CHAPTER 23

# quit through rule Commands

# quit

To exit the current configuration mode, or to log out from privileged or user EXEC modes, use the **quit** comman in user EXEC mode.

**quit**

## Syntax Description

This command has no arguments or keywords.

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| User EXEC | • | • | • | • | • |

## Command History

| Release | Modification |
|---|---|
| 1.1(1) | This command was introduced. |

## Usage Guidelines

You can also use the key sequence **Ctrl Z** to exit global configuration (and higher) modes. This key sequence does not work with privileged or user EXEC modes.

When you enter the **quit** command in privileged or user EXEC modes, you log out from the FWSM. Use the **disable** command to return to user EXEC mode from privileged EXEC mode.

## Examples

The following example shows how to use the **quit** command to exit global configuration mode, and then logout from the session:

```
hostname(config)# quit
hostname# quit

Logoff
```

The following example shows how to use the **quit** command to exit global configuration mode, and then use the **disable** command to exit privileged EXEC mode:

```
hostname(config)# quit
hostname# disable
hostname>
```

## Related Commands

| Command | Description |
|---|---|
| **exit** | Exits a configuration mode or logs out from privileged or user EXEC modes. |

# radius-common-pw

To specify a common password to be used for all users whose VPN access is authorized by a RADIUS authorization server, use the **radius-common-pw** command in aaa-server host configuration mode. To remove this specification, use the **no** form of this command:

**radius-common-pw** *password*

**no radius-common-pw**

## Syntax Description

|  |  |
|---|---|
| *password* | A case-sensitive, alphanumeric keyword of up to 127 characters to be used as a common password for all authorization transactions with the RADIUS server specified with the **aaa-server host** command. |

## Defaults

No default behaviors or values.

## Command Modes

The following table shows the modes in which you can enter the command:

|  | Firewall Mode |  | Security Context |  |  |
|---|---|---|---|---|---|
|  |  |  |  | Multiple |  |
| Command Mode | Routed | Transparent | Single | Context | System |
| Aaa-server host configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Usage Guidelines

This command is valid only for RADIUS authorization servers.

The RADIUS authorization server requires a password and username for each connecting user. The FWSM provides the username automatically. You enter the password here. The RADIUS server administrator must configure the RADIUS server to associate this password with each user authorizing to the server via this FWSM. Be sure to provide this information to your RADIUS server administrator.

If you do not specify a common user password, each user password is the username of the user. For example, the default RADIUS authorization for a user with the username “user1” is “user1”. If you are using usernames for the common user passwords, as a security precaution do not use this RADIUS server for authorization anywhere else on your network.

**Note** The password field is required by the RADIUS protocol and the RADIUS server requires it; however, users do not need to know it.

## Examples

The following example configures a RADIUS AAA server group named “svrgrp1” on host “209.165.200.225”, sets the timeout interval to 9 seconds, sets the retry interval to 7 seconds, and configures the RADIUS commnon password as “allauthpw”.

```
hostname(config)# aaa-server svrgrp1 protocol radius
hostname(config-aaa-server-group)# aaa-server svrgrp1 host 209.165.200.225
hostname(config-aaa-server-host)# timeout 9
hostname(config-aaa-server-host)# retry 7
hostname(config-aaa-server-host)# radius-common-pw allauthpw
```

## Related Commands

| Command | Description |
|---|---|
| **aaa-server host** | Enter aaa server host configuration mode so that you can configure AAA server parameters that are host-specific. |
| **clear configure aaa-server** | Remove all AAA command statements from the configuration. |
| **show running-config aaa-server** | Displays AAA server statistics for all AAA servers, for a particular server group, for a particular server within a particular group, or for a particular protocol. |

# radius-with-expiry

To have the FWSM use MS-CHAPv2 to negotiate a password update with the user during authentication, use the **radius-with-expiry** command in tunnel-group ipsec-attributes configuration mode. The FWSM ignores this command if RADIUS authentication has not been configured. To return to the default value, use the **no** form of this command.

**radius-with-expiry**

**no radius-with-expiry**

## Syntax Description

This command has no arguments or keywords.

## Defaults

The default setting for this command is disabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Tunnel-group ipsec-attributes configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Usage Guidelines

You can apply this attribute to IPSec remote-access tunnel-group type only.

## Examples

The following example entered in config-ipsec configuration mode, configures Radius with Expiry for the remote-access tunnel group named remotegrp:

```
hostname(config)# tunnel-group remotegrp type ipsec_ra
hostname(config)# tunnel-group remotegrp ipsec-attributes
hostname(config-ipsec)# radius-with-expiry
hostname(config-ipsec)#
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure tunnel-group** | Clears all configured tunnel groups. |
| **show running-config tunnel-group** | Shows the indicated certificate map entry. |
| **tunnel-group-map default-group** | Associates the certificate map entries created using the **crypto ca certificate map** command with tunnel groups. |

# reactivation-mode

To specify the method (reactivation policy) by which failed servers in a group are reactivated, use the **reactivation-mode** command in aaa-server group configuration mode. To remove this specification, use the **no** form of this command.

**reactivation-mode depletion** [**deadtime** *minutes*]

**reactivation-mode timed**

**no reactivation-mode**

## Syntax Description

| | |
|---|---|
| **deadtime** *minutes* | (Optional) Specifies the amount of time that elapses between the disabling of the last server in the group and the subsequent reenabling of all servers. |
| **depletion** | Reactivates failed servers only after all of the servers in the group are inactive. |
| **timed** | Reactivates failed servers after 30 seconds of down time. |

## Defaults

The default reactivation mode is depletion, and the default deadtime value is 10. The supported range of values for deadtime is 0-1440 minutes.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Aaa-server group configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Usage Guidelines

Each server group has an attribute that specifies the reactivation policy for its servers.

In **depletion** mode, when a server is deactivated, it remains inactive until all other servers in the group are inactive. When and if this occurs, all servers in the group are reactivated. This approach minimizes the occurrence of connection delays due to failed servers. When **depletion** mode is in use, you can also specify the **deadtime** parameter. The **deadtime** parameter specifies the amount of time (in minutes) that will elapse between the disabling of the last server in the group and the subsequent re-enabling of all servers. This parameter is meaningful only when the server group is being used in conjunction with the local fallback feature.

In **timed** mode, failed servers are reactivated after 30 seconds of down time. This is useful when customers use the first server in a server list as the primary server and prefer that it is online whenever possible. This policy breaks down in the case of UDP servers. Because UDP is a connectionless protocol, the FWSM cannot determine if the server is present; therefore, UDP servers are put back on line blindly. This could lead to slowed connection times or connection failures if a server list contains multiple servers that are not reachable.

Accounting server groups that have simultaneous accounting enabled are forced to use the **timed** mode. This implies that all servers in a given list are equivalent.

## Examples

The following example configures a TACACS+ AAA server named "svrgrp1" to use the depletion reactivation mode, with a deadtime of 15 minutes:

```
hostname(config)# aaa-server svrgrp1 protocol tacacs+
hostname(config-aaa-sersver-group)# reactivation-mode depletion deadtime 15
```

The following example configures a TACACS+ AAA server named "svrgrp1" to use timed reactivation mode:

```
hostname(config)# aaa-server svrgrp2 protocol tacacs+
hostname(config-aaa-server)# reactivation-mode timed
```

## Related Commands

| | |
|---|---|
| **accounting-mode** | Indicates whether accounting messages are sent to a single server (single mode) or sent to all servers in the group (simultaneous mode). |
| **aaa-server protocol** | Enters aaa server group configuration mode so that you can configure AAA server parameters that are group-specific and common to all hosts in the group. |
| **max-failed-attempts** | Specifies the number of failures that will be tolerated for any given server in the server group before that server is deactivated. |
| **clear configure aaa-server** | Removes all AAA server configuration. |
| **show running-config aaa-server** | Displays AAA server statistics for all AAA servers, for a particular server group, for a particular server within a particular group, or for a particular protocol. |

# redistribute (router eigrp)

To redistribute routes from one routing domain into another routing domain, use the **redistribute** command in router configuration mode. To remove the redistribution, use the **no** form of this command.

**redistribute** {{**ospf** *pid* **nssa-external** [**1** | **2**]}} | **static** | **connected**} [**metric** *metric_value*] [**route-map** *map_name*]

**no redistribute** {{**ospf** *pid* **nssa-external** [**1** | **2**]}} | **static** | **connected**} [**metric** *metric_value*] [**route-map** *map_name*]

## Syntax Description

| | |
|---|---|
| **connected** | Specifies redistributing a network connected to an interface into an EIGRP routing process. |
| metric *metric_value* | (Optional) Specifies the OSPF default metric value from 0 to 16777214. |
| **nssa-external** *type* | Specifies the OSPF metric type for routes that are external to a not-so-stubby area (NSSA); valid values are **1** or **2**. |
| ospf *pid* | Used to redistribute an OSPF routing process into the current EIGRP routing process. The *pid* specifies the internally used identification parameter for an OSPF routing process; valid values are from 1 to 65535. |
| route-map *map_name* | (Optional) Name of the route map to apply. |
| **static** | Used to redistribute a static route into an EIGRP process. |

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Router configuration | • | — | • | — | — |

## Command History

| Release | Modification |
|---|---|
| 4.0(1) | This command was introduced. |

## Examples

This example shows how to redistribute static routes into the current EIGRP process:

```
hostname(config-router)# redistribute static
```

## Related Commands

| Command | Description |
|---|---|
| **router eigrp** | Enters router configuration mode. |
| **show running-config router** | Displays the commands in the global router configuration. |

# redistribute (router ospf)

To redistribute routes from one routing domain into another routing domain, use the **redistribute** command in router configuration mode. To remove the redistribution, use the **no** form of this command.

**redistribute** {{**eigrp** *pid* [**match** {**internal** | **external** [**1** | **2**] | **nssa-external** [**1** | **2**]}]} | **static** | **ospf** *pid* **connected**} [**metric** *metric_value*] [**metric-type** *metric_type*] [**route-map** *map_name*] [**tag** *tag_value*] [**subnets**]

**no redistribute** {{**eigrp** *pid* [**match** {**internal** | **external** [**1** | **2**] | **nssa-external** [**1** | **2**]}]} | **static** | **ospf** *pid* **connected**} [**metric** *metric_value*] [**metric-type** *metric_type*] [**route-map** *map_name*] [**tag** *tag_value*] [**subnets**]

## Syntax Description

| | |
|---|---|
| **connected** | Specifies redistributing a network connected to an interface into an OSPF routing process. |
| eigrp *pid* | Used to redistribute an EIGRP routing process into the current OSPF routing process. The *pid* specifies the internally used identification parameter for an EIGRP routing process; valid values are from 1 to 65535. |
| **external** *type* | Specifies the metric routes that are external to a specified autonomous system; valid values are **1** or **2**. |
| **internal** *type* | Specifies metric routes that are internal to a specified autonomous system. |
| match | (Optional) Specifies the conditions for redistributing routes from one routing protocol into another. |
| metric-type *metric_type* | (Optional) The external link type associated with the default route advertised into the OSPF routing domain. It can be either of the following two values: 1 (Type 1 external route) or 2 (Type 2 external route). |
| metric *metric_value* | (Optional) Specifies the EIGRP default metric value from 0 to 16777214. |
| **nssa-external** *type* | Specifies the EIGRP metric type for routes that are external to a not-so-stubby area (NSSA); valid values are **1** or **2**. |
| ospf *pid* | Used to redistribute an OSPF routing process into the current OSPF routing process. The *pid* specifies the internally used identification parameter for an EIGRP routing process; valid values are from 1 to 65535. |
| route-map *map_name* | (Optional) Name of the route map to apply. |
| **static** | Used to redistribute a static route into an OSPF process. |
| subnets | (Optional) For redistributing routes into EIGRP, scopes the redistribution for the specified protocol. If not used, only classful routes are redistributed. |
| tag *tag_value* | (Optional) A 32-bit decimal value attached to each external route. This value is not used by OSPF itself. It may be used to communicate information between ASBRs. If none is specified, then the remote autonomous system number is used for routes from BGP and EGP; for other protocols, zero (0) is used. Valid values range from 0 to 4294967295. |

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Router configuration | • | — | • | — | — |

## Command History

| Release | Modification |
|---|---|
| 4.0(1) | This command was introduced. |

## Examples

This example shows how to redistribute static routes into the current OSPF process:

```
hostname(config-router)# redistribute static
```

## Related Commands

| Command | Description |
|---|---|
| **router ospf** | Enters router configuration mode. This keyword is only available under router-ospf configuration mode. |
| **show running-config router** | Displays the commands in the global router configuration. |

# redistribute (route-inject)

To configure the type of routes or NAT pools to inject to the MSFC routing tables. use the **redistribute** command in route-inject configuration mode. To delete the configuration, use the **no** form of this command.

**redistribute {static | connected | nat} [route-map** *map-name* **| access-list** *acl-id* **| global-pool** *pool-id*] **interface** *interface-name*

**no redistribute**

## Syntax Description

| | |
|---|---|
| **access-list** | (Optional) Specifies which routes to inject based on matching an access list. |
| *acl-id* | The unique identifier of the access list to match when you inject the connected and static routes, or NAT pools. |
| **connected** | Injects the connected routes into the MSFC routing table. |
| **global-pool** | (Optional) Specifies the global-pool to inject. |
| **interface** | Specifies the interface on which the routes must be injected. |
| *interface-name* | Specifies the name of the interface on which the routes must be injected. |
| *map-name* | Specifies the name of the route-map to match for the injection. |
| **nat** | Injects a NAT pool into the MSFC routing table. |
| *pool-id* | The unique identifier of the global-pool. |
| **route-map** | (Optional) Specifies which routes to inject based on matching a route-map. |
| **static** | Injects the static routes into the MSFC routing table. |

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Route-inject configuration | • | — | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 4.0(1) | This command was introduced. |

## Usage Guidelines

The **redistribute** command allows you to configure the type of routes or NAT pools to inject to the MSFC routing tables.

FWSM injects the IP of the FWSM interface as the next-hop IP address for specific destination addresses to the connected and static routes and NAT pools configured on FWSM into the routing table of the local switch.

For example, if you wanted to configure a NAT pool on FWSM, the MFSC and other external routers do not know that those NAT pool addresses are on FWSM unless the user configures the static routes on MSFC to point to the FWSM interface. But by utilizing RHI, you can inject the NAT pool addresses to point to the FWSM interface so the MSFC can automatically forward that traffic to the FWSM.

Because FWSM only supports OSPF or other dynamic routing protocols in single routed-mode, RHI can be used in multi-mode to inject routes (connected/static) to the MSFC, which can then redistribute these routes through OSPF or other dynamic routing protocols. This allows FWSM to redistribute FWSM routes through OSPF or other dynamic routing protocols even when running multi-mode, by utilizing the MSFC's routing protocols and RHI.

**Note** The connected and static routes and NAT pools can be selectively injected by configuring a redistribute policy using a standard access list, route-map or global pool ID (only for NAT).

RHI is supported in both single and multi-mode, but not Transparent mode. Additionally, RHI is supported with HA (Active/Standby and Active/Active).

## Examples

### Configuring RHI for NAT with Standard ACL

In this example, only a perfect match will be injected. The **acl1**, 23.10.143.20/30 is injected with nexthop of 20.22.211.21 (Active IP of “outside”) on vlan 20 (vlan of “outside”).

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 20.22.211.21 255.255.255.0 standby 20.22.211.22
hostname(config-if)# exit
hostname(config)# access-list acl1 standard permit 23.10.143.20 255.255.255.252
hostname(config)# global (outside) 10 23.10.143.20-23.10.143.23 netmask 255.255.255.0
hostname(config)# global (outside) 10 23.10.143.40-23.10.143.45 netmask 255.255.255.0
hostname(config)# route-inject
hostname(config-route-inject)# redistribute nat access-list acl1 interface outside
```

### Configuring RHI for NAT with Global Pool ID

In this example, 23.11.111.1-23.11.111.7 and 23.11.111.10-23.11.111.20 injected with nexthop 20.11.111.11 on vlan 20. Be sure that the global interface and pool ID match the **redistribute** command.

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 20.11.111.11 255.255.255.0 standby 20.11.111.21
hostname(config-if)# exit
hostname(config)# global (dmz) 10 22.11.111.1-22.11.111.10 netmask 255.255.255.0
hostname(config)# global (outside) 10 23.11.111.1-23.11.111.7 netmask 255.255.255.0
hostname(config)# global (outside) 10 23.11.111.10-23.11.111.20 netmask 255.255.255.0
hostname(config)# global (outside) 20 23.11.111.30-23.11.111.40 netmask 255.255.255.0
hostname(config)# route-inject
hostname(config-route-inject)# redistribute nat global-pool 10 interface outside
```

### Configuring RHI for Static Route using route-map

In this example, 23.11.111.0/24 and 25.11.111.0/24 will be injected with nexthop of 20.11.111.11 on vlan 20. The **route-map** command can be used to match destination IP, nexthop IP, metric, or interface

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 20.11.111.11 255.255.255.0 standby 20.11.111.12
hostname(config-if)# exit
hostname(config)# access-list acl1 standard permit 23.11.111.0 255.255.255.0
hostname(config)# access-list acl2 standard permit 25.11.111.0 255.255.255.0
hostname(config)# route-map map1 permit 10
hostname(config-route-map)# match ip address acl1 acl2
hostname(config-route-map)# exit
hostname(config)# route outside 23.11.111.0 255.255.255.0 23.11.111.9
hostname(config)# route outside 24.11.111.0 255.255.255.0 24.11.111.9
hostname(config)# route outside 25.11.111.0 255.255.255.0 25.11.111.9
hostname(config)# route-inject
hostname(config-route-inject)# redistribute static route-map map1 interface outside
```

**Note** Route maps can only be used in single routed mode.

## Related Commands

| Command | Description |
|---|---|
| **clear configure route-inject** | Removes the conditions for the route injection. |
| **route-inject** | Allows you to inject the connected and static routes and NAT pools configured on the FWSM into the MSFC routing table. |
| **show route-inject** | Displays the routes and NAT pools that have been injected. |
| **show running-config route-inject** | Displays the route-injection running configuration. |

# regex

To create a regular expression to match text, use the **regex** command in global configuration mode. To delete a regular expression, use the **no** form of this command.

**regex** *name regular_expression*

**no regex** *name* [*regular_expression*]

## Syntax Description

| | |
|---|---|
| *name* | Specifies the regular expression name, up to 40 characters in length. |
| *regular_expression* | Specifies the regular expression up to 100 characters in length. See "Usage Guidelines" for a list of metacharacters you can use in the regular expression. |

## Defaults

No default behaviors or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Global configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 4.0(1) | This command was introduced. |

## Usage Guidelines

The **regex** command can be used for various features that require text matching. For example, you can configure special actions for application inspection using Modular Policy Framework using an *inspection policy map* (see the **policy map type inspect** command). In the inspection policy map, you can identify the traffic you want to act upon by creating an inspection class map containing one or more **match** commands or you can use **match** commands directly in the inspection policy map. Some **match** commands let you identify text in a packet using a regular expression; for example, you can match URL strings inside HTTP packets. You can group regular expressions in a regular expression class map (see the **class-map type regex** command).

A regular expression matches text strings either literally as an exact string, or by using *metacharacters* so you can match multiple variants of a text string. You can use a regular expression to match the content of certain application traffic; for example, you can match body text inside an HTTP packet.

**Note** As an optimization, the FWSM searches on the deobfuscated URL. Deobfuscation compresses multiple forward slashes (/) into a single slash. For strings that commonly use double slashes, like "http://", be sure to search for "http:/" instead.

Table 23-1 lists the metacharacters that have special meanings.

*Table 23-1 regex Metacharacters*

| Character | Description | Notes |
|---|---|---|
| **.** | Dot | Matches any single character. For example, **d.g** matches dog, dag, dtg, and any word that contains those characters, such as doggonnit. |
| **(***exp***)** | Subexpression | A subexpression segregates characters from surrounding characters, so that you can use other metacharacters on the subexpression. For example, **d(o\|a)g** matches dog and dag, but **do\|ag** matches do and ag. A subexpression can also be used with repeat quantifiers to differentiate the characters meant for repetition. For example, **ab(xy){3}z** matches abxyxyxyz. |
| **\|** | Alternation | Matches either expression it separates. For example, **dog\|cat** matches dog or cat. |
| **?** | Question mark | A quantifier that indicates that there are 0 or 1 of the previous expression. For example, **lo?se** matches lse or lose.<br>**Note** You must enter **Ctrl+V** and then the question mark or else the help function is invoked. |
| ***** | Asterisk | A quantifier that indicates that there are 0, 1 or any number of the previous expression. For example, **lo*se** matches lse, lose, loose, and so on. |
| **+** | Plus | A quantifier that indicates that there is at least 1 of the previous expression. For example, **lo+se** matches lose and loose, but not lse. |
| **{***x***}** or **{***x***,}** | Minimum repeat quantifier | Repeat at least *x* times. For example, **ab(xy){2,}z** matches abxyxyz, abxyxyxyz, and so on. |
| **[***abc***]** | Character class | Matches any character in the brackets. For example, **[abc]** matches a, b, or c. |
| **[^***abc***]** | Negated character class | Matches a single character that is not contained within the brackets. For example, **[^abc]** matches any character other than a, b, or c. **[^A-Z]** matches any single character that is not an uppercase letter. |
| **[***a***-***c***]** | Character range class | Matches any character in the range. **[a-z]** matches any lowercase letter. You can mix characters and ranges: **[abcq-z]** matches a, b, c, q, r, s, t, u, v, w, x, y, z, and so does **[a-cq-z]**.<br>The dash (-) character is literal only if it is the last or the first character within the brackets: **[abc-]** or **[-abc]**. |
| **""** | Quotation marks | Preserves trailing or leading spaces in the string. For example, **" test"** preserves the leading space when it looks for a match. |
| **^** | Caret | Specifies the beginning of a line. |

*Table 23-1 regex Metacharacters (continued)*

| Character | Description | Notes |
|---|---|---|
| \ | Escape character | When used with a metacharacter, matches a literal character. For example, **\[** matches the left square bracket. |
| *char* | Character | When character is not a metacharacter, matches the literal character. |
| **\r** | Carriage return | Matches a carriage return 0x0d. |
| **\n** | Newline | Matches a new line 0x0a. |
| **\t** | Tab | Matches a tab 0x09. |
| **\f** | Formfeed | Matches a form feed 0x0c. |
| **\x***NN* | Escaped hexadecimal number | Matches an ASCII character using hexadecimal (exactly two digits). |
| \*NNN* | Escaped octal number | Matches an ASCII character as octal (exactly three digits). For example, the character 040 represents a space. |

To test a regular expression to make sure it matches what you think it will match, enter the **test regex** command.

The regular expression performance impact is determined by two main factors:

- The length of text that needs to be searched for a regular expression match.

  The regular expression engine has only a small impact to the FWSM performance when the search length is small.

- The number of regular expression chained tables that need to be searched for a regular expression match.

**Note** The maximum number of regular expressions per context is 2048.

The **debug menu regex 40 10** command can be used to display how many chained tables there are in each regex database.

## Examples

The following example creates two regular expressions for use in an inspection policy map:

```
hostname(config)# regex url_example example\.com
hostname(config)# regex url_example2 example2\.com
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect** | Creates ain inspection class map to match traffic specific to an application. |
| **policy-map** | Creates a policy map by associating the traffic class with one or more actions. |
| **policy-map type inspect** | Defines special actions for application inspection. |

| Command | Description |
|---|---|
| **class-map type regex** | Creates a regular expression class map. |
| **test regex** | Tests a regular expression. |

# reload

To reboot and reload the configuration, use the **reload** command in privileged EXEC mode.

**reload** [**at** *hh***:***mm* [*month day* | *day month*]] [**cancel**] [**in** [*hh***:**]*mm*] [**max-hold-time** [*hh***:**]*mm*] [**noconfirm**] [**quick**] [**reason** *text*] [**save-config**]

## Syntax Description

| | |
|---|---|
| **at** *hh***:***mm* | (Optional) Schedules a reload of the software to take place at the specified time (using a 24-hour clock). If you do not specify the month and day, the reload occurs at the specified time on the current day (if the specified time is later than the current time), or on the next day (if the specified time is earlier than the current time). Specifying 00:00 schedules the reload for midnight. The reload must take place within 24 hours. |
| **cancel** | (Optional) Cancels a scheduled reload. |
| *day* | (Optional) Number of the day in the range from 1 to 31. |
| **in** [*hh***:**]*mm*] | (Optional) Schedules a reload of the software to take effect in the specified minutes or hours and minutes. The reload must occur within 24 hours. |
| **max-hold-time** [*hh***:**]*mm* | (Optional) Specifies the maximum hold time the FWSM waits to notify other subsystems before a shutdown or reboot. After this time elapses, a quick (forced) shutdown/reboot occurs. |
| *month* | (Optional) Specifies the name of the month. Enter enough characters to create a unique string for the name of the month. For example, "Ju" is not unique because it could represent June or July, but "Jul" is unique because no other month beginning with those exact three letters. |
| **noconfirm** | (Optional) Permits the FWSM to reload without user confirmation. |
| **quick** | (Optional) Forces a quick reload, without notifying or properly shutting down all the subsystems. |
| **reason** *text* | (Optional) Specifies the reason for the reload, 1 to 255 characters. The reason text is sent to all open IPSec VPN client, terminal, console, telnet, SSH, and ASDM connections/sessions.<br>**Note** Some applications, like isakmp, require additional configuration to send the reason text to IPSec VPN Clients. Refer to the appropriate section in the software configuration documentation for more information. |
| **save-config** | (Optional) Saves the running configuration to memory before shutting down. If you do not enter the **save-config** keyword, any configuration changes that have not been saved will be lost after the reload. |

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Privileged EXEC | • | • | • | — | • |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was modified to add the following new arguments and keywords: *day*, *hh*, *mm*, *month*, **quick**, **save-config**, and *text*. |

## Usage Guidelines

The  command lets you reboot the FWSM and reload the configuration from Flash.

By default, the **reload** command is interactive. The FWSM first checks whether the configuration has been modified but not saved. If so, the FWSM prompts you to save the configuration. In multiple context mode, the FWSM prompts for each context with an unsaved configuration. If you specify the **save-config** parameter, the configuration is saved without prompting you. The FWSM then prompts you to confirm that you really want to reload the system. Only a response of **y** or pressing the **Enter** key causes a reload. Upon confirmation, the FWSM starts or schedules the reload process, depending upon whether you have specified a delay parameter (**in** or **at**).

By default, the reload process operates in "graceful" (also known as "nice") mode. All registered subsystems are notified when a reboot is about to occur, allowing these subsystems to shut down properly before the reboot. To avoid waiting until for such a shutdown to occur, specify the **max-hold-time** parameter to specify a maximum time to wait. Alternatively, you can use the **quick** parameter to force the reload process to begin abruptly, without notifying the affected subsystems or waiting for a graceful shutdown.

You can force the **reload** command to operate noninteractively by specifying the **noconfirm** parameter. In this case, the FWSM does not check for an unsaved configuration unless you have specified the **save-config** parameter. The FWSM does not prompt the user for confirmation before rebooting the system. It starts or schedules the reload process immediately, unless you have specified a delay parameter, although you can specify the **max-hold-time** or **quick** parameters to control the behavior or the reload process.

Use **reload cancel** to cancel a scheduled reload. You cannot cancel a reload that is already in progress.

**Note** Configuration changes that are not written to the Flash partition are lost after a reload. Before rebooting, enter the **write memory** command to store the current configuration in the Flash partition.

## Examples

This example shows how to reboot and reload the configuration:

```
hostname# reload
Proceed with ?  [confirm] y

Rebooting...

XXX Bios VX.X
...
```

## Related Commands

| Command | Description |
|---|---|
| **show reload** | Displays the reload status of the FWSM. |

# remote-access threshold session-threshold-exceeded

To set threshold values, use the **remote-access threshold session-threshold-exceeded** command in global configuration mode. To remove threshold values, use the **no** form of this command. This command specifies the number of remote access sessions that need to be active for the FWSM to send traps.

**remote-access threshold session-threshold-exceeded** {*threshold-value*}

**no remote-access threshold session-threshold-exceeded**

## Syntax Description

| | |
|---|---|
| *threshold-value* | Specifies an integer less than or equal to the session limit the FWSM supports. |

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Global configuration | • | • | • | — | — |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Examples

The following example shows how to set a threshold value of 1500:

```
hostname# remote-access threshold session-threshold-exceeded 1500
```

## Related Commands

| Command | Description |
|---|---|
| **snmp-server enable trap remote-access** | Enables threshold trapping. |

# rename

To rename a file or a directory from the source filename to the destination filename, use the **rename** command in privileged EXEC mode.

**rename *[/noconfirm] [*flash:*] source-path* [flash:*] destination-path***

## Syntax Description

| | |
|---|---|
| /noconfirm | (Optional) Suppresses the confirmation prompt. |
| *destination-path* | Specifies the path of the destination file. |
| **flash:** | (Optional) Specifies the internal Flash memory, followed by a colon. |
| *source-path* | Specifies the path of the source file. |

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| **Command Mode** | **Routed** | **Transparent** | **Single** | **Context** | **System** |
| Privileged EXEC | • | • | • | — | • |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | Support for this command was introduced. |

## Usage Guidelines

The **rename flash: flash:** command prompts you to enter a source and destination filename.

You cannot rename a file or directory across file systems.

For example:

```
hostname# rename flash: disk1:
Source filename []? new-config
Destination filename []? old-config
%Cannot rename between filesystems
```

## Examples

The following example shows how to rename a file named “test” to “test1”:

```
hostname# rename flash: flash:
Source filename [running-config]? test
Destination filename [n]? test1
```

## Related Commands

| Command | Description |
|---|---|
| **mkdir** | Creates a new directory. |
| **rmdir** | Removes a directory. |
| **show file** | Displays information about the file system. |

# rename (class-map)

To rename a class map, enter the **rename** command in class-map configuration mode.

**rename** *new_name*

## Syntax Description

| | |
|---|---|
| *new_name* | Specifies the new name of the class map, up to 40 characters in length. The name “class-default” is reserved. |

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| **Command Mode** | **Routed** | **Transparent** | **Single** | **Context** | **System** |
| Class-map configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Examples

The following example shows how to rename a class map from test to test2:

```
hostname(config)# class-map test
hostname(config-cmap)# rename test2
```

## Related Commands

| Command | Description |
|---|---|
| **class-map** | Creates a class map. |

# replication http

To enable HTTP connection replication for the failover group, use the **replication http** command in failover group configuration mode. To disable HTTP connection replication, use the **no** form of this command.

**replication http**

**no replication http**

## Syntax Description

This command has no arguments or keywords.

## Defaults

Disabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Failover group configuration | • | • | — | — | • |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Usage Guidelines

By default, the FWSM does not replicate HTTP session information when Stateful Failover is enabled. Because HTTP sessions are typically short-lived, and because HTTP clients typically retry failed connection attempts, not replicating HTTP sessions increases system performance without causing serious data or connection loss. The **replication http** command enables the stateful replication of HTTP sessions in a Stateful Failover environment, but could have a negative effect on system performance.

This command is available for Active/Active failover only. It provides the same functionality as the **failover replication http** command for Active/Standby failover, except for failover groups in Active/Active failover configurations.

## Examples

The following example shows a possible configuration for a failover group:

```
hostname(config)# failover group 1
hostname(config-fover-group)# primary
hostname(config-fover-group)# preempt 100
hostname(config-fover-group)# replication http
hostname(config-fover-group)# exit
```

## Related Commands

| Command | Description |
|---|---|
| **failover group** | Defines a failover group for Active/Active failover. |
| **failover replication http** | Configures Stateful Failover to replicate HTTP connections. |

# request-command deny

To disallow specific commands within FTP requests, use the **request-command deny** command in ftp map configuration mode, which is accessible by using the **ftp-map** command. To remove the configuration, use the **no** form of this command.

**request-command deny** { **appe** | **cdup** | **dele** | **get** | **help** | **mkd** | **put** | **rmd** | **rnfr** | **rnto** | **site** | **stou** }

**no request-command deny** { **appe** | **cdup** | **help** | **retr** | **rnfr** | **rnto** | **site** | **stor** | **stou** }

## Syntax Description

| | |
|---|---|
| **appe** | Disallows the command that appends to a file. |
| **cdup** | Disallows the command that changes to the parent directory of the current working directory. |
| **dele** | Disallows the command that deletes a file on the server. |
| **get** | Disallows the client command for retrieving a file from the server. |
| **help** | Disallows the command that provides help information. |
| **mkd** | Disallows the command that makes a directory on the server. |
| **put** | Disallows the client command for sending a file to the server. |
| **rmd** | Disallows the command that deletes a directory on the server. |
| **rnfr** | Disallows the command that specifies rename-from filename. |
| **rnto** | Disallows the command that specifies rename-to filename. |
| **site** | Disallows the command that are specific to the server system. Usually used for remote administration. |
| **stou** | Disallows the command that stores a file using a unique filename. |

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Ftp map configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Usage Guidelines

This command is used for controlling the commands allowed within FTP requests traversing the FWSM when using strict FTP inspection.

## Examples

The following example causes the FWSM to drop FTP requests containing **stor**, **stou**, or **appe** commands:

```
hostname(config)# ftp-map inbound_ftp
hostname(config-ftp-map)# request-command deny put stou appe
```

## Related Commands

| Commands | Description |
|---|---|
| **class-map** | Defines the traffic class to which to apply security actions. |
| **ftp-map** | Defines an FTP map and enables ftp map configuration mode. |
| **inspect ftp** | Applies a specific FTP map to use for application inspection. |
| **mask-syst-reply** | Hides the FTP server response from clients. |
| **policy-map** | Associates a class map with specific security actions. |

# request-method

To restrict HTTP traffic based on the HTTP request method, use the **request-method** command in http map configuration mode, which is accessible using the **http-map** command. To disable this feature, use the **no** form of this command.

**request-method** {{ **ext** *ext_methods* | **default**} | { **rfc** *rfc_methods* | **default**}} **action** {**allow** | **reset** | **drop**} [**log**]

**no request-method** { **ext** *ext_methods* | **rfc** *rfc_methods* } **action** {**allow** | **reset** | **drop**} [**log**]

## Syntax Description

| | |
|---|---|
| **action** | Identifies the action taken when a message fails this command inspection. |
| **allow** | Allows the message. |
| **default** | Specifies the default action taken by the FWSM when the traffic contains a supported request method that is not on a configured list. |
| **drop** | Closes the connection. |
| **ext** | Specifies extension methods. |
| *ext-methods* | Identifies one of the extended methods you want to allow to pass through the FWSM. |
| **log** | (Optional) Generates a syslog. |
| **reset** | Sends a TCP reset message to client and server. |
| **rfc** | Specifies RFC 2616 supported methods. |
| *rfc-methods* | Identifies one of the RFC methods you want to allow to pass through the FWSM (see Table 23-2). |

## Defaults

This command is disabled by default. When the command is enabled and a supported request method is not specified, the default action is to allow the connection without logging. To change the default action, use the **default** keyword and specify a different default action.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Http map configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Usage Guidelines

When you enable the **request-method** command, the FWSM applies the specified action to HTTP connections for each supported and configured request method.

The FWSM applies the **default** action to all traffic that does *not* match the request methods on the configured list. The **default** action is to **allow** connections without logging. Given this preconfigured default action, if you specify one or more request methods with the action of **drop** and **log**, the FWSM drops connections containing the configured request methods, logs each connection, and allows all connections containing other supported request methods.

If you want to configure a more restrictive policy, change the default action to **drop** (or **reset**) and **log** (if you want to log the event). Then configure each permitted method with the **allow** action.

Enter the **request-method** command once for each setting you wish to apply. You use one instance of the **request-method** command to change the default action or to add a single request method to the list of configured methods.

When you use the **no** form of the command to remove a request method from the list of configured methods, any characters in the command line after the request method keyword are ignored.

Table 23-2 lists the methods defined in RFC 2616 that you can add to the list of configured methods:

*Table 23-2 RFC 2616 Methods*

| Method | Description |
|---|---|
| connect | Used with a proxy that can dynamically switch to being a tunnel (for example SSL tunneling). |
| delete | Requests that the origin server delete the resource identified by the Request-URI. |
| get | Retrieves whatever information or object is identified by the Request-URI. |
| head | Identical to GET except that the server does not return a message-body in the response. |
| options | Represents a request for information about the communication options available on server identified by the Request-URI. |
| post | Request that the origin server accept the object enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. |
| put | Requests that the enclosed object be stored under the supplied Request-URI. |
| trace | Invokes a remote, application-layer loop-back of the request message. |

## Examples

The following example provides a permissive policy, using the preconfigured default, which allows all supported request methods that are not specifically prohibited.

```
hostname(config)# http-map inbound_http
hostname(config-http-map)# request-method rfc options drop log
hostname(config-http-map)# request-method rfc post drop log
```

In this example, only the **options** and **post** request methods are dropped and the events are logged.

The following example provides a restrictive policy, with the default action changed to **reset** the connection and **log** the event for any request method that is not specifically allowed.

```
hostname(config)# http-map inbound_http
hostname(config-http-map)# request-method rfc default action reset log
hostname(config-http-map)# request-method rfc get allow
hostname(config-http-map)# request-method rfc put allow
```

In this case, the **get** and **put** request methods are allowed. When traffic is detected that uses any other methods, the FWSM resets the connection and creates a syslog entry.

### Related Commands

| Commands | Description |
|---|---|
| **class-map** | Defines the traffic class to which to apply security actions. |
| **debug appfw** | Displays detailed information about traffic associated with enhanced HTTP inspection. |
| **http-map** | Defines an HTTP map for configuring enhanced HTTP inspection. |
| **inspect http** | Applies a specific HTTP map to use for application inspection. |
| **policy-map** | Associates a class map with specific security actions. |

# request-queue

To specify the maximum number of GTP requests that will be queued waiting for a response, use the **request-queue** command in gtp map configuration mode, which is accessed by using the **gtp-map** command. To return this number to the default of 200, use the **no** form of this command.

**request-queue** *max_requests*

**no request-queue** *max_requests*

## Syntax Description

| | |
|---|---|
| *max_requests* | The maximum number of GTP requests that will be queued waiting for a response. The range values is 1 to 4294967295. |

## Defaults

The *max_requests* default is 200.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Gtp map configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Usage Guidelines

The **gtp request-queue** command specifies the maximum number of GTP requests that are queued waiting for a response. When the limit has been reached and a new request arrives, the request that has been in the queue for the longest time is removed. The Error Indication, the Version Not Supported and the SGSN Context Acknowledge messages are not considered as requests and do not enter the request queue to wait for a response.

## Examples

The following example specifies a maximum request queue size of 300 bytes:

```
hostname(config)# gtp-map qtp-policy
hostname(config-gtpmap)# request-queue-size 300
```

## Related Commands

| Commands | Description |
|---|---|
| **clear service-policy inspect gtp** | Clears global GTP statistics. |
| **debug gtp** | Displays detailed information about GTP inspection. |

| Commands | Description |
|---|---|
| **gtp-map** | Defines a GTP map and enables gtp map configuration mode. |
| **inspect gtp** | Applies a specific GTP map to use for application inspection. |
| **show service-policy inspect gtp** | Displays the GTP configuration. |

# reset

When using the Modular Policy Framework, drop packets, close the connection, and send a TCP reset for traffic that matches a **match** command or class map by using the **reset** command in match or class configuration mode. This reset action is available in an inspection policy map (the **policy-map type inspect** command) for application traffic; however, not all applications allow this action. To disable this action, use the **no** form of this command.

**reset** [**log**]

**no reset** [**log**]

## Syntax Description

| | |
|---|---|
| **log** | Logs the match. The system log message number depends on the application. |

## Defaults

No default behaviors or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Match and class configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 4.0(1) | This command was introduced. |

## Usage Guidelines

An inspection policy map consists of one or more **match** and **class** commands. The exact commands available for an inspection policy map depends on the application. After you enter the **match** or **class** command to identify application traffic (the **class** command refers to an existing **class-map type inspect** command that in turn includes **match** commands), you can enter the **reset** command to drop packets and close the connection for traffic that matches the **match** command or **class** command.

If you reset a connection, then no further actions are performed in the inspection policy map. For example, if the first action is to reset the connection, then it will never match any further **match** or **class** commands. If the first action is to log the packet, then a second action, such as resetting the connection, can occur. You can configure both the **reset** and the **log** action for the same **match** or **class** command, in which case the packet is logged before it is reset for a given match.

When you enable application inspection using the **inspect** command in a Layer 3/4 policy map (the **policy-map** command), you can enable the inspection policy map that contains this action, for example, enter the **inspect http** *http_policy_map* command where *http_policy_map* is the name of the inspection policy map.

## Examples

The following example resets the connection and sends a log when they match the http-traffic class map. If the same packet also matches the second **match** command, it will not be processed because it was already dropped.

```
hostname(config-cmap)# policy-map type inspect http http-map1
hostname(config-pmap)# class http-traffic
hostname(config-pmap-c)# reset log
hostname(config-pmap-c)# match req-resp content-type mismatch
hostname(config-pmap-c)# reset log
```

## Related Commands

| Commands | Description |
|---|---|
| **class** | Identifies a class map name in the policy map. |
| **class-map type inspect** | Creates an inspection class map to match traffic specific to an application. |
| **policy-map** | Creates a Layer 3/4 policy map. |
| **policy-map type inspect** | Defines special actions for application inspection. |
| **show running-config policy-map** | Display all current policy map configurations. |

# resource acl-partition

To reduce the number of memory partitions in multiple context mode from the maximum of 12, use the **resource acl-partition** command in global configuration mode. To restore the number of partitions to 12, use the **no** form of this command.

**resource acl-partition** *number*

**no resource acl-partition** *number*

## Syntax Description

| | |
|---|---|
| *number* | Specifies the number of partitions, between 1 and 12.<br>**Note** If you assign a context to a partition, the partition numbering starts with 0. So if you have 12 partitions, the partition numbers are 0 through 11. |

## Defaults

The FWSM uses 12 memory partitions by default.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Global configuration | N/A | N/A | — | — | • |

## Command History

| Release | Modification |
|---|---|
| 2.3(1) | This command was introduced. |

## Usage Guidelines

### Information About Partitions

In multiple context mode, the FWSM partitions the memory allocated to rule configuration, and assigns each context to a partition. You might want to reduce the number of partitions to better match the number of contexts you have. By default, a context belongs to one of 12 partitions that offers a maximum number rules, including ACEs, AAA rules, and others. See the **resource rule** command for a list of rule limits. The FWSM assigns contexts to the partitions in the order they are loaded at startup. For example, if you have 12 contexts and the maximum number of rules is 14,103, each context is assigned to its own partition, and can use 14,103 rules. If you add one more context, then context number 1 and the new context number 13 are both assigned to partition 1, and can use 14,103 rules divided between them; the other 11 contexts continue to use 14,103 rules each. If you delete contexts, the partition membership does not shift, so you might have some unequal distribution until you reboot, at which time the contexts are evenly distributed.

**Note** Rules are used up on a first come, first served basis, so one context might use more rules than another context.

You can manually assign a context to a partition with the **allocate-acl-partition** command.

### How Repartitioning Works

When increasing the number of partitions, the default size of each partition is reduced. If you manually configured the partition sizes using the **size** command, the sizes you set might not be compatible with the new smaller partition sizes. If the current configured sizes do not fit into the new partitions, then the FWSM rejects the **resource acl-partition** command. The FWSM also checks the rule allocation (see the **resource rule** or **rule** command). If you manually allocated rules between features so that the total number of rules allocated is now greater than those available, then the FWSM rejects the **resource acl-partition** command. Similarly, if the absolute maximum number of rules for a feature is now exceeded, then the FWSM rejects the **resource acl-partition** command.

**Note** Changing the number of partitions requires you to reload the FWSM.

### Important Guidelines

**Caution** Failure to follow the following guidelines might result in dropped access list configuration as well as other anomalies, including ACL tree corruption.

- The target partition and rule allocation settings must be carefully calculated, planned, and preferably tested in a non-production environment prior to making the change to ensure that all existing contexts and rules can be accommodated.
- When failover is used, both FWSMs need to be reloaded at the same time after making partition changes. Reloading both FWSMs causes an outage with no possibility for a zero-downtime reload. At no time should two FWSMs with a mismatched number of partitions or rule limits synchronize over failover.

### Clearing the Configuration

If you later enter the **clear configure all** command to restore the default configuration, the **resource acl-partition** command is not changed back to the default. You must enter the **no resource acl-partition** command to restore the default for this command.

## Examples

The following example partitions the memory into 8 parts:

```
hostname(config)# resource acl-partition 8

WARNING: This command leads to re-paritioning of ACL Memory.
It will not take affect until you save the configuration and reboot.
```

## Related Commands

| Command | Description |
|---|---|
| **allocate-acl-partition** | Assigns a context to a specific memory partition. |

| Command | Description |
|---|---|
| **context** | Configures a security context. |
| **show resource acl-partition** | Shows the contexts assigned to each memory partition and the number of rules used. |

# resource partition

To customize a memory partition, including changing the size or reallocating rules between features, use the **resource partition** command in global configuration mode. To remove the resource partition configuration, use the **no** form of this command.

**resource partition** *number*

**no resource partition** *number*

## Syntax Description

| | |
|---|---|
| *number* | Specifies the partition number, between 0 and 11 by default. If you changed the number of partitions using the **resource acl-partition** command, the partition numbering starts with 0. So if you have 10 partitions, the partition numbers are 0 through 9. |

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Global configuration | • | • | — | — | • |

## Command History

| Release | Modification |
|---|---|
| 4.0(1) | This command was introduced. |

## Usage Guidelines

After you enter resource partition configuration mode, you can customize the partition using the **size** and **rule** commands, for example.

### Important Guidelines

**Caution** Failure to follow these guidelines might result in dropped access list configuration as well as other anomalies, including ACL tree corruption.

- The target partition and rule allocation settings must be carefully calculated, planned, and preferably tested in a non-production environment prior to making the change to ensure that all existing contexts and rules can be accommodated.
- When failover is used, both FWSMs need to be reloaded at the same time after making partition changes. Reloading both FWSMs causes an outage with no possibility for a zero-downtime reload. At no time should two FWSMs with a mismatched number of partitions or rule limits synchronize over failover.

## Examples

The following example enters resource partition configuration mode, and changes the size 5000 rules:

```
hostname(config)# resource partition 0
hostname(config-partition)# size 5000
```

## Related Commands

| Command | Description |
|---|---|
| **allocate-acl-partition** | Assigns a context to a specific memory partition. |
| **clear configure resource partition** | Clears the current memory partition configuration. |
| **resource acl-partition** | Sets the total number of memory partitions. |
| **resource rule** | Reallocates rules between features globally for all partitions. |
| **rule** | Reallocates rules between features for a specific partition. |
| **show resource acl-partition** | Shows the current memory partition characteristics, including the sizes and allocated contexts. |
| **show resource partition** | Shows the memory partition sizes. |
| **show resource rule** | Shows the current allocation of rules. |
| **show running-config resource partition** | Shows the current memory partition configuration. |
| **size** | Changes the size of a memory partition. |

# resource rule

To reallocate rules between features, use the **resource rule** command in global configuration mode. To restore the default values, use the **no** form of this command.

**resource rule nat** {*max_policy_nat_rules* | **current** | **default** | **max**}
**acl** {*max_ace_rules* | **current** | **default** | **max**}
**filter** {*max_filter_rules* | **current** | **default** | **max**}
**fixup** {*max_inspect_rules* | **current** | **default** | **max**}
**est** {*max_established_rules* | **current** | **default** | **max**}
**aaa** {*max_aaa_rules* | **current** | **default** | **max**}
**console** {*max_console_rules* | **current** | **default** | **max**}

**no resource rule**

## Syntax Description

| | |
|---|---|
| **aaa** *max_aaa_rules* | Sets the maximum number of AAA rules, between 0 and 10000. |
| **acl** *max_ace_rules* | Sets the maximum number of ACEs, between 0 and 74188. |
| **console** *max_console_rules* | Sets the maximum number of ICMP, Telnet, SSH, and HTTP rules, between 0 and 4000. |
| **current** | Keeps the current value set. |
| **default** | Sets the maximum rules to the default. To view the defaults, use the **show resource rule** command. |
| **est** *max_established_rules* | Sets the maximum number of **established** commands, between 0 and 716. The **established** command creates two types of rules, control and data. You allocate both rules by setting the number of **established** commands; you do not set each rule separately. However, both of these types are shown in the **show resource rule** and **show np 3 acl count** displays, so be sure to double the **est** value when comparing the total number of rules configured with the display in the **show** commands. |
| **filter** *max_filter_rules* | Sets the maximum number of filter rules, between 0 and 6000. |
| **fixup** *max_inspect_rules* | Sets the maximum number of inspect rules, between 0 and 10000. |
| **max** | Sets the rules to the maximum allowed for the feature. Be sure to set other features lower to accommodate this value. |
| **nat** *max_policy_nat_rules* | Sets the maximum number of policy NAT ACEs, between 0 and 10000. |

## Defaults

Use the **show resource rule** command to view default values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Global configuration | • | • | • | — | • |

## Command History

| Release | Modification |
|---|---|
| 3.2(1) | This command was introduced. |

## Usage Guidelines

### Information About Rules

There are a fixed number of rules available on the FWSM, so you might want to reallocate rules between features depending on usage. Features that use rules include access lists, inspections, AAA, and more.

If you increase the value for one feature, then you must decrease the value by the same amount for one or more features so the total number of rules does not exceed the system limit. Use the **show resource rule** command to view the total number of rules available, the default values, current rule allocation, and the absolute maximum number of rules you can allocate per feature.

### Important Guidelines

**Caution** Failure to follow these guidelines might result in dropped access list configuration as well as other anomalies, including ACL tree corruption.

- The target partition and rule allocation settings must be carefully calculated, planned, and preferably tested in a non-production environment prior to making the change to ensure that all existing contexts and rules can be accommodated.
- When failover is used, both FWSMs need to be reloaded at the same time after making partition changes. Reloading both FWSMs causes an outage with no possibility for a zero-downtime reload. At no time should two FWSMs with a mismatched number of partitions or rule limits synchronize over failover.
- You must enter all arguments in this command.
- This command takes effect immediately.
- If you increase the size of a partition (the **size** command, or if you decrease the number of partitions using the **resource acl-partition** command) but have not yet reloaded, the maximum number of rules remains at the old smaller size. You have to reload to see the increased limits. If you decrease the size of a partition but have not yet reloaded, the new smaller number of rules is reflected right away.

### Viewing Rules

To view the number of rules currently being used so you can plan your reallocation, enter one of the following commands.

- In single mode or within a context, enter the following command:

```
hostname(config)# show np 3 acl count 0
```

- In multiple context mode system execution space, enter the following command:

  ```
  hostname(config)# show np 3 acl count partition_number
  ```

For example, the following display shows the number of inspections (Fixup Rule) close to the maximum of 9216. You might choose to reallocate some access list rules (ACL Rule) to inspections.

```
hostname(config)# show np 3 acl count 0

-------------- CLS Rule Current Counts --------------
CLS Filter Rule Count       :            0
CLS Fixup Rule Count        :           32
CLS Est Ctl Rule Count      :            0
CLS AAA Rule Count          :            0
CLS Est Data Rule Count     :            0
CLS Console Rule Count      :            1
CLS Policy NAT Rule Count   :            0
CLS ACL Rule Count          :            0
CLS ACL Uncommitted Add     :            0
CLS ACL Uncommitted Del     :            0
...
```

### Rules in Multiple Context Mode

In multiple context mode with the default of 12 memory partitions, each context supports the maximum number of rules; the actual number of rules supported in a context might be more or less, depending on how many contexts you have and how many partitions you configure. See the **resource acl-partition** command for information about memory distribution among contexts.

If you reduce the number of partitions, the maximum number of rules is recalculated and might not match the total system number available for 12 partitions. To view the maximum number of rules for partitions, enter the following command in the system execution space:

```
hostname(config)# show resource rule
```

For example, the following display shows the maximum rules as 19219 per partition with 12 partitions (this is an example only, and might differ from the actual number of rules for your system):

```
hostname(config)# show resource rule

             Default  Configured  Absolute
 CLS Rule     Limit      Limit      Max
-----------+---------+----------+---------
 Policy NAT     384        384        833
 ACL          14801      14801      14801
 Filter         576        576       1152
 Fixup         1537       1537       3074
 Est Ctl         96         96         96
 Est Data        96         96         96
 AAA           1345       1345       2690
 Console        384        384        768
-----------+---------+----------+---------
 Total        19219      19219

Partition Limit - Configured Limit = Available to allocate
      19219     -      19219       =           0
```

To override the global setting for rule reallocation, use the rule command to set the **rule** allocation for a specific partition.

## Examples

The following example reallocates 1000 rules from the single-mode default 74,188 ACEs to inspections (default 4147):

```
hostname(config)# resource rule nat default acl 73188 filter default fixup 5157 est
default aaa default console default
```

In multiple context mode with 12 partitions, to reallocate 100 ACEs (default 10,633) to inspections (default 1417) as well as all but one established rule (default 70) to filter (default 425), enter the following command:

```
hostname(config)# resource rule nat default acl 10533 filter 494 fixup 1517 est 1 aaa
default console default
```

## Related Commands

| Command | Description |
|---|---|
| **allocate-acl-partition** | Assigns a context to a specific memory partition. |
| **context** | Configures a security context. |
| **resource acl-partition** | Sets the number of memory partitions for rules. |
| **rule** | Sets the resource rule allocation for a specific partition. |
| **show np 3 acl count** | Shows the number of rules in use. |
| **show resource acl-partition** | Shows the contexts assigned to each memory partition and the number of rules used. |
| **show resource rule** | Shows the total number of rules available, the default values, current rule allocation, and the absolute maximum number of rules you can allocate per feature. |

# retry-interval

To configure the amount of time between retry attempts for a particular AAA server designated in a prior **aaa-server host** command, use the **retry-interval** command in aaa-server host mode. To reset the retry interval to the default value, use the **no** form of this command.

**retry-interval** *seconds*

**no retry-interval**

## Syntax Description

| | |
|---|---|
| *seconds* | Specify the retry interval (1-10 seconds) for the request. This is the time the FWSM waits before retrying a connection request. |

## Defaults

The default retry interval is 10 seconds.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Aaa-server host configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Usage Guidelines

Use the **retry-interval** command to specify or reset the number of seconds the FWSM waits between connection attempts. Use the **timeout** command to specify the length of time during which the FWSM attempts to make a connection to a AAA server.

## Examples

The following examples show the retry-interval command in context:

```
hostname(config)# aaa-server svrgrp1 protocol radius
hostname(config-aaa-server-group)# aaa-server svrgrp1 host 209.165.200.225
hostname(config-aaa-server-host)# timeout 7
hostname(config-aaa-server-host)# retry-interval 9
```

## Related Commands

| Command | Description |
|---|---|
| **aaa-server host** | Enters aaa server host configuration mode so that you can configure AAA server parameters that are host-specific. |

| | |
|---|---|
| **clear configure aaa-server** | Removes all AAA command statements from the configuration. |
| **show running-config aaa-server** | Displays AAA server statistics for all AAA servers, for a particular server group, for a particular server within a particular group, or for a particular protocol. |
| **timeout** | Specifies the length of time during which the FWSM attempts to make a connection to a AAA server. |

# re-xauth

To require that users reauthenticate on IKE rekey, issue the **re-xauth enable** command in group-policy configuration mode. To disable user reauthentication on IKE rekey, use the **re-xauth disable** command. To remove the re-xauth attribute from the running configuration, use the **no** form of this command. This enables inheritance of a value for reauthentication on IKE rekey from another group policy.

**re-xauth** {**enable** | **disable**}

**no re-xauth**

## Syntax Description

| | |
|---|---|
| **disable** | Disables reauthentication on IKE rekey. |
| **enable** | Enables reauthentication on IKE rekey. |

## Defaults

Reauthentication on IKE rekey is disabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Group policy configuration | • | — | • | — | — |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | This command was introduced. |

## Usage Guidelines

If you enable reauthentication on IKE rekey, the FWSM prompts the user to enter a username and password during initial Phase 1 IKE negotiation and also prompts for user authentication whenever an IKE rekey occurs. Reauthentication provides additional security.

If the configured rekey interval is very short, users might find the repeated authorization requests inconvenient. In this case, disable reauthentication. To check the configured rekey interval, in monitoring mode, issue the **show crypto ipsec sa** command to view the security association lifetime in seconds and lifetime in kilobytes of data.

**Note** The reauthentication fails if there is no user at the other end of the connection.

## Examples

The following example shows how to enable reauthentication on rekey for the group policy named FirstGroup:

```
hostname(config) #group-policy FirstGroup attributes
hostname(config-group-policy)# re-xauth enable
```

# rip

To enable and change RIP settings, use the **rip** command in global configuration mode. To disable the FWSM RIP routing table updates, use the **no** form of this command.

**rip** *if_name* {**default** | **passive**} [**version** {**1** | **2** [**authentication** {**text** | **md5**} *key key_id*]}]

**no rip** *if_name* {**default** | **passive**} [**version** {**1** | **2** [**authentication {text** | **md5}** *key key_id*]}]

## Syntax Description

| | |
|---|---|
| **authentication** | (Optional) Enables RIP version 2 authentication. |
| **default** | Broadcast a default route on the interface. |
| *if_name* | The interface on which RIP is being enabled. |
| *key* | Key to authenticate RIP updates. |
| *key_id* | Key identification value; valid values range from 1 to 255. |
| **md5** | Uses MD5 for RIP message authentication. |
| **passive** | Enables passive RIP on the interface. The interface listens for RIP routing broadcasts and uses that information to populate the routing tables but does not broadcast routing updates. |
| **text** | Uses clear text for RIP message authentication (not recommended). |
| **version** | (Optional) Specifies the RIP version; valid values are **1** and **2**. |

## Defaults

RIP is disabled.

If you do not specify a version, RIP version 1 is enabled by default.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Global configuration | • | — | • | — | — |

## Command History

| Release | Modification |
|---|---|
| 1.1(1) | This command was introduced. |

## Usage Guidelines

The **rip** command lets you to enable the sending and receiving of RIP routing updates on an interface. You configure RIP update transmission and reception independently; you can enable transmission only, reception only, or both transmission and reception on each interface. Use the **passive** keyword with the **rip** command to enable RIP update reception. Use the **default** keyword with the **rip** command to enable the broadcast of a default route. To enable both transmission and reception of RIP updates on an interface, you must two **rip** commands for the interface, one with the **default** keyword, enabling the sending of RIP routing updates, and one with the **passive** keyword, enabling the interface to receive RIP updates and to populate the routing table with those updates.

**Note** The FWSM cannot pass RIP updates between interfaces.

If you specify RIP version 2, you can enable neighbor authentication and use MD5-based encryption to authenticate the RIP updates. When you enable neighbor authentication, you must ensure that the *key* and *key_id* arguments are the same as those used by neighbor devices that provide RIP version 2 updates. The *key* is a text string of up to 16 characters.

Configuring RIP Version 2 registers the multicast address 224.0.0.9 on the respective interface to be able to accept multicast RIP Version 2 updates. When RIP Version 2 is configured in passive mode, the FWSM accepts RIP Version 2 multicast updates with an IP destination of 224.0.0.9. When RIP Version 2 is configured in default mode, the FWSM transmits default route updates using an IP multicast destination of 224.0.0.9. Removing the RIP version 2 commands for an interface unregisters the multicast address from the interface card.

**Note** Only Intel 10/100 and Gigabit interfaces support multicasting.

RIP is not supported under transparent mode. By default, the FWSM denies all RIP broadcast and multicast packets. To permit these RIP messages to pass through a FWSM operating in transparent mode you must define access list entries to permit this traffic. For example, to permit RIP version 2 traffic through the security appliance, create an access list entry like `access-list myriplist extended permit ip any host 224.0.0.9`. To permit RIP version 1 broadcasts, create an access list entry like `access-list myriplist extended permit udp any any eq rip`. Apply these access list entries to the appropriate interface using the **access-group** command.

## Examples

The following example shows how to combine version 1 and version 2 commands and list the information with the **show running-config rip** command after entering the **rip** commands. The **rip** commands let you to do the following.

- Enable version 2 passive and default RIP using MD5 authentication on the outside interface to encrypt the key that is used by the FWSM and other RIP peers, such as routers.
- Enable version 1 passive RIP listening on the inside interface of the FWSM.
- Enable version 2 passive RIP listening on the dmz (demilitarized) interface of the FWSM.

```
hostname(config)# rip outside passive version 2 authentication md5 thisisakey 2
hostname(config)# rip outside default version 2 authentication md5 thisisakey 2
hostname(config)# rip inside passive
hostname(config)# rip dmz passive version 2

hostname# show running-config rip
rip outside passive version 2 authentication md5 thisisakey 2
rip outside default version 2 authentication md5 thisisakey 2
rip inside passive version 1
rip dmz passive version 2
```

The following example shows how to use the version 2 feature that passes the encryption key in text form:

```
hostname(config)# rip out default version 2 authentication text thisisakey 3
hostname# show running-config rip
```

```
rip outside default version 2 authentication text thisisakey 3
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure rip** | Clears all RIP commands from the running configuration. |
| **debug rip** | Displays debug information for RIP. |
| **show running-config rip** | Displays the RIP commands in the running configuration. |

# rmdir

To remove the existing directory, use the **rmdir** command in privileged EXEC mode.

**rmdir** [**/noconfirm**] [**flash:**]*path*

## Syntax Description

| | |
|---|---|
| flash: | (Optional) Specifies the nonremovable internal Flash, followed by a colon. |
| noconfirm | (Optional) Suppresses the confirmation prompt. |
| *path* | (Optional) The absolute or relative path of the directory to remove. |

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Privileged EXEC | • | • | • | — | • |

## Command History

| Release | Modification |
|---|---|
| 3.1(1) | Support for this command was introduced. |

## Usage Guidelines

If the directory is not empty, the **rmdir** command fails.

## Examples

The following example shows how to remove an existing directory named “test”:

```
hostname# rmdir test
```

## Related Commands

| Command | Description |
|---|---|
| **dir** | Displays the directory contents. |
| **mkdir** | Creates a new directory. |
| **pwd** | Displays the current working directory. |
| **show file** | Displays information about the file system. |

# route

To enter a static or default route for the specified interface, use the **route** command in global configuration mode. To remove routes from the specified interface, use the **no** form of this command.

**route** *interface_name ip_address netmask gateway_ip* [*metric*]

**no route** *interface_name ip_address netmask gateway_ip* [*metric*]

## Syntax Description

| | |
|---|---|
| *gateway_ip* | Specifies the IP address of the gateway router (the next-hop address for this route).<br>**Note** The *gateway_ip* argument is optional in transparent mode. |
| *interface_name* | Internal or external network interface name. |
| *ip_address* | Internal or external network IP address. |
| *metric* | (Optional) The administrative distance for this route. Valid values range from 1 to 255. The default value is 1. |
| *netmask* | Specifies a network mask to apply to *ip_address*. |

## Defaults

The *metric* default is 1.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| **Command Mode** | **Routed** | **Transparent** | **Single** | **Context** | **System** |
| Global configuration | • | • | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 1.1(1) | This command was introduced. |

## Usage Guidelines

Use the **route** command to enter a default or static route for an interface. To enter a default route, set *ip_address* and *netmask* to **0.0.0.0,** or use the shortened form of **0**. All routes that are entered using the **route** command are stored in the configuration when it is saved.

Create static routes to access networks that are connected outside a router on any interface. For example, the FWSM sends all packets that are destined to the 192.168.42.0 network through the 192.168.1.5 router with this static **route** command.

```
hostname(config)# route dmz 192.168.42.0 255.255.255.0 192.168.1.5 1
```

Once you enter the IP address for each interface, the FWSM creates a CONNECT route in the route table. This entry is not deleted when you use the **clear route** or **clear configure route** commands.

If the **route** command uses the IP address from one of the interfaces on the FWSM as the gateway IP address, the FWSM will ARP for the destination IP address in the packet instead of ARPing for the gateway IP address.

## Examples

The following example shows how to specify one default **route** command for an outside interface:

```
hostname(config)# route outside 0 0 209.165.201.1 1
```

The following example shows how to add these static **route** commands to provide access to the networks:

```
hostname(config)# route dmz1 10.1.2.0 255.0.0.0 10.1.1.4 1
hostname(config)# route dmz1 10.1.3.0 255.0.0.0 10.1.1.4 1
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure route** | Removes statically configured **route** commands. |
| **clear route** | Removes routes learned through dynamic routing protocols such as RIP. |
| **show route** | Displays route information. |
| **show running-config route** | Displays configured routes. |

# route-inject

To inject the connected and static routes and NAT pools configured on FWSM into the MSFC routing table, use the **route-inject** command in global configuration mode. To delete the connection, use the **no** form of this command or the **clear configure route-inject** command.

**route-inject**

**no route-inject**

## Syntax Description

This command has no arguments or keywords.

## Defaults

No default behavior or values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| **Command Mode** | **Routed** | **Transparent** | **Single** | **Context** | **System** |
| Global configuration | • | — | • | • | — |

## Command History

| Release | Modification |
|---|---|
| 4.0(1) | This command was introduced. |

## Usage Guidelines

The **route-inject** command allows you to inject the connected and static routes and NAT pools configured on FWSM into the MSFC routing table.

FWSM injects the IP address of the FWSM interface as the next-hop IP address for specific destination addresses to the connected and static routes and NAT pools configured on FWSM into the routing table of the local switch.

For example, if you wanted to configure a NAT pool on FWSM, the MFSC and other external routers do not know that those NAT pool addresses are on FWSM unless the user configures the static routes on MSFC to point to the FWSM interface. But by utilizing RHI, you can inject the NAT pool addresses to point to the FWSM interface so the MSFC can automatically forward that traffic to the FWSM.

Because FWSM only supports OSPF or other dynamic routing protocols in single routed-mode, RHI can be used in multi-mode to inject routes (connected/static) to the MSFC, which can then redistribute these routes through OSPF or other dynamic routing protocols. This allows FWSM to redistribute FWSM routes through OSPF or other dynamic routing protocols even when running multi-mode, by utilizing the MSFC routing protocols and RHI.

**Note** The connected and static routes and NAT pools can be selectively injected by configuring a redistribute policy using a standard access-list, route-map or global pool ID (only for NAT).

RHI is supported in both single and multi-mode, but not Transparent mode. Additionally, RHI is supported with HA (Active/Standby and Active/Active).

## Examples

### Configuring RHI for NAT with Standard ACL

In this example, only a perfect match will be injected. The **acl1**, 23.10.143.20/30 is injected with nexthop of 20.22.211.21 (Active IP of "outside") on vlan 20 (vlan of "outside").

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 20.22.211.21 255.255.255.0 standby 20.22.211.22
hostname(config-if)# exit
hostname(config)# access-list acl1 standard permit 23.10.143.20 255.255.255.252
hostname(config)# global (outside) 10 23.10.143.20-23.10.143.23 netmask 255.255.255.0
hostname(config)# global (outside) 10 23.10.143.40-23.10.143.45 netmask 255.255.255.0
hostname(config)# route-inject
hostname(config-route-inject)# redistribute nat access-list acl1 interface outside
```

### Configuring RHI for NAT with Global Pool ID

In this example, 23.11.111.1-23.11.111.7 and 23.11.111.10-23.11.111.20 injected with nexthop 20.11.111.11 on vlan 20. Be sure that the global interface and pool ID match the **redistribute** command.

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 20.11.111.11 255.255.255.0 standby 20.11.111.21
hostname(config-if)# exit
hostname(config)# global (dmz) 10 22.11.111.1-22.11.111.10 netmask 255.255.255.0
hostname(config)# global (outside) 10 23.11.111.1-23.11.111.7 netmask 255.255.255.0
hostname(config)# global (outside) 10 23.11.111.10-23.11.111.20 netmask 255.255.255.0
hostname(config)# global (outside) 20 23.11.111.30-23.11.111.40 netmask 255.255.255.0
hostname(config)# route-inject
hostname(config-route-inject)# redistribute nat global-pool 10 interface outside
```

### Configuring RHI for Static Route using route-map

In this example, 23.11.111.0/24 and 25.11.111.0/24 will be injected with nexthop of 20.11.111.11 on vlan 20. The **route-map** command can be used to match destination IP, nexthop IP, metric, or interface.

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 20.11.111.11 255.255.255.0 standby 20.11.111.12
hostname(config-if)# exit
hostname(config)# access-list acl1 standard permit 23.11.111.0 255.255.255.0
hostname(config)# access-list acl2 standard permit 25.11.111.0 255.255.255.0
hostname(config)# route-map map1 permit 10
hostname(config-route-map)# match ip address acl1 acl2
hostname(config-route-map)# exit
hostname(config)# route outside 23.11.111.0 255.255.255.0 23.11.111.9
hostname(config)# route outside 24.11.111.0 255.255.255.0 24.11.111.9
hostname(config)# route outside 25.11.111.0 255.255.255.0 25.11.111.9
hostname(config)# route-inject
hostname(config-route-inject)# redistribute static route-map map1 interface outside
```

**Note** Route maps can only be used in single routed mode.

## Related Commands

| Command | Description |
|---|---|
| **clear configure route-inject** | Removes the routes/NAT pools that were injected into the MSFC routing tables. Additionally, removes the redistribute and route-inject configuration for the user context if you are in multi-mode or system context if in single routed mode. |
| **debug route-inject** | Enables debugging of the route-injections that have been configured on the FWSM. |
| **redistribute** | Configures the type of route or NAT pools to inject. |
| **show route-inject** | Displays the routes and NAT pools that have been injected. |
| **show running-config route-inject** | Displays the route-injection running configuration. |

# route-map

To define the conditions for redistributing routes from one routing protocol into another, use the **route-map** command in global configuration mode. To delete a map, use the **no** form of this command.

**route-map** *map_tag* [**permit** | **deny**] [*seq_num*]

**no route-map** *map_tag* [**permit** | **deny**] [*seq_num*]

## Syntax Description

| | |
|---|---|
| **deny** | (Optional) Specifies that if the match criteria are met for the route map, the route is not redistributed. |
| *map_tag* | Text for the route map tag; the text can be up to 57 characters in length. |
| **permit** | (Optional) Specifies that if the match criteria is met for this route map, the route is redistributed as controlled by the set actions. |
| *seq_num* | (Optional) Route map sequence number; valid values are from 0 to 65535. Indicates the position that a new route map will have in the list of route maps already configured with the same name. |

## Defaults

The defaults are as follows:

- **permit**.
- If you do not specify a *seq_num*, a *seq_num* of 10 is assigned to the first route map.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Global configuration | • | — | • | — | — |

## Command History

| Release | Modification |
|---|---|
| 1.1(1) | This command was introduced. |

## Usage Guidelines

The **route-map** command lets you redistribute routes.

The **route-map** global configuration command and the **match** and **set** configuration commands define the conditions for redistributing routes from one routing protocol into another. Each **route-map** command has **match** and **set** commands that are associated with it. The **match** commands specify the match criteria that are the conditions under which redistribution is allowed for the current **route-map** command. The **set** commands specify the set actions, which are the redistribution actions to perform if the criteria enforced by the **match** commands are met. The **no route-map** command deletes the route map.

The **match route-map** configuration command has multiple formats. You can enter the **match** commands in any order, and all **match** commands must pass to cause the route to be redistributed according to the set actions given with the **set** commands. The **no** form of the **match** commands removes the specified match criteria.

Use route maps when you want detailed control over how routes are redistributed between routing processes. You specify the destination routing protocol with the **router ospf** global configuration command. You specify the source routing protocol with the **redistribute** router configuration command.

When you pass routes through a route map, a route map can have several parts. Any route that does not match at least one match clause relating to a **route-map** command is ignored; the route is not advertised for outbound route maps and is not accepted for inbound route maps. To modify only some data, you must configure a second route map section with an explicit match specified.

The *seq_number* argument is as follows:

1. If you do not define an entry with the supplied tag, an entry is created with the *seq_number* argument set to 10.
2. If you define only one entry with the supplied tag, that entry becomes the default entry for the following **route-map** command. The *seq_number* argument of this entry is unchanged.
3. If you define more than one entry with the supplied tag, an error message is printed to indicate that the *seq_number* argument is required.

If the **no route-map** *map-tag* command is specified (with no *seq-num* argument), the whole route map is deleted (all **route-map** entries with the same *map-tag* text).

If the match criteria are not met, and you specify the **permit** keyword, the next route map with the same *map_tag* is tested. If a route passes none of the match criteria for the set of route maps sharing the same name, it is not redistributed by that set.

## Examples

The following example shows how to configure a route map in OSPF routing:

```
hostname(config)# route-map maptag1 permit 8
hostname(config-route-map)# set metric 5
hostname(config-route-map)# match metric 5
hostname(config-route-map)# show running-config route-map
route-map maptag1 permit 8
    set metric 5
    match metric 5
hostname(config-route-map)# exit
hostname(config)#
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure route-map** | Removes the conditions for redistributing the routes from one routing protocol into another routing protocol. |
| **match interface** | Distributes any routes that have their next hop out one of the interfaces specified. |
| **router ospf** | Starts and configures an ospf routing process. |
| **set metric** | Specifies the metric value in the destination routing protocol for a route map. |
| **show running-config route-map** | Displays the information about the route map configuration. |

# route-monitor

To monitor routes and switch to an alternate path in the event a router goes down, use the **route-monitor** command in interface configuration mode. To remove route monitoring, use the **no** form of this command.

**route-monitor** *network_address network_mask* [**query interval** *interval*] [**max-failures** *failures*]

**no route-monitor** *network_address network_mask* [**query interval** *interval*] [**max-failures** *failures*]

## Syntax Description

| | |
|---|---|
| **max-failures** *failures* | (Optional) Specifies the number of ICMP queries that are not replied to before the route is considered down. Valid values are between 3 and 200; the default value is 5. |
| *network address* | Specifies the network address to be monitored. |
| *network mask* | Specifies the network mask for the address to be monitored. |
| **query interval** *interval* | (Optional) Specifies the interval value in milliseconds. Valid values are between **100** or **3000**; the default value is 300. |

## Defaults

The default value for maximum number of failures is 5, and the default value for query intervals is 300 milliseconds.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Interface configuration | • | • | • | — | — |

## Command History

| Release | Modification |
|---|---|
| 4.0(1) | This command was introduced. |

## Usage Guidelines

**Note** Currently, you can only monitor routes for one network as specified in the **route-monitor** command.

If you configured multiple static or default routes, FWSM lets you configure multiple routes to monitor whether there are any problems on the active route, and if so, switches to an alternate route on the network in the event a router goes down.

To do this, FWSM route monitoring process starts to send out ICMP queries to determine the best two static route for the destination network and a back up route at a configurable interval of time set. The interval of sending the ICMP query is set by the *interval* keyword; valid values are 100 to 3000, with the default value at 300 milliseconds. The query is always sent to both of the chosen routers, keeping the current available status locally.

The two routes chosen have the least metric distance, with the lowest chosen as the best path to send traffic. In the FWSM, the **route-monitor** command will automatically choose the best two routes among the static routes configured. The next best path always gets installed in the routing table when the current route goes down, and the current one becomes the backup router.

If the ICMP query does not receive a configurable threshold number set by the *failures* keyword, the router is determined to be unreachable. The *failures* keyword is the maximum number of ICMP queries that are not replied to before the router is determined to be down; the default value being five seconds. At this point the backup route takes precedence, provided this route was reachable, and becomes the best route. The original route then becomes the backup route.

If the original best route becomes reachable again, then FWSM switches back to that route and the current best route becomes the backup route. If in case both routes become unreachable, then both are made backup routes. However, there is no change in the routing table.

To monitor a static or default route, enter the following command:

```
hostname(config-if)# route-monitor network_address network_mask [query_interval interval]
[max-failures failures]
```

## Examples

This example shows how to monitor a static route:

```
hostname(config-if)# route-monitor 192.168.1.0 255.255.255.0
```

## Related Commands

| Command | Description |
|---|---|
| **show running-config router** | Displays the commands in the global router configuration. |

# router bgp

To start a BGP routing process and configure parameters for that process, use the **router bgp** command in global configuration mode. To disable BGP routing, use the **no** form of this command.

**router bgp** *as-number*

**no router bgp** *as-number*

## Syntax Description

| | |
|---|---|
| *as-number* | Number of an autonomous system that identifies the FWSM to other BGP routers and tags the routing information passed along. The *as-number* assigned to the BGP stub routing process must be the same as the BGP neighbor *as-number.* |

## Defaults

BGP routing is disabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context[1] | System |
| Global configuration | • | — | • | • | — |

[1] This command is only available in the admin context.

## Command History

| Release | Modification |
|---|---|
| 3.2(1) | This command was introduced. |

## Usage Guidelines

The **router bgp** command is the global configuration command for BGP routing processes running on the FWSM. Once you enter the **router bgp** command, the command prompt appears as `hostname(config-router)#`, indicating that you are in router configuration mode. The **no router bgp** command terminates the BGP routing process.

The AS number assigned to the BGP stub routing process must be the same as the BGP neighbor AS number.

The **router bgp** command is used with the following BGP-specific commands to configure BGP routing process:

- **bgp router id**—Specified the BGP router ID for the FWSM.
- **neighbor**—Specifies the neighbor BGP router.
- **network**—Specifies the networks that can be advertised by the BGP routing process.

In multiple context mode, this command is only available in the admin context. The admin context must be in routed mode. The BGP stub routing configuration entered in the admin context applies to all contexts configured on the device; you cannot configure BGP stub routing on a per-context basis.

## Examples

The following example shows how to enter the configuration mode for the BGP routing process. The FWSM belongs to AS 800:

```
hostname(config)# router bgp 800
hostname(config-router)#
```

## Related Commands

| Command | Description |
|---|---|
| **bgp router-id** | Specifies the BGP router ID for the FWSM. |
| **clear configure router** | Clears the **router** commands from the running configuration. |
| **neighbor remote-as** | Specifies the neighbor BGP router. |
| **network** | Specifies the networks that can be advertised by the BGP routing process. |
| **show running-config router** | Displays the **router** commands in the running configuration. |

# router eigrp

To start an EIGRP routing process and configure parameters for that process, use the **router eigrp** command in global configuration mode. To disable EIGRP routing, use the **no** form of this command.

**router eigrp** *as-number*

**no router eigrp** *as-number*

## Syntax Description

| | |
|---|---|
| *as-number* | Autonomous system number that identifies the routes to the other EIGRP routers. It is also used to tag the routing information. Valid values are from 1 to 65535. |

## Defaults

EIGRP routing is disabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Global configuration | • | — | • | — | — |

## Command History

| Release | Modification |
|---|---|
| 4.0(1) | This command was introduced. |

## Usage Guidelines

The **router eigrp** command creates an EIGRP routing process or enters router configuration mode for an existing EIGRP routing process. You can only create a single EIGRP routing process on the FWSM.

Use the following router configuration mode commands to configure the EIGRP routing processes:

- **auto-summary**—Enable/disable automatic route summarization.
- **default-information**—Enable/disable the reception and sending of default route information.
- **default-metric**—Define the default metrics for routes redistributed into the EIGRP routing process.
- **distance eigrp**—Configure the administrative distance for internal and external EIGRP routes.
- **distribute-list**—Filter the networks received and sent in routing updates.
- **eigrp log-neighbor-changes**—Enable/disable the logging of neighbor state changes.
- **eigrp log-neighbor-warnings**—Enable/disable the logging of neighbor warning messages.
- **eigrp router-id**—Creates a fixed router ID.
- **eigrp stub**—Configures the FWSM for stub EIGRP routing.
- **neighbor**—Statically define an EIGRP neighbor.

- **network**—Configure the networks that participate in the EIGRP routing process.
- **passive-interface**—Configure an interface to act as a passive interface.
- **redistribute**—Redistribute routes from other routing processes into EIGRP.

Use the following interface configuration mode commands to configure interface-specific EIGRP parameters:

- **authentication key eigrp**—Define the authentication key used for EIGRP message authentication.
- **authentication mode eigrp**—Define the authentication algorithm used for EIGRP message authentication.
- **delay**—Configure the delay metric for an interface.
- **hello-interval eigrp**—Change the interval at which EIGRP hello packets are sent out of an interface.
- **hold-time eigrp**—Change the hold time advertised by the FWSM.
- **split-horizon eigrp**—Enable/disable EIGRP split-horizon on an interface.
- **summary-address eigrp**—Manually define a summary address.

## Examples

The following example shows how to enter the configuration mode for the EIGRP routing process with the autonomous system number 100:

```
hostname(config)# router eigrp 100
hostname(config-router)#
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure router eigrp** | Clears the EIGRP router configuration mode commands from the running configuration. |
| **show running-config router eigrp** | Displays the EIGRP router configuration mode commands in the running configuration. |

# router ospf

To start an OSPF routing process and configure parameters for that process, use the **router ospf** command in global configuration mode. To disable OSPF routing, use the **no** form of this command.

**router ospf** *pid*

**no router ospf** *pid*

## Syntax Description

| | |
|---|---|
| *pid* | Internally used identification parameter for an OSPF routing process; valid values are from 1 to 65535. The *pid* does not need to match the ID of OSPF processes on other routers. |

## Defaults

OSPF routing is disabled.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Global configuration | • | — | • | — | — |

## Command History

| Release | Modification |
|---|---|
| 1.1(1) | This command was introduced. |

## Usage Guidelines

The **router ospf** command is the global configuration command for OSPF routing processes running on the FWSM. Once you enter the **router ospf** command, the command prompt appears as (config-router)#, indicating that you are in router configuration mode.

When using the **no router ospf** command, you do not need to specify optional arguments unless they provide necessary information. The **no router ospf** command terminates the OSPF routing process specified by its *pid.* You assign the *pid* locally on the FWSM. You must assign a unique value for each OSPF routing process.

The **router ospf** command is used with the following OSPF-specific commands to configure OSPF routing processes:

- **area**—Configures a regular OSPF area.
- **compatible rfc1583**—Restores the method used to calculate summary route costs per RFC 1583.
- **default-information originate**—Generates a default external route into an OSPF routing domain.
- **distance**—Defines the OSPF route administrative distances based on the route type.
- **ignore**—Suppresses the sending of syslog messages when the router receives a link-state advertisement (LSA) for type 6 Multicast OSPF (MOSPF) packets.

- **log-adj-changes**—Configures the router to send a syslog message when an OSPF neighbor goes up or down.
- **neighbor**—Specifies a neighbor router. Used to allow adjacency to be established over VPN tunnels.
- **network**—Defines the interfaces on which OSPF runs and the area ID for those interfaces.
- **redistribute**—Configures the redistribution of routes from one routing domain to another according to the parameters specified.
- **router-id**—Creates a fixed router ID.
- **summary-address**—Creates the aggregate addresses for OSPF.
- **timers lsa-group-pacing**—OSPF LSA group pacing timer (interval between group of LSA being refreshed or max-aged).
- **timers spf**—Delay between receiving a change to the SPF calculation.

You cannot configure OSPF when RIP is configured on the FWSM.

## Examples

The following example shows how to enter the configuration mode for the OSPF routing process numbered 5:

```
hostname(config)# router ospf 5
hostname(config-router)#
```

## Related Commands

| Command | Description |
|---|---|
| **clear configure router** | Clears the OSPF router commands from the running configuration. |
| **show running-config router ospf** | Displays the OSPF router commands in the running configuration. |

# router-id

To use a fixed router ID, use the **router-id** command in router configuration mode. To reset OSPF to use the previous router ID behavior, use the **no** form of this command.

**router-id** *addr*

**no router-id** [*addr*]

## Syntax Description

| | |
|---|---|
| *addr* | Router ID in IP address format. |

## Defaults

If not specified, the highest-level IP address on the FWSM is used as the router ID.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Router configuration | • | — | • | — | — |

## Command History

| Release | Modification |
|---|---|
| 1.1(1) | This command was introduced. |

## Usage Guidelines

If the highest-level IP address on the FWSM is a private address, then this address is sent in hello packets and database definitions. To prevent this address from being used, use the **router-id** command to specify a global address for the router ID.

Router IDs must be unique within an OSPF routing domain. If two routers in the same OSPF domain are using the same router ID, routing may not work correctly.

## Examples

The following example sets the router ID to 192.168.1.1:

```
hostname(config-router)# router-id 192.168.1.1
hostname(config-router)#
```

## Related Commands

| Command | Description |
|---|---|
| **router ospf** | Enters router configuration mode. |
| **show ospf** | Displays general information about the OSPF routing processes. |

# rule

To reallocate rules between features for a specific memory partition in multiple context mode, use the **rule** command in resource partition configuration mode. To restore the default values, use the **no** form of this command.

**rule nat** {*max_policy_nat_rules* | **current** | **default** | **max**}
**acl** {*max_ace_rules* | **current** | **default** | **max**}
**filter** {*max_filter_rules* | **current** | **default** | **max**}
**fixup** {*max_inspect_rules* | **current** | **default** | **max**}
**est** {*max_established_rules* | **current** | **default** | **max**}
**aaa** {*max_aaa_rules* | **current** | **default** | **max**}
**console** {*max_console_rules* | **current** | **default** | **max**}

**no rule**

## Syntax Description

| | |
|---|---|
| **aaa** *max_aaa_rules* | Sets the maximum number of AAA rules, between 0 and 10000. |
| **acl** *max_ace_rules* | Sets the maximum number of ACEs, between 0 and 74188. |
| **console** *max_console_rules* | Sets the maximum number of ICMP, Telnet, SSH, and HTTP rules, between 0 and 4000. |
| **current** | Keeps the current value set. |
| **default** | Sets the maximum rules to the default. To view the defaults, use the **show resource rule** command. |
| **est** *max_established_rules* | Sets the maximum number of **established** commands, between 0 and 716. The **established** command creates two types of rules, control and data. You allocate both rules by setting the number of **established** commands; you do not set each rule separately. However, both of these types are shown in the **show resource rule** and **show np 3 acl count** displays, so be sure to double the **est** value when comparing the total number of rules configured with the display in the **show** commands. |
| **filter** *max_filter_rules* | Sets the maximum number of filter rules, between 0 and 6000. |
| **fixup** *max_inspect_rules* | Sets the maximum number of inspect rules, between 0 and 10000. |
| **max** | Sets the rules to the maximum allowed for the feature. Be sure to set other features lower to accommodate this value. |
| **nat** *max_policy_nat_rules* | Sets the maximum number of policy NAT ACEs, between 0 and 10000. |

## Defaults

Use the **show resource rule** command to view default values.

## Command Modes

The following table shows the modes in which you can enter the command:

| | Firewall Mode | | Security Context | | |
|---|---|---|---|---|---|
| | | | | Multiple | |
| Command Mode | Routed | Transparent | Single | Context | System |
| Resource partition configuration | • | • | — | — | • |

## Command History

| Release | Modification |
|---|---|
| 4.0(1) | This command was introduced. |

## Usage Guidelines

### Information About Rules

There are a fixed number of rules available on the FWSM, so you might want to reallocate rules between features depending on usage. Features that use rules include access lists, inspections, AAA, and more.

If you increase the value for one feature, then you must decrease the value by the same amount for one or more features so the total number of rules does not exceed the system limit. Use the **show resource rule partition** command to view the total number of rules available, the default values, current rule allocation, and the absolute maximum number of rules you can allocate per feature.

### Important Guidelines

**Caution** Failure to follow these guidelines might result in dropped access list configuration as well as other anomalies, including ACL tree corruption.

- The target partition and rule allocation settings must be carefully calculated, planned, and preferably tested in a non-production environment prior to making the change to ensure that all existing contexts and rules can be accommodated.
- When failover is used, both FWSMs need to be reloaded at the same time after making partition changes. Reloading both FWSMs causes an outage with no possibility for a zero-downtime reload. At no time should two FWSMs with a mismatched number of partitions or rule limits synchronize over failover.
- You must enter all arguments in this command.
- This command takes effect immediately.
- This command overrides the global allocation settings set by the **resource rule** command.
- If you increase the size of a partition (the **size** command, or if you decrease the number of partitions using the **resource acl-partition** command) but have not yet reloaded, the maximum number of rules remains at the old smaller size. You have to reload to see the increased limits. If you decrease the size of a partition but have not yet reloaded, the new smaller number of rules is reflected right away.

### Viewing Rules

To view the total number of rules available per partition, the default values, current rule allocation, and the absolute maximum number of rules you can allocate per feature, enter the following command:

```
hostname(config)# show resource rule partition number
```

For example, the following display shows the maximum rules as 19219 for partition 0 (this is an example only, and might differ from the actual number of rules for your system):

```
hostname(config)# show resource rule partition 0

             Default  Configured  Absolute
 CLS Rule     Limit      Limit      Max
-----------+---------+----------+---------
 Policy NAT     384        384       833
 ACL          14801      14801     14801
 Filter         576        576      1152
 Fixup         1537       1537      3074
 Est Ctl         96         96        96
 Est Data        96         96        96
 AAA           1345       1345      2690
 Console        384        384       768
-----------+---------+----------+---------
 Total        19219      19219

Partition Limit - Configured Limit = Available to allocate
      19219     -      19219       =          0
```

To view the number of rules currently being used so you can plan your reallocation, enter the following command:

```
hostname(config)# show np 3 acl count partition_number
```

The following example shows the number of inspections (Fixup Rule) close to the maximum of 9216. You might choose to reallocate some access list rules (ACL Rule) to inspections.

```
hostname(config)# show np 3 acl count

-------------- CLS Rule Current Counts --------------
CLS Filter Rule Count       :            0
CLS Fixup Rule Count        :         9001
CLS Est Ctl Rule Count      :            4
CLS AAA Rule Count          :           15
CLS Est Data Rule Count     :            4
CLS Console Rule Count      :           16
CLS Policy NAT Rule Count   :            0
CLS ACL Rule Count          :        30500
CLS ACL Uncommitted Add     :            0
CLS ACL Uncommitted Del     :            0
...
```

## Examples

The following example shows how partition 0 reallocates 999 rules from the default 14,801 ACEs to inspections (default 9001):

```
hostname(config)# resource partition 0
hostname(config-partition)# rule nat default acl 13802 filter default fixup 10000 est
default aaa default console default
```

## Examples

| Command | Description |
|---|---|
| **allocate-acl-partition** | Assigns a context to a specific memory partition. |
| **context** | Configures a security context. |
| **resource acl-partition** | Sets the number of memory partitions for rules. |
| **resource rule** | Sets the resource rule allocation globally. |
| **show np 3 acl count** | Shows the number of rules in use. |
| **show resource acl-partition** | Shows the contexts assigned to each memory partition and the number of rules used. |
| **show resource rule** | Shows the total number of rules available, the default values, current rule allocation, and the absolute maximum number of rules you can allocate per feature. |