the **baseline** commands.

### ***baseline aaa***

- Use to set a statistics baseline for AAA statistics.
- The system implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **delta** keyword with the **show aaa statistics** command to specify that baselined statistics are to be shown.

***baseline dhcp relay***

- Use to set a statistics baseline for DHCP relay statistics.
- The system implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **delta** keyword with the **show dhcp relay statistics** command to specify that baselined statistics are to be shown.

***baseline dhcp server***

- Use to set a statistics baseline for DHCP proxy server statistics.
- The system implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **delta** keyword with the **show dhcp server statistics** command to specify that baselined statistics are to be shown.

***baseline local pool***

- Use to set a statistics baseline for local address pool statistics.
- The system implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **delta** keyword with the **show local pool statistics** command to specify that baselined statistics are to be shown.

***baseline radius***

- Use to set a statistics baseline for RADIUS statistics.
- The system implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.
- Use the **delta** keyword with the **show radius statistics** command to specify that baselined statistics are to be shown.

## Monitoring Remote Access

---

Use the commands in this section to monitor remote access. These commands provide information on the following:

- AAA profiles
- AAA statistics
- Address pools
- COPS protocol layer
- Domain name delimiters

- Name servers
- RADIUS servers
- RADIUS statistics
- SDX client connections
- Subscribers
- User domain mapping

Use the following commands to monitor PPP interfaces:

- **show ppp interface summary**
- **show ppp interface** *<selective control>*

See *ERX Physical and Link Layers Configuration Guide, Chapter 13, Configuring Point-to-Point Protocol* for details on the **show ppp** commands.

You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *ERX System Basics Configuration Guide, Chapter 2, Command Line Interface*.

### **show aaa**

- Use to display accounting or authentication information regarding PPP.
- Example

```
host1#show aaa authentication ppp default
radius
```

### **show aaa accounting**

- Use to display the AAA accounting configuration.
- Field descriptions
  - › Accounting duplication – name of the virtual router for which duplicate accounting records are sent to the accounting server
  - › send acct-stop on access failure – enabled, disabled
- Example

```
host1#show aaa accounting
No Accounting duplication configured
send acct-stop on ERX-AAA access deny is enabled
send acct-stop on authentication server access deny is
disabled
acct-interval (for PPP Clients) 0
```

### **show aaa accounting interval**

- Use to display the accounting interval.
- Example

```
host1#show aaa accounting interval
acct-interval (for PPP Clients) 10
```

### ***show aaa delimiters***

- Use to display the delimiters and parse order configured on the system.
- Example

```
host1#show aaa delimiters
domain delimiters "@"
realm delimiters "/"
parse-order realm first
```

### ***show aaa domain-map***

- Use to display the mapping between user domains and virtual routers.
- The following keywords have significance when used as user domains:
  - › **none** – all client requests with no user domain name are associated with the virtual router mapped to the **none** entry
  - › **default** – all client requests with a domain present that have no map are associated with the virtual router mapped to the **default** entry
- Example

```
host1#show aaa domain-map
Domain: xamian.com; virtual-router: default; loopback: 1; poolname: xamian;
IP hint enabled; override-username: roger; override-password: "H382crE"
Tunnel
Tag      Tunnel Peer      Type  Medium  Password  Id      Preference
---      -
1        172.31.1.98  l2tp  ipv4    temporary tunnel1  0
```

### ***show aaa duplicate-address-check***

- Use to display whether the routing table address lookup or duplicate address check is enabled or disabled.
- Example:

```
host1#show aaa duplicate-address-check
enabled
```

### ***show aaa model***

- Use to display the AAA model.
- Example:

```
host1#show aaa model
aaa model: old model
```

### ***show aaa profile***

- Use to display AAA profile names and the actions associated with each specified AAA profile name.
- Example

```

host1#show aaa profile
charlie:
    allow abc.com
    deny default

restrictToABC:
    allow abc.com
    deny default

```

### **show aaa name-servers**

- Use to display the IP addresses of the primary and secondary DNS and WINS name servers.
- Example

```

host1#show aaa name-servers
Name Server Addresses (for PPP Clients):
primary DNS Addr          1.2.3.4
secondary DNS Addr       5.6.7.8
primary NBNS (WINS) Addr 11.22.33.44
secondary NBNS (WINS) Addr 55.66.77.88

```

### **show aaa statistics**

- Use to display authentication and authorization statistics.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - › incoming initiate requests – number of incoming (from other ERX applications) AAA requests for user connect services
  - › incoming disconnect requests – number of incoming (from other ERX applications) AAA requests for user disconnect services
  - › outgoing grant (tunnel) responses – number of outgoing tunnel grant responses to AAA requests
  - › outgoing grant responses – number of outgoing grant responses to AAA requests
  - › outgoing deny responses – number of outgoing deny responses to AAA requests
  - › outgoing error responses – number of outgoing error responses to AAA requests
  - › incoming Authentication responses – number of authentication responses from the authentication task to AAA
  - › outgoing Authentication requests – number of authentication requests from AAA to the authentication task
  - › outgoing Re-Authentication requests – number of reauthentication requests from AAA to the authentication task
  - › incoming Re-Authentication responses – number of reauthentication responses from the authentication task to AAA

- › outgoing Accounting requests – number of accounting requests (starts, updates, stops) from AAA to the accounting task
  - › incoming Accounting responses – number of accounting responses (starts, updates, stops) from the accounting task to AAA
  - › outgoing Duplicate acct requests – number of duplicate accounting requests (starts, updates, stops) from AAA to the accounting task
  - › incoming Duplication acct responses – number of duplicate accounting responses (starts, updates, stops) from the accounting task to AAA
  - › outgoing Address requests – number of address allocation/release requests from AAA to address allocation task
  - › incoming Address responses – number of address allocation/release responses from the address allocation task to AAA
- Example

```

host1#show aaa statistics
                AAA Statistics
Statistic                               Count
-----
incoming initiate requests              2
incoming disconnect requests            2
outgoing grant (tunnel) responses       2
outgoing grant responses                0
outgoing deny responses                 0
outgoing error responses                0
outgoing Authentication requests        2
incoming Authentication responses       2
outgoing Accounting requests            4
incoming Accounting responses           4
outgoing Duplicate Acct requests        0
incoming Duplicate Acct responses       0
outgoing Address requests               0
incoming Address responses              0

```

### ***show aaa subscriber per-port-limit***

- Use to display the maximum number of active subscribers configured per port.
- Example

```

host1#show aaa subscriber per-port-limit
                Subscriber Port Limits
                -----
Port           Limit
-----
0/2            5
0/3            2
3/2            2

```

***show aaa subscriber per-vr-limit***

- Use to display the maximum number of active subscribers configured per virtual router.
- Example

```
host1#show aaa subscriber per-vr-limit
subscriber limit is 0
```

***show aaa timeout***

- Use to display information on idle and session timeouts.
- Example

```
host1#show aaa timeout
idle timeout (for PPP Clients) 0 seconds
session timeout (for PPP Clients) 0 seconds
```

***show cops info***

- Use to display information about the COPS layer over which the SDX connection is made.
- Field descriptions
  - › General Cops Information:
    - Session Created – number of COPS sessions created
    - Sessions Deleted – number of COPS sessions deleted
    - Current Sessions – number of current COPS sessions
    - Bytes Received – number of bytes received on all COPS sessions
    - Packets Received – number of packets received on all COPS sessions
    - Bytes Sent – number of bytes transmitted on all COPS sessions
    - Packets Sent – number of packets transmitted on all COPS sessions
    - Keep Alive Received – number of COPS keepalive messages received
    - Keep Alive Sent – number of COPS keepalive messages sent
  - › Session Information:
    - Remote IP Address – IP address of the remote peer
    - Remote TCP Port – TCP port number of the remote peer
    - Client Type – type of client for the session. For this release the client type must be 16640 (SDX client).
    - Bytes Received – number of bytes received for this COPS session
    - Packets Received – number of packets received for this COPS session
    - Bytes Sent – number of bytes sent on this COPS session
    - Packets Sent – number of packets sent on this COPS session
    - REQ Sent – number of Request packets sent on this COPS session
    - DEC Rcv – number of Decision packets received on this COPS session
    - RPT Sent – number of Report packets sent on this COPS session
    - DRQ Sent – number of Delete Requests sent on this COPS session
    - SSQ Rcv – number of Synch Requests received on this COPS session

- OPN Sent – number of Open messages sent on this COPS session
  - CAT Rcv – number of Client Accepts packets received on this COPS session
  - CC Sent – number of Client Closes packets sent on this COPS session
  - CC Rcv – number of Client Closes packets received on this COPS session
  - SSC Sent – number of Sync Complete packets sent on this COPS session
- Example

```
host1#show cops info
```

```
General Cops Information:
```

```
Sessions Created: 1
Sessions Deleted: 0
Current Sessions: 1
Bytes Received: 680
Packets Received: 17
Bytes Sent: 692
Packets Sent: 21
Keep Alive Received: 12
Keep Alive Sent: 12
```

```
Session Information
```

```
Remote Ip Address: 10.10.0.223
Remote TCP Port: 4001
Client Type: 16384
Bytes Received: 2224
Packets Received: 5
Bytes Sent: 596
Packets Sent: 9
REQ Sent: 4
DEC Rcv: 4
RPT Sent: 4
DRQ Sent: 0
SSQ Rcv: 0
OPN Sent: 1
CAT Rcv: 1
CC Sent: 0
CC Rcv: 0
SSC Sent: 0
```

### ***show dhcp relay***

- Use to display DHCP relay information.
- Example

```
host1#show dhcp relay
```

```

DHCP Relay Configuration
-----
Relay Agent Option - off

DHCP Server Addresses
-----
192.168.1.2
192.168.1.3

```

### ***show dhcp relay statistics***

- Use to display DHCP relay statistics.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - › Add DHCP Relay Agent Option – adds the DHCP Relay Agent Option to the DHCP packet
  - › unknown messages – number of DHCP packets received with an unrecognized command field
  - › bad messages – number of packets received that were not recognizable as valid DHCP packets
  - › packets with Relay Agent Option already present – number of packets received that already have the DHCP Relay Agent Option
  - › packets received with our IP address in giaddr – number of packets received that have the system address in the relay agent address field
- Example

```

host1#show dhcp relay statistics
                DHCP Relay Statistics
                -----
                Statistic                               Values
                -----
Add DHCP Relay Agent Option                            Off
unknown messages                                       0
bad messages                                           0
packets with Relay Agent Option already present        0
packets received with our IP address in giaddr        0

```

### ***show dhcp server***

- Use to display DHCP proxy statistics.
- Field descriptions
  - › O – read-only value that displays the operational status of the server:
    - E – enabled; indicates that the server is being actively used to supply IP addresses to clients
    - D – draining; indicates that the server is not accepting any new requests for addresses, but is maintaining the addresses that it has already assigned

- X – disabled: means that the server is not accepting any new requests for addresses and has no outstanding addresses
- › A – read/write value that displays the administrative status of the server:
  - E – enabled; indicates that the server is being actively used to supply IP addresses to clients
  - D – draining; indicates that the server is not accepting any new requests for addresses, but is maintaining the addresses that it has already assigned
  - X – disabled; means that the server is not accepting any new requests for addresses and had no outstanding addresses
- › Address – IP address of a DHCP server
- › Leases – number of IP address leases granted by the server
- › Offers – number of offers sent by the server
- › Requests – number of requests sent to the server
- › Acks – number of acknowledgments received from the server
- › Naks – number of negative acknowledgments received from the server
- › Declines – number of IP addresses rejected because they were already in use
- › Releases – number of IP addresses released back to the server
- Example

```
host1#show dhcp server
```

```
DHCP Proxy Client Status:
```

```
-----
```

O	A	Address	Leases	Offers	Requests	Acks	Naks	Declines	Releases
E	E	10.6.128.10	0	0	0	0	0	0	0
E	E	10.6.128.11	0	0	0	0	0	0	0

### **show dhcp server statistics**

- Use to display DHCP proxy statistics
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions
  - › DHCP Server Address – IP address of the server
  - › Discovers sent – number of discover messages sent by the server
  - › leases granted – number of leases granted by the server
  - › Offers received – number of offers sent by the server
  - › Requests sent – number of requests sent to the server
  - › Acks received – number of acknowledgments received from the server
  - › Naks received – number of negative acknowledgments received from the server
  - › addresses declined – number of IP addresses rejected because they were already in use
  - › addresses released – number of IP addresses released back to the server

- › Informs sent – number of inform messages sent to the server
- › unknown messages – number of illegal DHCP messages or messages that cannot be handled by the system
- › bad messages – number of messages not recognized as DHCP messages
- Example

```
host1#show dhcp server statistics
```

```
DHCP Proxy Global Statistics
Messages from Unknown Servers 0
```

DHCP Proxy Server Statistics

Statistic	Counts	Counts	Counts
DHCP Server Address	10.6.128.10	10.10.0.42	192.168.200.10
Discovers sent	0	0	0
leases granted	0	0	0
Offers received	0	0	0
Requests sent	0	0	0
Acks received	0	0	0
Naks received	0	0	0
addresses declined	0	0	0
addresses released	0	0	0
Informs sent	0	0	0
unknown messages	0	0	0
bad messages	0	0	0

**show ip local pool**

- Use to display information about the local address pools configured on your system.
- If you do not specify the name of a local address pool, the system displays all local address pools.
- Field descriptions
  - › Pool – user-specified name of the address pool
  - › High Thresh – high utilization threshold value
  - › Abated Thresh – abated utilization threshold value
  - › Trap – enable SNMP pool utilization traps: Y (yes) or N (no)
  - › Begin – starting IP address
  - › End – ending IP address
  - › Free – number of addresses available for use
  - › In Use – number of addresses currently in use

- Example

```

host1#show ip local pool
          High   Abated
Pool     Thresh  Thresh  Trap
-----  -
addr_test      95      85    Y
                                     In
          Begin      End      Free  Use
-----  -
192.41.50.1   192.41.50.51    51    0

```

### ***show ip local pool statistics***

- Use to display local address pool statistics.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Example

```

host1#show ip local pool statistics
Local Address Pool Statistics

          Statistic                               Values
-----  -
Requests denied (pool exhaustion)                0

```

### ***show license b-ras***

- Displays the B-RAS license string.
- Example

```

host1#show license b-ras
K4bZ16Lr

```

### ***show radius algorithm***

- Use to display information on the currently configured RADIUS server algorithm.
- Example

```

host1#show radius algorithm
direct

```

### ***show radius rollover-on-reject***

- Use to display the configuration of the RADIUS rollover feature.
- Example

```

host1#show radius rollover-on-reject
rollover-on-reject enabled

```

**show radius servers**

- Use to display RADIUS authentication and accounting server information.
- Field descriptions
  - › IP Address – IP address of authentication or accounting server
  - › UDP Port – number of the UDP of authentication or accounting server
  - › Retry Count – maximum number of times that the system retransmits a RADIUS packet to the authentication or accounting server
  - › Timeout – interval (in seconds) before the system retransmits a RADIUS packet to the authentication or accounting server
  - › Maximum Sessions – the number of outstanding requests to the authentication or accounting server
  - › Dead Time – amount of time to remove the authentication or accounting server from the available list when a timeout occurs
  - › Secret – configured authentication or accounting server secret
- Example

```

host1#show radius servers
                RADIUS Authentication Configuration
                -----
                UDP  Retry          Maximum  Dead
                Port  Count    Timeout  Sessions Time   Secret
                -----
10.10.0.40      1645    3         3         255    5     radius
192.14.23.4    1812    3         3         255    0     <null>
132.2.6.1      1812    3         3         255    0     <null>

                RADIUS Accounting Configuration
                -----
                UDP  Retry          Maximum  Dead
                Port  Count    Timeout  Sessions Time   Secret
                -----
10.10.0.40      1646    3         3         255    5     radius

```

**show radius statistics**

- Use to display statistics on RADIUS authentication and accounting services.
- Use the optional **delta** keyword to specify that baselined statistics are to be shown.
- Field descriptions



**Note:** All descriptions apply to the primary, secondary, and tertiary RADIUS authentication and accounting servers.

- › UDP Port – number of the UDP of a RADIUS server
- › Round Trip Time – number of hundreds of seconds from request to response
- › Access Requests – number of access requests sent to server
- › Rollover Requests – number of requests coming into server as a result of the previous server timing out

- › Retransmissions – number of retransmissions
  - › Access Accepts – number of access accepts received from the server
  - › Access Rejects – number of access rejects received from the server
  - › Access Challenges – number of access challenges received from the server
  - › Malformed Responses – number of malformed responses. Malformed Responses are those responses with attributes having an invalid length, or unexpected attributes in response.
  - › Bad Authenticators – number of bad authenticators, meaning that the authenticator in the response is incorrect for the matching request.
  - › Requests Pending – number of requests waiting for a response
  - › Request Timeouts – number of requests that timed out
  - › Unknown Responses – number of unknown responses
  - › Packets Dropped – number of packets dropped. Packets Dropped are either packets that are too short or responses that have no matching request.
- Example

```
host1#show radius statistics
      RADIUS Authentication Statistics
      -----
      Statistic          10.10.121.128
      -----
UDP Port                1812
Round Trip Time         0
Access Requests        0
Rollover Requests      0
Retransmissions        0
Access Accepts         0
Access Rejects         0
Access Challenges      0
Malformed Responses    0
Bad Authenticators     0
Requests Pending       0
Request Timeouts       0
Unknown Responses      0
Packets Dropped        0

      RADIUS Accounting Statistics
      -----
      Statistic          10.10.121.128
      -----
UDP Port                1646
Round Trip Time         2
Accounting Requests    1
Rollover Requests      0
Retransmissions        3
Responses               1
```

```

Malformed Responses    0
Bad Authenticators     0
Requests Pending       0
Request Timeouts       3
Unknown Responses      0
Packets Dropped        0

```

### ***show radius tunnel-accounting***

- Use to display information on RADIUS accounting for L2TP tunnels.
- Example

```

host1#show radius tunnel-accounting
disabled

```

### ***show radius udp-checksum***

- Use to display information on UDP checksums.
- Example

```

host1#show radius udp-checksum
enabled

```

### ***show radius update-source-addr***

- Use to display the IP address of the RADIUS servers.
- Example

```

host1#show radius update-source-address
192.168.1.228

```

### ***show ssc info***

- Use to display the current status of the SDX client (formerly SSC client) connection to the SDX servers.
- Field descriptions
  - › The SSC client configured servers: – IP addresses of the primary, secondary, and tertiary SDX client servers
  - › The configured retry timer is (seconds) – delay period the client waits for a response from the SDX server before submitting request again
  - › SSC Client Statistics – statistics about the connection between the SDX client and SDX server
    - Policy Commands received – number of policy commands received on the SDX client connection
    - Policy Commands(List) – number of Policy Commands with subtype List
    - Policy Commands(Acct) – number of Policy Commands with subtype Accounting
    - Bad Policy Cmds received – number of Policy Commands received with bad policies
    - Error Policy Cmds received – number of Policy Commands received with errors

- Policy Reports sent – number of Policy Reports sent
  - Connection Open requests – number of connections the SDX client has tried to open with a remote SDX server
  - Connection Open completed – number of connections successfully open to the SDX server
  - Connection Closed sent – number of connections the SDX client has closed
  - Connection Closed remotely – number of connections that were closed by the remote SDX server
  - Create Interfaces sent – number of create interface indications sent to the SDX server
  - Delete Interfaces sent – number of delete interface indications sent to the SDX server
  - Active IP Interfaces – current number of active IP interfaces the SDX client is aware of
  - IP Interface Transitions – number of IP interface transitions logged by the SDX client
  - Synchronizes received – number of synchronization requests received by the SDX client from the SDX server
  - Synchronize Complete sent – number of synchronization complete indications sent
  - Internal Errors – number of internal errors
  - Communication Errors – number of errors with lower-layer communications (such as socket errors)
- Example

```
host1#show ssc info
The SSC Client is currently unconnected
The SSC Client configured servers are:
  Primary: 0.0.0.0:0
  Secondary: 0.0.0.0:0
  Tertiary: 0.0.0.0:0
  The configured retry timer is (seconds): 90
SSC Client Statistics:
  Policy Commands received      0
  Policy Commands(List)        0
  Policy Commands(Acct)        0
  Bad Policy Cmds received     0
  Error Policy Cmds received   0
  Policy Reports sent           0
  Connection Open requests     0
  Connection Open completed    0
  Connection Closed sent       0
  Connection Closed remotely   0
  Create Interfaces sent       0
  Delete Interfaces sent       0
  Active IP Interfaces         2
```

```

IP Interface Transitions 0
Synchronizes received    0
Synchronize Complete sent 0
Internal Errors          0
Communication Errors     0
Tokens Seen              0
Active Tokens            0
Token Transitions        0
Token Creates Sent       0
Token Deletes Sent       0
Active Addresses         0
Address Transitions      0
Create Addresses Sent    0
Delete Addresses Sent    0
Authentication Successes 0
Authentication Failures 0

```

**show ssc version**

- Use to display the SDX client (formerly SSC client) version number.
- Example

```

host1#show ssc version
The SSC Client version is: 4.0

```

**show subscribers**

- Use to display the active subscribers on your system.
- If you specify a username, the system displays only the users that minimally match.
- You can use the **domain**, **port**, **summary**, **username**, or **virtual-router** keyword. If a keyword is not selected, all users are displayed.
- Field descriptions

- › User Name – name of the subscriber
- › Type – type of subscriber: ppp, tnl (tunnel), atm, ip, tst (test)
- › Addr | Endpt – IP address and source of the address: radius, local, dhcp, user
- › Virtual Router – name of the virtual router context
- › Interface – interface specifier over which the subscriber is connected
- › Login Time – time the subscriber logged in

- Example

```

host1#show subscribers

```

```

Subscriber List
-----

```

User Name	Type	Addr Endpt	Virtual Router
bert	tst	10.10.67.86/radius	default

User Name	Interface	Login Time
bert	atm 2/1:100.105	02/02/18 10:59:08

- **Example** – this example shows the number of PPP subscribers on each virtual router.

```
host1#show subscribers summary
Virtual
Router      Count
-----
default    20
abc.com     40
Total Subscribers : 10
Peak Subscribers : 10
```

- **Examples** – these examples show the number of PPP subscribers on each interface.

```
host1#show subscribers summary port
Interface    Count
-----
3/1          5

ATM Subscribers : 5
Total Subscribers : 10
Peak Subscribers : 10
```

```
host1#show subscribe summary domain
Domain Name  Count
-----
abc.com      5

ATM Subscribers : 5
Total Subscribers : 10
Peak Subscribers : 10
```

