



# Home Agent Commands

---

## aaa accounting

To enable authentication, authorization, and accounting (AAA) accounting of requested services for billing or security purposes when you use RADIUS or TACACS+, use the **aaa accounting** command in global configuration mode. To disable AAA accounting, use the **no** form of this command.

```
aaa accounting {auth-proxy | system | network | exec | connection | commands level | dot1x}
  {default | list-name} [vrf vrf-name] {start-stop | stop-only | none} [broadcast] group
  group-name
```

```
no aaa accounting {auth-proxy | system | network | exec | connection | commands level | dot1x}
  {default | list-name} [vrf vrf-name] {start-stop | stop-only | none} [broadcast] group
  group-name
```

### Syntax Description

<b>auth-proxy</b>	Provides information about all authenticated-proxy user events.
<b>system</b>	Performs accounting for all system-level events not associated with users, such as reloads.  <b>Note</b> When system accounting is used and the accounting server is unreachable at system startup time, the system will not be accessible for approximately two minutes.
<b>network</b>	Runs accounting for all network-related service requests, including Serial Line Internet Protocol (SLIP), PPP, PPP Network Control Protocols (NCPs), and AppleTalk Remote Access Protocol (ARAP).
<b>exec</b>	Runs accounting for the EXEC shell session. This keyword might return user profile information such as what is generated by the <b>autocommand</b> command.
<b>connection</b>	Provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler and disassembler (PAD), and rlogin.
<b>commands level</b>	Runs accounting for all commands at the specified privilege level. Valid privilege level entries are integers from 0 through 15.
<b>dot1x</b>	Provides information about all IEEE 802.1x-related user events.
<b>default</b>	Uses the listed accounting methods that follow this keyword as the default list of methods for accounting services.
<i>list-name</i>	Character string used to name the list of at least one of the following accounting methods: <ul style="list-style-type: none"> <li>• <b>group radius</b>—Uses the list of all RADIUS servers for authentication as defined by the <b>aaa group server radius</b> command.</li> <li>• <b>group tacacs+</b>—Uses the list of all TACACS+ servers for authentication as defined by the <b>aaa group server tacacs+</b> command.</li> <li>• <b>group group-name</b>—Uses a subset of RADIUS or TACACS+ servers for accounting as defined by the server group <i>group-name</i> argument.</li> </ul>
<b>vrf vrf-name</b>	(Optional) Specifies a virtual routing and forwarding (VRF) configuration. VRF is used <i>only</i> with system accounting.

<b>start-stop</b>	Sends a “start” accounting notice at the beginning of a process and a “stop” accounting notice at the end of a process. The “start” accounting record is sent in the background. The requested user process begins regardless of whether the “start” accounting notice was received by the accounting server.
<b>stop-only</b>	Sends a “stop” accounting notice at the end of the requested user process.
<b>none</b>	Disables accounting services on this line or interface.
<b>broadcast</b>	(Optional) Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.
<b>group</b> <i>group-name</i>	Specifies the accounting method list. Enter at least one of the following keywords: <ul style="list-style-type: none"> <li>• <b>auth-proxy</b>—Creates a method list to provide accounting information about all authenticated hosts that use the authentication proxy service.</li> <li>• <b>commands</b>—Creates a method list to provide accounting information about specific, individual EXEC commands associated with a specific privilege level.</li> <li>• <b>connection</b>—Creates a method list to provide accounting information about all outbound connections made from the network access server.</li> <li>• <b>exec</b>—Creates a method list to provide accounting records about user EXEC terminal sessions on the network access server, including username, date, and start and stop times.</li> <li>• <b>network</b>—Creates a method list to provide accounting information for SLIP, PPP, NCPs, and ARAP sessions.</li> <li>• <b>resource</b>—Creates a method list to provide accounting records for calls that have passed user authentication or calls that failed to be authenticated.</li> <li>• <b>tunnel</b>—Creates a method list to provide accounting records (Tunnel-Start, Tunnel-Stop, and Tunnel-Reject) for virtual private dialup network (VPDN) tunnel status changes.</li> <li>• <b>tunnel-link</b>—Creates a method list to provide accounting records (Tunnel-Link-Start, Tunnel-Link-Stop, and Tunnel-Link-Reject) for VPDN tunnel-link status changes.</li> </ul>

**Defaults**

AAA accounting is disabled.

**Command Modes**

Global configuration (config)

**Command History**

Release	Modification
10.3	This command was introduced.
12.0(5)T	Group server support was added.
12.1(1)T	The <b>broadcast</b> keyword was introduced on the Cisco AS5300 and Cisco AS5800 universal access servers.

Release	Modification
12.1(5)T	The <b>auth-proxy</b> keyword was added.
12.2(1)DX	The <b>vrf</b> keyword and <i>vrf-name</i> argument were introduced on the Cisco 7200 series and Cisco 7401ASR.
12.2(2)DD	This command was integrated into Cisco IOS Release 12.2(2)DD.
12.2(4)B	This command was integrated into Cisco IOS Release 12.2(4)B.
12.2(13)T	The <b>vrf</b> keyword and <i>vrf-name</i> argument were integrated into Cisco IOS Release 12.2(13)T.
12.2(15)B	The tunnel and tunnel-link accounting methods were introduced.
12.3(4)T	The tunnel and tunnel-link accounting methods were integrated into Cisco IOS Release 12.3(4)T.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(11)T	The <b>dot1x</b> keyword was integrated into Cisco IOS Release 12.4(11)T.
12.2(33)SXH	This command was integrated into Cisco IOS release 12.(33)SXH.

## Usage Guidelines

### General Information

Use the **aaa accounting** command to enable accounting and to create named method lists that define specific accounting methods on a per-line or per-interface basis.

[Table 1](#) contains descriptions of keywords for AAA accounting methods.

**Table 1** *aaa accounting Methods*

Keyword	Description
<b>group radius</b>	Uses the list of all RADIUS servers for authentication as defined by the <b>aaa group server radius</b> command.
<b>group tacacs+</b>	Uses the list of all TACACS+ servers for authentication as defined by the <b>aaa group server tacacs+</b> command.
<b>group</b> <i>group-name</i>	Uses a subset of RADIUS or TACACS+ servers for accounting as defined by the server <i>group group-name</i> argument.

In [Table 1](#), the **group radius** and **group tacacs+** methods refer to a set of previously defined RADIUS or TACACS+ servers. Use the **radius-server host** and **tacacs-server host** commands to configure the host servers. Use the **aaa group server radius** and **aaa group server tacacs+** commands to create a named group of servers.

Cisco IOS software supports the following two methods of accounting:

- **RADIUS**—The network access server reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server.
- **TACACS+**—The network access server reports user activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting AV pairs and is stored on the security server.

Method lists for accounting define the way accounting will be performed. Named accounting method lists enable you to designate a particular security protocol to be used on specific lines or interfaces for particular types of accounting services. Create a list by entering values for the *list-name* argument where *list-name* is any character string used to name this list (excluding the names of methods, such as RADIUS or TACACS+) and method list keywords to identify the methods to be tried in sequence as given.

If the **aaa accounting** command for a particular accounting type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines (where this accounting type applies) except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, then no accounting takes place.

**Note**

---

System accounting does not use named accounting lists; you can define the default list only for system accounting.

---

For minimal accounting, include the **stop-only** keyword to send a “stop” record accounting notice at the end of the requested user process. For more accounting, you can include the **start-stop** keyword, so that RADIUS or TACACS+ sends a “start” accounting notice at the beginning of the requested process and a “stop” accounting notice at the end of the process. Accounting is stored only on the RADIUS or TACACS+ server. The **none** keyword disables accounting services for the specified line or interface.

To specify an accounting configuration for a particular VRF, specify a default system accounting method list, and use the **vrf** keyword and *vrf-name* argument. System accounting does not have knowledge of VRF unless specified.

When AAA accounting is activated, the network access server monitors either RADIUS accounting attributes or TACACS+ AV pairs pertinent to the connection, depending on the security method you have implemented. The network access server reports these attributes as accounting records, which are then stored in an accounting log on the security server. For a list of supported RADIUS accounting attributes, see the appendix “RADIUS Attributes” in the [Cisco IOS Security Configuration Guide](#). For a list of supported TACACS+ accounting AV pairs, see the appendix “TACACS+ Attribute-Value Pairs” in the [Cisco IOS Security Configuration Guide](#).

**Note**

---

This command cannot be used with TACACS or extended TACACS.

---

### Cisco Service Selection Gateway Broadcast Accounting

To configure Cisco Service Selection Gateway (SSG) broadcast accounting, use `ssg_broadcast_accounting` for the *list-name* argument. For more information about configuring SSG, see the chapter “Configuring Accounting for SSG” in the [Cisco IOS Service Selection Gateway Configuration Guide](#), Release 12.4.

### Layer 2 LAN Switch Port

You must configure the RADIUS server to perform accounting tasks, such as logging start, stop, and interim-update messages and time stamps. To turn on these functions, enable logging of “Update/Watchdog packets from this AAA client” in your RADIUS server Network Configuration tab. Next, enable “CVS RADIUS Accounting” in your RADIUS server System Configuration tab.

You must enable AAA before you can enter the **aaa accounting** command. To enable AAA and 802.1X (port-based authentication), use the following global configuration mode commands:

- **aaa new-model**
- **aaa authentication dot1x default group radius**

- **dot1x system-auth-control**

Use the **show radius statistics** command to display the number of RADIUS messages that do not receive the accounting response message.

---

**Examples**

The following example defines a default commands accounting method list, where accounting services are provided by a TACACS+ security server, set for privilege level 15 commands with a stop-only restriction.

```
aaa accounting commands 15 default stop-only group tacacs+
```

The following example defines a default auth-proxy accounting method list, where accounting services are provided by a TACACS+ security server with a start-stop restriction. The **aaa accounting** command activates authentication proxy accounting.

```
aaa new-model
aaa authentication login default group tacacs+
aaa authorization auth-proxy default group tacacs+
aaa accounting auth-proxy default start-stop group tacacs+
```

The following example defines a default system accounting method list, where accounting services are provided by RADIUS security server “server1” with a start-stop restriction. The **aaa accounting** command specifies accounting for vrf “vrf1.”

```
aaa accounting system default vrf1 water start-stop group server1
```

The following example defines a default IEEE 802.1x accounting method list, where accounting services are provided by a RADIUS server. The **aaa accounting** command activates IEEE 802.1x accounting.

```
aaa new model
aaa authentication dot1x default group radius
aaa authorization dot1x default group radius
aaa accounting dot1x default start-stop group radius
```

The following example shows how to enable network accounting and send tunnel and tunnel-link accounting records to the RADIUS server. (Tunnel-Reject and Tunnel-Link-Reject accounting records are automatically sent if either start or stop records are configured.)

```
aaa accounting network tunnel start-stop group radius
aaa accounting network session start-stop group radius
```

The following example shows how to enable IEEE 802.1x accounting:

```
aaa accounting dot1x default start-stop group radius
aaa accounting system default start-stop group radius
```

---

**Related Commands**

Command	Description
<b>aaa authentication dot1x</b>	Specifies one or more AAA methods for use on interfaces running IEEE 802.1X.
<b>aaa authentication ppp</b>	Specifies one or more AAA authentication methods for use on serial interfaces running PPP.
<b>aaa authorization</b>	Sets parameters that restrict user access to a network.
<b>aaa group server radius</b>	Groups different RADIUS server hosts into distinct lists and distinct methods.
<b>aaa group server tacacs+</b>	Groups different server hosts into distinct lists and distinct methods.

<b>Command</b>	<b>Description</b>
<b>aaa new-model</b>	Enables the AAA access control model.
<b>dot1x</b> <b>system-auth-control</b>	Enables port-based authentication.
<b>radius-server host</b>	Specifies a RADIUS server host.
<b>show radius statistics</b>	Displays the RADIUS statistics for accounting and authentication packets.
<b>tacacs-server host</b>	Specifies a TACACS+ server host.

# aaa authorization ipmobile

To authorize Mobile IP to retrieve security associations from the AAA server using TACACS+ or RADIUS, use the **aaa authorization ipmobile** command in global configuration mode. To remove authorization, use the **no** form of this command.

```
aaa authorization ipmobile {[radius | tacacs+] | default} [group server-groupname]
```

```
no aaa authorization ipmobile {[radius | tacacs+] | default} [group server-groupname]
```

## Syntax Description

<b>radius</b>	Authorization list named radius.
<b>tacacs+</b>	Authorization list named tacacs+.
<b>default</b>	Default authorization list.
<b>group</b> <i>server-groupname</i>	(Optional) Name of the server group to use.

## Defaults

AAA is not used to retrieve security associations for authentication.

## Command Modes

Global configuration

## Command History

Release	Modification
12.0(1)T	This command was introduced.

## Usage Guidelines

Mobile IP requires security associations for registration authentication. The security associations are configured on the router or on a AAA server. This command is not needed for the former; but in the latter case, this command authorizes Mobile IP to retrieve the security associations from the AAA server.

Once the authorization list is named, it can be used in other areas such as login. You can only use one named authorization list; multiple named authorization lists are not supported.

The **aaa authorization ipmobile default group** *server-groupname* command is the most commonly used method to retrieve security associations from the AAA server.



### Note

The AAA server does not authenticate the user. It stores the security association that is retrieved by the router to authenticate registration.

## Examples

The following example uses TACACS+ to retrieve security associations from the AAA server:

```
aaa new-model
aaa authorization ipmobile tacacs+
tacacs-server host 1.2.3.4
tacacs-server key mykey
ip mobile host 10.0.0.1 10.0.0.5 virtual-network 10.0.0.0 255.0.0.0 aaa
```



The following example uses RADIUS as the default group to retrieve security associations from the AAA server:

```
aaa new-model
aaa authentication login default enable
aaa authorization ipmobile default group radius
aaa session-id common
radius-server host 128.107.162.173 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server key cisco
ip mobile host 10.0.0.1 10.0.0.5 virtual-network 10.0.0.0 255.0.0.0 aaa
```

#### Related Commands

Command	Description
<b>aaa new-model</b>	Enables the AAA access control model.
<b>ip mobile host</b>	Configures the mobile host or mobile node group.
<b>radius-server host</b>	Specifies a RADIUS server host.
<b>radius-server key</b>	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.
<b>show ip mobile host</b>	Displays mobile node information.
<b>tacacs-server host</b>	Specifies a TACACS host.
<b>tacacs-server key</b>	Sets the authentication encryption key used for all TACACS+ communications between the access server and the TACACS+ daemon.

# aaa pod server

To enable inbound user sessions to be disconnected when specific session attributes are presented, use the **aaa pod server** command in global configuration mode. To disable this feature, use the **no** form of this command.

```
aaa pod server [port port number] [auth-type {any | all | session-key}] server-key
[encryption-type] string
```

```
no aaa pod server
```

Syntax Description	
<b>port</b> <i>port number</i>	(Optional) Network access server User Datagram Protocol (UDP) port to use for packet of disconnect (POD) requests. Default value is 1700.
<b>auth-type</b>	(Optional) Type of authorization required for disconnecting sessions. If no authentication type is specified, <b>auth-type</b> is the default.
<b>any</b>	(Optional) Session that matches all of the attributes sent in the POD packet is disconnected. The POD packet may contain one or more of four key attributes (user-name, framed-IP-address, session-ID, and session-key).
<b>all</b>	(Optional) Only a session that matches all four key attributes is disconnected. The default is <b>all</b> .
<b>session-key</b>	(Optional) Session with a matching session-key attribute is disconnected. All other attributes are ignored.
<b>server-key</b>	Configures the shared-secret text string.
<i>encryption-type</i>	(Optional) Single-digit number that defines whether the text immediately following is encrypted, and, if so, what type of encryption is used. Currently defined encryption types are 0, which means that the text immediately following is not encrypted, and 7, which means that the text is encrypted using an encryption algorithm defined by Cisco.
<i>string</i>	Shared-secret text string that is shared between the network access server and the client workstation. This shared-secret string must be the same on both systems.

**Defaults** The POD server function is disabled.

**Command Modes** Global configuration

Command History	Release	Modification
	12.1(2)XH	This command was introduced.
	12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T.
	12.2(2)XB	The <i>encryption-type</i> argument was added, as well as support for the voice applications and the Cisco 3600 series, and Cisco AS5350, and Cisco AS5400 routers.
	12.2(2)XB1	Support for the Cisco AS5800 was added.

Release	Modification
12.2(11)T	The <i>encryption-type</i> argument and support for the voice applications were added.  <b>Note</b> Support for the Cisco AS5300, Cisco AS5350, Cisco AS5400, and Cisco AS5800 is not included in this release.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

### Usage Guidelines

To disconnect a session, the values in one or more of the key fields in the POD request must match the values for a session on one of the network access server ports. Which values must match depends on the **auth-type** attribute defined in the command. If no **auth-type** attribute is specified, all three values must match. If no match is found, all connections remain intact and an error response is returned. The key fields are as follows:

- An h323-conf-id vendor-specific attribute (VSA) with the same content as received from the gateway for this call.
- An h323-call-origin VSA with the same content as received from the gateway for the leg of interest.
- A 16-byte Message Digest 5 (MD5) hash value that is carried in the *authentication* field of the POD request.

### Examples

The following example enables POD and sets the secret key to “xyz123”:

```
aaa pod server server-key xyz123
```

### Related Commands

Command	Description
<b>aaa accounting delay-start</b>	Delays generation of the start accounting record until the user IP address is established.
<b>aaa accounting</b>	Enables accounting records.
<b>debug aaa pod</b>	Displays debug messages for POD packets.
<b>radius-server host</b>	Identifies a RADIUS host.

# access-list

To configure the access list mechanism for filtering frames by protocol type or vendor code, use the **access-list** command in global configuration mode. To remove the single specified entry from the access list, use the **no** form of this command.

```
access-list access-list-number {permit | deny} {type-code wild-mask | address mask}
```

```
no access-list access-list-number {permit | deny} {type-code wild-mask | address mask}
```

## Syntax Description

<i>access-list-number</i>	Integer that identifies the access list. If the <i>type-code</i> and <i>wild-mask</i> arguments are included, this integer ranges from 200 to 299, indicating that filtering is by protocol type. If the <i>address</i> and <i>mask</i> arguments are included, this integer ranges from 700 to 799, indicating that filtering is by vendor code.
<b>permit</b>	Permits the frame.
<b>deny</b>	Denies the frame.
<i>type-code</i>	16-bit hexadecimal number written with a leading 0x; for example, 0x6000. Specify either a Link Service Access Point (LSAP) type code for 802-encapsulated packets or a Subnetwork Access Protocol (SNAP) type code for SNAP-encapsulated packets. (LSAP, sometimes called SAP, refers to the type codes found in the DSAP and SSAP fields of the 802 header.)
<i>wild-mask</i>	16-bit hexadecimal number whose ones bits correspond to bits in the <i>type-code</i> argument. The <i>wild-mask</i> argument indicates which bits in the <i>type-code</i> argument should be ignored when making a comparison. (A mask for a DSAP/SSAP pair should always be 0x0101 because these two bits are used for purposes other than identifying the SAP code.)
<i>address</i>	48-bit Token Ring address written as a dotted triple of four-digit hexadecimal numbers. This field is used for filtering by vendor code.
<i>mask</i>	48-bit Token Ring address written as a dotted triple of four-digit hexadecimal numbers. The ones bits in <i>mask</i> are the bits to be ignored in <i>address</i> . This field is used for filtering by vendor code. For source address filtering, the mask always should have the high-order bit set. This is because the IEEE 802 standard uses this bit to indicate whether a Routing Information Field (RIF) is present, not as part of the source address.

## Defaults

No access list is configured.

## Command Modes

Global configuration

## Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

**Usage Guidelines**

For a list of type codes, refer to the “Ethernet Type Codes” appendix of this book.

**Examples**

In the following example, the access list permits only Novell frames (LSAP 0xE0E0) and filters out all other frame types. This set of access lists would be applied to an interface via the **source-bridge input-lsap list** or **source-bridge input-lsap list** command (described later in this chapter).

```
access-list 201 permit 0xE0E0 0x0101
access-list 201 deny 0x0000 0xFFFF
```

Combine the DSAP/LSAP fields into one number to do LSAP filtering; for example, 0xE0E0—not 0xE0. Note that the deny condition specified in the preceding example is not required; access lists have an implicit deny as the last statement. Adding this statement can serve as a useful reminder, however.

The following access list filters out only SNAP type codes assigned to Digital Equipment Corporation (DEC) (0x6000 to 0x6007) and lets all other types pass. This set of access lists would be applied to an interface using the **source-bridge input-type-list** or **source-bridge output-type-list** command (described later in this chapter).

```
access-list 202 deny 0x6000 0x0007
access-list 202 permit 0x0000 0xFFFF
```

**Note**

Use the last item of an access list to specify a default action; for example, to permit everything else or to deny everything else. If nothing else in the access list matches, the default action is to deny access; that is, filter out all other type codes.

Type code access lists will negatively affect system performance by greater than 30 percent. Therefore, we recommend that you keep the lists as short as possible and use wildcard bit masks whenever possible.

**Related Commands**

Command	Description
<b>access-expression</b>	Defines an access expression.
<b>source-bridge input-address-list</b>	Applies an access list to an interface configured for source-route bridging, and filters source-routed packets received from the router interface based on the source MAC address.
<b>source-bridge input-lsap-list</b>	Filters, on input, FDDI and IEEE 802-encapsulated packets that include the DSAP and SSAP fields in their frame formats.
<b>source-bridge input-type-list</b>	Filters SNAP-encapsulated packets on input.
<b>source-bridge output-address-list</b>	Applies an access list to an interface configured for SRB, and filters source-routed packets sent to the router interface based on the destination MAC address.
<b>source-bridge output-lsap-list</b>	Filters, on output, FDDI and IEEE 802-encapsulated packets that have DSAP and SSAP fields in their frame formats.
<b>source-bridge output-type-list</b>	Filters SNAP-encapsulated frames by type code on output.

# clear ip mobile binding

To remove mobility bindings, use the **clear ip mobile binding** command in privileged EXEC mode.

```
clear ip mobile binding {all [load standby-group-name] | ip-address [coa care-of-address] | nai
string [session-id string] | vrf realm realm} [synch]
```

## Syntax Description

<b>all</b>	Clears all mobility bindings.
<b>load</b> <i>standby-group-name</i>	(Optional) Downloads mobility bindings for a standby group after a clear operation.
<i>ip-address</i>	IP address of a mobile node or mobile router.
<b>coa</b> <i>care-of-address</i>	(Optional) The binding corresponding to the IP address and its care-of address.
<b>nai</b> <i>string</i>	Network access identifier (NAI) of the mobile node.
<b>session-id</b> <i>string</i>	(Optional) Session identifier. The string value must be fewer than 25 characters in length.
<b>vrf realm</b> <i>realm</i>	Specifies the VRF realm.
<b>synch</b>	(Optional) Specifies that the bindings that are administratively cleared on the active home agent are synchronized to the standby home agent, and the bindings will be deleted on the standby home agent.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.0(1)T	This command was introduced.
12.1(3)T	The following keywords and argument were added: <ul style="list-style-type: none"> <li><b>all</b></li> <li><b>load</b></li> <li><i>standby-group-name</i></li> </ul>
12.2(2)XC	The <b>nai</b> keyword was added.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.3(4)T	The <b>session-id</b> keyword was added.
12.4(9)T	The <b>coa</b> <i>care-of-address</i> keyword and argument combination were added.
12.4(11)T	The <b>vrf realm</b> <i>realm</i> and <b>synch</b> keywords and argument were added.

## Usage Guidelines

The home agent creates a mobility binding for each roaming mobile node. Associated with the mobility binding is the tunnel to the visited network and a host route to forward packets destined for the mobile node. Typically, there should be no need to clear the binding because it expires after the lifetime is reached or when the mobile node deregisters.

When the mobility binding is removed through use of this command, the number of users on the tunnel is decremented and the host route is removed from the routing table. The mobile node is not notified.

If the **nai *string* session-id *string*** option is specified, only the binding entry with that session identifier is cleared. If the **session-id** keyword is not specified, all binding entries (potentially more than one, with different session identifiers) for that NAI are cleared. You can determine the **session-id *string*** value by using the **show ip mobile binding** command.

When the **synch** option is specified, bindings that are administratively cleared on the active home agent are synchronized to the standby home agent, and the bindings will be deleted on the standby home agent. When the redundancy mode is active-standby, the **synch** option will not take effect if the clear command is issued on the standby home agent.

Use this command with care, because it will disrupt any sessions used by the mobile node. After you use this command, the mobile node will need to reregister to continue roaming.

### Examples

The following example administratively stops mobile node 192.168.100.10 from roaming:

```
Router# show ip mobile binding

Mobility Binding List:
Total 1
192.168.100.10:
  Care-of Addr 192.168.6.1, Src Addr 192.168.4.2,
  Lifetime granted 02:46:40 (10000), remaining 02:46:32
  Flags SbdmGvt, Identification B750FAC4.C28F56A8,
  Tunnel100 src 192.168.1.2 dest 192.168.6.1 reverse-allowed
  Routing Options - (G)GRE

Router# clear ip mobile binding 10.2.0.1

Router# show ip mobile binding
```

### Related Commands

Command	Description
<b>show ip mobile binding</b>	Displays the mobility binding table.

# clear ip mobile host-counters

To clear the mobility counters specific to each mobile node, use the **clear ip mobile host-counters** command in EXEC mode.

```
clear ip mobile host-counters [[ip-address | nai string] undo]
```

## Syntax Description

<i>ip-address</i>	(Optional) IP address of a mobile node.
<b>nai string</b>	(Optional) Network access identifier of the mobile node.
<b>undo</b>	(Optional) Restores the previously cleared counters.

## Command Modes

EXEC

## Command History

Release	Modification
12.0(1)T	This command was introduced.
12.2(2)XC	The <b>nai</b> keyword was added.
12.2(13)T	The <b>nai</b> keyword was integrated into Cisco IOS Release 12.2(13)T.

## Usage Guidelines

This command clears the counters that are displayed when you use the **show ip mobile host** command. The **undo** keyword restores the counters (this option is useful for debugging).

## Examples

The following example shows how the counters can be used for debugging:

```
Router# show ip mobile host

10.0.0.1:
  Allowed lifetime 10:00:00 (36000/default)
  Roaming status -registered-, Home link on virtual network 20.0.0.0/8
  Accepted 2, Last time 04/13/02 19:04:28
  Overall service time 00:04:42
  Denied 0, Last time -never-
  Last code '-never- (0)'
```

```
Router# clear ip mobile host-counters
```

```
Router# show ip mobile host-counters

20.0.0.1:
  Allowed lifetime 10:00:00 (36000/default)
  Roaming status -Unregistered-, Home link on virtual network 20.0.0.0/8
  Accepted 0, Last time -never-
  Overall service time -never-
  Denied 0, Last time -never-
  Last code '-never- (0)'
```



```
Total violations 0
Tunnel to MN - pkts 0, bytes 0
Reverse tunnel from MN - pkts 0, bytes 0
```

---

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>show ip mobile host</b>	Displays mobile node counters and information.

---

# clear ip mobile secure

To clear and retrieve remote security associations, use the **clear ip mobile secure** command in EXEC mode.

```
clear ip mobile secure {host lower [upper] | nai string | empty | all} [load]
```

## Syntax Description

<b>host</b>	Mobile node host.
<i>lower</i>	IP address of mobile node. Can be used alone, or as lower end of a range of IP addresses.
<i>upper</i>	(Optional) Upper end of a range of IP addresses.
<b>nai</b> <i>string</i>	Network access identifier of the mobile node.
<b>empty</b>	Load in only mobile nodes without security associations. Must be used with the <b>load</b> keyword.
<b>all</b>	Clears all mobile nodes.
<b>load</b>	(Optional) Reload the security association from the AAA server after security association has been cleared.

## Command Modes

EXEC

## Command History

Release	Modification
12.0(1)T	This command was introduced.
12.2(2)XC	The <b>nai</b> keyword was added.
12.2(13)T	The <b>nai</b> keyword was integrated into Cisco IOS Release 12.2(13)T.

## Usage Guidelines

Security associations are required for registration authentication. They can be stored on an AAA server. During registration, they may be stored locally after retrieval from the AAA server. The security association on the router may become stale or out of date when the security association on the AAA server changes.

This command clears security associations that have been downloaded from the AAA server.



### Note

Security associations that are manually configured on the router or not stored on the router after retrieval from the AAA server are not applicable.

## Examples

In the following example, the AAA server has the security association for user 10.2.0.1 after registration:

```
Router# show ip mobile secure host 10.2.0.1

Security Associations (algorithm,mode,replay protection,key) :
10.2.0.1:
  SPI 300, MD5, Prefix-suffix, Timestamp +/- 7,
  Key 'oldkey' 1230552d39b7c1751f86bae5205ec0c8
```

If you change the security association stored on the AAA server for this mobile node, the router clears the security association and reloads it from the AAA server:

```
Router# clear ip mobile secure host 10.2.0.1 load
```

```
Router# show ip mobile secure host 10.2.0.1
```

```
10.2.0.1:  
  SPI 300, MD5, Prefix-suffix, Timestamp +/- 7,  
  Key 'newkey' 1230552d39b7c1751f86bae5205ec0c8
```

---

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>ip mobile secure</b>	Specifies the mobility security associations for mobile host, visitor, home agent, and foreign agent.

---

# clear ip mobile traffic

To clear counters, use the **clear ip mobile traffic** command in EXEC mode.

**clear ip mobile traffic [undo]**

<b>Syntax Description</b>	<b>undo</b> (Optional) Restores the previously cleared counters.
---------------------------	--

<b>Command Modes</b>	EXEC
----------------------	------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.0(1)T	This command was introduced.

**Usage Guidelines** Mobile IP counters are accumulated during operation. They are useful for debugging and monitoring. This command clears all Mobile IP counters. The **undo** keyword restores the counters (which is useful for debugging). See the **show ip mobile traffic** command for a description of all counters.

**Examples** The following example shows how counters can be used for debugging:

```
Router# show ip mobile traffic
IP Mobility traffic:
Advertisements:
  Solicitations received 0
  Advertisements sent 0, response to solicitation 0
Home Agent Registrations:
  Register 8, Deregister 0 requests
  Register 7, Deregister 0 replied
  Accepted 6, No simultaneous bindings 0
  Denied 1, Ignored 1
  Unspecified 0, Unknown HA 0
  Administrative prohibited 0, No resource 0
  Authentication failed MN 0, FA 0
  Bad identification 1, Bad request form 0
.
Router# clear ip mobile traffic

Router# show ip mobile traffic
IP Mobility traffic:
Advertisements:
  Solicitations received 0
  Advertisements sent 0, response to solicitation 0
Home Agent Registrations:
  Register 0, Deregister 0 requests
  Register 0, Deregister 0 replied
  Accepted 0, No simultaneous bindings 0
  Denied 0, Ignored 0
  Unspecified 0, Unknown HA 0
  Administrative prohibited 0, No resource 0
  Authentication failed MN 0, FA 0
  Bad identification 0, Bad request form 0
```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>show ip mobile traffic</b>	Displays protocol counters.

## crypto map (global IPsec)

To enter crypto map configuration mode and create or modify a crypto map entry, to create a crypto profile that provides a template for configuration of dynamically created crypto maps, or to configure a client accounting list, use the **crypto map** command in global configuration mode. To delete a crypto map entry, profile, or set, use the **no** form of this command.

**crypto map** *map-name seq-num* [**ipsec-manual**]

**crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**]  
[**profile** *profile-name*]

**crypto map** *map-name* [**client-accounting-list** *aaalist*]

**crypto map** *map-name seq-num* [**gdoi**]

**no crypto map** *map-name seq-num*



### Note

Issue the **crypto map** *map-name seq-num* command without a keyword to modify an existing crypto map entry.

### Syntax Description

<i>map-name</i>	Name that identifies the crypto map set. This is the name assigned when the crypto map was created.
<i>seq-num</i>	Sequence number you assign to the crypto map entry. See additional explanation for using this argument in the “Usage Guidelines” section.
<b>ipsec-manual</b>	(Optional) Indicates that Internet Key Exchange (IKE) will not be used to establish the IP Security (IPSec) security associations (SAs) for protecting the traffic specified by this crypto map entry.
<b>ipsec-isakmp</b>	(Optional) Indicates that IKE will be used to establish the IPSec SAs for protecting the traffic specified by this crypto map entry.
<b>dynamic</b>	(Optional) Specifies that this crypto map entry is to reference a preexisting dynamic crypto map. Dynamic crypto maps are policy templates used in processing negotiation requests from a peer IPSec device. If you use this keyword, none of the crypto map configuration commands will be available.
<i>dynamic-map-name</i>	(Optional) Specifies the name of the dynamic crypto map set that should be used as the policy template.
<b>discover</b>	(Optional) Enables peer discovery. By default, peer discovery is not enabled.
<b>profile</b>	(Optional) Designates a crypto map as a configuration template. The security configurations of this crypto map will be cloned as new crypto maps are created dynamically on demand.
<i>profile-name</i>	(Optional) Name of the crypto profile being created.
<b>client-accounting-list</b>	(Optional) Designates a client accounting list.
<i>aaalist</i>	(Optional) List name.
<b>gdoi</b>	(Optional) Indicates that the key management mechanism is Group Domain of Interpretation (GDOI).

**Command Default** No crypto maps exist.  
Peer discovery is not enabled.

**Command Modes** Global configuration

Command History	Release	Modification
	11.2	This command was introduced.
	11.3T	The following keywords and arguments were added: <ul style="list-style-type: none"> <li>• <b>ipsec-manual</b></li> <li>• <b>ipsec-isakmp</b></li> <li>• <b>dynamic</b></li> <li>• <i>dynamic-map-name</i></li> </ul>
	12.0(5)T	The <b>discover</b> keyword was added to support Tunnel Endpoint Discovery (TED).
	12.2(4)T	The <b>profile</b> <i>profile-name</i> keyword and argument combination was added to allow the generation of a crypto map profile that is cloned to create dynamically created crypto maps on demand.
	12.2(11)T	This command was implemented on the Cisco 1760, Cisco AS5300, Cisco AS5400, and Cisco AS5800 platforms.
	12.2(15)T	The <b>client-accounting-list</b> <i>aaalist</i> keyword and argument combination was added.
	12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
	12.4(6)T	The <b>gdoi</b> keyword was added.
	12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB without support for the <b>gdoi</b> keyword.
	12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

**Usage Guidelines** Use this command to create a new crypto map entry, to create a crypto map profile, or to modify an existing crypto map entry or profile.

After a crypto map entry has been created, you cannot change the parameters specified at the global configuration level because these parameters determine which of the configuration commands are valid at the crypto map level. For example, after a map entry has been created using the **ipsec-isakmp** keyword, you cannot change it to the option specified by the **ipsec-manual** keyword; you must delete and reenter the map entry.

After you define crypto map entries, you can assign the crypto map set to interfaces using the **crypto map** (interface IPSec) command.

#### Crypto Map Functions

Crypto maps provide two functions: filtering and classifying traffic to be protected and defining the policy to be applied to that traffic. The first use affects the flow of traffic on an interface; the second affects the negotiation performed (via IKE) on behalf of that traffic.

IPSec crypto maps define the following:

- What traffic should be protected
- To which IPSec peers the protected traffic can be forwarded—these are the peers with which an SA can be established
- Which transform sets are acceptable for use with the protected traffic
- How keys and SAs should be used or managed (or what the keys are, if IKE is not used)

### Multiple Crypto Map Entries with the Same Map Name Form a Crypto Map Set

A crypto map set is a collection of crypto map entries, each with a different *seq-num* argument but the same *map-name* argument. Therefore, for a given interface, you could have certain traffic forwarded to one IPSec peer with specified security applied to that traffic and other traffic forwarded to the same or a different IPSec peer with different IPSec security applied. To accomplish differential forwarding you would create two crypto maps, each with the same *map-name* argument, but each with a different *seq-num* argument. Crypto profiles must have unique names within a crypto map set.

### Sequence Numbers

The number you assign to the *seq-num* argument should not be arbitrary. This number is used to rank multiple crypto map entries within a crypto map set. Within a crypto map set, a crypto map entry with a lower *seq-num* is evaluated before a map entry with a higher *seq-num*; that is, the map entry with the lower number has a higher priority.

For example, consider a crypto map set that contains three crypto map entries: mymap 10, mymap 20, and mymap 30. The crypto map set named “mymap” is applied to serial interface 0. When traffic passes through serial interface 0, the traffic is evaluated first for mymap 10. If the traffic matches any access list permit statement entry in the extended access list in mymap 10, the traffic will be processed according to the information defined in mymap 10 (including establishing IPSec SAs when necessary). If the traffic does not match the mymap 10 access list, the traffic will be evaluated for mymap 20, and then mymap 30, until the traffic matches a permit entry in a map entry. (If the traffic does not match a permit entry in any crypto map entry, it will be forwarded without any IPSec security.)

### Dynamic Crypto Maps

Refer to the “Usage Guidelines” section of the **crypto dynamic-map** command for a discussion on dynamic crypto maps.

Crypto map entries that reference dynamic map sets should be the lowest priority map entries, allowing inbound SA negotiation requests to try to match the static maps first. Only after the request does not match any of the static maps do you want it to be evaluated against the dynamic map set.

If a crypto map entry references a dynamic crypto map set, make it the lowest priority map entry by giving it the highest *seq-num* value of all the map entries in a crypto map set.

Create dynamic crypto map entries using the **crypto dynamic-map** command. After you create a dynamic crypto map set, add the dynamic crypto map set to a static crypto map set with the **crypto map** (global IPSec) command using the **dynamic** keyword.

### TED

TED is an enhancement to the IPSec feature. Defining a dynamic crypto map allows you to dynamically determine an IPSec peer; however, only the receiving router has this ability. With TED, the initiating router can dynamically determine an IPSec peer for secure IPSec communications.

Dynamic TED helps to simplify IPSec configuration on the individual routers within a large network. Each node has a simple configuration that defines the local network that the router is protecting and the IPSec transforms that are required.



**Note**

TED helps only in discovering peers; otherwise, TED does not function any differently from normal IPsec. Thus, TED does not improve the scalability of IPsec (in terms of performance or the number of peers or tunnels).

**Crypto Map Profiles**

Crypto map profiles are created using the **profile** *profile-name* keyword and argument combination. Crypto map profiles are used as configuration templates for dynamically creating crypto maps on demand for use with the L2TP Security feature. The relevant SAs in the crypto map profile will be cloned and used to protect IP traffic on the L2TP tunnel.

**Note**

The **set peer** and **match address** commands are ignored by crypto profiles and should not be configured in the crypto map definition.

**Examples**

The following example shows the minimum required crypto map configuration when IKE will be used to establish the SAs:

```
crypto map mymap 10 ipsec-isakmp
  match address 101
  set transform-set my_t_set1
  set peer 10.0.0.1
```

The following example shows the minimum required crypto map configuration when the SAs are manually established:

```
crypto transform-set someset ah-md5-hmac esp-des
crypto map mymap 10 ipsec-manual
  match address 102
  set transform-set someset
  set peer 10.0.0.5
  set session-key inbound ah 256 98765432109876549876543210987654
  set session-key outbound ah 256 fedcbafedcbafedcbafedcbafedcbafedc
  set session-key inbound esp 256 cipher 0123456789012345
  set session-key outbound esp 256 cipher abcdefabcdefabcd
```

The following example configures an IPsec crypto map set that includes a reference to a dynamic crypto map set.

Crypto map “mymap 10” allows SAs to be established between the router and either (or both) of two remote IPsec peers for traffic matching access list 101. Crypto map “mymap 20” allows either of two transform sets to be negotiated with the remote peer for traffic matching access list 102.

Crypto map entry “mymap 30” references the dynamic crypto map set “mydynamicmap,” which can be used to process inbound SA negotiation requests that do not match “mymap” entries 10 or 20. In this case, if the peer specifies a transform set that matches one of the transform sets specified in “mydynamicmap,” for a flow permitted by the access list 103, IPsec will accept the request and set up SAs with the remote peer without previously knowing about the remote peer. If the request is accepted, the resulting SAs (and temporary crypto map entry) are established according to the settings specified by the remote peer.

The access list associated with “mydynamicmap 10” is also used as a filter. Inbound packets that match any access list permit statement in this list are dropped for not being IPsec protected. (The same is true for access lists associated with static crypto maps entries.) Outbound packets that match a permit statement without an existing corresponding IPsec SA are also dropped.

```

crypto map mymap 10 ipsec-isakmp
  match address 101
  set transform-set my_t_set1
  set peer 10.0.0.1
  set peer 10.0.0.2
crypto map mymap 20 ipsec-isakmp
  match address 102
  set transform-set my_t_set1 my_t_set2
  set peer 10.0.0.3
crypto map mymap 30 ipsec-isakmp dynamic mydynamicmap
!
crypto dynamic-map mydynamicmap 10
  match address 103
  set transform-set my_t_set1 my_t_set2 my_t_set3

```

The following example configures TED on a Cisco router:

```
crypto map testtag 10 ipsec-isakmp dynamic dmap discover
```

The following example configures a crypto profile to be used as a template for dynamically created crypto maps when IPSec is used to protect an L2TP tunnel:

```
crypto map l2tpsec 10 ipsec-isakmp profile l2tp
```

The following example configures a crypto map for a GDOI group member:

```
crypto map diffint 10 gdoi
  set group diffint
```

## Related Commands

Command	Description
<b>crypto dynamic-map</b>	Creates a dynamic crypto map entry and enters crypto map configuration command mode.
<b>crypto isakmp profile</b>	Audits IPSec user sessions.
<b>crypto map (interface IPSec)</b>	Applies a previously defined crypto map set to an interface.
<b>crypto map local-address</b>	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
<b>match address (IPSec)</b>	Specifies an extended access list for a crypto map entry.
<b>set peer (IPSec)</b>	Specifies an IPSec peer in a crypto map entry.
<b>set pfs</b>	Specifies that IPSec should ask for PFS when requesting new SAs for this crypto map entry, or that IPSec requires PFS when receiving requests for new SAs.
<b>set session-key</b>	Specifies the IPSec session keys within a crypto map entry.
<b>set transform-set</b>	Specifies which transform sets can be used with the crypto map entry.
<b>show crypto map (IPSec)</b>	Displays the crypto map configuration.

## crypto map (interface IPsec)

To apply a previously defined crypto map set to an interface, use the **crypto map** command in interface configuration mode. To remove the crypto map set from the interface, use the **no** form of this command.

```
crypto map map-name [redundancy standby-group-name[stateful]]
```

```
no crypto map [map-name] [redundancy standby-group-name [stateful]]
```

### Syntax Description

<i>map-name</i>	Name that identifies the crypto map set. This is the name assigned when the crypto map was created.  When the <b>no</b> form of the command is used, this argument is optional. Any value supplied for the argument is ignored.
<b>redundancy</b>	(Optional) Defines a backup IP Security (IPSec) peer. Both routers in the standby group are defined by the redundancy <i>standby name</i> and share the same virtual IP address.
<i>standby-group-name</i>	(Optional) Refers to the name of the standby group as defined by Hot Standby Router Protocol (HSRP) standby commands.
<b>stateful</b>	(Optional) Enables IPSec stateful failover for the crypto map.

### Defaults

No crypto maps are assigned to interfaces.

### Command Modes

Interface configuration

### Command History

Release	Modification
11.2	This command was introduced.
12.1(9)E	The <b>redundancy</b> keyword and <i>standby-name</i> argument were added.
12.2(8)T	The <b>redundancy</b> keyword and <i>standby-name</i> argument were integrated into Cisco IOS Release 12.2(8)T.
12.2(11)T	This command was implemented on the Cisco AS5300 and Cisco AS5800 platforms.
12.2(9)YE	The <b>redundancy</b> keyword and <i>standby-name</i> argument were integrated into Cisco IOS Release 12.2(9)YE.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.
12.3(11)T	The <b>stateful</b> keyword was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

**Usage Guidelines**

Use this command to assign a crypto map set to an interface. You must assign a crypto map set to an interface before that interface can provide IPsec services. Only one crypto map set can be assigned to an interface. If multiple crypto map entries have the same map name but a different sequence number, they are considered to be part of the same set and will all be applied to the interface. The crypto map entry that has the lowest sequence number is considered the highest priority and will be evaluated first. A single crypto map set can contain a combination of **ipsec-isakmp** and **ipsec-manual crypto map** entries.

**Note**


---

A crypto map applied to loopback interface is not supported.

---

The standby name must be configured on all devices in the standby group, and the standby address must be configured on at least one member of the group. If the standby name is removed from the router, the IPsec security associations (SAs) will be deleted. If the standby name is added again, regardless of whether the same name or a different name is used, the crypto map (using the **redundancy** option) will have to be reapplied to the interface.

**Note**


---

A virtual IP address must be configured in the standby group to enable either stateless or stateful redundancy.

---

The **stateful** keyword enables stateful failover of IKE and IPsec sessions. Stateful Switchover (SSO) must also be configured for IPsec stateful failover to operate correctly.

**Examples**

The following example shows how all remote Virtual Private Network (VPN) gateways connect to the router via 192.168.0.3:

```
crypto map mymap 1 ipsec-isakmp
  set peer 10.1.1.1
  reverse-route
  set transform-set esp-3des-sha
  match address 102

Interface FastEthernet 0/0
 ip address 192.168.0.2 255.255.255.0
 standby name group1
 standby ip 192.168.0.3
 crypto map mymap redundancy group1

access-list 102 permit ip 192.168.1.0 0.0.0.255 10.0.0.0 0.0.255.255
```

The crypto map on the interface binds this standby address as the local tunnel endpoint for all instances of “mymap” and, at the same time, ensures that stateless HSRP failover is facilitated between an active and standby device that belongs to the same standby group, “group1.”

Reverse route injection (RRI) is also enabled to provide the ability for only the *active* device in the HSRP group to be advertising itself to inside devices as the next hop VPN gateway to the remote proxies. If a failover occurs, routes are deleted on the former active device and created on the new active device.

The following example shows how to configure IPsec stateful failover on the crypto map “to-peer-outside”:

```
crypto map to-peer-outside 10 ipsec-isakmp
  set peer 209.165.200.225
  set transform-set trans1
  match address peer-outside
```

```

interface Ethernet0/0
 ip address 209.165.201.1 255.255.255.224
 standby 1 ip 209.165.201.3
 standby 1 preempt
 standby 1 name HA-out
 standby 1 track Ethernet1/0
 crypto map to-peer-outside redundancy HA-out stateful

```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>crypto map (global IPSec)</b>	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
<b>crypto map local-address</b>	Specifies and names an identifying interface to be used by the crypto map for IPSec traffic.
<b>redundancy inter-device</b>	Configures redundancy and enters inter-device configuration mode.
<b>show crypto map (IPSec)</b>	Displays the crypto map configuration.
<b>standby ip</b>	Assigns an IP address that is to be shared among the members of the HSRP group and owned by the primary IP address.
<b>standby name</b>	Assigns a user-defined group name to the HSRP redundancy group.

# debug aaa accounting

To display information on accountable events as they occur, use the **debug aaa accounting** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

**debug aaa accounting**

**no debug aaa accounting**

## Syntax Description

This command has no arguments or keywords.

## Command Modes

Privileged EXEC

## Usage Guidelines

The information displayed by the **debug aaa accounting** command is independent of the accounting protocol used to transfer the accounting information to a server. Use the **debug tacacs** and **debug radius** protocol-specific commands to get more detailed information about protocol-level issues.

You can also use the **show accounting** command to step through all active sessions and to print all the accounting records for actively accounted functions. The **show accounting** command allows you to display the active “accountable events” on the system. It provides systems administrators a quick look at what is happening, and may also be useful for collecting information in the event of a data loss of some kind on the accounting server. The **show accounting** command displays additional data on the internal state of the authentication, authorization, and accounting (AAA) security system if **debug aaa accounting** is turned on as well.

## Examples

The following is sample output from the **debug aaa accounting** command:

```
Router# debug aaa accounting

16:49:21: AAA/ACCT: EXEC acct start, line 10
16:49:32: AAA/ACCT: Connect start, line 10, glare
16:49:47: AAA/ACCT: Connection acct stop:
task_id=70 service=exec port=10 protocol=telnet address=172.31.3.78 cmd=glare bytes_in=308
bytes_out=76 paks_in=45 paks_out=54 elapsed_time=14
```

## Related Commands

Command	Description
<b>debug aaa authentication</b>	Displays information on accountable events as they occur.
<b>debug aaa authorization</b>	Displays information on AAA/TACACS+ authorization.
<b>debug radius</b>	Displays information associated with the RADIUS.
<b>debug tacacs</b>	Displays information associated with the TACACS.

# debug aaa pod

To display debug messages related to packet of disconnect (POD) packets, use the **debug aaa pod** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

**debug aaa pod**

**no debug aaa pod**

## Syntax Description

This command has no keywords or arguments.

## Defaults

Debugging for POD packets is not enabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.1(3)T	This command was introduced.
12.2(2)XB	Support for the voice applications as well as support for the Cisco AS5350, Cisco AS5400 and the Cisco 3600 series was added.
12.2(2)XB1	Support for the Cisco AS5800 was added.
12.2(11)T	Support for the Cisco AS5850 was added. This command was integrated into Cisco IOS Release 12.2(11)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

## Examples

The following example shows output from a successful POD request when using the **show debug** command:

```
Router# debug aaa pod

AAA POD packet processing debugging is on

Router# show debug

General OS:
  AAA POD packet processing debugging is on
Router#
Apr 25 17:15:59.318:POD:172.19.139.206 request queued
Apr 25 17:15:59.318:voice_pod_request:
Apr 25 17:15:59.318:voip_populate_pod_attr_list:
Apr 25 17:15:59.318:voip_pod_get_guid:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=50
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr=h323-conf-id
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=50 value_len=35
Apr 25 17:15:59.318:voip_pod_get_guid:conf-id=FFA7785F F7F607BB
00000000 993FB1F4 n_bytes=35
Apr 25 17:15:59.318:voip_pod_get_guid:GUID = FFA7785F F7F607BB 00000000
993FB1F4
```

## ■ debug aaa pod

```
Apr 25 17:15:59.318:voip_populate_pod_attr_list:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=23
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr=h323-originate
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=23 value_len=6
Apr 25 17:15:59.318:voip_get_call_direction:
Apr 25 17:15:59.318:voip_get_call_direction:returning answer
Apr 25 17:15:59.318:voip_eval_pod_attr:
Apr 25 17:15:59.318:cc_api_trigger_disconnect:
Apr 25 17:15:59.322:POD:Sending ACK to 172.19.139.206/1700
Apr 25 17:15:59.322:voip_pod_clean:
```

---

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>aaa pod server</b>	Enables the POD feature.

---



# debug condition

To filter debugging output for certain **debug** commands on the basis of specified conditions, use the **debug condition** command in privileged EXEC mode. To remove the specified condition, use the **no** form of this command.

```
debug condition { called dial-string | caller dial-string | calling tid/imsi string | domain
domain-name | ip ip-address | mac-address hexadecimal-MAC-address | portbundle ip
ip-address bundle bundle-number | session-id session-number | username username | vcid
vc-id }
```

```
no debug condition { condition-id | all }
```

Syntax	Description
<b>called</b> <i>dial-string</i>	Filters output on the basis of the called party number.
<b>caller</b> <i>dial-string</i>	Filters output on the basis of the calling party number.
<b>calling</b> <i>tid/imsi string</i>	Filters debug messages for general packet radio service (GPRS) tunneling protocol (GTP) processing on the gateway GPRS support node (GGSN) based on the tunnel identifier (TID) or international mobile system identifier (IMSI) in a Packet Data Protocol (PDP) Context Create Request message.
<b>domain</b> <i>domain-name</i>	Filters output on the basis of the specified domain.
<b>ip</b> <i>ip-address</i>	Filters output on the basis of the specified IP address.
<b>mac-address</b> <i>hexadecimal-MAC-address</i>	Filters messages on the specified MAC address.
<b>portbundle ip</b> <i>IP-address</i>	Filters output on the basis of the port-bundle host key (PBHK) that uniquely identifies the session.
<b>bundle</b> <i>bundle-number</i>	Specifies the port bundle.
<b>session-id</b> <i>session-number</i>	Filters output on the specified Intelligent Service Architecture (ISA) session identifier.
<b>username</b> <i>username</i>	Filters output on the basis of the specified username.
<b>vcid</b> <i>vc-id</i>	Filters output on the basis of the specified VC ID.
<i>condition-id</i>	Removes the condition indicated.
<b>all</b>	Removes all debugging conditions, and conditions specified by the <b>debug condition interface</b> command. Use this keyword to disable conditional debugging and reenables debugging for all interfaces.

## Defaults

All debugging messages for enabled protocol-specific **debug** commands are generated.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
11.3(2)AA	This command was introduced.
12.0(23)S	This command was integrated into Cisco IOS Release 12.0(23)S. This command was updated with the <b>vcid</b> and <b>ip</b> keywords to support the debugging of Any Transport over MPLS (AToM) messages.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.3(2)XB	This command was introduced on the GGSN.
12.3(8)T	The <b>calling</b> keyword and <i>tid/imsi string</i> argument were added.
12.2(28)SB	The ability to filter output on the following conditions was added: domain, MAC address, PBHK, and ISA session ID.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

## Usage Guidelines

Use the **debug condition** command to restrict the debug output for some commands. If any **debug condition** commands are enabled, output is generated only for interfaces associated with the specified keyword. In addition, this command enables debugging output for conditional debugging events. Messages are displayed as different interfaces meet specific conditions.

If multiple **debug condition** commands are enabled, output is displayed if at least one condition matches. All the conditions do not need to match.

The **no** form of this command removes the debug condition specified by the condition identifier. The condition identifier is displayed after you use a **debug condition** command or in the output of the **show debug condition** command. If the last condition is removed, debugging output resumes for all interfaces. You will be asked for confirmation before removing the last condition or all conditions.

Not all debugging output is affected by the **debug condition** command. Some commands generate output whenever they are enabled, regardless of whether they meet any conditions.

The following components are supported for Intelligent Service Architecture (ISA) distributed conditional debugging:

- Authentication, authorization, and accounting (AAA) and RADIUS
- ATM components
- Feature Manager
- Policy Manager
- PPP
- PPP over Ethernet (PPPoE)
- Session Manager
- Virtual Private Dialup Network (VPDN)

Ensure that you enable TID/IMSI-based conditional debugging by entering **debug condition calling** before configuring **debug gprs gtp** and **debug gprs charging**. In addition, ensure that you disable the **debug gprs gtp** and **debug gprs charging** commands using the **no debug all** command before disabling conditional debugging using the **no debug condition** command. This will prevent a flood of debugging messages when you disable conditional debugging.

## Examples

### Example 1

In the following example, the router displays debugging messages only for interfaces that use a username of “user1”. The condition identifier displayed after the command is entered identifies this particular condition.

```
Router# debug condition username user1
```

```
Condition 1 set
```

### Example 2

The following example specifies that the router should display debugging messages only for VC 1000:

```
Router# debug condition vcid 1000
```

```
Condition 1 set
```

```
01:12:32: 1000 Debug: Condition 1, vcid 1000 triggered, count 1
```

```
01:12:32: 1000 Debug: Condition 1, vcid 1000 triggered, count 1
```

The following example enables other debugging commands. These debugging commands will only display information for VC 1000.

```
Router# debug mpls l2transport vc event
```

```
AToM vc event debugging is on
```

```
Router# debug mpls l2transport vc fsm
```

```
AToM vc fsm debugging is on
```

The following commands shut down the interface on which VC 1000 is established.

```
Router(config)# interface s3/1/0
```

```
Router(config-if)# shut
```

The debugging output shows the change to the interface where VC 1000 is established.

```
01:15:59: AToM MGR [13.13.13.13, 1000]: Event local down, state changed from established to remote ready
```

```
01:15:59: AToM MGR [13.13.13.13, 1000]: Local end down, vc is down
```

```
01:15:59: AToM SMGR [13.13.13.13, 1000]: Processing imposition update, vc_handle 6227BCF0, update_action 0, remote_vc_label 18
```

```
01:15:59: AToM SMGR [13.13.13.13, 1000]: Imposition Disabled
```

```
01:15:59: AToM SMGR [13.13.13.13, 1000]: Processing disposition update, vc_handle 6227BCF0, update_action 0, local_vc_label 755
```

```
01:16:01:%LINK-5-CHANGED: Interface Serial3/1/0, changed state to administratively down
```

```
01:16:02:%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/1/0, changed state to down
```

## Related Commands

Command	Description
<b>debug condition interface</b>	Limits output for some debugging commands based on the interfaces.

# debug ip mobile

To display IP mobility activities, use the **debug ip mobile** command in privileged EXEC mode.

```
debug ip mobile [advertise | host [access-list-number] | local-area | redundancy |
udp-tunneling]
```

## Syntax Description

<b>advertise</b>	(Optional) Advertisement information.
<b>host</b>	(Optional) The mobile node host.
<i>access-list-number</i>	(Optional) The number of an IP access list.
<b>local-area</b>	(Optional) The local area.
<b>redundancy</b>	(Optional) Redundancy activities.
<b>udp-tunneling</b>	(Optional) User Datagram Protocol (UDP) tunneling activities.

## Defaults

No default behavior or values.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
12.0(1)T	This command was introduced.
12.0(2)T	The <b>standby</b> keyword was added.
12.2(8)T	The <b>standby</b> keyword was replaced by the <b>redundancy</b> keyword.
12.2(13)T	This command was enhanced to display information about foreign agent reverse tunnels and the mobile networks attached to the mobile router.
12.3(8)T	The <b>udp-tunneling</b> keyword was added and the command was enhanced to display information about NAT traversal using UDP tunneling.
12.3(7)XJ	This command was enhanced to include the Resource Management capability.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

## Usage Guidelines

Use the **debug ip mobile redundancy** command to troubleshoot redundancy problems.

No per-user debugging output is shown for mobile nodes using the network access identifier (NAI) for the **debug ip mobile host** command. Debugging of specific mobile nodes using an IP address is possible through the access list.

## Examples

The following is sample output from the **debug ip mobile** command when foreign agent reverse tunneling is enabled:

```
MobileIP:MN 14.0.0.30 deleted from ReverseTunnelTable of Ethernet2/1(Entries 0)
```

The following is sample output from the **debug ip mobile advertise** command:

```
Router# debug ip mobile advertise

MobileIP: Agent advertisement sent out Ethernet1/2: type=16, len=10, seq=1,
lifetime=36000,
flags=0x1400 (rbhFmGv-rsv-),
Care-of address: 68.0.0.31
Prefix Length ext: len=1 (8 )
FA Challenge value:769C808D
```

Table 2 describes the significant fields shown in the display.

**Table 2** *debug ip mobile advertise Field Descriptions*

Field	Description
type	Type of advertisement.
len	Length of extension (in bytes).
seq	Sequence number of this advertisement.
lifetime	Lifetime (in seconds).
flags	Capital letters represent bits that are set; lowercase letters represent unset bits.
Care-of address	IP address.
Prefix Length ext	Number of prefix lengths advertised. This is the bits in the mask of the interface sending this advertisement. Used for roaming detection.
FA Challenge value	Foreign Agent challenge value (randomly generated by the foreign agent.)

The following is sample output from the **debug ip mobile host** command:

```
Router# debug ip mobile host

MobileIP: HA received registration for MN 20.0.0.6 on interface Ethernet1 using COA
68.0.0.31 HA 66.0.0.5 lifetime 30000 options sbdmgvT
MobileIP: Authenticated FA 68.0.0.31 using SPI 110 (MN 20.0.0.6)
MobileIP: Authenticated MN 20.0.0.6 using SPI 300

MobileIP: HA accepts registration from MN 20.0.0.6
MobileIP: Mobility binding for MN 20.0.0.6 updated
MobileIP: Roam timer started for MN 20.0.0.6, lifetime 30000
MobileIP: MH auth ext added (SPI 300) in reply to MN 20.0.0.6
MobileIP: HF auth ext added (SPI 220) in reply to MN 20.0.0.6

MobileIP: HA sent reply to MN 20.0.0.6
```

The following is sample output from the **debug ip mobile redundancy** command. In this example, the active home agent receives a registration request from mobile node 20.0.0.2 and sends a binding update to peer home agent 1.0.0.2:

```
MobileIP:MN 20.0.0.2 - sent BindUpd to HA 1.0.0.2 HAA 20.0.0.1
MobileIP:HA standby maint started - cnt 1
MobileIP:MN 20.0.0.2 - sent BindUpd id 3780410816 cnt 0 elapsed 0
adjust -0 to HA 1.0.0.2 in grp 1.0.0.10 HAA 20.0.0.1
```

In this example, the standby home agent receives a binding update for mobile node 20.0.0.2 sent by the active home agent:

```
MobileIP:MN 20.0.0.2 - HA rcv BindUpd from 1.0.0.3 HAA 20.0.0.1
```

The following is sample output from the **debug ip mobile udp-tunneling** command and displays the registration, authentication, and establishment of UDP tunneling of a mobile node (MN) with a foreign agent (FA):

```

Dec 31 12:34:25.707: UDP: rcvd src=10.10.10.10(434),dst=10.30.30.1(434), length=54
Dec 31 12:34:25.707: MobileIP: ParseRegExt type MHAE(32) addr 2000FEEC end 2000FF02
Dec 31 12:34:25.707: MobileIP: ParseRegExt skipping 10 to next
Dec 31 12:34:25.707: MobileIP: FA rcv registration for MN 10.10.10.10 on Ethernet2/2 using
COA 10.30.30.1 HA 10.10.10.100 lifetime 65535 options sbdmg-T-identification
C1BCOD4FB01AC0D8
Dec 31 12:34:25.707: MobileIP: Ethernet2/2 glean 10.10.10.10 accepted
Dec 31 12:34:25.707: MobileIP: Registration request byte count = 74
Dec 31 12:34:25.707: MobileIP: FA queued MN 10.10.10.10 in register table
Dec 31 12:34:25.707: MobileIP: Visitor registration timer started for MN 10.10.10.10,
lifetime 120
Dec 31 12:34:25.707: MobileIP: Adding UDP Tunnel req extension
Dec 31 12:34:25.707: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:25.707: MobileIP: MN 10.10.10.10 FHAE added to HA 10.10.10.100 using SPI 1000
Dec 31 12:34:25.707: MobileIP: FA forwarded registration for MN 10.10.10.10 to HA
10.10.10.100
Dec 31 12:34:25.715: UDP: rcvd src=10.10.10.100(434), dst=10.30.30.1(434), length=94
Dec 31 12:34:25.715: MobileIP: ParseRegExt type NVSE(134) addr 20010B28 end 20010B6A
Dec 31 12:34:25.715: MobileIP: ParseRegExt type MN-config NVSE(14) subtype 1 (MN prefix
length) prefix length (24)
Dec 31 12:34:25.715: MobileIP: ParseRegExt skipping 12 to next
Dec 31 12:34:25.715: MobileIP: ParseRegExt type MHAE(32) addr 20010B36 end 20010B6A
Dec 31 12:34:25.715: MobileIP: ParseRegExt skipping 10 to next
Dec 31 12:34:25.715: MobileIP: ParseRegExt type UDPTUNREPE(44) addr 20010B4C end 20010B6A
Dec 31 12:34:25.715: Parsing UDP Tunnel Reply Extension - length 6
Dec 31 12:34:25.715: MobileIP: ParseRegExt skipping 6 to next
Dec 31 12:34:25.715: MobileIP: ParseRegExt type FHAE(34) addr 20010B54 end 20010B6A
Dec 31 12:34:25.715: MobileIP: ParseRegExt skipping 20 to next
Dec 31 12:34:25.715: MobileIP: FA rcv accept (0) reply for MN 10.10.10.10 on Ethernet2/3
using HA 10.10.10.100 lifetime 65535
Dec 31 12:34:25.719: MobileIP: Authenticating HA 10.10.10.100 using SPI 1000
Dec 31 12:34:25.719: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:25.719: MobileIP: Authenticated HA 10.10.10.100 using SPI 1000 and 16 byte
key
Dec 31 12:34:25.719: MobileIP: HA accepts UDP Tunneling
Dec 31 12:34:25.719: MobileIP: Update visitor table for MN 10.10.10.10
Dec 31 12:34:25.719: MobileIP: Enabling UDP Tunneling
Dec 31 12:34:25.719: MobileIP: Tunnel0 (MIPUDP/IP) created with src 10.30.30.1 dst
10.10.10.100
Dec 31 12:34:25.719: MobileIP: Setting up UDP Keep-Alive Timer for tunnel 10.30.30.1:0 -
10.10.10.100:0 with keep-alive 30
Dec 31 12:34:25.719: MobileIP: Starting the tunnel keep-alive timer
Dec 31 12:34:25.719: MobileIP: ARP entry for MN 10.10.10.10 using 10.10.10.10 inserted on
Ethernet2/2
Dec 31 12:34:25.719: MobileIP: FA route add 10.10.10.10 successful. Code = 0
Dec 31 12:34:25.719: MobileIP: MN 10.10.10.10 added to ReverseTunnelTable of Ethernet2/2
(Entries 1)
Dec 31 12:34:25.719: MobileIP: FA dequeued MN 10.10.10.10 from register table
Dec 31 12:34:25.719: MobileIP: MN 10.10.10.10 using 10.10.10.10 visiting on Ethernet2/2
Dec 31 12:34:25.719: MobileIP: Reply in for MN 10.10.10.10 using 10.10.10.10, accepted
Dec 31 12:34:25.719: MobileIP: registration reply byte count = 84
Dec 31 12:34:25.719: MobileIP: FA forwarding reply to MN 10.10.10.10 (10.10.10.10 mac
0060.70ca.f021)
Dec 31 12:34:26.095: MobileIP: agent advertisement byte count = 48
Dec 31 12:34:26.095: MobileIP: Agent advertisement sent out Ethernet2/2: type=16, len=10,
seq=55, lifetime=65535, flags=0x1580(rbhFmG-TU),
Dec 31 12:34:26.095: Care-of address: 10.30.30.1
Dec 31 12:34:26.719: MobileIP: swif coming up Tunnel0

```

```
!
Dec 31 12:34:35.719: UDP: sent src=10.30.30.1(434), dst=10.10.10.100(434)
Dec 31 12:34:35.719: UDP: rcvd src=10.10.10.100(434), dst=10.30.30.1(434), length=32d0
```

The following is sample output from the **debug ip mobile udp-tunneling** command and displays the registration, authentication, and establishment of UDP tunneling of a MN with a home agent (HA):

```
Dec 31 12:34:26.167: MobileIP: ParseRegExt skipping 20 to next
Dec 31 12:34:26.167: MobileIP: ParseRegExt type UDPTUNREQE(144) addr 2001E762 end 2001E780
Dec 31 12:34:26.167: MobileIP: Parsing UDP Tunnel Request Extension - length 6
Dec 31 12:34:26.167: MobileIP: ParseRegExt skipping 6 to next
Dec 31 12:34:26.167: MobileIP: ParseRegExt type FHAE(34) addr 2001E76A end 2001E780
Dec 31 12:34:26.167: MobileIP: ParseRegExt skipping 20 to next
Dec 31 12:34:26.167: MobileIP: HA 167 rcv registration for MN 10.10.10.10 on Ethernet2/1
  using HomeAddr 10.10.10.10 COA 10.30.30.1 HA 10.10.10.100 lifetime 65535 options
  sbdmg-T-identification C1BC0D4FB01AC0D8
Dec 31 12:34:26.167: MobileIP: NAT detected SRC:10.10.10.50 COA: 10.30.30.1
Dec 31 12:34:26.167: MobileIP: UDP Tunnel Request accepted 10.10.10.50:434
Dec 31 12:34:26.167: MobileIP: Authenticating FA 10.30.30.1 using SPI 1000
Dec 31 12:34:26.167: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:26.167: MobileIP: Authentication algorithm MD5 and truncated key
Dec 31 12:34:26.167: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:26.167: MobileIP: Authenticated FA 10.30.30.1 using SPI 1000 and 16 byte key
Dec 31 12:34:26.167: MobileIP: Authenticating MN 10.10.10.10 using SPI 1000
Dec 31 12:34:26.167: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:26.167: MobileIP: Authentication algorithm MD5 and truncated key
Dec 31 12:34:26.167: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:26.167: MobileIP: Authenticated MN 10.10.10.10 using SPI 1000 and 16 byte key
Dec 31 12:34:26.167: MobileIP: Mobility binding for MN 10.10.10.10 created
Dec 31 12:34:26.167: MobileIP: NAT detected for MN 10.10.10.10. Terminating tunnel on
  10.10.10.50
Dec 31 12:34:26.167: MobileIP: Tunnel0 (MIPUDP/IP) created with src 10.10.10.100 dst
  10.10.10.50
Dec 31 12:34:26.167: MobileIP: Setting up UDP Keep-Alive Timer for tunnel 10.10.10.100:0 -
  10.10.10.50:0 with keep-alive 30
Dec 31 12:34:26.167: MobileIP: Starting the tunnel keep-alive timer
Dec 31 12:34:26.167: MobileIP: MN 10.10.10.10 Insert route for 10.10.10.10/255.255.255.255
  via gateway 10.10.10.50 on Tunnel0
Dec 31 12:34:26.167: MobileIP: MN 10.10.10.10 is now roaming
Dec 31 12:34:26.171: MobileIP: Gratuitous ARPs sent for MN 10.10.10.10 MAC 0002.fca5.bc39
Dec 31 12:34:26.171: MobileIP: Mask for address is 24
Dec 31 12:34:26.171: MobileIP: HA accepts registration from MN 10.10.10.10
Dec 31 12:34:26.171: MobileIP: Dynamic and Static Network Extension Length 0 - 0
Dec 31 12:34:26.171: MobileIP: Composed mobile network extension length:0
Dec 31 12:34:26.171: MobileIP: Added prefix length vse in reply
Dec 31 12:34:26.171: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:26.171: MobileIP: MN 10.10.10.10 MHAE added to MN 10.10.10.10 using SPI 1000
Dec 31 12:34:26.171: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:26.171: MobileIP: MN 10.10.10.10 FHAE added to FA 10.10.10.50 using SPI 1000
Dec 31 12:34:26.171: MobileIP: MN 10.10.10.10 - HA sent reply to 10.10.10.50
Dec 31 12:34:26.171: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:26.171: MobileIP: MN 10.10.10.10 HHAE added to HA 10.10.10.3 using SPI 1000
Dec 31 12:34:26.175: MobileIP: ParseRegExt type CVSE(38) addr 2000128C end 200012AE
Dec 31 12:34:26.175: MobileIP: ParseRegExt type HA red. version CVSE(6)
Dec 31 12:34:26.175: MobileIP: ParseRegExt skipping 8 to next
Dec 31 12:34:26.175: MobileIP: ParseRegExt type HHAE(35) addr 20001298 end 200012AE
Dec 31 12:34:26.175: MobileIP: ParseRegExt skipping 20 to next
Dec 31 12:34:26.175: MobileIP: Authenticating HA 10.10.10.3 using SPI 1000
Dec 31 12:34:26.175: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:26.175: MobileIP: Authentication algorithm MD5 and truncated key
Dec 31 12:34:26.175: MobileIP: Authentication algorithm MD5 and 16 byte key
Dec 31 12:34:26.175: MobileIP: Authenticated HA 10.10.10.3 using SPI 1000 and 16 byte key
Dec 31 12:34:27.167: MobileIP: swif coming up Tunnel0d0
```

# debug ip mobile advertise

The **debug ip mobile advertise** command was consolidated with the **debug ip mobile** command. See the description of the **debug ip mobile** command in the “Debug Commands” chapter for more information.

To display advertisement information, use the **debug ip mobile advertise EXEC** command .

**debug ip mobile advertise**

**no debug ip mobile advertise**

**Syntax Description** This command has no arguments or keywords.

**Defaults** No default values.

**Command Modes** EXEC mode

Command History	Release	Modification
	12.0(1)T	This command was introduced.

**Examples** The following is sample output from the **debug ip mobile advertise** command. [Table 3](#) describes significant fields shown in the display.

```
Router# debug ip mobile advertise
```

```
MobileIP: Agent advertisement sent out Ethernet1/2: type=16, len=10, seq=1,
lifetime=36000,
flags=0x1400 (rbhFmGv-rsv-),
Care-of address: 14.0.0.31
Prefix Length ext: len=1 (8 )
```

**Table 3** Debug IP Mobile Advertise Field Descriptions

Field	Description
type	Type of advertisement.
len	Length of extension in bytes.
seq	Sequence number of this advertisement.
lifetime	Lifetime in seconds.
flags	Capital letters represent bits that are set, lower case letters represent unset bits.
Care-of address	IP address.
Prefix Length ext	Number of prefix lengths advertised. This is the bits in the mask of the interface sending this advertisement. Used for roaming detection.



# debug ip mobile host

The **debug ip mobile host** command was consolidated with the **debug ip mobile** command. See the description of the **debug ip mobile** command in the “Debug Commands” chapter for more information.

Use the **debug ip mobile host EXEC** command to display IP mobility events.

```
debug ip mobile host [[access-list-number]][nai {NAI username | username@realm}]
```

```
no debug ip mobile host [[access-list-number]][nai {NAI username | username@realm}]
```

## Syntax Description

<b>host</b>	(Optional) The mobile node host. [ <i>access-list-number</i> ]
<b>nai {NAI username   username@realm}</b>	(Optional) Mobile host identified by NAI.

## Defaults

No default values.

## Command History

Release	Modification
12.0(1)T	This command was introduced.

## Examples

The following is sample output from the **debug ip mobile host** command:

```
Router# debug ip mobile host

MobileIP: HA received registration for MN 10.0.0.6 on interface Ethernet1 using COA
14.0.0.31 HA 15.0.0.5 lifetime 30000 options sbdmgvT
MobileIP: Authenticated FA 15.0.0.31 using SPI 110 (MN 20.0.0.6)
MobileIP: Authenticated MN 11.0.0.6 using SPI 300

MobileIP: HA accepts registration from MN 11.0.0.6
MobileIP: Mobility binding for MN 11.0.0.6 updated
MobileIP: Roam timer started for MN 11.0.0.6, lifetime 30000
MobileIP: MH auth ext added (SPI 300) in reply to MN 11.0.0.6
MobileIP: HF auth ext added (SPI 220) in reply to MN 11.0.0.6

MobileIP: HA sent reply to MN 11.0.0.6
```

# debug ip mobile redundancy

The **debug ip mobile redundancy** command was consolidated with the **debug ip mobile** command. See the description of the **debug ip mobile** command in the “Debug Commands” chapter for more information.

Use the **debug ip mobile redundancy EXEC** command to display IP mobility events.

**debug ip mobile redundancy**

**no debug ip mobile redundancy**

---

**Syntax Description** This command has no keywords or arguments.

---

**Defaults** No default values.

---

Command History	Release	Modification
	12.0(1)T	This command was introduced.

---



---

**Examples** The following is sample output from the debug ip mobile redundancy command:

```
Router# debug ip mobile redundancy

00:19:21: MobileIP: Adding MN service flags to bindupdate
00:19:21: MobileIP: Adding MN service flags 0 init registration flags 1
00:19:21: MobileIP: Adding a hared version cvse - bindupdate
00:19:21: MobileIP: HARelayBindUpdate version number 2MobileIP: MN 14.0.0.20 - sent
BindUpd to HA 11.0.0.3 HAA 11.0.0.4
00:19:21: MobileIP: HA standby maint started - cnt 1
00:19:21: MobileIP: MN 14.0.0.20 - HA rcv BindUpdAck accept from 11.0.0.3 HAA 11.0.0.4
00:19:22: MobileIP: HA standby maint started - cnt 1
```

# debug radius

To display information associated with RADIUS, use the **debug radius** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

**debug radius [brief | hex]**

**no debug radius [brief | hex]**

## Syntax Description

<b>brief</b>	(Optional) Displays abbreviated debug output.
<b>hex</b>	(Optional) Displays debugging output in hexadecimal notation.

## Defaults

Debugging output in ASCII format is enabled.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
11.2(1)T	This command was introduced.
12.2(11)T	The <b>brief</b> and <b>hex</b> keywords were added. The default output format became ASCII rather than hexadecimal.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

## Usage Guidelines

RADIUS is a distributed security system that secures networks against unauthorized access. Cisco supports RADIUS under the authentication, authorization, and accounting (AAA) security system. When RADIUS is used on the router, you can use the **debug radius** command to display detailed debugging and troubleshooting information in ASCII format. Use the **debug radius brief** command for abbreviated output displaying client/server interaction and minimum packet information. Use the **debug radius hex** command to display packet dump information that has not been truncated in hex format.

## Examples

The following is sample output from the **debug radius** command:

```
Router# debug radius

Radius protocol debugging is on
Radius packet hex dump debugging is off
Router#
00:02:50: RADIUS: ustruct sharecount=3
00:02:50: Radius: radius_port_info() success=0 radius_nas_port=1
00:02:50: RADIUS: Initial Transmit ISDN 0:D:23 id 0 10.0.0.1:1824, Accounting-Request, len
358
00:02:50: RADIUS:  NAS-IP-Address      [4]  6  10.0.0.0
00:02:50: RADIUS:  Vendor, Cisco          [26] 19 VT=02 TL=13 ISDN 0:D:23
00:02:50: RADIUS:  NAS-Port-Type       [61] 6  Async
00:02:50: RADIUS:  User-Name            [1] 12  "4085554206"
00:02:50: RADIUS:  Called-Station-Id  [30] 7  "52981"
```

```

00:02:50: RADIUS: Calling-Station-Id [31] 12 "4085554206"
00:02:50: RADIUS: Acct-Status-Type [40] 6 Start
00:02:50: RADIUS: Service-Type [6] 6 Login
00:02:50: RADIUS: Vendor, Cisco [26] 27 VT=33 TL=21 h323-gw-id=5300_43.
00:02:50: RADIUS: Vendor, Cisco [26] 55 VT=01 TL=49
h323-incoming-conf-id=8F3A3163 B4980003 0 29BD0
00:02:50: RADIUS: Vendor, Cisco [26] 31 VT=26 TL=25 h323-call-origin=answer
00:02:50: RADIUS: Vendor, Cisco [26] 32 VT=27 TL=26 h323-call-type=Telephony
00:02:50: RADIUS: Vendor, Cisco [26] 57 VT=25 TL=51 h323-setup-time=*16:02:48.681
PST Fri Dec 31 1999
00:02:50: RADIUS: Vendor, Cisco [26] 46 VT=24 TL=40 h323-conf-id=8F3A3163
B4980003 0 29BD0
00:02:50: RADIUS: Acct-Session-Id [44] 10 "00000002"
00:02:50: RADIUS: Delay-Time [41] 6 0
00:02:51: RADIUS: Received from id 0 1.7.157.1:1824, Accounting-response, len 20
00:02:51: %ISDN-6-CONNECT: Interface Serial0:22 is now connected to 4085274206
00:03:01: RADIUS: ustruct sharecount=3
00:03:01: Radius: radius_port_info() success=0 radius_nas_port=1
00:03:01: RADIUS: Initial Transmit ISDN 0:D:23 id 1 1.7.157.1:1823, Access-Request, len
171
00:03:01: RADIUS: NAS-IP-Address [4] 6 10.0.0.0
00:03:01: RADIUS: Vendor, Cisco [26] 19 VT=02 TL=13 ISDN 0:D:23
00:03:01: RADIUS: NAS-Port-Type [61] 6 Async
00:03:01: RADIUS: User-Name [1] 8 "123456"
00:03:01: RADIUS: Vendor, Cisco [26] 46 VT=24 TL=40 h323-conf-id=8F3A3163
B4980003 0 29BD0
00:03:01: RADIUS: Calling-Station-Id [31] 12 "4085554206"
00:03:01: RADIUS: User-Password [2] 18 *
00:03:01: RADIUS: Vendor, Cisco [26] 36 VT=01 TL=30 h323-ivr-out=transactionID:0
00:03:01: RADIUS: Received from id 1 1.7.157.1:1823, Access-Accept, len 115
00:03:01: RADIUS: Service-Type [6] 6 Login
00:03:01: RADIUS: Vendor, Cisco [26] 29 VT=101 TL=23 h323-credit-amount=45
00:03:01: RADIUS: Vendor, Cisco [26] 27 VT=102 TL=21 h323-credit-time=33
00:03:01: RADIUS: Vendor, Cisco [26] 26 VT=103 TL=20 h323-return-code=0
00:03:01: RADIUS: Class [25] 7 6C6F63616C
00:03:01: RADIUS: saved authorization data for user 62321E14 at 6233D258
00:03:13: %ISDN-6-DISCONNECT: Interface Serial0:22 disconnected from 4085274206, call
lasted 22 seconds
00:03:13: RADIUS: ustruct sharecount=2
00:03:13: Radius: radius_port_info() success=0 radius_nas_port=1
00:03:13: RADIUS: Sent class "local" at 6233D2C4 from user 62321E14
00:03:13: RADIUS: Initial Transmit ISDN 0:D:23 id 2 1.7.157.1:1824, Accounting-Request,
len 775
00:03:13: RADIUS: NAS-IP-Address [4] 6 10.0.0.0
00:03:13: RADIUS: Vendor, Cisco [26] 19 VT=02 TL=13 ISDN 0:D:23
00:03:13: RADIUS: NAS-Port-Type [61] 6 Async
00:03:13: RADIUS: User-Name [1] 8 "123456"
00:03:13: RADIUS: Called-Station-Id [30] 7 "52981"
00:03:13: RADIUS: Calling-Station-Id [31] 12 "4085274206"
00:03:13: RADIUS: Acct-Status-Type [40] 6 Stop
00:03:13: RADIUS: Class [25] 7 6C6F63616C
00:03:13: RADIUS: Undebuggable [45] 6 00000001
00:03:13: RADIUS: Service-Type [6] 6 Login
00:03:13: RADIUS: Vendor, Cisco [26] 27 VT=33 TL=21 h323-gw-id=5300_43.
00:03:13: RADIUS: Vendor, Cisco [26] 55 VT=01 TL=49
h323-incoming-conf-id=8F3A3163 B4980003 0 29BD0
00:03:13: RADIUS: Vendor, Cisco [26] 31 VT=26 TL=25 h323-call-origin=answer
00:03:13: RADIUS: Vendor, Cisco [26] 32 VT=27 TL=26 h323-call-type=Telephony
00:03:13: RADIUS: Vendor, Cisco [26] 57 VT=25 TL=51 h323-setup-time=*16:02:48.681
PST Fri Dec 31 1999
00:03:13: RADIUS: Vendor, Cisco [26] 59 VT=28 TL=53
h323-connect-time=*16:02:48.946 PST Fri Dec 31 1999
00:03:13: RADIUS: Vendor, Cisco [26] 62 VT=29 TL=56in=0
00:03:13: RADIUS: Vendor, Cisco [26] 23 VT=01 TL=17 pre-bytes-out=0

```

```

00:03:13: RADIUS: Vendor, Cisco [26] 21 VT=01 TL=15 pre-paks-in=0
00:03:13: RADIUS: Vendor, Cisco [26] 22 VT=01 TL=16 pre-paks-out=0
00:03:13: RADIUS: Vendor, Cisco [26] 22 VT=01 TL=16 nas-rx-speed=0
00:03:13: RADIUS: Vendor, Cisco [26] 22 VT=01 TL=16 nas-tx-speed=0
00:03:13: RADIUS: Delay-Time [41] 6 0
00:03:13: RADIUS: Received from id 2 1.7.157.1:1824, Accounting-response, len 20
h323-disconnect-time=*16:03:11.306 PST Fri Dec 31 1999
00:03:13: RADIUS: Vendor, Cisco [26] 32 VT=30 TL=26 h323-disconnect-cause=10
00:03:13: RADIUS: Vendor, Cisco [26] 28 VT=31 TL=22 h323-voice-quality=0
00:03:13: RADIUS: Vendor, Cisco [26] 46 VT=24 TL=40 h323-conf-id=8F3A3163
B4980003 0 29BD0
00:03:13: RADIUS: Acct-Session-Id [44] 10 "00000002"
00:03:13: RADIUS: Acct-Input-Octets [42] 6 0
00:03:13: RADIUS: Acct-Output-Octets [43] 6 88000
00:03:13: RADIUS: Acct-Input-Packets [47] 6 0
00:03:13: RADIUS: Acct-Output-Packets [48] 6 550
00:03:13: RADIUS: Acct-Session-Time [46] 6 22
00:03:13: RADIUS: Vendor, Cisco [26] 30 VT=01 TL=24 subscriber=RegularLine
00:03:13: RADIUS: Vendor, Cisco [26] 35 VT=01 TL=29 h323-ivr-out=Tariff:Unknown
00:03:13: RADIUS: Vendor, Cisco [26] 22 VT=01 TL=16 pre-bytes-

```

The following is sample output from the **debug radius brief** command:

```
Router# debug radius brief
```

```

Radius protocol debugging is on
Radius packet hex dump debugging is off
Radius protocol in brief format debugging is on
00:05:21: RADIUS: Initial Transmit ISDN 0:D:23 id 6 10.0.0.1:1824, Accounting-Request, len
358
00:05:21: %ISDN-6-CONNECT: Interface Serial0:22 is now connected to 4085274206
00:05:26: RADIUS: Retransmit id 6
00:05:31: RADIUS: Tried all servers.
00:05:31: RADIUS: No valid server found. Trying any viable server
00:05:31: RADIUS: Tried all servers.
00:05:31: RADIUS: No response for id 7
00:05:31: RADIUS: Initial Transmit ISDN 0:D:23 id 8 10.0.0.0:1823, Access-Request, len 171
00:05:36: RADIUS: Retransmit id 8
00:05:36: RADIUS: Received from id 8 1.7.157.1:1823, Access-Accept, len 115
00:05:47: %ISDN-6-DISCONNECT: Interface Serial0:22 disconnected from 4085274206, call
lasted 26 seconds
00:05:47: RADIUS: Initial Transmit ISDN 0:D:23 id 9 10.0.0.1:1824, Accounting-Request, len
775
00:05:47: RADIUS: Received from id 9 1.7.157.1:1824, Accounting-response, len 20

```

The following example shows **debug radius hex** output:

```
Router# debug radius hex
```

```

Radius protocol debugging is on
Radius packet hex dump debugging is on
Router#
17:26:52: RADIUS: ustruct sharecount=3
17:26:52: Radius: radius_port_info() success=0 radius_nas_port=1
17:26:52: RADIUS: Initial Transmit ISDN 0:D:23 id 10 10.0.0.1:1824, Accounting-Request,
len 361
17:26:52: Attribute 4 6 01081D03
17:26:52: Attribute 26 19 00000009020D4953444E20303A443A3233
17:26:52: Attribute 61 6 00000000
17:26:52: Attribute 1 12 34303835323734323036
17:26:52: Attribute 30 7 3532393831
17:26:52: Attribute 31 12 34303835323734323036
17:26:52: Attribute 40 6 00000001
17:26:52: Attribute 6 6 00000001

```

```

17:26:52:      Attribute 26 27 000000092115683332332D67772D69643D353330305F34332E
17:26:52:      Attribute 26 57
000000090133683332332D696E636F6D696E672D636F6E662D69643D3846334133313633204234393830303046
20302033424537314238
17:26:52:      Attribute 26 31
000000091A19683332332D63616C6C2D6F726967696E3D616E73776572
17:26:52:      Attribute 26 32
000000091B1A683332332D63616C6C2D747970653D54656C6570686F6E79
17:26:52:      Attribute 26 56
000000091932683332332D73657475702D74696D653D2A30393A32363A35322E3838302050535420536174204A
616E20312032303030
17:26:52:      Attribute 26 48
00000009182A683332332D636F6E662D69643D3846334133313633204234393830303046203020334245373142
38
17:26:52:      Attribute 44 10 3030303030303035
17:26:52:      Attribute 41 6 00000000
17:26:52: %ISDN-6-CONNECT: Interface Serial0:22 is now connected to 4085274206
17:26:52: RADIUS: Received from id 10 10.0.0.1:1824, Accounting-response, len 20
17:27:01: RADIUS: ustruct sharecount=3
17:27:01: Radius: radius_port_info() success=0 radius_nas_port=1
17:27:01: RADIUS: Initial Transmit ISDN 0:D:23 id 11 10.0.0.0:1823, Access-Request, len
173
17:27:01:      Attribute 4 6 01081D03
17:27:01:      Attribute 26 19 00000009020D4953444E20303A443A3233
17:27:01:      Attribute 61 6 00000000
17:27:01:      Attribute 1 8 313233343536
17:27:01:      Attribute 26 48
00000009182A683332332D636F6E662D69643D3846334133313633204234393830303046203020334245373142
38
17:27:01:      Attribute 31 12 34303835323734323036
17:27:01:      Attribute 2 18 C980D8D0E9A061B3D783C61AA6F27214
17:27:01:      Attribute 26 36
00000009011E683332332D6976722D6F75743D7472616E73616374696F6E49443A33
17:27:01: RADIUS: Received from id 11 1.7.157.1:1823, Access-Accept, len 115
17:27:01:      Attribute 6 6 00000001
17:27:01:      Attribute 26 29 000000096517683332332D6372656469742D616D6F756E743D3435
17:27:01:      Attribute 26 27 000000096615683332332D6372656469742D74696D653D3333
17:27:01:      Attribute 26 26 000000096714683332332D72657475726E2D636F64653D30
17:27:01:      Attribute 25 7 6C6F63616C
17:27:01: RADIUS: saved authorization data for user 61AA0698 at 6215087C
17:27:09: %ISDN-6-DISCONNECT: Interface Serial0:22 disconnected from 4085554206, call
lasted 17 seconds
17:27:09: RADIUS: ustruct sharecount=2
17:27:09: Radius: radius_port_info() success=0 radius_nas_port=1
17:27:09: RADIUS: Sent class "local" at 621508E8 from user 61AA0698
17:27:09: RADIUS: Initial Transmit ISDN 0:D:23 id 12 1.7.157.1:1824, Accounting-Request,
len 776
17:27:09:      Attribute 4 6 01081D03
17:27:09:      Attribute 26 19 00000009020D4953444E20303A443A3233
17:27:09:      Attribute 61 6 00000000
17:27:09:      Attribute 1 8 313233343536
17:27:09:      Attribute 30 7 3532393831
17:27:09:      Attribute 31 12 34303835323734323036
17:27:09:      Attribute 40 6 00000002
17:27:09:      Attribute 25 7 6C6F63616C
17:27:09:      Attribute 45 6 00000001
17:27:09:      Attribute 6 6 00000001
17:27:09:      Attribute 26 27 000000092115683332332D67772D69643D353330305F34332E
17:27:09:      Attribute 26 57
000000090133683332332D696E636F6D696E672D636F6E662D69643D3846334133313633204234393830303046
20302033424537314238
17:27:09:      Attribute 26 31
000000091A19683332332D63616C6C2D6F726967696E3D616E73776572

```

```

17:27:09:      Attribute 26 32
000000091B1A683332332D63616C6C2D747970653D54656C6570686F6E79
17:27:09:      Attribute 26 56
000000091932683332332D73657475702D74696D653D2A30393A32363A35322E3838302050535420536174204A
616E20312032303030
17:27:09:      Attribute 26 58
000000091C34683332332D636F6E6E6563742D74696D653D2A30393A32363A35322E3930372050535420536174
204A616E20312032303030
17:27:09:      Attribute 26 61
000000091D37683332332D646973636F6E6E6563742D74696D653D2A30393A32373A31302E3133372050535420
536174204A616E20312032303030
17:27:09:      Attribute 26 32
000000091E1A683332332D646973636F6E6E6563742D63617573653D3130
17:27:09:      Attribute 26 28 000000091F16683332332D766F6963652D7175616C6974793D30
17:27:09:      Attribute 26 48
00000009182A683332332D636F6E662D69643D3846334133313633204234393830303046203020334245373142
38
17:27:09:      Attribute 44 10 3030303030303035
17:27:09:      Attribute 42 6 00000000
17:27:09:      Attribute 43 6 00012CA0
17:27:09:      Attribute 47 6 00000000
17:27:09:      Attribute 48 6 000001E1
17:27:09:      Attribute 46 6 00000011
17:27:09:      Attribute 26 30 000000090118737562736372696265723D526567756C61724C696E65
17:27:09:      Attribute 26 35
00000009011D683332332D6976722D6F75743D5461726966663A556E6B6E6F776E
17:27:09:      Attribute 26 22 0000000901107072652D62797465732D696E3D30
17:27:09:      Attribute 26 23 0000000901117072652D62797465732D6F75743D30
17:27:09:      Attribute 26 21 00000009010F7072652D70616B732D696E3D30
17:27:09:      Attribute 26 22 0000000901107072652D70616B732D6F75743D30
17:27:09:      Attribute 26 22 0000000901106E61732D72782D73706565643D30
17:27:09:      Attribute 26 22 0000000901106E61732D74782D73706565643D30
17:27:09:      Attribute 41 6 00000000
17:27:09: RADIUS: Received from id 12 10.0.0.1:1824, Accounting-response, len 20

```

**Related Commands**

Command	Description
<b>debug aaa accounting</b>	Displays information on accountable events as they occur.
<b>debug aaa authentication</b>	Displays information on AAA/TACACS+ authentication.

# debug tacacs

To display information associated with TACACS, use the **debug tacacs** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

**debug tacacs**

**no debug tacacs**

## Syntax Description

This command has no arguments or keywords.

## Command Modes

Privileged EXEC

## Usage Guidelines

TACACS is a distributed security system that secures networks against unauthorized access. Cisco supports TACACS under the authentication, authorization, and accounting (AAA) security system.

Use the **debug aaa authentication** command to get a high-level view of login activity. When TACACS is used on the router, you can use the **debug tacacs** command for more detailed debugging information.

## Examples

The following is sample output from the **debug aaa authentication** command for a TACACS login attempt that was successful. The information indicates that TACACS+ is the authentication method used.

```
Router# debug aaa authentication
14:01:17: AAA/AUTHEN (567936829): Method=TACACS+
14:01:17: TAC+: send AUTHEN/CONT packet
14:01:17: TAC+ (567936829): received authen response status = PASS
14:01:17: AAA/AUTHEN (567936829): status = PASS
```

The following is sample output from the **debug tacacs** command for a TACACS login attempt that was successful, as indicated by the status PASS:

```
Router# debug tacacs
14:00:09: TAC+: Opening TCP/IP connection to 192.168.60.15 using source 10.116.0.79
14:00:09: TAC+: Sending TCP/IP packet number 383258052-1 to 192.168.60.15 (AUTHEN/START)
14:00:09: TAC+: Receiving TCP/IP packet number 383258052-2 from 192.168.60.15
14:00:09: TAC+ (383258052): received authen response status = GETUSER
14:00:10: TAC+: send AUTHEN/CONT packet
14:00:10: TAC+: Sending TCP/IP packet number 383258052-3 to 192.168.60.15 (AUTHEN/CONT)
14:00:10: TAC+: Receiving TCP/IP packet number 383258052-4 from 192.168.60.15
14:00:10: TAC+ (383258052): received authen response status = GETPASS
14:00:14: TAC+: send AUTHEN/CONT packet
14:00:14: TAC+: Sending TCP/IP packet number 383258052-5 to 192.168.60.15 (AUTHEN/CONT)
14:00:14: TAC+: Receiving TCP/IP packet number 383258052-6 from 192.168.60.15
14:00:14: TAC+ (383258052): received authen response status = PASS
14:00:14: TAC+: Closing TCP/IP connection to 192.168.60.15
```



The following is sample output from the **debug tacacs** command for a TACACS login attempt that was unsuccessful, as indicated by the status FAIL:

Router# **debug tacacs**

```
13:53:35: TAC+: Opening TCP/IP connection to 192.168.60.15 using source
192.48.0.79
13:53:35: TAC+: Sending TCP/IP packet number 416942312-1 to 192.168.60.15
(AUTHEN/START)
13:53:35: TAC+: Receiving TCP/IP packet number 416942312-2 from 192.168.60.15
13:53:35: TAC+ (416942312): received authen response status = GETUSER
13:53:37: TAC+: send AUTHEN/CONT packet
13:53:37: TAC+: Sending TCP/IP packet number 416942312-3 to 192.168.60.15
(AUTHEN/CONT)
13:53:37: TAC+: Receiving TCP/IP packet number 416942312-4 from 192.168.60.15
13:53:37: TAC+ (416942312): received authen response status = GETPASS
13:53:38: TAC+: send AUTHEN/CONT packet
13:53:38: TAC+: Sending TCP/IP packet number 416942312-5 to 192.168.60.15
(AUTHEN/CONT)
13:53:38: TAC+: Receiving TCP/IP packet number 416942312-6 from 192.168.60.15
13:53:38: TAC+ (416942312): received authen response status = FAIL
13:53:40: TAC+: Closing TCP/IP connection to 192.168.60.15
```

#### Related Commands

Command	Description
<b>debug aaa accounting</b>	Displays information on accountable events as they occur.
<b>debug aaa authentication</b>	Displays information on AAA/TACACS+ authentication.

# ip mobile home-agent

To enable and control home agent (HA) services, use the **ip mobile home-agent** command in global configuration mode. To disable these services, use the **no** form of this command.

**ip mobile home-agent** [**address** *ip-address*] [**broadcast**] [**care-of-access** *access-list*] [**lifetime** *seconds*] [**nat-detect**] [**replay** *seconds*] [**reverse-tunnel** {**off** | **private-address**}] [**roam-access** *access-list*] [**strip-realm**] [**suppress-unreachable**] [**local-timezone**] [**unknown-ha** [**accept** | **reply**] | **deny**]] [**send-mn-address**]

**no ip mobile home-agent** [**address** *ip-address*] [**broadcast**] [**care-of-access** *access-list*] [**lifetime** *seconds*] [**nat-detect**] [**replay** *seconds*] [**reverse-tunnel** {**off** | **private-address**}] [**roam-access** *access-list*] [**strip-realm**] [**suppress-unreachable**] [**local-timezone**] [**unknown-ha** [**accept** | **reply**] | **deny**]] [**send-mn-address**]

## Syntax Description

<b>address</b> <i>ip-address</i>	(Optional) Specifies the IP address of the HA.  <b>Note</b> This option is only applicable when HA redundancy is used for virtual networks.
<b>broadcast</b>	(Optional) Enables forwarding of broadcast datagrams to the mobile node (MN). By default, broadcasting is disabled.
<b>care-of-access</b> <i>access-list</i>	(Optional) Controls which care-of addresses (CoAs) in registration requests are permitted by the HA. By default, all CoAs are permitted. The <i>access-list</i> argument can be a string or number from 1 to 99.
<b>lifetime</b> <i>seconds</i>	(Optional) Specifies the global registration lifetime for an MN in seconds. Range is from 3 to 65535 (infinity). Default is 36000 (10 hours).  <b>Note</b> This configuration can be overridden by the individual MN configuration. Registrations requesting a lifetime greater than this value will still be accepted, but will use this lifetime value.
<b>nat-detect</b>	(Optional) Allows the HA to detect registration requests from a MN traversing a Network Address Translation (NAT)-enabled device and apply a tunnel to reach the MN. By default, NAT detection is disabled.
<b>replay</b> <i>seconds</i>	(Optional) Sets the replay protection time-stamp value in seconds. A registration received within the router clock time plus or minus 7 is valid.
<b>reverse-tunnel</b> { <b>off</b>   <b>private-address</b> }	(Optional) Enables support of reverse tunnel by the HA. By default, reverse tunnel support is enabled. The keywords are as follows: <ul style="list-style-type: none"> <li><b>off</b>—Disables support of reverse tunnel.</li> <li><b>private-address</b>—Reverse tunnel mandatory for private Mobile IP addresses.</li> </ul>
<b>roam-access</b> <i>access-list</i>	(Optional) Controls which MNs are permitted or denied to roam. By default, all specified MNs can roam.
<b>strip-realm</b>	(Optional) Strips the realm part of the Network access identifier (NAI) before authentication is performed. This option is useful if the majority of MNs have the identical realm, for example, in the case of enterprise networks.
<b>suppress-unreachable</b>	(Optional) Disables sending Internet Control Message Protocol (ICMP) unreachable messages to the source when an MN on the virtual network is not registered. By default, ICMP unreachable messages are sent.

<b>local-timezone</b>	(Optional) Uses the local time zone to generate identification fields.
<b>unknown-ha</b> [ <b>accept</b>   <b>reply</b> ]   <b>deny</b> ]	<p>Accepts or denies an unknown HA registration request. The keywords are as follows:</p> <ul style="list-style-type: none"> <li>• <b>accept</b>—(Optional) HA accepts the registration request with an HA address different from the IP destination of the registration request. The HA address set in the registration reply is that of the IP destination address.</li> <li>• <b>reply</b>—(Optional) HA uses the received HA address in reply.</li> <li>• <b>deny</b>—(Optional) HA denies the registration request with an HA address different from the IP destination of the registration request with error code Unknown HomeAgent. The HA address set in the reject registration reply is that of the IP destination address.</li> </ul>

**Note**

This command option can be used in a testing environment when the home agent is in private addressing space behind a NAT gateway.

<b>send-mn-address</b>	<p>Sends the home address as received in the registration request and in the access request messages for the HA Challenge Handshake Authentication Protocol (CHAP).</p> <p><b>Note</b> You must configure this keyword in the HA to send <b>radius-server vsa send authentication 3gpp2</b> attributes. This keyword is available only on PDSN platforms running specific PDSN code images.</p>
------------------------	---

**Defaults**

The command is disabled. Broadcasting is disabled. Reverse tunnel support is enabled. ICMP unreachable messages are sent. NAT detection is disabled.

**Command Modes**

Global configuration

**Command History**

Release	Modification
12.0(1)T	This command was introduced.
12.2(2)XC	The <b>strip-nai-realm</b> and <b>local-timezone</b> keywords were added.
12.2(13)T	The <b>nat-detect</b> keyword was added.
12.3(4)T	The <b>unknown-ha</b> , <b>accept</b> , <b>reply</b> , <b>deny</b> and <b>send-mn-address</b> keywords were added.

**Usage Guidelines**

This command enables and controls HA services on a router. Changes to service take effect immediately; however, broadcast and lifetime settings for previously registered MNs are unaffected. Tunnels are shared by MNs registered with the same endpoints, so the **reverse-tunnel-off** keyword also affects registered MNs.

The HA processes registration requests from the MN and sets up tunnels and routes to the CoA. Packets to the MN are forwarded to the visited network.

The HA will forward broadcast packets to MNs if the MNs are registered with the service. However, heavy broadcast traffic uses the CPU of the router.

The HA can control where the MNs roam by the **care-of-access** keyword, and which MN is allowed to roam by the **roam-access** keyword.

When a registration request comes in, the HA ignores requests when HA service is not enabled or the security association of the MN is not configured. The latter condition occurs because the security association must be available for the MH authentication extension in the reply. If a security association exists for the FA (IP source address or CoA in the request), the FA is authenticated, and then the MN is authenticated. The Identification field is verified to protect against replay attack. The HA checks the validity of the request (see [Table 4](#)) and sends a reply. (Reply codes are listed in [Table 5](#).) A security violation is logged when FA authentication, MH authentication, or identification verification fails. (The violation reasons are listed in [Table 6](#).)

After registration is accepted, the HA creates or updates the mobility binding of the MN, which contains the expiration timer. If no binding existed before this registration, a virtual tunnel is created, a host route to the MN via the care-of address is added to the routing table, and gratuitous ARPs are sent out. For deregistration, the host route is removed from the routing table, the virtual tunnel interface is removed (if no MNs are using it), and gratuitous ARP messages are sent out if the MN is back home. Mobility binding is removed (along with its associated host route and tunnel) when registration lifetime expires or deregistration is accepted.

By default, the HA uses the entire NAI string as the username for authentication (which may be with local security association or retrieved from the AAA server). The **strip-nai-realm** keyword instructs the HA to strip off the realm part of NAI (if it exists) before performing authentication. Basically, the MN is identified by only the user name part of the NAI. This option is useful if the majority of MNs belong to the same realm, for example, in the case of enterprise networks.

When the packet destined for the MN arrives on the HA, the HA encapsulates the packet and tunnels it to the care-of address. If the Don't Fragment (DF) bit is set in the packet via the **ip mobile tunnel path-mtu-discovery** global configuration command, the HA will copy the DF bit from the original packet to the new tunnel IP header. This allows the path MTU discovery to set the MTU of the tunnel. Subsequent packets greater than the MTU of the tunnel will be dropped and an ICMP datagram too big message will be sent to the source (correspondent node). If the HA loses the route to the tunnel endpoint, the host route to the MN will be removed from the routing table until the tunnel route is available. Packets destined for the MN without a host route will be sent out the interface (home network) or to the virtual network (see the description of the **suppress-unreachable** keyword). For subnet-directed broadcasts to the home link, the HA will send a copy to all MNs registered with the broadcast routing option.

Some companies block ICMP datagram too big messages. If the message does not reach the original correspondent node sending the packet, the correspondent node will simply resend the same size packet. To work around this problem, turn off Path MTU Discovery with the **no ip mobile tunnel path-mtu-discovery** command. The DF bit will not be copied from the original packet and the tunnel packet can be fragmented.

The **ip mobile home-agent nat-detect** option is supported for MNs using a collocated care-of address and registering through the FA. The MN will use the NAT inside address as the collocated care-of address used in its registration requests. If a MN is using a FA CoA address, the MN can be detected behind a NAT gateway.

The **ip mobile home-agent unknown-ha** option can be useful in a testing environment when the HA is using a private address behind a NAT gateway. A MN would need to access the HA through the NAT box while it is on a public network domain. However, NAT will translate the destination IP address of the

registration request to the private address of the HA. When the HA checks the HA field in the registration request, it does not match one of the interfaces. The packet can not be processed properly and the tunnels are not set up properly. The **ip mobile home-agent unknown-ha** command allows the HA to accept the unknown (translated) address and process the registration request.

The **send-mn-address** keyword is available only on PDSN platforms running specific PDSN code images; consult Feature Navigator for your Cisco IOS software release.

The MN requests services from the HA by setting bits in the registration request. [Table 4](#) shows the services the MN can request.

**Table 4 HA Registration Bitflags**

Bit Set	Definition
S	Accept with code 1 (no simultaneous binding).
B	Accept. Broadcast can be enabled or disabled.
D	Accept. Tunnel endpoint is a colocated care-of address.
M	Deny. Minimum IP encapsulation is not supported.
G	Accept. GRE encapsulation is supported.
V	Deny if this bit is set.
T	Accept if the <b>reverse-tunnel-off</b> parameter is not set.
reserved	Deny. Reserved bit must not be set.

[Table 5](#) lists the HA registration reply codes. The codes tell the MN whether the registration was accepted or denied. If registration is denied, the reply code gives the reason.

**Table 5 HA Registration Reply Codes**

Code	Reason
0	Accept.
1	Accept. No simultaneous bindings.
128	Reason unspecified.
129	Administratively prohibited.
130	Insufficient resource.
131	MN failed authentication.
132	FA failed authentication.
133	Registration identification mismatched (timestamp is off).
134	Poorly formed request.
136	Unknown HA address.
137	Reverse tunnel is unavailable.
138	Reverse tunnel is mandatory and T bit not set.
139	Unsupported encapsulation.
140	Unsupported vendor id or unable to interpret registration request extensions sent by the MN to the home agent.

**Table 5** HA Registration Reply Codes (continued)

Code	Reason
141	Unsupported vendor id or unable to interpret registration request extensions sent by the FA to the home agent.
142	Active home agent failed authentication.

Table 6 lists security violation codes.

**Table 6** Security Violation Codes

Code	Reason
1	No mobility security association.
2	Bad authenticator.
3	Bad identifier.
4	Bad SPI.
5	Missing security extension.
6	Other.
7	Stale request.

**Examples**

The following example enables broadcast routing and specifies a global registration lifetime of 7200 seconds (2 hours):

```
ip mobile home-agent broadcast lifetime 7200
```

**Related Commands**

Command	Description
<b>ip mobile tunnel</b>	Specifies the setting of tunnels created by Mobile IP.
<b>show ip mobile binding</b>	Displays the mobility binding table.
<b>show ip mobile globals</b>	Displays global information for mobile agents.

# ip mobile home-agent accounting

To enable home agent accounting services on the router, use the **ip mobile home-agent accounting** command in global configuration mode. To disable these services, use the **no** form of this command.

```
ip mobile home-agent accounting { default | list-name }
```

```
no ip mobile home-agent accounting { default | list-name }
```

Syntax Description	default	Uses the listed accounting methods that follow this argument as the default list of methods for accounting services.
	<i>list-name</i>	Character string used to name the list of at least one of the accounting methods.

**Defaults** The command is disabled.

**Command Modes** Global configuration

Command History	Release	Modification
	12.2(15)T	This command was introduced.

**Usage Guidelines** This command enables and controls home agent accounting services on the router. First, use the **aaa accounting** global configuration command to define the accounting method list. Next, apply the same accounting method list on the home agent using the **ip mobile home-agent accounting** global configuration command.

**Examples** The following example enables home agent accounting for the list named mobile-list:

```
ip mobile home-agent accounting mobile-list
```

Related Commands	Command	Description
	<b>aaa accounting</b>	Enables AAA accounting of requested services for billing or security purposes.

# ip mobile home-agent dynamic-address

To set the home agent address field in a Registration Response packet, use the **ip mobile home-agent dynamic-address** command in global configuration. To disable this functionality, or to reset the field use the **no** form of this command.

**ip mobile home-agent dynamic-address** *ip-address*

**no ip mobile home-agent dynamic-address** *ip-address*

## Syntax Description

ip-address	The IP address of the Home Agent.
------------	-----------------------------------

## Defaults

The Home Agent Address field will be set to the values specified by the *ip-address* argument.

## Command Modes

Global configuration

## Command History

Release	Modification
12.3(11)YF	This command was introduced.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.

## Examples

In the following example, the dynamic home-agent address is set to 10.1.1.1:

```
Router# ip mobile home-agent dynamic-address 10.1.1.1
```



# ip mobile home-agent redundancy

To configure the home agent for redundancy by using the Hot Standby Router Protocol (HSRP) group name, use the **ip mobile home-agent redundancy** command in global configuration mode. To remove the address, use the **no** form of this command.

**ip mobile home-agent redundancy** *hsrp-group-name* [[**virtual-network**] **address** *address*] [**mode active-standby**] [**swact-notification**]

**no ip mobile home-agent redundancy** *hsrp-group-name* [[**virtual-network**] **address** *address*] [**mode active-standby**] [**swact-notification**]

## Syntax Description

<i>hsrp-group-name</i>	Specifies the HSRP group name.
<b>virtual-network</b>	(Optional) Specifies that the HSRP group is used to support virtual networks.
<b>address</b> <i>address</i>	(Optional) Home agent address.
<b>mode active-standby</b>	(Optional) Allows the bindings to come up (with local pool addressing for virtual-networks) with the home agent IP address specified under the loopback interface.
<b>swact-notification</b>	(Optional) Notifies the RADIUS server of a home agent failover.

## Defaults

No global home agent addresses are specified.

## Command Modes

Global configuration

## Command History

Release	Modification
12.0(2)T	This command was introduced.
12.2(8)T	The command changed from <b>ip mobile home-agent standby</b> to <b>ip mobile home-agent redundancy</b> .
12.4(11)T	The <b>mode active-standby</b> and <b>swact-notification</b> keywords were added.

## Usage Guidelines

The **virtual-network** keyword specifies that the HSRP group supports virtual networks.



### Note

Redundant home agents must have identical Mobile IP configurations. You can use a standby group to provide HA redundancy for either physical or virtual networks, but not both at the same time.

When Mobile IP standby is configured, the home agent can request mobility bindings from the peer home agent. When Mobile IP standby is deconfigured, the home agent can remove mobility bindings. Operation of home agent redundancy on physical and virtual networks is described as follows:

- **Physical network**—Only the active home agent will receive registrations on a physical network. It updates the standby home agent. The standby home agent requests the mobility binding table from the active home agent. When Mobile IP standby is deconfigured, the standby home agent removes all bindings, but the active home agent keeps all bindings.
- **Virtual network**—Both active and standby home agents receive registrations if the loopback interface is used; each will update the peer after accepting a registration. Otherwise, the active home agent receives registrations. Both active and standby home agents request mobility binding tables from each other. When Mobile IP standby is deconfigured, the standby or active home agent removes all bindings.

**Note**

The **swact-notification** option notifies the RADIUS server of a home agent failover. This is achieved by including the `cisco-avpair radius` attribute “mobileip-rfswat=1” in RADIUS accounting records. This attribute is included only in the first accounting record of a binding generated after a failover, and if that binding was created before the failover.

**Examples**

The following example specifies an HSRP group named SanJoseHA:

```
ip mobile home-agent redundancy SanJoseHA
```

**Related Commands**

Command	Description
<code>show ip mobile globals</code>	Displays global information for mobile agents.

# ip mobile home-agent redundancy periodic-sync

To synchronize the byte and packet counters for each binding to the standby unit using an accounting update event, use the **ip mobile home-agent redundancy periodic-sync** command in global configuration mode. To disable this functionality, use the **no** form of this command.

```
ip mobile home-agent redundancy hsrp-group-name [[virtual-network] address address]
periodic-sync
```

```
no ip mobile home-agent redundancy hsrp-group-name [[virtual-network] address address]
periodic-sync
```

## Syntax Description

<b>hsrp-group-name</b>	Specifies the HSRP group name.
<b>virtual-network</b>	(Optional) Specifies that the HSRP group is used to support virtual networks.
<b>address</b> <i>address</i>	(Optional) Home agent address.

## Defaults

There are no default values for this command.

## Command Modes

Global configuration

## Command History

Release	Modification
12.3(14)YX	This command was introduced.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.

## Usage Guidelines

The byte and packet counters for each binding are synchronized to the standby unit using an accounting update event only if the byte counts have changed since the last synchronization.

## Examples

In the following example, the byte and packet counters for each binding will be periodically synchronized between the active and standby unit:

```
Router# ip mobile home-agent redundancy group1 periodic-sync
```

# ip mobile home-agent reject-static-addr

To configure the HA to reject Registration Requests from MNs under certain conditions, use the **ip mobile home-agent reject-static-addr** sub-command under the **ip mobile home-agent** global configuration command.

## **ip mobile home-agent reject-static-addr**

**Syntax Description** This command has not arguments or keywords

**Command Modes** Sub-command of the **ip mobile home-agent** global configuration command.

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.2(8)BY	This command was introduced.
	12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.

**Usage Guidelines** You must first configure the **ip mobile home-agent** command to use this sub-command.

If an MN that has a binding to the HA with a static address tries to register with the same static address again, then the HA rejects the second RRQ from the MN.

**Examples** The following example illustrates the **ip mobile home-agent reject-static-addr** command:

```
Router# ip mobile home-agent reject-static-addr
```

# ip mobile home-agent resync-sa

To configure the home agent to clear out the old cached security associations and requery the AAA server for a new security association when the mobile node fails authentication, use the **ip mobile home-agent resync-sa** command in global configuration mode. To disable this functionality, use the **no** form of this command.

**ip mobile home-agent resync-sa** *seconds*

**no ip mobile home-agent resync-sa** *seconds*

<b>Syntax Description</b>	<i>seconds</i>	Specifies the time in which the home agent will wait to initiate a resynchronization.
---------------------------	----------------	---

<b>Defaults</b>	This command is off by default. The normal behavior of the home agent is to never requery the AAA server for a new security association.
-----------------	--

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.2	This command was introduced.

<b>Usage Guidelines</b>	You must enable security association caching for the <b>ip mobile home-agent resync-sa</b> command to work. Use the <b>ip mobile host aaa load-sa</b> global configuration command to enable caching of security associations retrieved from a AAA server.
-------------------------	--

When a security association is downloaded for a mobile node from a AAA server, the security association is time stamped. If the mobile node fails reregistration and the time interval since the security association was cached is greater than *sec* seconds, the home agent will clear out the old security association and requery the AAA server. If the time period is less than the *sec* value, the home agent will not requery the AAA server for the security association of the mobile node.

The *sec* value represents the number of seconds the home agent will consider the downloaded security association synchronized with the AAA server. After that time period, it is considered old and can be replaced by a new security association from the AAA server.

This time-based resynchronization process helps prevent denial-of-service attacks on the AAA server and provides a way to synchronize the home agent's cached security association entry when a change to the security association for the mobile node is made at the AAA server and on the mobile node. By using this process, once the mobile node fails reregistration with the old cached security association, the home agent will clear the cache for that mobile node, and resynchronize with the AAA server.

---

**Examples**

In the following example, if a registration fails authentication, the home agent retrieves a new security association from the AAA server if the existing security association was downloaded more than 10 seconds ago:

```
ip mobile home-agent resync-sa 10
```

---

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>ip mobile host</b>	Configures the mobile node or mobile host group.

---

# ip mobile home-agent revocation

To enable support for MIPv4 registration revocation on the home agent, use the **ip mobile home-agent revocation** command in global configuration mode. To disable support for registration revocation, use the **no** form of the command.

**ip mobile home-agent revocation** [*timeout seconds*] [*retransmit retries*] [*timestamp msec*]

**no ip mobile home-agent revocation** [*timeout seconds*] [*retransmit retries*] [*timestamp msec*]

## Syntax Description

<i>timeout seconds</i>	(Optional) Configures the time interval (in seconds) between retransmission of MIPv4 registration revocation message. The <b>no</b> version restores the time interval between retransmission of MIPv4 registration revocation Message to the default value. The default is 3 seconds. The range is from 1 to 100 seconds
<i>retransmit retries</i>	(Optional) Configures the number of times MIPv4 registration revocation messages are retransmitted. The <b>no</b> version of this command restores the retransmit number to the default value. The default is 3 retransmissions. The range is from 1 to 100 retransmissions.
<i>timestamp msec</i>	(Optional) Configures the units in which the timestamp value in the revocation support extension and revocation message should be encoded. By default the timestamp value will be sent as seconds. If the <b>msec</b> option is specified, the values will be encoded in milliseconds.

## Command Default

The home agent does not support registration revocation.

## Command Modes

Global configuration

## Command History

Release	Modification
12.3(7)XJ	This command was introduced.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.

## Examples

In the following example, the MIPv4 registration message will be retransmitted a maximum of 5 times with a time interval of 4 seconds in between retransmissions:

```
Router(config)#ip mobile home-agent revocation timeout 4 retransmit 5
```

# ip mobile home-agent template tunnel

To configure a home agent to use the template tunnel, use the **ip mobile home-agent template tunnel** command in global configuration. To disable the use of the template tunnel, use the **no** form of the command.

**ip mobile home-agent template tunnel** *interface-id* **address** *ha-address*

**no ip mobile home-agent template tunnel** *interface-id* **address** *ha-address*

## Syntax Description

<b>interface-id</b>	Specifies the template tunnel interface ID from which to apply ACLs.
<b>address</b>	Specifies the home agent address. ACLs will be applied to tunnels with
<b>ha-address</b>	<i>ha-address</i> as the local end point.

## Command Default

The home agent does not use a template tunnel.

## Command Modes

Global configuration

## Command History

Release	Modification
12.3(8)XJW	This command was introduced.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.

## Examples

In the following example, the home agent is configured to use the template tunnel:

```
Router(config)# interface tunnel 10
!
Router(config)# ip mobile home-agent template tunnel 10 address 10.0.0.1
```